SOLUTION OF A TRUSS TOPOLOGY BILEVEL PROGRAMMING PROBLEM BY MEANS OF AN INEXACT RESTORATION METHOD.

ANA FRIEDLANDER AND FRANCISCO A. M. GOMES

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

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

ABSTRACT. We formulate a truss topology optimization problem as a bilevel programming problem and solve it by means of a line search type inexact restoration algorithm. We discuss details of the implementation and show results of numerical experiments.

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

Key words: Inexact Restoration – Bilevel Programming – Truss Topology Optimization.

1. INTRODUCTION

Bilevel programming problems model two-level hierarchical systems and have been studied since the seventies. In [4, 5, 6], history, applications, algorithms, theoretical questions and almost all relevant references can be found.

In this paper we address the following bilevel programming problem:

$$
(1)\qquad
\begin{array}{ll}
\max\limits_{x,s,u} & W(x,u) \\
\text{s.t.} & q(x,s)=0 \\
 & 0 \le x \le x_U \\
 & s \ge 0 \\
 & u = \arg\min\limits_{\bar{u}} \quad g(x,\bar{u}) \\
 & \qquad\qquad \text{s.t.} \quad u_L \le \bar{u} \le u_U.
\end{array}
$$

where $x \in \mathbb{R}^n$, $s \in \mathbb{R}$. u, $\bar{u} \in \mathbb{R}^m$, and W, q and g are real functions.

This work was supported by PRONEX-Optimization (PRONEX-CNPq/FAPERJ E-26/171.164/2003-APQ1), FAPESP (Grant 06/53768-0) and CNPq.

In [2] the authors studied the resolution of bilevel programmimg problems using the Inexact Restoration (IR) algorithm introduced in [10]. In this paper we follow the approach for the IR algorithm introduced in [7].

The present paper is organized as follows. In Section 2, we describe the IR algorithm and discuss its application for solving bilevel problems. In Section 3, we state and comment the truss topology optimization problem that will be solved. Since the resolution of bilevel problems requires specifications that are problem dependent, we discuss in section 4 how this IR algorithm can be adapted to our problems. In Section 5, we present numerical experiments and we state some conclusions in Section 6.

2. The line-search IR algorithm

A general nonlinear programming problem can be stated as

$$\begin{array}{rl} \underset{x}{\text{Minimize}} & f(z) \\ \text{s.t.} & C(z) = 0 \\ & l \leq z \leq u. \end{array} \tag{2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $C : \mathbb{R}^n \to \mathbb{R}^m$.

Inexact restoration algorithms for nonlinear programming (see [8, 10]) belong to the class of *feasible* algorithms because they emphasize the importance of the feasibility of an approximate solution. They were introduced to avoid the excessive computational effort that must be spent to achieve constraint fulfillment at iterations that are still very far from optimality, in problems with strongly nonlinear constraints.

In [7], the authors introduce a line search IR algorithm that captures the essential features of the IR technique. Moreover, its convergence theory is straightforward and allows for considerable freedom in the choice of implementation details.

The basic steps of the algorithm presented in [7] are shown below, assuming that $r \in [0,1)$, $\beta, \gamma, \tau > 0$ are fixed parameters, $h : [l,u] \to [0,\infty)$ is a function such that $\|C(z)\| \leq h(z)$, and $\Phi(z,p) = f(z) + (1-p)h(z)$.

Algorithm 1

- Step 0: *Initialization.*
  Choose $l \leq z^0 \leq u$ and $p_0 \in (0,1)$. Set $k := 0$.
- Step 1: *Inexact Restoration.*
  Compute $y^k \in [l,u]$ so that
  $$h(y^k) \leq rh(z^k),$$
  $$f(y^k) \leq f(z^k) + \beta h(z^k).$$
- Step 2: *Penalty parameter determination.*
  Determine $p_{k+1} \in \{2^{-i} p_k,\ i = 0,1,\ldots\}$ as large as possible so that
  $$\Phi(y^k, p_{k+1}) - \Phi(z^k, p_{k+1}) \leq \frac{1}{2}(1-r)(h(y^k) - h(z^k)).$$
- Step 3: *Computation of the search direction.*
  Compute d^k so that $y^k + d^k \in [l,u]$ and the following inequalities hold for $t \in [0,\tau]$,
  $$f(y^k + td^k) \leq f(y^k) - \gamma t \|d^k\|^2$$
  $$h(y^k + td^k) \leq h(y^k) + \gamma t^2 \|d^k\|^2$$
- Step 4: *Line search.*
  Determine $t_k \in \{2^{-i} i = 0,1,\ldots\}$ as large as possible so that the first inequality in Step 3 holds and
  $$\Phi(y^k + t_k d^k, p_{k+1}) - \Phi(z^k, p_{k+1}) \leq \frac{1}{2}(1-r)(h(y^k) - h(z^k)).$$
- Step 5: *Step update*
  Set $z^{k+1} := y^k + td^k$ and $k := k+1$. Go to step 1.

The equality constraints that define the feasible region of the bilevel problem (1) are given by

$$\begin{aligned} & q(x,s) = 0 \\ & u = \arg\min_{\bar{u}} \quad g(x,\bar{u}) \\ & \qquad\quad \text{s.t.} \quad u_L \leq \bar{u} \leq u_U. \end{aligned} \tag{3}$$

At the inexact restoration step, we need to determine a point that is more feasible than $(x,s,u)^T$. For $q(x,s)$, this is clearly stated in step 1 of the algorithm. However, for the low level problem

$$
u = \arg\min_{\bar{u}} \quad g(x,\bar{u}) \qquad \text{s.t.} \quad u_L \leq \bar{u} \leq u_U, \tag{4}
$$

we have to explain carefully what we mean by a better value of u.

At the beginning of iteration k, the point $z^k = (x^k, s^k, u^k, \mu_L^k, \mu_U^k)^T$ is available, where μ_L^k and μ_U^k are estimates of the Lagrange multipliers associated with the bounds on $\bar{u}$. For a fixed value of the first two variables, we can use u^k as an initial point and apply an adequate minimization algorithm to solve problem (4). The choice of the algorithm is problem dependent. For the truss topology optimization problem presented in Section 3, we discuss this issue in Section 4.

The inexact restoration step of the algorithm does not require full feasibility. Instead, a sufficient decrease of the feasibility error measured by a majorant of the norm of the constraint vector $C(z)$ given in (2) is required. In the definition of $C(z)$, the infeasibility related to (4) is measured by equalities in the KKT optimality conditions of this minimization problem, i.e.

$$
C(z) = C(x,s,u,\mu_L,\mu_U) = \begin{pmatrix} q(x,s) \\ \nabla g_u(x,s,u) - \mu_L + \mu_U \\ (u-u_L)\mu_L \\ (u_U-u)\mu_U \end{pmatrix}. \tag{5}
$$

The bound constraints and the non-negativeness of the Lagrange multipliers are forced at each iteration. Other details of the algorithm, such as the choice of h, the search direction in Step 3 and the line search in Step 4 are problem dependent and will be presented in Section 4.

3. The truss topology problem

The simplest truss topology optimization problem consists in finding the stiffest truss for a given volume. Various formulations of this problem can be found in [1] and [3]. More sophisticated problems can be generated if, for example, we include upper limits on the displacements and on the stresses, or multiple loads.

In this work, we consider the optimization of a truss subject to external loads and frictionless contact supports. We restrict our analysis to a rectangular bidimensional domain discretized into a mesh with $n_x \times n_y$ nodes. The truss bars are generated connecting nodes from this mesh. The set of potentials bars is called the *ground structure*. The size of the ground structure depends on the *connectivity level* adopted, that determines to which other nodes each node can be connected.

If a node can only be connected to its immediate neighbors, the connectivity level, hereafter denoted n_p, is set to one. If a node can also be connected to the nodes in the neighborhood of its immediate neighbors, $n_p = 2$. Generalizing this idea, n_p can assume any value between 1 and the maximum of $\{n_x - 1, n_y - 1\}$. In the latter case, the ground structure is said to be fully connected. Figure 1 shows the bars of a ground structure that connect one node to its neighbors in the NE quadrant for $n_p = 4$.

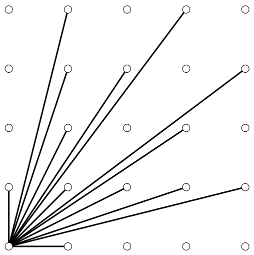

FIGURE 1. The bars that connect a node to its northeast neighbors, for $n_p = 4$.

Once n_p is chosen, the number of potential bars of the structure is given by

$$\begin{aligned} n_b &= 4n_xn_y - 3n_x - 3n_y + 2 + \\ &\sum_{p=1}^{n_p} \left[4r_pn_xn_y - 2pr_p(n_x + n_y) + 2(2p - n_x - n_y) \sum_{i=1}^{r_p} k_i^p \right], \end{aligned}$$

where r_p is the number of integers between 1 and $p - 1$ that are relatively prime to p, and k_i^p is the i-th of these numbers.

Table 1 shows the growth of the number of bars as a function of the number of nodes and the connectivity level, for a square structure. The number of bars of a fully connected base structure with $n_x = n_y = 100$ is 30398894. Thus, for highly discretized structures, it is important to keep n_p small.

TABLE 1. The number of bars as a function of n_x and n_p, for a square structure.

n_x	n_p				
	1	2	3	4	5
10	342	630	1106	1490	2090
100	39402	78210	154646	229910	378110
1000	3994002	7982010	15946046	23898110	39778310

3.1. **The truss topology problem with frictionless contact as a bilevel programming problem.**

In the topology optimization of a truss formed by m potential bars that link n potential nodes, we consider two sets of variables: the vector of cross sectional areas of the bars, $x \in \mathbb{R}^m$, and the vector of horizontal and vertical displacements of the nodes, $u \in \mathbb{R}^{2n}$. The objective is to maximize the stiffness of the structure, i.e. minimize the compliance, subject to volume and contact constraints.

The truss topology optimization problem with frictionless contact conditions is stated as

$$\begin{array}{lll} \max\limits_{x,s,u} & W(x,u) = \frac{1}{2}u^T K(x)u - P^T u & \\ \text{s.t.} & \ell^T x + s = V & \\ & 0 \leq x \leq x_U & \\ & s \geq 0 & \\ & u = \arg\min\limits_{\bar{u}} & \frac{1}{2}\bar{u}^T K(x)\bar{u} - P^T \bar{u} \\ & \qquad\qquad \text{s.t.} & u_L \leq \bar{u} \leq u_U . \end{array} \tag{6}$$

where the vector ℓ contains the lengths of the bars, V is the upper limit for the structure volume, s is the slack variable of the volume constraint, x_U is the vector of upper limits for the cross sectional areas of the bars, u_L and u_U are the lower and upper limits for the nodal displacements and P is the vector of nodal forces. The global stiffness matrix $K(x)$ is given by $K(x) = \sum_{j=1}^{m} K_j(x)$, where $K_j(x) = (\frac{E}{\ell_j})x_j b_j b_j^T$ and b_j is the j-th column of the compatibility matrix B that relates the nodal forces to the bar forces.

The optimal solution of (6) contains only a few of the ground structure bars. Therefore many components of vector x are expected to vanish near the optimum. However, to avoid numerical difficulties that arise when the cross sectional areas of the truss bars are allowed to become zero, it is a common practice to define a positive lower level x_{min} for these areas. Following this practice, we also require that $x_i \in [x_{min}, x_{max}]$, for $i = 1, \ldots, m$.

4. THE ALGORITHM OF INEXACT RESTORATION FOR TRUSS OPTIMIZATION WITH FRICTIONLESS CONTACT

The crucial steps of Algorithm 1 are the inexact restoration, the definition of the search direction and the line search. In this section, we show how these steps may be efficiently computed for the bilevel problem (6).

4.1. **The inexact restoration step.**

An approximate solution z for (6) is said to be feasible if

$$\begin{aligned} C(z) = 0, \\ z \in \Omega \end{aligned} \tag{7}$$

where $z = (x, s, u, \mu_L, \mu_U)^T$, μ_L and μ_U are the Lagrange multipliers of the lower level problem in (6), s is the slack variable of the volume constraint,

$$C(z) = \begin{bmatrix} \ell^T x + s - V \\ K(x)u - P - \mu_L + \mu_U \\ M_L(u - u_L) \\ M_U(u_U - u) \end{bmatrix},$$

and

$$\Omega = z \in \mathbb{R}^{m+1+6n} \mid s, \mu_L, \mu_U \geq 0,\ 0 \leq x \leq x_U,\ u_L \leq u \leq u_U\}.$$

Here, M_L and M_U are the diagonal matrices containing μ_L and μ_U, respectively. One may notice that the first component of $C(z)$ is the volume constraint, while the remaining components are the equalities in the KKT conditions of the lower level problem.

In the inexact restoration phase, we seek a solution that satisfies the conditions stated at Step 1 of the algorithm. In [7] the authors suggest that this can be accomplished defining

$$h(z) = \|C(z)\| + \sqrt{\mu_l^T(u-u_L) + \mu_u^T(u_U-u)} \tag{8}$$

and requiring the restored point at iteration k, y^k to satisfy $h(y^k) \le rh(z^k)$ and

$$\|y^k - z^k\| \le \beta h(z^k).$$

The bounds on the variables defined by Ω and the linear volume constraint are forced at each iteration. Therefore, the components x^k and s^k of the restored step y^k take the same values as in z^k.This implies that the restoration is restricted to find a vector of nodal displacements u^R that solves the problem

$$\begin{array}{ll} \min\limits_{u} & \frac{1}{2}u^T K(x^k)u - P^T u \\ \text{s.t.} & u_L \le u \le u_U \\ & \|u-u^k\|_\infty \le \beta h(z^k) \end{array} \tag{9}$$

After obtaining u^R, the Lagrange multipliers are estimated and forced to verify $\|\mu_L^R - \mu_L^k\|_\infty \le \beta h(z^k)$ and $\|\mu_U^R - \mu_U^k\|_\infty \le \beta h(z^k)$.

If we define a positive lower bound for the cross sectional areas of the bars, (9) turns out to be a box constrained strictly convex quadratic programming problem that can be easily solved.

4.2. **The search direction.** Algorithm 1 requires the search direction $d^k = (d_x, d_s, d_u, d_{\mu_L}, d_{\mu_U})$ to satisfy

$$f(y^k + td^k) \le f(y^k) - \gamma t\|d^k\|^2 \tag{10}$$

and

$$h(y^k + td^k) \le h(y^k) + \gamma t^2\|d^k\|^2. \tag{11}$$

In [7], the authors show that this can be obtained if d^k is the projection of $-\nabla W(x^k, u^R)$ onto the set of tangent directions

$$\mathscr{T} = \{d \in \mathbb{R}^{m+1+6n} \mid \nabla C(y^k)^T d = 0,\, y^k + d \in \Omega\}.$$

In this case, d^k is the solution of the quadratic programming problem

$$\begin{array}{ll} \min & \frac{1}{2} d^T d + \bar{c}_k^T d \\ \text{s.t.} & \nabla C(y^k) d = 0 \\ & y^k + d \in \Omega \end{array} \tag{12}$$

where

$$\bar{c}_k^T = [\nabla_x W(x^R, u^R)^T \;\; 0 \;\; (K(x^R)u^R + P)^T \;\; 0 \;\; 0],$$

$$\nabla C(y^k) = \left[\begin{array}{ccccc} \ell^T & 1 & 0 & 0 & 0 \\ \nabla^2_{xu} W(x^k, u^R) & 0 & K(x^k) & -I & I \\ 0 & 0 & M_L & (U^R - U_L) & 0 \\ 0 & 0 & -M_U & 0 & (U_U - u^R) \end{array} \right],$$

and U^R, U_L and U_U are diagonal matrices containing u^R, u_L and u_U, respectively.

However, defining d^k this way may drastically reduce the convergence rate of the algorithm. Therefore, in our algorithm the objective function of (12) is replaced by the quadratic approximation of $W(x^k + d_x, u^R + d_u)$, i.e.

$$q(d) = \frac{1}{2}(y^k + d)^T B(y^k)(y^k + d) + \bar{c}_k^T d + W(x^k, u^R),$$

where

$$B(y^k) = \left[\begin{array}{ccccc} 0 & 0 & -\nabla^2_{xu} W(x^k, u^R)^T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\nabla^2_{xu} W(x^k, u^R) & 0 & -K(x^k) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right],$$

Since the solution of this problem may not satisfy (11), we include a trust region constraint, so the step is recomputed whenever it fails to verify this condition. Thus, our search direction is the solution of

$$
\begin{array}{ll}
\min & \frac{1}{2}(y^k+d)^T B(y^k)(y^k+d)+\bar{c}_k^T d \\
\text{s.t.} & \nabla C(y^k)d=0 \\
& y^k+d\in\Omega \\
& \|d\|_\infty\leq\Delta_k
\end{array}
\tag{13}
$$

With this strategy all the convergence results are essentially preserved as shown in [9].

The solution of (13) may be obtained using any quadratic programming library that works with sparse matrices. However, if the routine requires the decomposition of a matrix formed of some or all of the columns of $\nabla C(y^k)$ or $\nabla C(y^k)^T\nabla C(y^k)$, it is better to reorder the variables in a way that the factors generated by the decomposition will remain very sparse. A good reordering can be obtained if we replace d^k by $\bar{d^k}=(d_{\mu_L},d_{\mu_U},d_u,d_s,d_x)$. Moreover, a good node ordering should also be used, in order to reduce the fill-in produced by the factorization of $K(x^k)$.

Following the usual approach of IR algorithms (see [8] and [10], for example), we also tried to define d^k as the solution of the nonlinear programming problem that consists in minimizing the original objective function subject to a linearization of the constraints, i.e.

$$
\begin{array}{ll}
\max & \frac{1}{2}(u^R+d_u)^T K(x^k+d_x)(u^R+d_u)-P^T(u^R+d_u) \\
\text{s.t.} & \nabla C(y^k)d=0 \\
& y^k+d\in\Omega \\
& \|d\|_\infty\leq\Delta_k
\end{array}
\tag{14}
$$

However, the direction generated by this problem, besides being more difficult to obtain, did not show a better performance than the one computed solving (13).

4.3. **The line search.** The backtracking scheme adopted in Algorithm 1 seems to be quite simple to code. However, to compute a step length that satisfies the condition stated at Step 4 of the algorithm, we need to evaluate the objective function at some intermediate points $(x_+,u_+)=(x^k+\lambda d_x,u^R+\lambda d_u)$. This task may be costly if a naive approach is adopted, since it requires the computation of the stiffness matrix $K(x_+)$. Fortunately, K varies linearly with x, so we may write

$$K(x_+) = K(x^k) + \lambda K(d_x). \tag{15}$$

Thus, once we have computed $K(d_x)$, we can avoid building K from scratch each time λ is changed. Moreover, we can also reduce the cost of obtaining $W_+ = W(x_+, u_+)$ if we write

$$\begin{aligned} W_+ &= \frac{1}{2}(u^R + \lambda d_u)^T (K(x^k) + \lambda K(d_x))(u^R + \lambda d_u) + P^T (u^R + \lambda d_u) \\ &= \frac{1}{2} u_+^T [K(x^k)u^R + \lambda (K(x^k)d_u + K(d_x)u^R + \lambda K(d_x)d_u)] + P^T u_+ \\ &= \frac{1}{2} u_+^T [v_1 + \lambda (v_2 + \lambda v_3)] + P^T u_+. \end{aligned}$$

Vector $v_1 = K(x^k)u^R$ is already available, since it was used to calculate $W(x^k, u^R)$. Therefore, after computing vectors u_+, $v_2 = K(x^k)d_u + K(d_x)u^R$ and $v_3 = K(d_x)d_u$, the objective function value may be obtained with two saxpys and two inner products.

After determining the optimal value for λ, the stiffness matrix may also be updated using (15). To keep the roundoff errors under control, K should be recomputed from scratch once in a while. In our algorithm, this recalculation is done at each 50 iterations.

If $\lambda = 1$ satisfies the condition defined in Step 4, then the approach presented here is more expensive than computing $K(x_+)$ directly. However, usually the step length needs to be reduced, so it pays for computing three extra matrix-vector products and a matrix update.

Finally, it must be noticed that due to the linearity of the volume constraint, x_+ will always satisfy this constraint whenever it is satisfied by x^k, independently of the value of λ. This property of the problem allows us to perform an inexact restoration based only on the nodal displacements.

5. NUMERICAL RESULTS

In this section, we show the results obtained using the algorithm for nonlinear bilevel problems presented above to optimize the topology of some plane trusses supported by a frictionless contact foundation.

In all of the examples, the material used in the trusses has elasticity modulus $E = 200000 N/mm^2$. The cross sectional areas of the bars are required to belong to the interval $[0.00001mm^2,\ 4mm^2]$. The parameters used in the inexact restoration step are $r = 0.5$ and $\beta = 10.0$.

5.1. **First problem.** The ground structure of our first problem is shown in Figure 2. Due to symmetry, only the right half of the frame is considered. The structure has 668 potential bars and 189 potential nodes. The connectivity level is one.

Only one external vertical force of magnitude $10N$ is applied at node 9. The volume is limited to 2% of the ground structure's volume. The foundation is not parallel to the bottom line of the structure. In fact, the gap between node 91 and the foundation is equal to $10mm$, and this distance is linearly reduced to $0mm$ at node 181 (the bottom right node). It must be noticed that no fixed supports are defined for the structure, but only the frictionless foundation that limits downward node displacements.

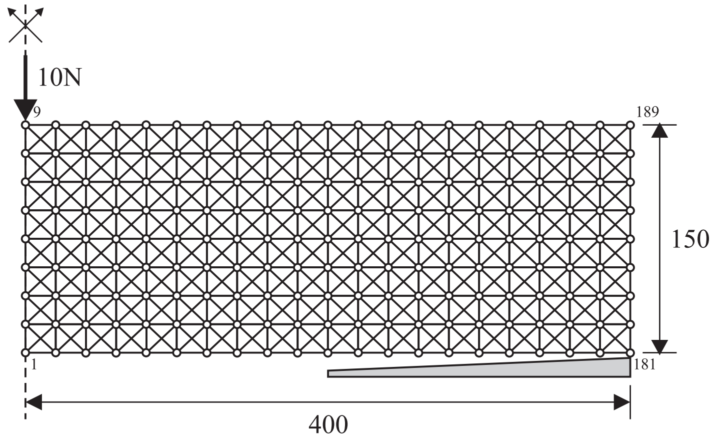

FIGURE 2. First problem.

The solution of the first problem is shown in Figure 3. The truss obtained resembles the MBB beam that is usually obtained solving the topology optimization problem where the lower right corner of the structure is fixed.

Increasing the connectivity level to 2, we obtain the structure shown in Figure 4.

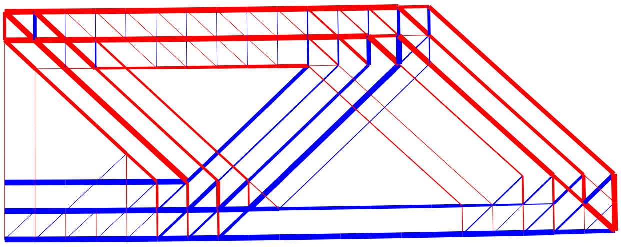

FIGURE 3. Solution of the first problem for $n_p = 1$.

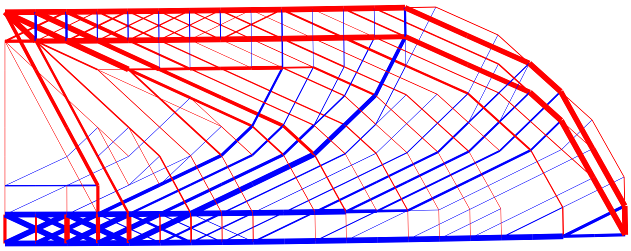

FIGURE 4. Solution of the first problem for $n_p = 2$.

5.2. **Second problem.** The domain of the second problem is the same of the first problem. However, an external vertical force of magnitude $1N$ is applied at node 1, instead of at node 9. The volume is limited to 2% of the ground structure's volume. Also, the frictionless foundation is flat and covers only one fourth of the ground structure basis.

A schematic representation of the problem is presented in Figure 5. The solution obtained after applying Algorithm 1 is shown in Figure 6. Again, the truss obtained is similar to the usual MBB beam with a fixed lower right node. However, the structure is shorter, due to the presence of the flat foundation.

5.3. **Third problem.** In the third problem, there are two external forces applied at the lower corners of the structure. A small frictionless foundation prevents the downward displacement of one single node of the structure's basis. The volume is limited to 0,05% of the ground structure's volume. The connectivity level is set to 2.

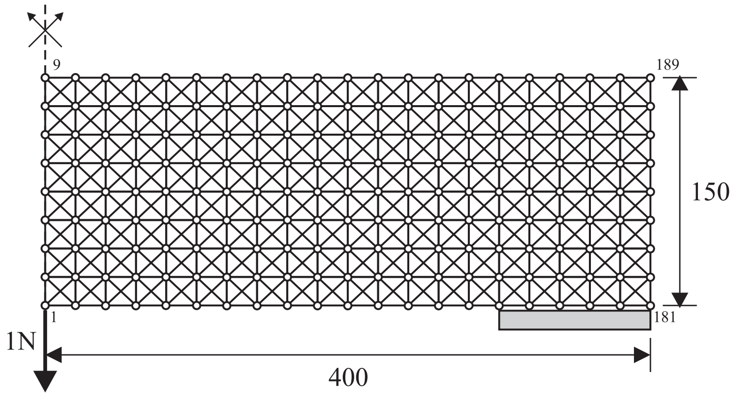

FIGURE 5. Second problem.

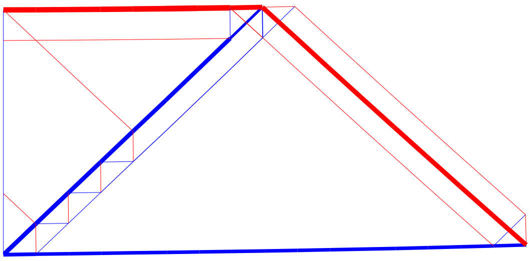

FIGURE 6. Solution of the second problem.

The problem is shown in Figure 7 and the solution obtained after applying Algorithm 1 is given in Figure 8. As we can see, the upper right border of the structure is curved, and is linked to the node that is in contact with the foundation by a group of thin bars.

It must be noticed that, due to the less stringent boundary conditions adopted for the problems presented in this section, the stiffness matrix is always singular. Thus, these problems are harder to solve than usual truss topology problems. Even though, the structures shown in Figures 3, 4, 6 and 8 are quite reasonable.

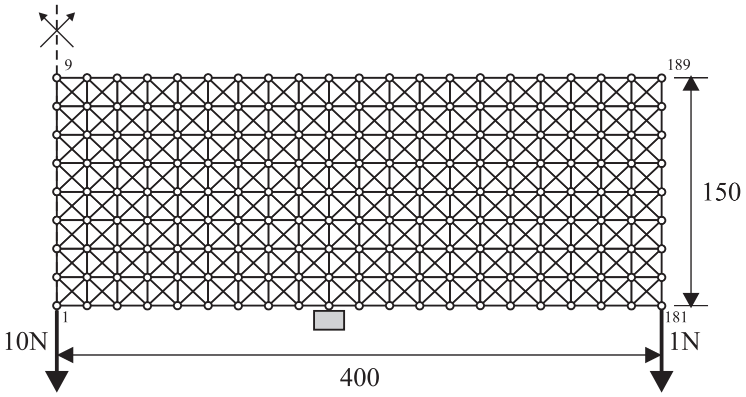

FIGURE 7. Third problem.

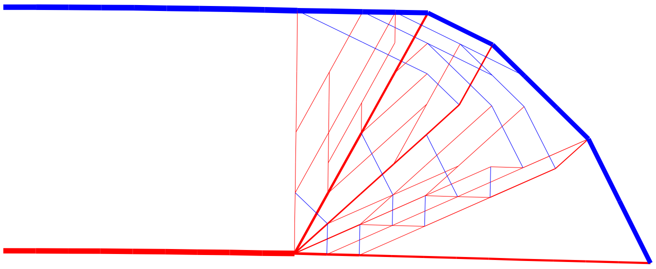

FIGURE 8. Solution of the third problem.

6. CONCLUSIONS

The IR approach for solving bilevel problems allows for a lot of freedom in the choice of the method used to solve the lower level optimization problem. For particular problems, some special nonlinear programming algorithms may be devised. In this paper, we illustrate this fact dealing with a simple truss topology optimization problem.

We do not claim that the lower level strategy presented in Section 4 is better than others. Another strategy can be used in this framework. It is worth to mention that the feasibility of the iterates can be explicitly controlled at all iterations. We do not need to solve for feasibility when we are far from optimality. On the other hand, when we judge that we are sufficiently near to the optimal value, we may solve the problems in Step 1 with stronger stopping criteria by taking smaller values of the parameter r. This may be important in practice if these constraints need to be satisfied at an approximate solution.

With our approach, we were able to solve truss topology optimization problems with singular stiffness matrices and frictionless contact conditions. Besides, it would be easy to include other constraints on the bar volumes, on the displacements or even on the stresses. It would also be possible to consider other two level formulations of the truss topology problem, such as the one presented in [11].

REFERENCES

[1] W. Achtziger, On simultaneous optimization of truss geometry and topology. Strucutral and Multidisciplinary Optimization 33, 285–304, 2007.

[2] Andreani, R., Castro, S.L.C., Chela, J., Friedlander, A., Santos, S.A., An inexact-restoration method for nonlinear bilevel programming problems, Comput. Optim. Appl. Comp. 43, 307–328, 2009.

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

[4] S. Dempe, *Foundations of Bilevel Programming*, Nonconvex Optimization and its Applications Series, 61, Kluwer Academic Publishers, The Netherlands, 2002.

[5] S. Dempe, Annottated Bibliography on Bilevel Programming and Mathematical Problems with Equilibrium Constraints", Optimization, 52, 333–359, 2003.

[6] S. Dempe, Bilevel Programming - A survey, Technical Report TU 2003-11, Bergakademie Freiberg, 2003.

[7] Fischer A, Friedlander A, A new line search inexact restoration approach for nonlinear programming, Comp. Optim. Appl. 46, 333-346, 2010.

[8] Gonzaga, C.C., Karas, E.W., Vanti, M. A globally convergent filter method for nonlinear programming. SIAM J. Optim. 14, 646–669, 2003.

[9] G. Haeser, Condies sequenciais de otimalidade, Doctoral Thesis, Unicamp 2009. Available for download in www.sjc.unifesp.br/docente/haeser/.

[10] J. M. Martínez, Inexact-Restoration Method with Lagrangian Tangent Decrease and New Merit Function for Nonlinear Programming, Journal of Optimization Theory and Applications, 111(1), 39–58, 2001.

[11] M. Kočvara, J. V. Outrata, Effective reformulations of the truss topology design problem, Optimization and Engineering 7, 201–219, 2006.