

INVERSE TRUSS DESIGN AS A CONIC MATHEMATICAL PROGRAM WITH EQUILIBRIUM CONSTRAINTS

MICHAL KOČVARA*

School of Mathematics, University of Birmingham
Birmingham B15 2TT, United Kingdom
and

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4
18208 Praha 8, Czech Republic

JIŘÍ V. OTRATA

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4
18208 Praha 8, Czech Republic

Dedicated to Tomáš Roubíček on the occasion of his 60th birthday

ABSTRACT. We formulate an inverse optimal design problem as a Mathematical Programming problem with Equilibrium Constraints (MPEC). The equilibrium constraints are in the form of a second-order conic optimization problem. Using the so-called Implicit Programming technique, we reformulate the bilevel optimization problem as a single-level nonsmooth nonconvex problem. The major part of the article is devoted to the computation of a subgradient of the resulting composite objective function. The article is concluded by numerical examples demonstrating, for the first time, that the Implicit Programming technique can be efficiently used in the numerical solution of MPECs with conic constraints on the lower level.

1. Introduction. A truss is an assemblage of m pin-jointed uniform straight bars. The bars are subjected to only axial tension and compression when the truss is loaded at the joints; the load is denoted by $f \in \mathbb{R}^n$. With a given load and a given set of joints at which the truss is fixed, the goal of the designer is to find a truss that is as light as possible and satisfies the equilibrium conditions. In the simplest, yet meaningful, approach, the number of the joints (nodes) and their position are kept fixed. The design variables are the bar cross-sectional areas assembled in a vector $a \in \mathbb{R}^m$ and the only constraints are the equilibrium equation and an upper bound on the weighted sum of the displacements of loaded nodes, so-called *compliance*. This model (or its equivalent reformulations) has been extensively analyzed in the mathematical and engineering literature (see, e.g., [1, 2, 17] and the references therein).

2010 *Mathematics Subject Classification.* Primary: 49K40, 74P05; Secondary: 49M05, 90C30.

Key words and phrases. Mathematical programs with equilibrium constraints, conic optimization, truss topology optimization.

The second author is supported by the Grant Agency of the Czech Republic project 15-00735S.

* Corresponding author.

Assume that we solved, in the past, the truss optimization problem for some load vector f^* ; the solution (bar cross-sectional areas) is denoted by a^* . We are now facing the question of solving the inverse problem: given an optimal design $a^* \in \mathbb{R}^m$, we want to find a load vector $f \in \mathbb{R}^n$ such that a^* is a solution of the truss optimization problem for this $f \in \mathbb{R}^n$. This might be useful, for instance, for existing constructions, to determine the loading (position and magnitude of forces) under which the structure performs best.

The resulting problem is a typical Mathematical Program with Equilibrium Constraints (MPEC) where on the upper level we minimize, with respect to the load vector f , the distance of the solution a from a given a^* and on the lower level we solve the truss optimization problem for a given f . One of the established techniques for the solution of a class of MPECs is the so-called implicit programming (ImP) technique; see [22, 28]. This technique replaces the initial bi-level problem by a single-level one, by using the *solution map* implicitly defined via the lower-level problem. The resulting single-level problem (in variable f only) is typically nonsmooth and one has to use a nonsmooth optimization algorithm for its solution.

For the implicit programming technique to work, some assumptions have to be satisfied. In particular, the lower-level optimization problem (in our case the truss design problem) must satisfy a certain constraint qualification which enables us to work with respective optimality conditions. Further, its solution map (assigning a to f) must be at least locally unique and locally Lipschitzian. In case of truss design, the problem is usually formulated as a nonlinear nonconvex optimization problem that, unfortunately, does not satisfy the required constraint qualifications. The problem can be reformulated as a convex quadratically constrained quadratic problem which, however, does not include a as a (primal) variable. We thus use yet another reformulation of the truss design problem as a conic optimization problem with a Cartesian product of Lorentz cones. We will show that this problem (in variable a) satisfies the required constraint qualification and arrive in this way at a mathematical program with constraints in the form of a conic optimization problem, or *conic MPEC*. Under the assumption that its solution of the lower level conic problem is unique we will apply the ImP technique. The conic MPECs have been rarely studied in the literature, due to their technical complexity. The main difficulty lies in finding a subgradient of the (nonsmooth) composite objective that includes the solution map to the conic optimization problem. For that we will use recent results based on the generalized differentiable calculus of B. Mordukhovich ([23, 25]) and on the coderivatives of the projections onto the Lorentz cone ([30]).

To the best of our knowledge, we present one of the first approaches to the solution of a conic MPEC that leads to a computationally approachable problem. In standard (not conic) MPECs one typically replaces the lower-level equilibrium problem by the corresponding KKT conditions (if possible), in which the (difficult) complementarity condition is either relaxed or added to the upper-level objective via a suitable (exact) penalty. In this way one eventually arrives at a nonlinear optimization problem in all primal variables and, additionally, the multipliers; see [16, 32]. Alternatively, under additional assumptions, the ImP approach mentioned above can be used; see [19] for the comparison of these two techniques. The advantage of the first technique is the fact that one does not need the often complicated formula for the subdifferential of the composite function, as in the ImP technique. On the other hand, the resulting nonlinear optimization problem may be difficult

to solve numerically because it is nonconvex, despite the original lower-level problem being convex or even linear. Moreover, to guarantee theoretical properties of this technique, one often needs similar assumptions as in ImP. The disadvantages of the first technique are magnified when considering a linear conic optimization problem on the lower level, as in this paper. While avoiding the complex analysis and technical formulas presented in Section 4, it leads to a nonlinear nonconvex conic optimization problem, for which there is no available software so far. Compared to that, the ImP approach used here requires solutions of linear second-order conic problems and of a nonconvex nonsmooth optimization problem in the control variable only. Numerical results presented in Section 5 show that, indeed, with this approach we can successfully find local (and often even global) optima in the considered MPECs. The reader is kindly asked to look at the practical relevance of our application from this point of view, as a demonstration of solvability of conic MPECs.

The plan of the paper is as follows. After introducing necessary notation, in Section 3 we define the truss design problem, give its various reformulations and show that the final formulation as a convex conic optimization problem could be amenable for the solution by the ImP technique. In this section we also define our main problem, the inverse truss design, as a conic MPEC. Section 4 presents details of the implicit programming technique, in particular, the computation of the subgradient information for the composite objective function. In Section 5 we show results of two numerical examples.

2. Notation and preliminaries. For a Lipschitz continuous function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\bar{\partial}f(x)$ denotes the *Clarke subdifferential* at x , defined by

$$\bar{\partial}f(x) = \text{conv}\left\{ \lim_{i \rightarrow \infty} \nabla f(x_i) \mid x_i \xrightarrow{\Omega_f} x \right\},$$

where Ω_f is the set of points at which f is differentiable.

In Section 4 we use the following notions of variational analysis. Given a closed set $A \subset \mathbb{R}^n$ and a point $\bar{x} \in A$, we denote by $T_A(\bar{x})$ the *Bouligand tangent cone* to A at \bar{x} , formed by vectors $h \in \mathbb{R}^n$ such that there exist sequences $h_k \rightarrow h$ and $t_k \downarrow 0$ for which $\bar{x} + t_k h_k \in A$ for all k .

We further denote by $\widehat{N}_A(\bar{x})$ the *Fréchet (regular) normal cone* to A at \bar{x} , defined as the (negative) polar cone to $T_A(\bar{x})$, i.e., by

$$\widehat{N}_A(\bar{x}) = (T_A(\bar{x}))^0.$$

The *limiting (Mordukhovich) normal cone* to A at \bar{x} , denoted $N_A(\bar{x})$, is defined by

$$N_A(\bar{x}) := \text{Limsup}_{x \xrightarrow{A} \bar{x}} \widehat{N}_A(x),$$

where “Limsup” is the Kuratowski-Painlevé outer limit of sets (see [31]). If A is convex, then $N_A(\bar{x}) = \widehat{N}_A(\bar{x})$ amounts to the classic normal cone in the sense of convex analysis. We say that \bar{x} is a (*normally*) *regular point* of A , provided $N_A(\bar{x}) = \widehat{N}_A(\bar{x})$.

Let $\Phi : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ be a multifunction with closed graph and (\bar{x}, \bar{y}) belong to the graph of Φ . The multifunction $D^*\Phi(\bar{x}, \bar{y}) : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$, defined by

$$D^*\Phi(\bar{x}, \bar{y})(y^*) := \{x^* \in \mathbb{R}^n \mid (x^*, -y^*) \in N_{\text{gph } \Phi}(\bar{x}, \bar{y})\}$$

is called *limiting (Mordukhovich) coderivative* of Φ at (\bar{x}, \bar{y}) . If Φ happens to be single-valued, we simply write $D^*\Phi(\bar{x})(\cdot)$. If Φ is continuously differentiable, then $D^*\Phi(\bar{x})$ amounts to the adjoint Jacobian of Φ at \bar{x} .

Let $K \in \mathbb{R}^m$ be a closed convex cone with non-empty interior. For $a \in \mathbb{R}^m$ we will use the notation $a \succeq_K 0$ for $a \in K$ and $a \succ_K 0$ for $a \in \text{int}(K)$. Consider the following linear conic optimization problem with equality constraints:

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} c^\top x & (1) \\ & \text{subject to} \\ & Ax = b \\ & Cx - d \succeq_K 0, \end{aligned}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$, $C \in \mathbb{R}^{m \times n}$, $d \in \mathbb{R}^m$.

Definition 2.1. We say that problem (1) satisfies the *Conic Mangasarian-Fromovitz constraint qualification (C-MFCQ)* if A has linearly independent rows and there exists a vector $\xi \in \mathbb{R}^n$ such that

$$\begin{aligned} A\xi &= b \\ C\xi - d &\succ_K 0. \end{aligned}$$

C-MFCQ is equivalent to the Robinson constraint qualification for a general optimization problem; see, e.g., [5, 15]. In the absence of equality constraints, it reduces to the Slater constraint qualification.

Definition 2.2. The cone $L^n \in \mathbb{R}^n$ defined by

$$L^n = \left\{ (x_1, \dots, x_{n-1}, x_n)^\top \mid \sqrt{x_1^2 + \dots + x_{n-1}^2} \leq x_n \right\}$$

is called the *n-dimensional Lorentz cone* or the *second-order cone*.

Linear conic problems (1) with the K being the Lorentz cone are called second-order conic programming problems (SOCP).

The rest of our notation is standard; in particular, $\text{gph } f$ denotes the graph of a map f , $\text{Bd}(\Omega)$ is the boundary of a set Ω , and $\text{Proj}_{\mathcal{K}}(\cdot)$ denotes the metric projection onto a convex set $\mathcal{K} \subset \mathbb{R}^n$. Finally, for $x \in \mathbb{R}$, $x_+ := \max\{0, x\}$.

3. Optimum truss design. By truss we understand an assemblage of pin-jointed uniform straight bars in two or three spatial dimensions. The bars can only carry axial tension and compression when the truss is loaded at the joints. A truss is determined by positions of nodes and cross-section areas of bars, i.e., by vectors $y \in \mathbb{R}^{\tilde{n}}$ and $a \in \mathbb{R}^m$. The bar lengths are denoted by ℓ_i . The material properties of bars are characterized by their Young's moduli E_i . The response of the truss to the vector of nodal forces $f \in \mathbb{R}^{\tilde{n}}$ is measured by the vector of nodal displacements $u \in \mathbb{R}^{\tilde{n}}$. Some of the nodes are assumed to be fixed, that is, some components of u are forced to be zero; the simplest way to treat these components is to omit them and work with reduced vectors $u \in \mathbb{R}^n$ and $f \in \mathbb{R}^n$, $n < \tilde{n}$. We will always assume that the nodal coordinates y are fixed.

We now denote by

$$q_i = \frac{E_i a_i}{\ell_i} r_i^\top u$$

the axial force in i -th bar and introduce bar stiffness matrices and assemble them in the global stiffness matrix of the truss

$$K(a) = \sum_{i=1}^m a_i K_i = \sum_{i=1}^m a_i \frac{E_i}{\ell_i} r_i r_i^\top, \quad i = 1, \dots, m$$

and introduce the equilibrium equation

$$K(a)u = f. \tag{2}$$

Here r_i denotes the vectors of directional cosines of the i -th bar; see [1]. To simplify the notation, we define $b_i = \sqrt{\frac{E_i}{\ell_i}} r_i$, so that $K(a) = \sum_{i=1}^m a_i b_i b_i^\top$. Finally, we denote by $\hat{a} \in \mathbb{R}^m$ an upper bound for vector a and by V the maximum volume of all bars.

Assumption 1. *The rows of the matrix composed column-wise from vectors b_i , i.e., the $n \times m$ matrix (b_1, b_2, \dots, b_m) , are linearly independent.*

Assumption 2. *The maximum volume V must satisfy $\frac{V}{m} < \hat{a}_i \ell_i, i = 1, \dots, m$, and thus $V < \sum_{i=1}^m \hat{a}_i \ell_i$*

It follows from Assumption 1 that $K(1_m)$ is positive definite, where $1_m \in \mathbb{R}^m$ is the vector of all ones. From the point of view of Mechanics, Assumption 1 excludes rigid body movements of the truss composed of bars of unit cross-sectional areas.

We first recall the standard truss topology design problem. The design variables are the bar areas a_i and we try to find such a design that the truss of limited volume is as stiff as possible, with respect to the given load f . This is formulated as the following standard nonlinear program (NLP), the so-called *minimum compliance problem*

$$\begin{aligned} \min_{a,u} f^\top u & \tag{3} \\ \text{subject to} & \\ \hat{a}_i \geq a_i \geq 0, \quad i = 1, \dots, m & \\ \sum_{i=1}^m \ell_i a_i \leq V & \\ K(a)u = f. & \end{aligned}$$

Lemma 3.1. *The optimal solution a^* of (3) is invariant with respect to the scaling of the load vector f , whereas the second component of the solution, u^* , is directly proportional to the scaling of f .*

Proof. This follows directly from the KKT conditions of (3). □

The next theorem shows that the seemingly straightforward optimization problem (3) does not satisfy (standard) Mangasarian-Fromovitz constraint qualification (MFCQ).

Theorem 3.2. *Let (\bar{a}, \bar{u}) be feasible for problem (3), and $K(\bar{a})$ be singular and such that some of its rows and columns are zero. Then MFCQ is violated at (\bar{a}, \bar{u}) .*

Proof. MFCQ is equivalent to the fact that the Lagrangian multipliers associated with the constraints of the optimization problem are bounded [5]. We will show that the multipliers in problem (3) may be unbounded.

We claim that MFCQ is violated when some components of a are equal to zero. Hence, without loss of generality, we will assume that $\hat{a} = \infty$, so that the first constraint reduces to $a_i \geq 0$. Let $\bar{\mu} \in \mathbb{R}^n$, $\bar{v} \in \mathbb{R}^m$, $\bar{\lambda} \in \mathbb{R}$ be the Lagrangian multipliers associated with the constraints in (3). The Lagrangian of (3) is then

$$f^\top \bar{u} - \bar{\mu}^\top (K(\bar{a})\bar{u} - f) + \bar{\lambda} \left(\sum_{i=1}^m \ell_i \bar{a}_i - V \right) - \bar{v}^\top \bar{a}$$

and the Karush-Kuhn-Tucker (KKT) conditions can be written as

$$\begin{aligned} f - K(\bar{a})\bar{\mu} &= 0 \\ \bar{\mu}^\top K_i \bar{u} - \ell_i \bar{\lambda} + \bar{v}_i &= 0, \quad i = 1, \dots, m \\ K(\bar{a})\bar{u} - f &= 0 \\ \sum_{i=1}^m \ell_i \bar{a}_i \leq V, \quad \bar{\lambda} \geq 0, \quad \left(\sum_{i=1}^m \ell_i \bar{a}_i - V \right) \bar{\lambda} &= 0 \\ \bar{a}_i \geq 0, \quad \bar{v}_i \geq 0, \quad \bar{a}^\top \bar{v} &= 0. \end{aligned}$$

The first condition shows that $\bar{\mu}$ solves the equilibrium equation for optimal \bar{a} . However, we have assumed that $K(\bar{a})$ is singular, in particular, that some of the rows and columns of $K(\bar{a})$ are zero (see the example below the proof). That is, after a proper re-ordering, the stiffness matrix is of the form

$$K(\bar{a}) = \begin{pmatrix} K_{1:n',1:n'} & 0_{n',n''} \\ 0_{n'',n'} & 0_{n'',n''} \end{pmatrix}$$

with some $0 < n', n'' < n$ such that $n' + n'' = n$. The last n'' components of vector $\bar{\mu}$ may be then arbitrary and thus unbounded.

Even unbounded, vector $\bar{\mu}$ must still satisfy the second KKT condition for all bars $i = 1, \dots, m$. Recall that $K(\bar{a}) = \sum_{i=1}^m \bar{a}_i K_i$ and assume that the last n'' components of vector $\bar{\mu}$ are indeed unbounded. Then, for every $i = 1, \dots, m$, either $(K_i)_{n'+1:n, n'+1:n}$ is a zero matrix and then $\bar{\mu}^\top K_i \bar{u}$ is finite, or $(K_i)_{n'+1:n, n'+1:n}$ is not zero. In the latter case, $\bar{\mu}^\top K_i \bar{u}$ is unbounded; however, the corresponding \bar{a}_i must be then equal to zero (as $(K(\bar{a}))_{n'+1:n, n'+1:n} = 0$ and the diagonal elements of K_i are nonnegative), the complementarity condition $\bar{a}_i \bar{v}_i = 0$ is trivially satisfied and thus also \bar{v}_i can be (nonnegative) unbounded for the second KKT condition to hold (see Example 1). □

Example 1. Consider the truss shown in Figure 1 with $V = 1$. The element

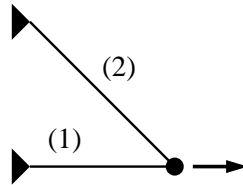


FIGURE 1. Two-bar truss.

stiffness matrices of the two bars and the right-hand side vector are

$$K_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad K_2 = \frac{1}{4} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad \text{and} \quad f = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Consider the point

$$a = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad u = \begin{pmatrix} 1 \\ u_2 \end{pmatrix}$$

with some (fixed) $u_2 \in \mathbb{R}$. The global stiffness matrix is then

$$K(a) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

and the equilibrium equation is satisfied by any μ of the form $\mu = (1, \mu_2)^\top$, $\mu_2 \in \mathbb{R}$. The second KKT condition reads then as

$$\begin{aligned} \mu^\top K_1 u - \lambda &= 0 \\ \mu^\top K_2 u - \lambda + \nu_2 &= 0. \end{aligned}$$

Clearly, $\lambda = 1$. Now, the latter equation reduces to

$$\frac{1}{4}(1 - u_2 - \mu_2 + u_2\mu_2) - 1 + \nu_2 = 0.$$

Without loss of generality, we may assume that $u_2 < 0$ (this component is arbitrary). The equation then remains valid when ν_2 and μ_2 go both to plus infinity with the same rate. \triangle

3.1. Dual problem (QCQP and QP formulations). We now introduce a reformulation of (3) that has a great impact on the numerical solution of truss design problems. It is the following quadratically constrained quadratic program (QCQP):

$$\min_{u \in \mathbb{R}^n, \alpha \in \mathbb{R}, \rho \in \mathbb{R}^m} \alpha V - f^\top u + \sum_{i=1}^m \hat{a}_i \rho_i \tag{4}$$

subject to

$$\begin{aligned} \frac{1}{2} u^\top K_i u &\leq \alpha \ell_i + \rho_i, \quad i = 1, \dots, m \\ \rho &\geq 0. \end{aligned}$$

Then next theorem is a compilation of several theorems from [1].

Theorem 3.3 ([1]). *The NLP and QCQP problems (3) and (4) are equivalent in the following sense:*

(i) *If one problem has a solution, then also the other problem has a solution and*

$$\min(3) = \min(4).$$

(ii) *Let (u^*, α^*, ρ^*) be a solution to (4). Let further τ^* be the vector of Lagrangian multipliers for the inequality constraints associated with this solution. Then (u^*, τ^*) is a solution of (3).*

(iii) *Let (u^*, a^*) be a solution of (3). Let further r^* be the Lagrangian multipliers associated with the upper bounds on a , respectively, and let α^* be the multiplier for the volume constraint. Then (u^*, α^*, r^*) is a solution of (4).*

The QCQP problem (4) can further be rewritten as a standard (linearly constrained) quadratic programming problem. We will use the fact that α is always nonnegative. This follows from the quadratic constraints in (4) and the fact that, by Assumption 2, $\rho_i = 0$ for at least one i .

Theorem 3.4. *The QCQP problem (4) is equivalent to the quadratic program*

$$\min_{u \in \mathbb{R}^n, \mu \in \mathbb{R}, s \in \mathbb{R}^m} (s^\top, \mu) C (s^\top, \mu)^\top - f^\top u \tag{5}$$

subject to

$$\begin{aligned} \frac{1}{\sqrt{2\ell_i}} b_i^\top u &\leq \mu + s_i, \quad i = 1, \dots, m \\ -\frac{1}{\sqrt{2\ell_i}} b_i^\top u &\leq \mu + s_i, \quad i = 1, \dots, m \\ s &\geq 0 \\ \mu &\geq 0, \end{aligned}$$

where

$$C = \begin{pmatrix} \hat{a}_1 \ell_1 & & & \hat{a}_1 \ell_1 \\ & \ddots & & \vdots \\ & & \hat{a}_m \ell_m & \hat{a}_m \ell_m \\ \hat{a}_1 \ell_1 & \dots & \hat{a}_m \ell_m & V \end{pmatrix}.$$

Proof. We first apply transformation of variables α, ρ in (4) to new variables $\mu \in \mathbb{R}, s \in \mathbb{R}^m$,

$$\alpha = \mu^2, \quad \rho_i = \ell_i(2\mu s_i + s_i^2),$$

to get a problem

$$\min_{u \in \mathbb{R}^n, \mu \in \mathbb{R}, s \in \mathbb{R}^m} \mu^2 V - f^\top u + \sum_{i=1}^m \hat{a}_i \ell_i (2\mu s_i + s_i^2)$$

subject to

$$\begin{aligned} \frac{1}{2} \frac{1}{\ell_i} u^\top K_i u &\leq (\mu + s_i)^2, \quad i = 1, \dots, m \\ s &\geq 0 \\ \mu &\geq 0. \end{aligned}$$

Now recalling that $K_i = b_i b_i^\top$, we immediately get (5). □

Remark 1. Matrix C in Theorem 3.4 may be indefinite, in general. Indeed, if V is not big enough, relative to $\hat{a}_i \ell_i$, it may have one negative eigenvalue and m non-negative ones. Only when $V \geq \sum_{i=1}^m \hat{a}_i \ell_i$ (and this is *not* a typical case and contradicts Assumption 2), C will become positive semidefinite, according to the Gershgorin theorem [14]. It was shown in [1] that problem (5) is convex nonetheless, as C is positive semidefinite on the feasible set of (5). However, the consequence is that (5) cannot be readily solved by available software for convex quadratic programming, such as Gurobi [11].

3.2. Primal SOCP formulation. We recall a simple but useful lemma that shows the relation between convex quadratic constraints and second-order conic constraints. The proof follows from direct evaluation of the expression on the right-hand side.

Lemma 3.5. *Let $x \in \mathbb{R}^n, t \in \mathbb{R}$ and $s \in \mathbb{R}, s > 0$. Then*

$$\frac{x^\top x}{s} \leq t \iff \left\| \begin{pmatrix} x \\ \frac{t-s}{2} \end{pmatrix} \right\|_2 \leq \frac{t+s}{2}.$$

Using this lemma, QCQP (4) can be immediately re-written as an SOCP problem

$$\min_{u \in \mathbb{R}^n, \alpha \in \mathbb{R}, \rho \in \mathbb{R}^m} \alpha V - f^\top u + \sum_{i=1}^m \hat{a}_i \rho_i \tag{6}$$

subject to

$$\left\| \begin{pmatrix} \frac{\sqrt{2}}{2} b_i^\top u \\ \frac{\alpha \ell_i + \rho_i - 1}{2} \end{pmatrix} \right\|_2 \leq \frac{\alpha \ell_i + \rho_i + 1}{2}, \quad i = 1, \dots, m$$

$$\rho \geq 0.$$

3.3. Dual SOCP formulation. The primal SOCP formulation (6) does not include the variable a , the cross-section areas. This will be desirable in the next section, so we have to work with its dual.

Proposition 3.6. *The dual problem to (6) can be written as*

$$\min_{a \in \mathbb{R}^m, \tau \in \mathbb{R}^m, q \in \mathbb{R}^m} \frac{1}{2} \sum_{i=1}^m \tau_i \tag{7}$$

subject to

$$\sum_{i=1}^m a_i \ell_i = V$$

$$\sum_{i=1}^m q_i b_i = f$$

$$\left\| \begin{pmatrix} \sqrt{2} q_i \\ \frac{2a_i - \tau_i}{2} \end{pmatrix} \right\|_2 \leq \frac{2a_i + \tau_i}{2}, \quad i = 1, \dots, m$$

$$0 \leq a_i \leq \hat{a}_i, \quad i = 1, \dots, m,$$

where a are the bar areas, q the bar axial forces and the objective function is equal to the compliance.

Proof. Let μ_i, ν_i be Lagrangian multipliers to the conic constraints, where $\mu_i = (\mu_{i,1}, \mu_{i,2}) \in \mathbb{R}^2$ and $\nu_i \in \mathbb{R}$, $i = 1, \dots, m$. Further, let $\kappa \in \mathbb{R}^m$ be the multiplier to the bound constraint on ρ . The Lagrangian dual to (6) reads as

$$\max_{\mu \in \mathbb{R}^{m \times 2}, \nu \in \mathbb{R}^m, \kappa \in \mathbb{R}^m} \frac{1}{2} \sum_{i=1}^m \mu_{i,2} - \nu_i$$

subject to

$$\frac{1}{2} \sum_{i=1}^m (\mu_{i,2} + \nu_i) \ell_i = V$$

$$\sum_{i=1}^m \mu_{i,1} \frac{\sqrt{2}}{2} b_i = f$$

$$\|\mu_i\|_2 \leq \nu_i, \quad i = 1, \dots, m$$

$$\frac{1}{2} \mu_{i,2} + \frac{1}{2} \nu_i + \kappa_i = \hat{a}_i, \quad i = 1, \dots, m$$

$$\kappa_i \geq 0, \quad i = 1, \dots, m.$$

Setting $q_i = \frac{\sqrt{2}}{2}\mu_{i,1}$, $a_i = \frac{1}{2}(\mu_{i,2} + \nu_i)$ and $\tau = \nu_i - \mu_{i,2}$, we first see that $a_i \leq \widehat{a}_i$ (from the last two constraints), and that $a_i \geq 0$ (as $|\mu_{i,2}| \leq \nu_i$ from the conic constraint). Consequently, we get (7). \square

An important property of the above reformulation is stated in the next proposition.

Proposition 3.7. *The dual SOCP problem (7) satisfies C-MFCQ from Definition 2.1.*

Proof. By Assumption 1, the gradients of the equality constraints are linearly independent. We need to find a point (a, τ, q) feasible with respect to the equality constraints and strictly feasible with respect to the conic inequalities. (The standard inequalities $a_i \geq 0$ and $\widehat{a}_i - a_i \geq 0$, $i = 1, \dots, m$, can of course be viewed as conic inequalities with the cone \mathbb{R}_+^m .)

Choose

$$\tilde{a}_i = \frac{V}{m\ell_i}, \quad i = 1, \dots, m.$$

Then $\sum_{i=1}^m \tilde{a}_i \ell_i = V$ and, by Assumption 2, $0 < \tilde{a}_i < \widehat{a}_i$, $i = 1, \dots, m$. By Assumption 1, the matrix $K(\tilde{a})$ with $\tilde{a}_i = \tilde{a}_i \ell_i^{-\frac{1}{2}}$ is positive definite, so that we can define a (finite) unique vector

$$\tilde{u} = (K(\tilde{a}))^{-1} f$$

and the corresponding axial forces

$$\tilde{q}_i = \frac{E_i}{\ell_i} \tilde{a}_i b_i^\top \tilde{u}, \quad i = 1, \dots, m.$$

By Lemma 3.5, the conic inequalities can be written as

$$\frac{a_i}{\ell_i} u^\top K_i u \leq \tau_i, \quad i = 1, \dots, m.$$

For \tilde{a} and \tilde{u} the left-hand side is always finite, so that we can find $\tilde{\tau}_i$, $i = 1, \dots, m$, such that

$$\frac{\tilde{a}_i}{\ell_i} \tilde{u}^\top K_i \tilde{u} < \tilde{\tau}_i, \quad i = 1, \dots, m.$$

The triple $(\tilde{a}, \tilde{\tau}, \tilde{q})$ is the vector satisfying Definition 2.1. \square

3.4. Uniqueness of solution to SOCP (7). It is well known that the optimal solution of problem (7), in particular, the design vector a may be non-unique. This is demonstrated in Figure 2: it shows a 3×3 truss with all nodes connected by potential bars. Both trusses presented in this figure and all their convex combinations are optimal solutions of problem (7) (this can easily be confirmed numerically). We have to emphasize, however, that this is a very rare situation requiring a special combination of the nodal positions and the load vector. Indeed, in the above example, a minor change in either the nodal positions or the force direction would result in unique solution a^* of (7).

In the MPEC problem (8) below we want to identify some a^* , an optimal solution of (7) for a given f^* . Let $S(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{3m}$ be the *solution map* assigning a load vector f the (unique) optimal solution (a, τ, q) of (7). Denote by S_a a “restriction” of this map satisfying

$$\text{gph } S_a = \{(f, a) \mid \exists \tau, q : (f, a, \tau, q) \in \text{gph } S\}.$$

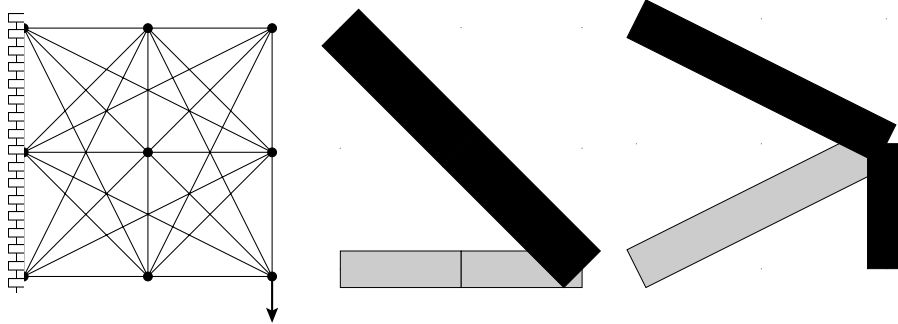


FIGURE 2. Two solutions of the 3×3 truss design problem with all nodes connected.

In the following, we will assume that the truss nodal positions and load vector f^* are chosen in such a way that S_a has a Lipschitzian single-valued localization around (f^*, a^*) , see [9, p.4]. In our case this means, by virtue of the convex valuedness of S_a , that $S_a(f^*) = a^*$ and there is a neighborhood \mathcal{U} of f^* such that S_a is single-valued and Lipschitz continuous on \mathcal{U} . It is readily seen from the proof of Proposition 3.6 and the definition of the axial force q that the single-valuedness and the Lipschitz continuity of S_a over \mathcal{U} amounts, in fact, the single-valuedness and Lipschitz continuity of S over \mathcal{U} . This may be ensured, e.g., via the property of nondegeneracy introduced in the next section and the appropriate strong second-order sufficient condition, see [6, Thm. 5].

3.5. MPEC for inverse truss design. Assume that we solved, in the past, the dual SOCP formulation (7) of the truss design problem for some load vector f^* . We are now facing the question of solving the inverse problem: given an optimal design $a^* \in \mathbb{R}^m$, we want to find a load vector $f \in \mathbb{R}^n$ such that a^* is a solution of (7) for this $f \in \mathbb{R}^n$. As mentioned in the Introduction, this may be useful, for instance, for existing constructions, when we want to determine the loading (positions and magnitudes of forces) under which the structure performs best.

As usual in such problems, we will try to find a vector $a \in \mathbb{R}^m$ that minimizes the 2-norm of the difference between a and a^* . The problem can be formulated as follows:

$$\begin{aligned} & \min_{f \in \mathbb{R}^n, a \in \mathbb{R}^m} (a - a^*)^\top (a - a^*) & (8) \\ & \text{subject to} \\ & \quad a \text{ solves (7) with load vector } f \\ & \quad f \in \mathcal{F}, \end{aligned}$$

where $\mathcal{F} \subset \mathbb{R}^n$ is a feasible set for f . Problem (8) is a conic MPEC, as defined in the Introduction. As also mentioned there, one of the established techniques for the solution of MPECs is the so-called implicit programming technique (see [22, 28]).

The idea of the ImP approach is to replace problem (8) by the following non-smooth optimization problem, referred to as ImP-MPEC:

$$\min_{f \in \mathbb{R}^n} (S_a(f) - a^*)^\top (S_a(f) - a^*)$$

$$\begin{aligned} &\text{subject to} \\ &f \in \mathcal{F}. \end{aligned} \tag{9}$$

The new problem (9) can now be solved by algorithms and software of nonsmooth optimization. In particular, the ImP technique is known to couple well, e.g., with the bundle algorithm described in [7]. This algorithm requires at each iteration, i.e., for any choice of $f \in \mathcal{F}$, the knowledge of the objective function value of (9) and of one arbitrary element of its Clarke subdifferential. While the objective function value is immediately available, the computation of an element of the subdifferential is highly nontrivial and, in case of conic MPEC, has not been conducted in the literature yet. The next section is devoted to this issue.

4. Computation of subgradients of the composite objective. Assume that f belongs to a neighborhood of f^* on which $S : f \mapsto (a, \tau, q)$ generated by the dual SOCP problem (7) is single-valued and Lipschitz continuous, as discussed in the previous section. In the computation of the “subgradient” information needed for the minimization of the respective composite objective function via the used bundle method, we take advantage of the generalized differential calculus of B. Mordukhovich and employ the chain rule [23, Thm. 1.110]. It follows that for an arbitrary continuously differentiable function φ of argument a and for a fixed quadruple $(\bar{f}, \bar{a}, \bar{\tau}, \bar{q}) \in \text{gph } S$ one has the inclusion

$$\bar{\partial}(\varphi \circ S)(\bar{f}) \supset D^*S(\bar{f}, \bar{a}, \bar{\tau}, \bar{q})(\nabla\varphi(\bar{a}), 0, 0), \tag{10}$$

whenever S is single-valued and Lipschitzian around \bar{f} .

So any element from the right-hand side of (10) provides a workable subgradient information at the considered point. Unfortunately, the currently available tools of variational analysis do not enable us to compute such an element at all points of $\text{gph } S$. In this section we will explain the used technique and discuss the properties of the resulting information which will be supplied to the bundle algorithm as a desired Clarke subgradient of the composite objective function in ImP-MPEC (9). To this purpose, we replace SOCP (7) by a slightly more general convex conic program which will allow us to simplify both the notation and the final formulas.

This conic program, dependent on parameter $f \in \mathbb{R}^n$, attains the form

$$\begin{aligned} &\min_y h(y) \\ &\text{subject to} \\ &g(f, y) = 0 \\ &b(y) \in \mathcal{K} \\ &y \in \Omega, \end{aligned} \tag{11}$$

where $y \in \mathbb{R}^o$ stands for the triple (a, τ, q) , $g : \mathbb{R}^n \times \mathbb{R}^o \rightarrow \mathbb{R}^r$ is affine linear and models the map in the equality constraints in (7), $\Omega \in \mathbb{R}^o$ is a closed convex set replacing the bound constraints on a and the constraint $b(y) \in \mathcal{K}$ generalizes the conic constraints in (7). We will assume that $h : \mathbb{R}^o \rightarrow \mathbb{R}$ is convex and continuously differentiable, $\mathcal{K} \subset \mathbb{R}^p$ is a closed cone with vertex at 0 and non-empty interior and $b : \mathbb{R}^o \rightarrow \mathbb{R}^p$ is twice continuously differentiable and \mathcal{K} -convex [5, Def. 2.103]. These assumptions are satisfied by the dual SOCP problem (7) and imply, in particular, that (11) is a convex program. Let Γ be the constraint multifunction in (11) defined by

$$\Gamma(f) = \{y \mid \Psi(f, y) \in D\}, \tag{12}$$

where

$$\Psi(f, y) = \begin{bmatrix} g(f, y) \\ b(y) \\ y \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} \{0\} \\ \mathcal{K} \\ \Omega \end{bmatrix}.$$

Further, let $\Sigma : \mathbb{R}^n \rightrightarrows \mathbb{R}^o$ be the “solution map” of (11) which assigns each f the corresponding set of minimizers y . Clearly, by virtue of the assumed convexity, for $f = \bar{f}$, \bar{y} is a solution of (11) if and only if it is a solution to the GE

$$0 \in \nabla h(y) + N_{\Gamma(\bar{f})}(y),$$

and, with $Q(f, y) := N_{\Gamma(f)}(y)$, one has

$$\text{gph } \Sigma = \left\{ (f, y) \in \mathbb{R}^n \times \mathbb{R}^o \mid \begin{bmatrix} f \\ y \\ -\nabla h(y) \end{bmatrix} \in \text{gph } Q \right\}.$$

Define now the associated *perturbation mapping* $M : \mathbb{R}^{n+2o} \rightarrow \mathbb{R}^{n+o}$ via

$$M(p) := \left\{ (f, y) \mid \begin{bmatrix} f \\ y \\ -\nabla h(y) \end{bmatrix} - p \in \text{gph } Q \right\}. \tag{13}$$

According to [12, Thm. 4.1] one has that

$$N_{\text{gph } \Sigma}(\bar{f}, \bar{y}) \subset \{ (a, d + \nabla^2 h(\bar{y})c) \mid (a, d) \in D^*Q(\bar{f}, \bar{y}, -\nabla h(\bar{y}))(c) \}, \tag{14}$$

whenever M has the calmness property at $(0, \bar{f}, \bar{y})$; see [31, p. 399]. This property is fulfilled under the “standard” 2nd-order qualification condition

$$\left. \begin{array}{l} d + \nabla^2 h(\bar{y})c = 0 \\ (0, d) \in D^*Q(\bar{f}, \bar{y}, -\nabla h(\bar{y}))(c) \end{array} \right\} \implies d = 0, \quad c = 0, \tag{15}$$

which ensures even the stronger Aubin property of M around $(0, \bar{f}, \bar{y})$ ([31, Def.9.36]).

In most applications of the ImP technique to various MPECs, the coderivative $D^*Q(\bar{f}, \bar{y}, -\nabla h(\bar{y}))$ can be computed and condition (15) can be verified. Even then, however, the subgradient information based on the right-hand side in (14) need not be always correct because it is an upper estimate of $N_{\text{gph } \Sigma}(\bar{f}, \bar{y})$. In our case, the situation is even more complicated because, under the imposed C-MFCQ, we dispose only with an upper approximation of $D^*Q(\bar{f}, \bar{y}, -\nabla h(\bar{y}))$ derived in [25, Thm. 4.3]. Notice that this upper approximation requires the fulfillment of an additional calmness assumption and knowledge of all Lagrange multipliers associated with the constraint system (12). Fortunately, recent results in [10] indicate that certain conic programs fulfill generically the so-called nondegeneracy condition and under this condition $D^*Q(\bar{f}, \bar{y}, -\nabla h(\bar{y}))$ can be computed exactly. (More precisely, in [10] this has been proved for nonlinear semidefinite programming (SDP) problems and thus, as every SOCP problem can be formulated as an SDP problem, also for our dual SOCP (7).)

To introduce the nondegeneracy in the case of problem (11), notice first that C-MFCQ at (\bar{f}, \bar{y}) for this problem amounts to the implication

$$\left. \begin{array}{l} 0 \in (\nabla_y g(\bar{f}, \bar{y}))^\top \lambda + (\nabla b(\bar{y}))^\top \mu + N_{\Omega}(\bar{y}) \\ g(\bar{f}, \bar{y}) = 0, \quad \mu \in N_{\mathcal{K}}(b(\bar{y})) \end{array} \right\} \implies \lambda = 0, \quad \mu = 0. \tag{16}$$

This follows from [5, Prop. 2.89] because both these conditions are equivalent with the metric regularity of the canonically perturbed conic constraint system

$$\Psi(\bar{f}, y) \in D \tag{17}$$

around $(\bar{y}, 0)$. Following [5, Def. 4.70] and [29, p. 804], we may now say that \bar{y} is a *nondegenerate* point of (17), provided (16) is strengthened to the form

$$\left. \begin{aligned} 0 \in (\nabla_y g(\bar{f}, \bar{y}))^\top \lambda + (\nabla b(\bar{y}))^\top \mu + \text{span } N_\Omega(\bar{y}) \\ g(\bar{f}, \bar{y}) = 0, \mu \in \text{span } N_{\mathcal{K}}(b(\bar{y})) \end{aligned} \right\} \Rightarrow \lambda = 0, \mu = 0. \tag{18}$$

It is easy to prove that if \bar{y} is a nondegenerate point of (17), then there are unique Lagrange multipliers $\bar{\lambda} \in \mathbb{R}^r$, $\bar{\mu} \in N_{\mathcal{K}}(b(\bar{y}))$ and $\bar{\nu} \in N_\Omega(\bar{y})$, satisfying the KKT conditions

$$\begin{aligned} \nabla_y \mathcal{L}(\bar{f}, \bar{y}, \bar{\lambda}, \bar{\mu}, \bar{\nu}) &= 0 \\ g(\bar{f}, \bar{y}) &= 0, \end{aligned} \tag{19}$$

where $\mathcal{L}(f, u, \lambda, \mu, \nu) := \nabla h(y) + \nabla_y g(f, y)^\top \lambda + \nabla b(y)^\top \mu + \nu$ is the Lagrangian associated with problem (11).

We may now invoke the recent results in [24, 26] and arrive at the following statement.

Theorem 4.1. *Let \bar{y} be a nondegenerate point of (17) and $\bar{\lambda} \in \mathbb{R}^r$, $\bar{\mu} \in N_{\mathcal{K}}(b(\bar{y}))$ and $\bar{\nu} \in N_\Omega(\bar{y})$ be the unique Lagrange multipliers satisfying (19). Then one has*

$$\begin{aligned} D^*Q(\bar{f}, \bar{y}, -\nabla h(\bar{y}))(v) &= \left\{ (a, w) \in \mathbb{R}^n \times \mathbb{R}^o \mid a = (\nabla_f g(\bar{f}, \bar{y}))^\top d, \right. \\ & \quad w \in \nabla_{yy}^2 \langle g, \bar{\lambda} \rangle (\bar{f}, \bar{y}) v + \nabla^2 \langle b, \bar{\mu} \rangle (\bar{y}) v + (\nabla_y g(\bar{f}, \bar{y}))^\top d \\ & \quad \left. + (\nabla b(\bar{y}))^\top D^*N_{\mathcal{K}}(b(\bar{y}), \bar{\mu})(\nabla b(\bar{y})v) + D^*N_\Omega(\bar{y}, \bar{\nu})(v) \right. \\ & \quad \left. \text{with some vectors } d \in \mathbb{R}^r, v \in \ker \nabla_y g(\bar{f}, \bar{y}) \right\}. \end{aligned} \tag{20}$$

Proof. The proof follows essentially from [26, Thm. 3.1], where this type of a second-order chain rule has been derived under surjectivity of $\nabla_y \Psi(\bar{f}, \bar{y})$. The replacement of this requirement by the nondegeneracy has been conducted in [24, Prop. 5.1]. The restriction $v \in \ker \nabla_y g(\bar{f}, \bar{y})$ comes from the fact that

$$D^*N_{\{0\}}(g(\bar{f}, \bar{y}), \bar{\lambda})(\nabla_y g(\bar{f}, \bar{y})v) = \begin{cases} \mathbb{R}^r & \text{provided } \nabla_y g(\bar{f}, \bar{y})v = 0 \\ \emptyset & \text{otherwise,} \end{cases}$$

so that we have no restrictions on d , but have to require that $v \in \ker \nabla_y g(\bar{f}, \bar{y})$.

Vector a amounts to the single term $(\nabla_f g(\bar{f}, \bar{y}))^\top d$ because $\nabla_y g(\bar{f}, \bar{y})$ does not depend on \bar{f} . □

Under the assumptions of Theorem 4.1 we thus immediately obtain that the right-hand side of (14) amounts to

$$\begin{aligned} \left\{ (\xi, \eta) \in \mathbb{R}^n \times \mathbb{R}^o \mid \xi &= (\nabla_f g(\bar{f}, \bar{y}))^\top d, \eta \in \nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}, \bar{\mu}, \bar{\nu})^\top v \right. \\ & \quad \left. + (\nabla_y g(\bar{f}, \bar{y}))^\top d + (\nabla b(\bar{y}))^\top D^*N_{\mathcal{K}}(b(\bar{y}), \bar{\mu})(\nabla b(\bar{y})v) + D^*N_\Omega(\bar{y}, \bar{\nu})(v) \right. \\ & \quad \left. \text{with some vectors } d \in \mathbb{R}^r, v \in \ker \nabla_y g(\bar{f}, \bar{y}) \right\}. \end{aligned} \tag{21}$$

It now follows from the definition of the coderivative that $D^*\Sigma(\bar{f}, \bar{y})(y^*)$ can be approximated by the set of vectors ξ such that $(\xi, -y^*)$ belongs to the set (21). This

type of approximation has been used in numerous applications of the ImP technique to MPECs with complicated variational systems on the lower level [3, 4, 18].

So, given a vector $\eta \in \mathbb{R}^p$, let us define the “adjoint system” in variables $d \in \mathbb{R}^r$, $v \in \mathbb{R}^p$:

$$\left. \begin{aligned} \eta &\in \nabla_y \mathcal{L}(\bar{y}, \bar{\lambda}, \bar{\mu}, \bar{\nu})^\top v + (\nabla_y g(\bar{f}, \bar{y}))^\top d \\ &\quad + (\nabla b(\bar{y}))^\top D^* N_{\mathcal{K}}(b(\bar{y}), \bar{\mu})(\nabla b(\bar{y})v) + D^* N_{\Omega}(\bar{y}, \bar{\nu})(v) \\ v &\in \ker \nabla_y g(\bar{f}, \bar{y}). \end{aligned} \right\} \quad (22)$$

On the basis of (21) we propose the following three-step procedure for the computation of a point that will be supplied as an element from $\bar{\partial}(h \circ \Sigma)(\bar{f})$.

Algorithm 1.

- (i) Solve (11) with $f = \bar{f}$ to obtain \bar{y} and associated multipliers $\bar{\lambda}, \bar{\mu}, \bar{\nu}$.
- (ii) Solve (22) with these data and $\eta = -\nabla h(\bar{y})$ to obtain a solution $(\bar{d}, \bar{v}) \in \mathbb{R}^r \times \mathbb{R}^p$.
- (iii) Put

$$\xi = (\nabla_f g(\bar{f}, \bar{y}))^\top \bar{d}. \quad (23)$$

Next we observe that the vector ξ from (23) belongs to $D^* \Sigma(\bar{f}, \bar{y})(\nabla h(\bar{y}))$, provided

- (a) M (defined in (13)) is calm at $(0, \bar{f}, \bar{y})$;
- (b) \bar{y} is a nondegenerate point of (17);
- (c) the points $(b(\bar{y}), \bar{\mu})$ and $(\bar{y}, \bar{\nu})$ are regular points of the sets $\text{gph } N_{\mathcal{K}}$ and $\text{gph } N_{\Omega}$, respectively.

Indeed, following [31, Thm.6.14] and [23, Thm.4.1], it can be shown that $\widehat{N}_{\text{gph } \Sigma}(\bar{f}, \bar{y})$ contains the set (21), provided the limiting coderivatives $D^* N_{\mathcal{K}}(b(\bar{y}), \bar{\mu})$ and $D^* N_{\Omega}(\bar{y}, \bar{\nu})$ are replaced by their regular counterparts. This fact, together with inclusion (14) produces a chain of inclusions which, under condition (c), become equalities. The nature of conditions (a)–(c) indicates that in the course of the iteration process we meet almost always points where our methods leads to a correct subgradient. This behaviour has been observed also in the case of the MPEC problem (8). Further notice that the algorithm of the used nonsmooth code BTNCLC is sufficiently robust so that it can cope with occasionally incorrect subgradients. Concerning (a),(b), one could attempt to verify implications (15) and (18) at the considered points. This would, however, be extremely time-consuming and definitely not practicable during the iteration process. It is substantially easier to verify the stationarity of the obtained results on the basis of a suitable stationarity concept. Finally notice that in many of our numerical examples presented in the next section, BTNCLC converged to points giving optimal objective values smaller than 10^{-7} . Due to the nature of problem, without any further verification we may conclude that these points are (approximations of) global minima.

We conclude this section with a specification of the above procedure to our MPEC problem (8). With respect to the structure of $\mathcal{K}(= \prod_{i=1}^m L^3)$, it is convenient to put

$$y = (y_1, y_2, \dots, y_m) \in \mathbb{R}^{3m} \quad \text{with} \quad y_i = (a_i, \tau_i, q_i) \in \mathbb{R}^3, \quad i = 1, \dots, m.$$

Also vector $b(y)$, multipliers μ, ν , adjoint variable v and vector \mathcal{L} have the same structure, where $v = (v_1, v_2, \dots, v_m) \in \mathbb{R}^{3m}$ with $v_i = (v_i^1, v_i^2, v_i^3)$,

$$\mathcal{L}(y, \lambda, \mu, \nu) = (\mathcal{L}_1(y_1, \lambda, \mu_1, \nu_1), \dots, \mathcal{L}_m(y_m, \lambda, \mu_m, \nu_m))$$

with

$$\mathcal{L}_i(y_i, \lambda, \mu_i, \nu_i) = \begin{bmatrix} 0 \\ \frac{1}{2} \\ 0 \end{bmatrix} + B_i^\top \lambda + A^\top \mu_i + \begin{bmatrix} \nu_i \\ 0 \\ 0 \end{bmatrix}, \quad i = 1, \dots, m,$$

and

$$A = \begin{bmatrix} 0 & 0 & \sqrt{2} \\ 1 & -\frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 0 \end{bmatrix}, \quad B_i = \begin{bmatrix} \ell_i & 0 & 0 \\ 0 & 0 & b_i \end{bmatrix},$$

i.e., $B_i \in \mathbb{R}^{(1+n) \times 3}$. Finally, let $\lambda = (\lambda^1, \lambda^2) \in \mathbb{R} \times \mathbb{R}^n$, where λ_1 and λ_2 are related to the first and the second equality in the dual SOCP problem (7), respectively. Putting everything together, we arrive at a system of linear equations in $7m + n + 1$ variables $\alpha = (\alpha_1^1, \alpha_1^2, \alpha_1^3, \dots, \alpha_m^1, \alpha_m^2, \alpha_m^3) \in \mathbb{R}^{3m}$, $d \in \mathbb{R}^{1+n}$, $\gamma \in \mathbb{R}^m$, $v = (v_1^1, v_1^2, v_1^3, \dots, v_m^1, v_m^2, v_m^3) \in \mathbb{R}^{3m}$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a_i - a_i^* \\ 0 \\ 0 \end{bmatrix} + B_i^\top \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + A^\top \begin{bmatrix} \alpha_i^1 \\ \alpha_i^2 \\ \alpha_i^3 \end{bmatrix} + \begin{bmatrix} \gamma_i \\ 0 \\ 0 \end{bmatrix} \quad \text{for } i = 1, \dots, m \quad (24)$$

$$\sum_{i=1}^m B_i v_i = 0 \quad (25)$$

together with the relations

$$\alpha_i \in D^* N_{L^3}(A\bar{y}_i, \bar{\mu}_i)(Av_i) \quad \text{for } i = 1, \dots, m \quad (26)$$

$$\gamma_i \in D^* N_{[0, \hat{a}_i]}(\bar{a}_i, \bar{v}_i)(v_i^1) \quad \text{for } i = 1, \dots, m. \quad (27)$$

System (24)–(27) represents a specific form of adjoint system (22). By a suitable choice of selections from the coderivative mappings $D^* N_{L^3}(A\bar{y}_i, \bar{\mu}_i)$ and $D^* N_{[0, \hat{a}_i]}(\bar{a}_i, \bar{v}_i)$, we obtain a square linear system in (α, d, γ, v) ; see [19, pp. 138–139].

The limiting coderivatives of N_{L^3} and $N_{[0, \hat{a}_i]}$ are well-known (see, e.g., [30, 8] and the Appendix) and so the choice of suitable vectors α_i, γ_i (dependent on the position of considered points in the graph of N_{L^3} and $N_{[0, \hat{a}_i]}$, respectively) does not make any difficulties. In step (iii) of Algorithm 1 we then put $\xi = -\bar{d}_2$, where $(\bar{\alpha}, \bar{d}, \bar{\gamma}, \bar{v})$ is a solution of system (24)–(27).

Finally notice that under assumptions (a)–(c) stated above, the system (24)–(27) possesses a solution. This follows from the fact that the set of all vectors $-d_2$ arising in the solutions of (24)–(27) amounts under (a)–(c) to the limiting (Mordukhovich) subdifferential of $\varphi \circ S$ at \bar{f} , where φ is the objective in the ImP-MPEC problem (9). Since $\varphi \circ S$ is Lipschitz around \bar{f} , the claim is implied by the non-emptiness of this subdifferential, see [23, p. 86].

5. Numerical experiments. In this section we present results of numerical examples which, on the one hand, demonstrate the correctness of our formulas and, on the other hand, the difficulties connected with this problem.

Intentionally, we do not specify realistic physical properties, like material elastic modulus E or density ρ . In all examples, they are assumed to be equal to one. This can be done without any loss of generality or applicability due to the linear dependence of the volume, compliance, and stiffness matrices on a , E and ρ . Switching from our default values to realistic physical values (say, $E = 2.1 \cdot 10^{11}$ Pa and $\rho = 7.8 \cdot 10^3$ kg/m³) is a matter of a simple linear scaling of our results. Further, when we speak of an $i \times j$ truss, we have in mind a regular equidistant

grid of i times j nodes, i in the horizontal direction and j in the vertical direction. Thus the dimensions of the nodal grid are $(i - 1)(j - 1)$.

The truss topology software together with formulas for the subgradients were implemented in MATLAB. The SOCP problems (7) were defined using YALMIP [21] and solved by the conic solver MOSEK [27]. The nonsmooth optimization problem (9) was solved by the Bundle-Trust code BTNCLC [33] for nonsmooth nonconvex optimization with linear constraints. Stopping criterion for BTNCLC was set to $\varepsilon = 10^{-4}$.

The examples were solved on a notebook with Intel Core i7 CPU M620 2.67GHz running 64-bit Windows 7.

In all examples, we have first solved the dual SOCP problem (7) with a given load vector f^* to get an optimal design a^* . Then, by solving MPEC (8), we tried to reconstruct this design, starting with a random load vector. We did not apply any constraints on f , i.e., we set $\mathcal{F} = \mathbb{R}^n$ in the ImP-MPEC (9). Notice that, using Lemma 3.1, the optimal solution a of the NLP problem (3) (and thus SOCP (7)) is invariant to the scaling of the load vector f . This means that problem (8), without any additional constraints on f , has always infinitely many optimal solutions giving the same value of the objective function.

The ImP-MPEC optimization problem (9) is nonconvex, hence BTNCLC may converge to a local minimum or just to a stationary point. The solution obtained (whether local or global) and the iteration process depends significantly on the initial iteration. We have adopted three strategies for choosing the initial point f_{ini} :

INI1: random vector of sum one;

INI2: random vector of zero sum with norm $0.1\|f^*\|$ added to the original load f^* ;

INI3: random vector of sum one applied only at nodes with nonzero components of the original load f^* ; here we assume that we know the loaded nodes but not the actual load vector.

Comparison with a derivative-free method. Is the complicated computation of the subgradient information (and the theory behind) really necessary? After all, with $\mathcal{F} = \mathbb{R}^n$ the implicit programming problem (9) is a nonsmooth unconstrained optimization problem. We may thus try to solve it by an appropriate derivative-free optimization (DFO) method. In this the examples below, we thus present also results obtained by a DFO algorithm and compare them with the BTNCLC results. Our DFO method of choice was the universal Nelder-Mead “simplex” algorithm and its implementation by Higham [13], available in File Exchange by Mathworks as `nmsmax.m`. We have used the default settings of `nmsmax.m`, in particular, of the stopping criterion: the algorithm stopped when the simplex size shrank below 10^{-4} . We refer to this method as NMSMAX in the examples below.

For NMSMAX we set the INI3 starting point equal to zero. With the original INI3 MOSEK sometimes failed to solve the SOCP subproblem .

Example 2. Consider a 5-by-3 truss with all nodes connected by 74 potential bars. The bottom-corner nodes are fixed in both directions, and horizontal force $(0, -1)$ is applied at all remaining bottom-line nodes; see Figure 3-left. The upper bound on a was chosen as $\hat{a} = 0.25$. The optimal design a^* is shown in Figure 3-right.

Table 1 summarizes the results obtained by BTNCLC and NMSMAX from our three initial points. We present the computed optimal objective value c^* and, in the next four columns, number of iterations needed to reach the objective values 10^{-3} – 10^{-6} . The last column gives the number of iterations needed to reach the

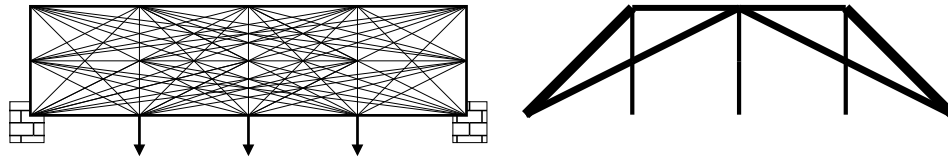


FIGURE 3. Five-by-three truss (Ex. 2): initial layout and optimal design

value c^* , i.e., the number of iterations needed to fulfill the stopping criterion of the respective algorithm. The superiority of BTNCLC is clearly demonstrated by these results.

TABLE 1. Results of Example 2. The last columns show the number of iterations needed to obtain the value of the objective function smaller than 10^{-3} – 10^{-6} and (the last column) to fulfill the stopping criterion.

method	INI	c^*	number of iterations to reach				c^*
			10^{-3}	10^{-4}	10^{-5}	10^{-6}	
BTNCLC	1	$2.4 \cdot 10^{-8}$	23	36	50	63	88
BTNCLC	2	$2.0 \cdot 10^{-8}$	6	15	25	43	64
BTNCLC	3	$7.6 \cdot 10^{-8}$	9	11	16	23	26
NMSMAX	1	$7.7 \cdot 10^{-6}$	547	1018	3631	–	6617
NMSMAX	2	$1.5 \cdot 10^{-4}$	694	–	–	–	4406
NMSMAX	3	$5.7 \cdot 10^{-6}$	33	66	106	–	192

Example 3. Consider a 5-by-5 truss with all nodes connected by 200 potential bars. The nodes on the left-hand side are fixed in both directions, and forces $(1, -0.333)$ and $(1, 0.333)$ are applied at the bottom-right and top-right nodes, respectively; see Figure 4-left. The upper bound on a is again $\hat{a} = 0.25$. The optimal design a^* is shown in Figure 4-right.

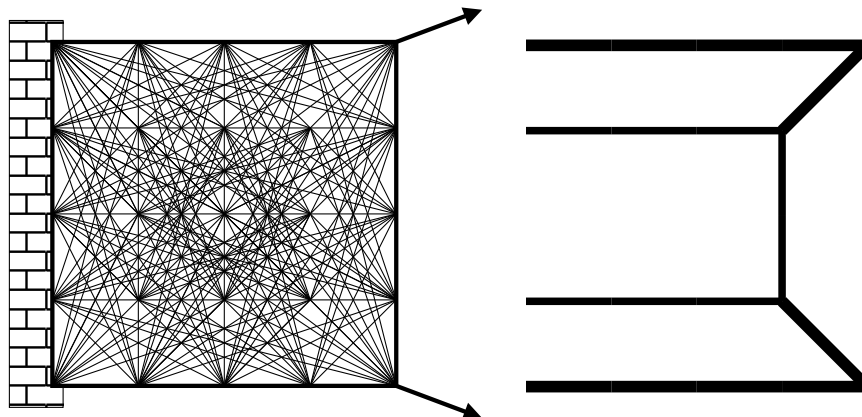


FIGURE 4. Five-by-five truss (Ex. 3): initial layout and optimal design

Table 2 presents the results in the same format as Table 1. We can see that, in case of INI3, BTNCLC converged to a local optimum with $c^* = 0.44$; the corresponding design is shown in Figure 5-right. The derivative-free code had serious problems in this example and only obtained an acceptable solution with INI3, however, in much bigger number of iterations than BTNCLC.

TABLE 2. Results of Example 3. The last columns show the number of iterations needed to obtain the value of the objective function smaller than 10^{-3} – 10^{-6} and (the last column) to fulfill the stopping criterion.

method	INI	c^*	number of iterations to reach				
			10^{-3}	10^{-4}	10^{-5}	10^{-6}	c^*
BTNCLC	1	$2.6 \cdot 10^{-7}$	67	92	111	313	143
BTNCLC	2	$4.8 \cdot 10^{-8}$	9	29	46	71	99
BTNCLC	3	$4.4 \cdot 10^{-1}$	–	–	–	–	11
NMSMAX	1	$5.3 \cdot 10^{-3}$	–	–	–	–	13487
NMSMAX	2	$1.7 \cdot 10^{-2}$	–	–	–	–	7850
NMSMAX	3	$2.0 \cdot 10^{-6}$	68	157	171	–	207

We reiterate that the load vector f leading to the same design vector a is not unique. Not only is the design a invariant with respect to scaling of f , it may differ substantially from the original given load f^* . An example is shown in Figure 5-left which presents the optimal solution obtained by BTNCLC from initial point INI1. As seen in Table 2 (and indeed in Figure 5 itself), this load vector leads to the same design as the original load f^* depicted in Figure 4.

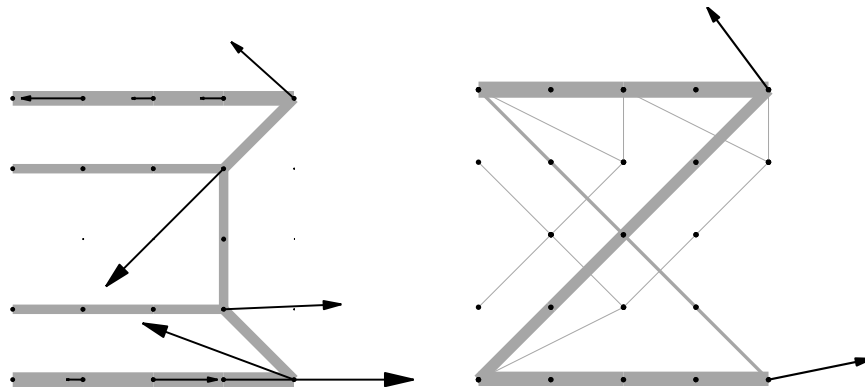


FIGURE 5. Five-by-five truss (Ex. 3): optimal load and corresponding design as computed by BTNCLC with INI1 (left) and INI3 (right).

Appendix. To compute vectors α, γ satisfying (26),(27), we make use of the well-known relationship

$$a \in N_{\mathcal{K}}(b) \Leftrightarrow b = \text{Proj}_{\mathcal{K}}(a + b), \tag{28}$$

which is valid for any closed convex set \mathcal{K} . This enables us to express the limiting coderivative of $N_{L^n}(\cdot)$ in terms of the limiting coderivative of $\text{Proj}_{L^n}(\cdot)$ and employ the results of [30]. Since the formulas used to compute an element of the limiting coderivative of the metric projection $\text{Proj}_{L^n}(\cdot)$ onto the second order cone L^n are rather complicated, we decided to give them in full detail in this Appendix.

For $x = (\hat{x}, x_n)$ with $\hat{x} \in \mathbb{R}^{n-1}$, $x_n \in \mathbb{R}$, we define the spectral decomposition

$$x = \lambda_1(x)c_1(x) + \lambda_2(x)c_2(x)$$

with

$$\lambda_1(x) = x_n + \|\hat{x}\|_2, \quad \lambda_2(x) = x_n - \|\hat{x}\|_2$$

and, for $x \neq 0$,

$$c_1(x) = \frac{1}{2} \begin{bmatrix} \frac{\hat{x}}{\|\hat{x}\|_2} \\ 1 \end{bmatrix}, \quad c_2(x) = \frac{1}{2} \begin{bmatrix} -\frac{\hat{x}}{\|\hat{x}\|_2} \\ 1 \end{bmatrix}.$$

If $x = 0$ then

$$c_1(x) = \frac{1}{2} \begin{bmatrix} w \\ 1 \end{bmatrix}, \quad c_2(x) = \frac{1}{2} \begin{bmatrix} -w \\ 1 \end{bmatrix},$$

with any $w \in \mathbb{R}^{n-1}$ such that $\|w\|_2 = 1$. The metric projection of x onto L^n can then be written as

$$\text{Proj}_{L^n}(x) = (\lambda_1(x))_+ c_1(x) + (\lambda_2(x))_+ c_2(x).$$

We further define the symmetric matrix of divided differences $\Psi(x) = \begin{bmatrix} \Psi_{11}(x) & \Psi_{12}(x) \\ \Psi_{12}(x) & \Psi_{22}(x) \end{bmatrix}$ by

$$\Psi_{ii}(x) = \begin{cases} 1 & \text{if } \lambda_i(x) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i = 1, 2$$

$$\Psi_{12}(x) = \frac{(\lambda_1(x))_+ - (\lambda_2(x))_+}{\lambda_1(x) - \lambda_2(x)}.$$

Assume that $x \notin L^n \cup -L^n$. Then we know ([20]) that the projection of x onto L^n is Fréchet differentiable at x and

$$\nabla \text{Proj}_{L^n}(x) = 2 \sum_{i=1}^2 \Psi_{ii}(x) c_i(x) (c_i(x))^\top + \Psi_{12}(x) \begin{bmatrix} I - \frac{\hat{x}\hat{x}^\top}{\|\hat{x}\|_2^2} & 0 \\ 0 & 0 \end{bmatrix}. \quad (29)$$

We now recall formula (25)

$$\alpha_i \in D^* N_{L^3}(A\bar{y}_i, \bar{\mu}_i)(Av_i) \quad \text{for } i = 1, \dots, m$$

and show its particular forms for various values of \bar{y} and $\bar{\mu}$. Using (28), we can write the above relation equivalently as

$$-Av_i \in D^* \text{Proj}_{L^3}(A\bar{y}_i + \bar{\mu}_i)(-Av_i - \alpha_i) \quad \text{for } i = 1, \dots, m, \quad (30)$$

see [29, Lemma 19].

We recognize three main cases in which Proj_{L^3} is differentiable and ∇Proj_{L^3} is symmetric:

Case 1. $A\bar{y}_i + \bar{\mu}_i \notin L^n \cup -L^n$. Then we can use (29) and relations (30) to obtain the form of equations in α_i, v_i :

$$\nabla \text{Proj}_{L^3}(A\bar{y}_i + \bar{\mu}_i)(Av_i + \alpha_i) - Av_i = 0 \quad \text{for } i = 1, \dots, m.$$

Case 2. $A\bar{y}_i \in \text{int}(L^n)$, $\bar{\mu}_i = 0$. Then we project an element from the interior of the cone and the derivative of the projection is trivially $\nabla \text{Proj}_{L^3}(A\bar{y}_i + \bar{\mu}_i) = I$, giving us the equations

$$\alpha_i = 0 \quad \text{for } i = 1, \dots, m.$$

Case 3. $A\bar{y}_i = 0$, $\bar{\mu}_i \in \text{int}(-L^n)$. Now we project an element from the interior of the polar cone, thus $\nabla \text{Proj}_{L^3}(A\bar{y}_i + \bar{\mu}_i) = 0$ and (30) gets the form

$$Av_i = 0 \quad \text{for } i = 1, \dots, m.$$

In the remaining three cases, the formulas for the limiting coderivatives of Proj_{L^3} are substantially more complicated; see [30, Thms. 3,4]. Nevertheless, for our purpose it suffices to make the following choices which keep the adjoint system (24)–(27) not too difficult to solve. In particular:

- if $A\bar{y}_i \in \text{Bd}(L^3) \setminus \{0\}$, $\bar{\mu}_i = 0$, proceed as in Case 1 or Case 2;
- if $A\bar{y}_i = 0$, $\bar{\mu}_i \in \text{Bd}(-L^3) \setminus \{0\}$, proceed as in Case 1 or Case 3;
- if $A\bar{y}_i = 0$, $\bar{\mu}_i = 0$, proceed as in any of Cases 1,2,3.

REFERENCES

- [1] W. Achtziger, M. Bendsøe, A. Ben-Tal and J. Zowe, [Equivalent displacement based formulations for maximum strength truss topology design](#), *IMPACT of Computing in Science and Engineering*, **4** (1992), 315–345.
- [2] M. Bendsøe and O. Sigmund, *Topology Optimization. Theory, Methods and Applications*, Springer-Verlag, Berlin, 2003.
- [3] P. Beremlijski, J. Haslinger, M. Kocvara, R. Kucera and J. V. Outrata, [Shape optimization in three-dimensional contact problems with coulomb friction](#), *SIAM Journal on Optimization*, **20** (2009), 416–444.
- [4] P. Beremlijski, J. Haslinger, J. V. Outrata and R. Pathó, [Shape optimization in contact problems with coulomb friction and a solution-dependent friction coefficient](#), *SIAM Journal on Control and Optimization*, **52** (2014), 3371–3400.
- [5] J.-F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer Series in Operations Research. Springer-Verlag, New York, 2000.
- [6] J. F. Bonnans and H. Ramírez, [Perturbation analysis of second-order cone programming problems](#), *Mathematical Programming*, **104** (2005), 205–227.
- [7] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal and C. A. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects*, Springer Science & Business Media, 2006.
- [8] A. L. Dontchev and R. T. Rockafellar, [Characterizations of strong regularity for variational inequalities over polyhedral convex sets](#), *SIAM Journal on Optimization*, **6** (1996), 1087–1105.
- [9] A. L. Dontchev and R. T. Rockafellar, *Implicit Functions and Solution Mappings*, Springer Monographs in Mathematics. Springer, Dordrecht, 2009.
- [10] D. Dorsch, W. Gómez and V. Shikhman, [Sufficient optimality conditions hold for almost all nonlinear semidefinite programs](#), *Mathematical Programming*, **158** (2016), 77–97.
- [11] G. Inc, *Gurobi Optimizer Reference Manual. Version 6.5*, 2016.
- [12] R. Henrion, A. Jourani and J. Outrata, [On the calmness of a class of multifunctions](#), *SIAM Journal on Optimization*, **13** (2002), 603–618.
- [13] N. J. Higham, [Optimization by direct search in matrix computations](#), *SIAM J. Matrix Anal. Appl.*, **14** (1993), 317–333.
- [14] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 2013.
- [15] F. Jarre, [Elementary optimality conditions for nonlinear SDPs](#), in *Handbook on Semidefinite, Conic and Polynomial Optimization* (eds. M. Anjos and J. Lasserre), Springer, **166** (2012), 455–470.

- [16] C. Kanzow and A. Schwartz, [Mathematical programs with equilibrium constraints: Enhanced fritz john-conditions, new constraint qualifications, and improved exact penalty results](#), *SIAM Journal on Optimization*, **20** (2010), 2730–2753.
- [17] M. Kočvara, M. Zibulevsky and J. Zowe, [Mechanical design problems with unilateral contact](#), *M2AN Mathematical Modelling and Numerical Analysis*, **32** (1998), 255–281.
- [18] M. Kočvara, M. Kružík and J. V. Outrata, [On the control of an evolutionary equilibrium in micromagnetics](#), in *Optimization with multivalued mappings*, Springer, **2** (2006), 143–168.
- [19] M. Kočvara and J. V. Outrata, [Optimization problems with equilibrium constraints and their numerical solution](#), *Mathematical Programming*, **101** (2004), 119–149.
- [20] A. Korányi, [Monotone functions on formally real Jordan algebras](#), *Mathematische Annalen*, **269** (1984), 73–76.
- [21] J. Löfberg, YALMIP : A toolbox for modeling and optimization in MATLAB, in *Proceedings of the 2004 IEEE International Symposium on Computer Aided Control Systems Design*, Taipei, Taiwan, 2004, 284–289.
- [22] Z.-Q. Luo, J.-S. Pang and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, 1996.
- [23] B. S. Mordukhovich, *Variational Analysis and Generalized Differentiation I: Basic Theory*, vol. 330, Springer Science & Business Media, 2006.
- [24] B. S. Mordukhovich, N. M. Nam and N. T. Yen Nhi, [Partial second-order subdifferentials in variational analysis and optimization](#), *Numerical Functional Analysis and Optimization*, **35** (2014), 1113–1151.
- [25] B. S. Mordukhovich and J. V. Outrata, [Coderivative analysis of quasi-variational inequalities with applications to stability and optimization](#), *SIAM Journal on Optimization*, **18** (2007), 389–412.
- [26] B. S. Mordukhovich and R. T. Rockafellar, [Second-order subdifferential calculus with applications to tilt stability in optimization](#), *SIAM Journal on Optimization*, **22** (2012), 953–986.
- [27] MOSEK ApS, *The MOSEK Optimization Toolbox for MATLAB Manual. Version 7.1 (Revision 28)*, 2015, URL <http://docs.mosek.com/7.1/toolbox/index.html>.
- [28] J. V. Outrata, M. Kočvara and J. Zowe, *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications and Numerical Results*, vol. 28, Springer Science & Business Media, 1998.
- [29] J. V. Outrata and H. Ramírez C, [On the Aubin property of critical points to perturbed second-order cone programs](#), *SIAM Journal on Optimization*, **21** (2011), 798–823.
- [30] J. V. Outrata and D. Sun, [On the coderivative of the projection operator onto the second-order cone](#), *Set-Valued Analysis*, **16** (2008), 999–1014.
- [31] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer, Berlin-Heidelberg, 1998.
- [32] H. Scheel and S. Scholtes, [Mathematical programs with equilibrium constraints: Stationarity, optimality and sensitivity](#), *Mathematics of Operations Research*, **25** (2000), 1–22.
- [33] H. Schramm and J. Zowe, [A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results](#), *SIAM Journal on Optimization*, **2** (1992), 121–152.

Received July 2016; revised December 2016.

E-mail address: m.kocvara@bham.ac.uk

E-mail address: outrata@utia.cas.cz