Optim Eng (2008) 9: 273–290
DOI 10.1007/s11081-007-9034-y

Cost minimization of a multiple section power cable supplying several remote telecom equipment

Joaquim J. Júdice · Victor Anunciada · Carlos P. Baptista

Received: 30 January 2006 / Accepted: 20 November 2007 / Published online: 5 December 2007

Abstract An optimization problem is described, that arises in telecommunications and is associated with multiple cross-sections of a single power cable used to supply remote telecom equipments. The problem consists of minimizing the volume of copper material used in the cables and consequently the total cable cost. Two main formulations for the problem are introduced and some properties of the functions and constraints involved are presented. In particular it is shown that the optimization problems are convex and have a unique optimal solution. A Projected Gradient algorithm is proposed for finding the global minimum of the optimization problem, taking advantage of the particular structure of the second formulation. An analysis of the performance of the algorithm for given real-life problems is also presented.

Keywords Nonlinear optimization · Convex programming · Energy systems · Telecommunications

V. Anunciada passed away on September 29, 2007.

J.J. Júdice (✉)
Departamento de Matemática, Universidade de Coimbra and Instituto de Telecomunicações, Coimbra, Portugal
e-mail: Joaquim.Judice@co.it.pt

V. Anunciada
Instituto Superior Técnico, UTL and Instituto de Telecomunicações, Lisboa, Portugal
e-mail: avaa@lx.it.pt

C.P. Baptista
Escola Superior de Tecnologia, Instituto Politécnico de Tomar, Tomar, Portugal
e-mail: Carlos.Perquilhas@aim.estt.ipt.pt

1 Introduction

This work describes the formulation and resolution of an engineering optimization problem that arose in a project of the *Instituto de Telecomunicações* (Portuguese research institute for telecommunications) with the company *Portugal Telecom* (Anunciada 2003). The problem consists of optimizing the multiple cross-sections of a single power cable used to supply remote telecom equipment.

Electric power transmission is usually operated at an almost constant voltage. In fact, electric appliances and equipment require a constant supply voltage, named the *nominal operating voltage*. In alternating current (AC) distribution networks, the nominal operating voltage is 230 V in Europe and 110 V in USA. Those values are the root mean square value of a sinusoidal voltage that is a function of time and frequency (frequencies of 50 Hz in Europe and 60 Hz in USA).

Electric appliances may accept only narrow variations of those voltages, typically tolerances of $+10\%$, -15%. If an electrical generator operates at a fixed voltage V and a distribution line with several nodes, where loads are connected, there are voltage drops along the line, described by Ohm's law (the voltage drop is proportional to the line impedance and line current, and line impedance is proportional to line length). However, when currents are at a maximum value in any one of the nodes, the voltage must be kept in the interval of acceptable voltages. Consequently, power distribution is made with oversized copper or aluminum cables. The maximum length of the distribution cables is limited, in order to avoid excessive voltage drops or excessive cross sections of the cables. Usually the cable length is limited to be less than 2 km. The cables cross sections are standardized considering the maximum current in the system and the maximum allowed voltage drops.

For larger distances, the distribution networks use transformers in order to increase operating voltage to 10 kV or 15 kV, reducing the line currents in the same proportion. For very long distances or large power, the voltage increases up to 440 kV or more.

Power distribution operates at an almost constant voltage. Consequently, power lines are interconnected with transformers in order to keep acceptable sizes of the cables. Industry has standardized cables, maximum distances and operating voltages in order to obtain acceptable values for the system cost and voltage drops.

A different situation occurs when a small amount of energy is required very far from any possible supply point. This happens on a motorway, far from any town where it is required to power a video camera and the radio transmitter. The use of optic fibers in long distance telecommunications implies the use of electronic signal conditioning equipment placed along the fibers, in places that can be more than 100 km from the nearest power supply. It is possible to construct a copper cable for energy supply along the fiber cable. However, such long distances imply very large voltage drops or bulky and costly cables, assuming a constant voltage operation. In the above mentioned project, the authors considered the possibility of having very large voltage drops in the cable, in order to avoid large cable costs. This option is possible because telecom equipment require voltage regulators and storage batteries that can operate with voltage tolerances up to 50%. When the power cable supplies several devices in different places along the cable, the cross section of each part of the cable should be optimized.

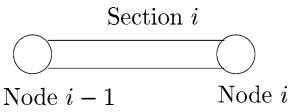
As shown in this paper, the model under consideration can be formulated as a nonlinear programming problem. We are able to show that the constraint set of this program is bounded and the objective function is strictly convex on this constraint set. Hence the optimization problem has a unique optimal solution, which can easily be found by a nonlinear program code, such as MINOS (Murtagh and Saunders 1987). By exploiting the structure of this optimization problem, it is also possible to reformulate it as a strictly convex nonlinear program on a simplex. This latter optimization problem can be efficiently processed by a Projected Gradient algorithm (Bertsekas 1976, 1999; Birgin et al. 2000) that fully exploits the structure of the constraint set. Computational experience with small instances of the problem illustrates the validity of the formulation for its purpose and of the Projected Gradient algorithm to process the associated nonlinear program.

The organization of the paper is as follows. In Sect. 2 the model and its formulation are introduced. A heuristic procedure used in Anunciada (2003) is reviewed in Sect. 3 to find a feasible solution to the associated optimization problem. The solution of the nonlinear program under consideration is fully discussed in Sect. 4. The alternative formulation with simplex constraints and the Projected Gradient algorithm for its solution are described in Sects. 5 and 6. Finally some computational experience and conclusions are reported.

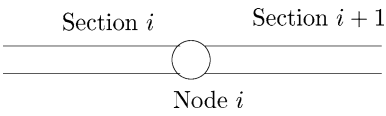
2 Model description

A long cable with two conducting wires is supplied at one end by a constant voltage generator with a known voltage v_I. The cable is constituted by n sections. The cross-sections of the conductors may differ from section to section (although in each section of the cable the cross-section of both conducting wires should be the same). If l_i is the length of each section i $(i = 1, 2, \ldots, n)$, then the total length of the cable is $\sum_{i=1}^{n} l_i$.

Besides the voltage generator and the n sections, we assume the existence of n nodes along the cable, each representing an electrical load. Section i connects two consecutive nodes $i - 1$ and i, thus



where $i = 1, 2, \ldots, n$ (and node 0 coincides with the voltage generator). Therefore node i connects two consecutive sections i and $i + 1$:



where $i = 1, 2, \ldots, n - 1$, with the exception of node n.

The load in each node i is known and is characterized by its constant power load p_i. The load current l_i^c in each node i depends on the voltage v_i in that node, accord-

ing to

$$l_i^c = \frac{p_i}{v_i}, \quad i = 1, 2, \ldots, n. \tag{1}$$

The voltage decrease Δv_i along each section i ($\Delta v_i = v_{i-1} - v_i$) is, according to Ohm's law (Feynman et al. 1963),

$$\Delta v_i = v_{i-1} - v_i = r_i c_i, \tag{2}$$

where r_i and c_i are the resistance of the cable and the current in section i, $i = 1, 2, \ldots, n$.

As the cable is made of two conducting wires, the resistance of each cable section is twice the resistance of each conductor in that section. Hence

$$r_i = 2\rho \frac{l_i}{s_i}, \tag{3}$$

where ρ is the material resistivity, l_i is the length of the section, and s_i is the area of the cross section.

The current c_i in section i is the sum of the currents in the loads at nodes i, $i = 1, \ldots, n$

$$c_i = \sum_{j=i}^{n} l_j^c = \sum_{j=i}^{n} \frac{p_j}{v_j}. \tag{4}$$

Consequently the current in section i ($i = 1, 2, \ldots, n$) depends on the voltage in all later nodes of this section, as the power loads p_i are constant. Now, using the expressions of r_i and s_i in (3) and (4), we obtain from (2)

$$v_{i-1} - v_i = 2\rho \frac{l_i}{s_i} \sum_{j=i}^{n} \frac{p_j}{v_j}. \tag{5}$$

Therefore

$$v_i = v_{i-1} - 2\rho \frac{l_i}{s_i} \sum_{j=i}^{n} \frac{p_j}{v_j} \tag{6}$$

for $i = 1, 2, \ldots, n$.

Another data item is the voltage v_n in the last node n of the cable. Considering that the voltage v_0 of node 0 coincides with the initial voltage v_I, then

$$v_n = a v_0, \tag{7}$$

where a is a positive given constant satisfying $\frac{1}{2} \leq a < 1$.[1]

[1] It is possible to show that if $a < \frac{1}{2}$, the total volume of the cable increases as the voltage v_n decreases (DeCarlo 2001).

The volume of each section i of the cable is equal to $2l_i s_i$. Therefore, the total volume of cable material is

$$V = \sum_{i=1}^{n} 2l_i s_i. \tag{8}$$

As the price of the cable is almost proportional to its volume, its total cost C is given by $C = cV$, where c is a given positive constant

$$C = c \sum_{i=1}^{n} 2l_i s_i = \sum_{i=1}^{n} 2cl_i s_i. \tag{9}$$

The objective of the problem is to determine the cross-section area s_i of the conducting wires in each section that minimizes the total cost of the cable. This is equivalent to minimizing its volume V. The values p_i and l_i, $i = 1, 2, \ldots, n$, as well as v_0, v_n, a, ρ and c are data of the optimization problem, and we have to find the cross-section areas $s_i, i = 1, 2, \ldots, n$, that minimize C (or equivalently V). The problem also contains the variables $v_1, v_2, \ldots, v_{n-1}$, which are related to $s_i, i = 1, 2, \ldots, n$ by (6).

The model described can then be formulated as the following nonlinear optimization problem:

$$\text{Minimize } \sum_{i=1}^{n} 2cl_i s_i \tag{10}$$

$$\text{subject to } v_i = v_{i-1} - 2\rho \frac{l_i}{s_i} \sum_{j=i}^{n} \frac{p_j}{v_j}, \quad i = 1, 2, \ldots, n, \tag{11}$$

$$v_n = a v_0, \tag{12}$$

$$v_{i-1} > v_i, \quad i = 1, 2, \ldots, n, \tag{13}$$

$$s_i, v_i > 0, \quad i = 1, 2, \ldots, n, \tag{14}$$

where the power loads p_i $(i = 1, 2, \ldots, n)$, the lengths l_i $(i = 1, 2, \ldots, n)$, the voltages v_0 and v_n, and the constants a, ρ and c, are data, and the cross-sections s_i $(i = 1, 2, \ldots, n)$ and the voltages $v_1, v_2 \ldots, v_{n-1}$ are the variables of the optimization problem. The objective in (10) is the minimization of the total cost of the cable or equivalently of its total volume.

3 An initial feasible solution for the optimization problem

As stated in Anunciada (2003), a first feasible solution for the optimization problem can be constructed by assuming that in each cable section $i, i = 1, 2, \ldots, n$, the cross section s_i is proportional to the current c_i. Thus $s_i = Sc_i$, and from (4), $s_i = S \sum_{j=i}^{n} I_j^c$, with $S > 0$. The process of finding such a solution reduces to the computation of the proportionality constant S. Suppose that all the loads along the

cable are concentrated in a unique node, so that the cable is made of a single section with length $L = \sum_{i=1}^{n} l_i$ and by two nodes, one at each end, namely the voltage generator and the current node. This last node concentrates the electrical loads, whence it has a load power $P = \sum_{i=1}^{n} p_i$. Since the voltage in this node is av_0, we can easily compute the load current c in the last node, the resistance r in the section, and the cross-section s in the conducting wires of that section by

$$c = \frac{P}{av_0}, \tag{15}$$

$$r = \frac{v_0 - av_0}{c} = \frac{a(1-a)v_0^2}{P}, \tag{16}$$

$$s = \frac{2\rho L}{r} = \frac{2\rho LP}{a(1-a)v_0^2}. \tag{17}$$

Therefore the reference cross-section S is defined by

$$S = \frac{s}{c} = \frac{2\rho L}{(1-a)v_0}. \tag{18}$$

The cross-sections s_i are computed from

$$s_i = S \sum_{j=i}^{n} l_j^c, \quad i = 1, 2, \ldots, n. \tag{19}$$

The voltages v_i can now be obtained recursively from (6) by

$$v_{i-1} = v_i + 2\rho \frac{l_i}{s_i} \sum_{j=i}^{n} \frac{p_j}{v_j}, \quad i = 1, 2, \ldots, n. \tag{20}$$

In the particular case where $p_1 = p_2 = \cdots = p_n$, with $v_n = \frac{v_0}{2}$, the voltages can be determined as follows:

$$v_{i-1} - v_i = \frac{l_i}{\sum_{i=1}^{n} l_i}(v_0 - v_n), \quad i = 1, 2, \ldots, n. \tag{21}$$

Then by (20) the cross-sections satisfy

$$s_i = 2\rho l_i \frac{\sum_{j=i}^{n} \frac{p_j}{v_j}}{v_{i-1} - v_i}, \quad i = 1, 2, \ldots, n. \tag{22}$$

Computational experience (Sect. 7) shows that this process finds a feasible solution that is in general a tight upper bound to the global minimum value of the optimization problem.

It is also important to note that, since $s_i = S \sum_{j=i}^{n} l_j^c$, (5) can be rewritten as $\Delta v_i = v_{i-1} - v_i = \frac{2\rho}{S} l_i$. This means that for the feasible solution obtained through this heuristic method, the voltage decrease Δv_i in each section i is proportional to the length l_i of this section.

4 Solution of the nonlinear optimization problem

Consider again the model described in Sect. 2. Looking carefully at this formulation, we note that the constraints $s_i, v_i > 0, i = 1, 2, \ldots, n$, are redundant. In fact, according to (12) and (13), we have $v_0 > v_1 > v_2 > \cdots > v_n = av_0$. Since a and v_0 are positive, then $v_i \geq av_0 > 0$, for all $i = 1, 2, \ldots, n$. As $v_{i-1} - v_i > 0$, $\rho > 0$, $l_i > 0$ and $p_j > 0$, for all j, then $s_i > 0$, for all $i = 1, 2, \ldots, n$. Then the variables s_i can be assumed to be unrestricted in sign and can be eliminated from the problem to obtain the following nonlinear program in the variables v_i:

$$\textbf{NLP:} \quad \underset{v \in \mathbb{R}^n}{\text{Minimize}} \; 4\rho c \sum_{i=1}^{n} l_i^2 \frac{\sum_{j=i}^{n} \frac{p_j}{v_j}}{v_{i-1} - v_i} \tag{23}$$

$$\text{subject to } v_{i-1} > v_i, \quad i = 1, 2, \ldots, n, \tag{24}$$

$$v_n = av_0, \tag{25}$$

where $v_0, a, \rho, c, l_i, p_j$ are positive data. After a global minimum is obtained for this optimization problem NLP, then the cross-section areas s_i can be found from (11).

The feasible set of this program is not closed, whence the Weierstrass Theorem (Bazaraa et al. 1993) cannot be used to guarantee the existence of a minimum. To overcome this difficulty, we consider the additional variables z_i, defined by $z_i = v_{i-1} - v_i, i = 1, 2, \ldots, n$, and the nonlinear program:

$$\textbf{NLP1:} \quad \text{Minimize } f(z, v) = 4\rho c \sum_{i=1}^{n} l_i^2 \frac{\sum_{j=i}^{n} \frac{p_j}{v_j}}{z_i} \tag{26}$$

$$\text{subject to } z_i - v_{i-1} + v_i = 0, \quad i = 1, 2, \ldots, n, \tag{27}$$

$$z_i \geq \varepsilon, \quad i = 1, 2, \ldots, n, \tag{28}$$

$$v_n = av_0, \tag{29}$$

where ε is a positive and small real number (in practice $\varepsilon = 10^{-6}$). As for each $i = 1, 2, \ldots, n$,

$$v_{i-1} > v_i \Leftrightarrow z_i = v_{i-1} - v_i > 0,$$

then we guarantee that the function f of NLP1 has a global minimum on the set

$$\mathrm{K} = \{(z, v) \in \mathbb{R}^{n \times n} \text{ satisfying (27–29)}\}.$$

Consider now the objective function of NLP1, the n functions f_i defined by

$$f_i(z_i, v_i, \ldots, v_n) = \frac{\sum_{j=i}^{n} \frac{p_j}{v_j}}{z_i}, \quad i = 1, 2, \ldots, n \tag{30}$$

and the sets

$$K_i = \{(z_i, v_i, \ldots, v_n) \in \mathbb{R}^{n-i+2} : z_i > 0; v_j > 0, j = i, \ldots, n\},$$
$$i = 1, 2, \ldots, n. \tag{31}$$

Therefore for all $(z, v) \in \mathrm{K}$,

$$f(z, v) = 4\rho c \sum_{i=1}^{n} l_i^2 f_i(z_i, v_i, \ldots, v_n). \tag{32}$$

Then f is strictly convex on K, as the following property holds:

Theorem *The function f_i $(i = 1, 2, \ldots, n)$ defined by (30) is strictly convex on K_i.*

Proof The gradient of f_i at each point of K_i always exists and is given by

$$\nabla f_i(z_i, v_i, \ldots, v_n) = \begin{bmatrix} -\frac{1}{z_i^2}\left(\sum_{j=i}^{n} \frac{p_j}{v_j}\right) \\ -\frac{1}{z_i}\left(\frac{p_i}{v_i^2}\right) \\ \vdots \\ -\frac{1}{z_i}\left(\frac{p_n}{v_n^2}\right) \end{bmatrix}.$$

The Hessian of f_i is

$$\nabla^2 f_i(z_i, v_i, \ldots, v_n) = A_1 + A_2,$$

where

$$A_1 = \mathrm{diag}\left(\frac{1}{z_i^3}\left(\sum_{j=i}^{n} \frac{p_j}{v_j}\right), \frac{p_i}{z_i v_i^3}, \ldots, \frac{p_n}{z_i v_n^3}\right)$$

and A_2 is given by

$$A_2 = \begin{bmatrix} \frac{1}{z_i^3}\left(\sum_{j=i}^{n} \frac{p_j}{v_j}\right) & \frac{p_i}{z_i^2 v_i^2} & \frac{p_{i+1}}{z_i^2 v_{i+1}^2} & \cdots & \frac{p_n}{z_i^2 v_n^2} \\ \frac{p_i}{z_i^2 v_i^2} & \frac{p_i}{z_i v_i^3} & 0 & \cdots & 0 \\ \frac{p_{i+1}}{z_i^2 v_{i+1}^2} & 0 & \frac{p_{i+1}}{z_i v_{i+1}^3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{p_n}{z_i^2 v_n^2} & 0 & 0 & \cdots & \frac{p_n}{z_i v_n^3} \end{bmatrix}.$$

Now A_2 can be written as follows:

$$A_2 = \begin{bmatrix} \alpha & v^T \\ v & D \end{bmatrix},$$

where $\alpha > 0$, $v \in \mathbb{R}^{n-i+1}$ and

$$D = \operatorname{diag}\left(\frac{p_i}{z_i v_i^3}, \frac{p_{i+1}}{z_i v_{i+1}^3}, \ldots, \frac{p_n}{z_i v_n^3} \right).$$

Since all diagonal elements of D are positive and the Schur Complement $(A_2|D)$ of D in A_2 (Cottle et al. 1992) is zero, then A_2 is Symmetric Positive Semi-Definite. As A_1 is Symmetric Positive Definite, then $\nabla^2 f_i(z_i, v_i, \ldots, v_n)$ is Symmetric Positive Definite in K_i (Cottle et al. 1992) and f_i is strictly convex on K_i. □

Since K is compact and f is strictly convex on K, then f has a unique stationary point on K, which is exactly its unique global minimum (Bazaraa et al. 1993). As a stationary point of f on K may be found by a local nonlinear programming algorithm, it is easy to obtain the unique global minimum of the function on K and thus to solve the optimization problem.

5 An alternative formulation on the simplex

Consider the program NLP1, introduced in the previous section. We can write the constraints of the program as follows:

$$\begin{aligned} Lv + z &= b, \\ v_n &= a v_0, \\ z_i &\geq \varepsilon, \quad i = 1, 2, \ldots, n, \end{aligned}$$

where

$$L = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & & \ddots & \\ & & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n \times n},$$

$$b = \begin{bmatrix} v_0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^n, \qquad z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_n \end{bmatrix} \in \mathbb{R}^n.$$

As L is a triangular nonsingular matrix, we can eliminate the variables v_i by solving $Lv = b - z$. Then

$$v_i = v_0 - \sum_{k=1}^{i} z_k, \quad i = 1, 2, \ldots, n. \tag{33}$$

Furthermore

$$\begin{aligned} v_n = a v_0 &\Leftrightarrow v_0 - (z_1 + z_2 + \cdots + z_n) = a v_0 \\ &\Leftrightarrow z_1 + z_2 + \cdots + z_n = (1-a) v_0 \\ &\Leftrightarrow \sum_{j=1}^{n} z_j = (1-a) v_0. \end{aligned}$$

Then NLP is equivalent to

$$\begin{aligned} \textbf{NLP2:} \quad \text{Minimize } & f(z) = 4\rho c \sum_{i=1}^{n} \frac{l_i^2}{z_i} \left(\sum_{j=i}^{n} \frac{p_j}{v_0 - \sum_{k=1}^{j} z_k} \right) \\ \text{subject to } & e^T z = (1-a) v_0, \\ & z_i \geq \varepsilon, \quad i = 1, 2, \ldots, n, \end{aligned}$$

where $e \in \mathbb{R}^n$ is a vector of ones. The feasible set of this program is given by

$$\mathrm{K} = \{ z \in \mathbb{R}^n : e^T z = (1-a) v_0; z_i \geq \varepsilon, i = 1, 2, \ldots, n \}. \tag{34}$$

Hence K is a compact set and by Weierstrass Theorem, the function f has a global minimum on K. Moreover, the function of NLP2 is obtained from the function of NLP by a linear transformation with a nonsingular matrix. Therefore this function f is strictly convex on K (Martos 1975). Hence the computation of the unique global minimum of NLP2 may be done by a local nonlinear programming method. In the next section we introduce a Projected Gradient algorithm that finds the unique stationary point of f on K by taking advantage of the structure of the program NLP2. The original variables v_i and s_i can be obtained afterwards by using formulas (33) and (11).

6 A projected gradient algorithm for the nonlinear program on the simplex

Consider the program NLP2 introduced in the previous section and its constraint set K given by (34). The projected gradient of f in the point $z \in \mathrm{K}$ is defined as the vector in $\mathbb{R}^n$

$$g(z) = P_K(z - \eta \nabla f(z)) - z, \tag{35}$$

where $P_K(\cdot)$ is the orthogonal projection in K and $\eta > 0$ (Bertsekas 1999).

It is known that for a given $\eta > 0$, $g(z) = 0$ *if and only if* z is a stationary point of f on K (Bertsekas 1999). The Projected Gradient algorithm is a modification of the Steepest Descent algorithm (Dennis and Schnabel 1996) in which the iterates are forced to stay in K by a projection (Bertsekas 1976). It can be shown (Bertsekas 1999; Birgin et al. 2000) that this algorithm possesses global convergence towards a stationary point under reasonable hypotheses. The steps of the algorithm are presented below.

Projected gradient algorithm for NLP2:

Step 0: Find an initial solution $z \in \mathrm{K}$.

Step 1: (**Computation of search direction**) Compute the direction $d \in \mathbb{R}^n$ by

$$d = y - z,$$

where $y = P_K(z - \eta \nabla f(z))$, with $\eta > 0$.

Step 2: (**Stopping test**) If $\|d\|_2 < \text{tol}$, then stop: z is a stationary point of f on K. Otherwise, go to Step 3.

Step 3: (**Armijo criterion**) Find $\alpha \in \mathbb{R}^+$ such that

$$f(z + \alpha d) \leq f(z) + \beta \alpha \nabla f(z)^T d,$$

with $0 < \beta < 1$.

Step 4: (**Iterate updating**) Update the current solution

$$z \leftarrow z + \alpha d$$

and return to Step 2.

Next, we discuss some details of the algorithm.

(I) *Computation of the gradient of* f

By simple manipulation, it is possible to write $f(z)$ as follows:

$$f(z) = 4\rho c \sum_{k=1}^{n} \frac{p_k \times \sum_{j=1}^{k} \frac{l_j^2}{z_j}}{v_0 - \sum_{j=1}^{k} z_j}.$$

Therefore it is easy to obtain the following formulas for the components of the gradient vector $\nabla f(z)$:

$$\underset{i=1,2,\ldots,n}{\forall} \quad \frac{\partial f(z)}{\partial z_i} = 4\rho c \sum_{k=i}^{n} \frac{p_k \times \sum_{j=1}^{k} \frac{l_j^2}{z_j}}{(v_0 - \sum_{j=1}^{k} z_j)^2} - \frac{l_i^2}{z_i^2} \sum_{k=i}^{n} \frac{p_k}{v_0 - \sum_{j=1}^{k} z_j}. \tag{36}$$

(II) *Computation of the projection* $y = P_K(z - \eta \nabla f(z))$

1. Find $u = z - \eta \nabla f(z)$ using (36) to compute $\nabla f(z)$.
2. The vector y is the unique optimal solution of the strictly convex quadratic program

$$\begin{aligned} \underset{y \in \mathbb{R}^n}{\text{Minimize}} \quad & \frac{1}{2}\|u - y\|_2^2 \\ \text{subject to} \quad & e^T y = (1 - a) v_0, \\ & y_i \geq \varepsilon, \quad i = 1, 2, \ldots, n. \end{aligned}$$

But

$$\|u-y\|_2^2 = (u-y)^T(u-y) = u^T u - 2u^T y + y^T y$$

and therefore this program is equivalent to

$$\begin{aligned} \underset{z\in\mathbb{R}^n}{\text{Minimize}} \;\; & q^T y + \frac{1}{2} y^T y \\ \text{subject to} \;\; & e^T y = b_0, \\ & y_i \geq \varepsilon, \quad i = 1, 2, \ldots, n, \end{aligned}$$

where $b_0 = (1-a)v_0$ and $q = -u$.

There are several algorithms described in the literature to process this kind of quadratic programs (Dussault et al. 1986; Pardalos and Kovoor 1990; Júdice and Pires 1992). Among these processes, the Block Pivotal Principal Pivoting (BLOCK) Algorithm described in (Júdice and Pires 1992) is strongly polynomial and very efficient in practice. The steps of this process are presented below.

BLOCK algorithm:

Step 0: Let $F = \{1, 2, \ldots, n\}$.

Step 1: Compute $\lambda = -\frac{b_0 + \sum_{i\in F} q_i}{|F|}$.

Step 2: (**Stopping test**)
Let

$$H = \{i \in F : q_i + \lambda > \varepsilon\}.$$

If $H = \phi$, stop: the vector

$$z = (z_i)_{i=1,2,\ldots,n}, \quad \text{with } z_i = \begin{cases} \varepsilon, & \text{if } i \notin F, \\ -(q_i + \lambda), & \text{if } i \in F \end{cases}$$

is the unique optimal solution of the quadratic program. Otherwise set $F = F - H$ and go to Step 1.

(III) *Computation of the optimal values for the variables v_i^* and s_i^**

Let $z^* = (z_i^*)_{i=1,2,\ldots,n} \in \mathbb{R}^n$ be the optimal solution of the problem NLP2. The variables $v_i^*, i = 1, 2, \ldots, n$, of the original problem satisfy $Lv^* = b - z^*$. Therefore

$$v_i^* = v_{i-1}^* - z_i^*, \quad i = 1, 2, \ldots, n \; (v_0^* = v_0). \tag{37}$$

Furthermore, (11) provides the values of the variables s_i^*:

$$s_i^* = \frac{2\rho l_i \sum_{j=i}^n \frac{p_j}{v_j^*}}{z_i^*}, \quad i = 1, 2, \ldots, n.$$

7 Computational experience

The computational experience presented in this section was performed using a PC with 3 GHz Pentium 4 processor and 2 GB RAM memory, running Linux 2.6.10. The active–set code `MINOS` (Murtagh and Saunders 1987) (one of the nonlinear solvers of `GAMS`) has been used for processing both the nonlinear programs NLP1 and NLP2. Furthermore, the Projected Gradient algorithm for the program NLP2 was implemented in Fortran 77 (Hehl 1987), using the Gnu Fortran (g77) compiler, version 3.4.3, with the options `-02 -malign-double -funroll-loops`. Running times presented in this section are given in CPU seconds, excluding *inputs* and *outputs*. The times for the Projected Gradient algorithm were measured by the `etime()` function.

The results reported in this section are concerned with two actual instances of the problem found in practice, the first (Example 1) presenting reduced voltage decrease and the second (Example 2) presenting high voltage decrease. The data for these problems are stated below and in Table 1:

$$n = 40; \qquad \rho = 20\ (\Omega\,\mathrm{km}^{-1}\,\mathrm{mm}^2); \qquad c > 0.$$

For the first example the cross-section value is $S = 9.680$ (with $L = 48.40$ and $P = 7750$), and for the second example, is $S = 8.811$ (with $L = 15.42$ and $P = 119850$). The heuristic procedure described in section 3 has found the following solutions:

For Example 1:

$$\begin{aligned} v &= (v_i)_{i=0,1,\ldots,40} \\ &= (500.000, 499.587, 497.107, 495.868, 483.058, 474.793, 468.182, \\ &\quad 464.876, 461.983, 457.025, 450.826, 443.388, 441.322, 440.083, \\ &\quad 438.843, 437.603, 434.711, 431.405, 426.860, 417.355, 386.364, \\ &\quad 381.818, 376.033, 372.314, 371.901, 371.488, 370.661, 369.835, \\ &\quad 369.421, 367.355, 363.223, 358.678, 353.306, 348.760, 317.769, \\ &\quad 304.132, 303.719, 303.306, 302.066, 300.413, 300.000); \end{aligned}$$

$$\begin{aligned} s &= (s_i)_{i=1,2,\ldots,40} \\ &= (195.930, 194.961, 191.066, 187.162, 186.160, 175.966, 165.628, \\ &\quad 163.546, 162.498, 161.439, 159.292, 154.926, 150.539, 146.140, \\ &\quad 141.728, 130.668, 129.555, 128.433, 127.299, 126.139, 121.128, \\ &\quad 116.058, 103.187, 97.987, 96.685, 95.382, 94.077, 92.768, \\ &\quad 91.458, 88.823, 83.493, 78.095, 64.396, 50.518, 35.287, \\ &\quad 32.104, 25.730, 19.347, 17.744, 16.133) \end{aligned}$$

with total cost $C = 11105.270 \times c$.

Table 1 Problem data for Examples 1 and 2

i	Example 1 (reduced voltage decrease)		Example 2 (high voltage decrease)	
	$V_0 = 500$ V	$a = 0.6$	$V_0 = 260$ V	$a \approx 0.731$
	$V_{40} = 300$ V		$V_{40} = 190$ V	
	length l_i (km)	power p_i (W)	length l_i (km)	power p_i (W)
1	0.1	50	0.1	1150
2	0.6	200	0.1	1150
3	0.3	200	0.2	2300
4	3.1	50	0.5	2300
5	2.0	500	0.4	1150
6	1.6	500	1.0	6900
7	0.8	100	0.2	1150
8	0.7	50	0.3	1150
9	1.2	50	0.2	2300
10	1.5	100	0.1	2300
11	1.8	200	0.1	1150
12	0.5	200	0.05	1150
13	0.3	200	0.05	2300
14	0.3	200	0.03	1150
15	0.3	500	0.4	6900
16	0.7	50	0.5	6900
17	0.8	50	0.5	1350
18	1.1	50	0.3	2300
19	2.3	50	0.4	1150
20	7.5	200	0.4	2300
21	1.1	200	0.9	6900
22	1.4	500	1.0	3450
23	1.9	200	0.5	1350
24	0.1	50	0.1	6900
25	0.1	50	0.1	1350
26	0.2	50	0.03	1350
27	0.2	50	0.05	1350
28	0.1	50	0.05	1350
29	0.5	100	1.1	6900
30	1.0	200	1.2	3450
31	1.1	200	0.6	3450
32	1.3	500	0.45	3450
33	1.1	500	0.33	6900
34	7.5	500	0.21	1350
35	3.3	100	0.09	10350
36	0.1	200	0.06	6900
37	0.1	200	1.25	1150
38	0.3	50	0.15	1150
39	0.4	50	0.12	1150
40	0.1	500	1.3	1150

For Example 2:

$$
\begin{aligned}
v &= (v_i)_{i=0,1,\ldots,40} \\
&= (260.000, 259.546, 259.092, 258.184, 255.914, 254.099, 249.559, \\
&\quad 248.651, 247.289, 246.381, 245.927, 245.473, 245.246, 245.019, \\
&\quad 244.883, 243.067, 240.798, 238.528, 237.166, 235.350, 233.534, \\
&\quad 229.449, 224.909, 222.639, 222.185, 221.732, 221.595, 221.368, \\
&\quad 221.141, 216.148, 210.700, 207.977, 205.934, 204.436, 203.482, \\
&\quad 203.074, 202.802, 197.127, 196.446, 195.901, 190.000);
\end{aligned}
$$

$$
\begin{aligned}
s &= (s_i)_{i=1,2,\ldots,40} \\
&= (4714.404, 4675.363, 4636.252, 4557.757, 4478.565, 4438.686, 4195.061, \\
&\quad 4154.309, 4113.332, 4031.076, 3948.669, 3907.389, 3866.070, 3783.357, \\
&\quad 3741.978, 3491.846, 3239.357, 3189.487, 3104.035, 3060.979, 2974.198, \\
&\quad 2709.221, 2574.057, 2520.628, 2246.988, 2193.340, 2139.659, 2085.924, \\
&\quad 2032.132, 1750.849, 1606.571, 1460.403, 1312.786, 1015.388, 956.928, \\
&\quad 507.839, 208.044, 156.640, 105.058, 53.332)
\end{aligned}
$$

with total cost $C = 74762.948 \times c$.

As mentioned earlier, each of these values is an upper bound for the optimal value given by the nonlinear programs NLP1 and NLP2. The unique optimal solution of those programs found by `MINOS` is given below:

For Example 1:

$$
\begin{aligned}
v^* &= (v_i^*)_{i=0,1,\ldots,40} \\
&= (500.000, 499.470, 496.298, 494.727, 478.658, 468.316, 460.255, \\
&\quad 456.334, 452.922, 447.091, 439.823, 431.152, 428.772, 427.361, \\
&\quad 425.968, 424.593, 421.494, 417.966, 413.132, 403.062, 370.347, \\
&\quad 365.618, 359.689, 356.031, 355.632, 355.235, 354.445, 353.659, \\
&\quad 353.267, 351.321, 347.470, 343.325, 338.543, 334.769, 311.298, \\
&\quad 302.215, 301.949, 301.704, 301.048, 300.203, 300.000);
\end{aligned}
$$

$$
\begin{aligned}
s^* &= (s_i^*)_{i=1,2,\ldots,40} \\
&= (156.856, 156.477, 154.934, 153.366, 152.944, 148.488, 143.766, \\
&\quad 142.792, 142.295, 141.785, 140.726, 138.507, 136.244, 133.944, \\
&\quad 131.608, 125.623, 125.007, 124.379, 123.734, 123.055, 119.820, \\
&\quad 116.475, 107.708, 104.070, 103.153, 102.231, 101.303, 100.368, \\
&\quad 99.429, 97.521, 93.586, 89.506, 78.755, 67.253, 53.127, 49.982, \\
&\quad 43.473, 36.549, 34.720, 32.831).
\end{aligned}
$$

For Example 2:

$$
\begin{aligned}
v^* &= (v_i^*)_{i=0,1,\ldots,40} \\
&= (260.000, 259.337, 258.676, 257.360, 254.097, 251.507, 245.062, \\
&\quad 243.805, 241.930, 240.685, 240.068, 239.457, 239.153, 238.851, \\
&\quad 238.671, 236.286, 233.396, 230.601, 228.936, 226.742, 224.561, \\
&\quad 219.715, 214.542, 212.010, 211.508, 211.029, 210.887, 210.653, \\
&\quad 210.421, 205.379, 200.211, 197.717, 195.919, 194.657, 193.937, \\
&\quad 193.636, 193.485, 191.440, 191.226, 191.085, 190.000);
\end{aligned}
$$

$$
\begin{aligned}
s^* &= (s_i^*)_{i=1,2,\ldots,40} \\
&= (3351.859, 3338.425, 3324.873, 3297.323, 3268.836, 3254.211, 3160.536, \\
&\quad 3144.546, 3128.242, 3095.103, 3061.525, 3044.547, 3027.457, 2992.976, \\
&\quad 2975.600, 2867.608, 2753.897, 2730.719, 2690.347, 2669.620, 2627.048, \\
&\quad 2491.010, 2418.020, 2388.453, 2233.547, 2202.441, 2171.058, 2139.357, \\
&\quad 2107.330, 1931.019, 1834.938, 1733.832, 1627.834, 1401.373, 1354.395, \\
&\quad 950.698, 588.979, 507.413, 412.098, 290.011).
\end{aligned}
$$

For both examples, we have used as an initial point for MINOS the default choice given by the code and the solution computed by the heuristic procedure. The values of the variables v_i show that the constraints $z_i \geq \varepsilon$ ($\varepsilon = 10^{-6}$) are inactive at the optimal solutions of both problems. Tables 2 and 3 report the numerical results under these two choices and lead to the conclusion that it is recommended to start with the feasible solution given by the heuristic procedure. It is also important to add that Formulation NLP1 is preferable if an active-set method such as MINOS is employed to process the nonlinear program. The reason for this arises from the more complex formulas of the objective function in the Formulation NLP2.

The implementation of the Projected Gradient algorithm for problem NLP2 requires a choice for the parameters β, θ and η used by the procedure. For both examples, some tests have been performed with the algorithm in order to make a sensitivity

Table 2 Comparative performance of the formulations NLP1 and NLP2 for Example 1, solved by GAMS/MINOS

Example 1						
Formulations	Initial solution					
	By default (MINOS)			Heuristic solution		
	Iterations	CPU	Cost	Iterations	CPU	Cost
NLP1	820	1.470	$10764.6896 \times c$	84	0.266	$10764.6896 \times c$
NLP2	789	12.423	$10764.6896 \times c$	121	2.447	$10764.6896 \times c$

Table 3 Comparative performance of the formulations NLP1 and NLP2 for Example 2, solved by GAMS/MINOS

Example 2						
Formulations	Initial solution					
	By default (MINOS)			Heuristic solution		
	Iterations	CPU	Cost	Iterations	CPU	Cost
NLP1	562	1.014	$66585.2396 \times c$	98	0.286	$66585.2396 \times c$
NLP2	530	8.474	$66585.2396 \times c$	153	3.014	$66585.2396 \times c$

analysis for these parameters. The best choices for the Projected Gradient algorithm have been achieved with $\beta = 10^{-4}$, $\theta = \frac{1}{2}$, $\eta = 10^{-2}$ for Example 1 and $\beta = 10^{-4}$ (or $\beta = 10^{-2}$), $\theta = \frac{1}{10}$, $\eta = 10^{-3}$ for Example 2.

Table 4 indicates the performance of the Projected Gradient algorithm (with $tol = 10^{-6}$) for the best choice of the parameters and for two initial solutions. These computational results show that the Projected Gradient algorithm is efficient to process the nonlinear program NLP2. The number of iterations required by the projected gradient method is slightly bigger than those of the active-set method (MINOS) displayed in Tables 2 and 3. However, the algorithm performs faster than MINOS. Another interesting conclusion is that the use of the initial solution given by the heuristic procedure does not seem to help for this method. As the Projected Gradient algorithm fully exploits the structure of the nonlinear program and requires minimal storage, we believe that this method can be useful for the solution of larger instances of this optimization model.

The unique optimal solution found by the algorithms gives about 3% savings for Example 1 and about 12% savings for Example 2, when compared with the heuristic solutions used in practice in the project reported in this paper. We believe that more substantial savings can be achieved for larger instances of the energy model if the unique optimal solution of NLP1 or NLP2 is used instead of the solution that is nowadays employed in this project.

Table 4 Best performances of the Projected Gradient algorithm

Example 1			
Initial solution	Iterations	CPU	Cost
$z_i = \frac{b_0}{40}, i = 1, 2, \ldots, 40$	399	0.003	$10764.6896 \times c$
Heuristic solution	512	0.005	$10764.6896 \times c$
Example 2			
Initial solution	Iterations	CPU	Cost
$z_i = \frac{b_0}{40}, i = 1, 2, \ldots, 40$	114	0.001	$66585.2396 \times c$
Heuristic solution	129	0.001	$66585.2396 \times c$

8 Conclusion

This paper describes formulations and solutions of an engineering optimization problem in telecommunications area, that consists of optimizing the cross section of conducting energy cables, used to supply remote telecom equipments, in order to minimize the volume of copper material used in the cables and consequently the cost. Two alternative formulations for the problem have been introduced and a Projected Gradient algorithm was proposed for the second formulation, taking advantage of its particular structure. Computational experience with an implementation of this algorithm has shown that the proposed methodology is efficient in practice. We believe that this type of model and the Projected Gradient algorithm can also be useful in other engineering models used in energy supply, piping and fluid distribution with a similar structure, particularly when the number of nodes is quite large.

References

Anunciada V (2003) Cost minimization of a multiple section energy cable supplying remote telecom equipments. Technical report, Instituto de Telecomunicações, Portugal

Bazaraa M, Sherali H, Shetty C (1993) Nonlinear programming: theory and algorithms. Wiley, New York

Bertsekas D (1999) Nonlinear programming. Athena Scientific, Massachusetts

Bertsekas DP (1976) On the Goldstein–Levitin–Polyak gradient projection method. IEEE Trans Autom Control 21:174–184

Birgin EG, Martinez JM, Raydan M (2000) Nonmonotone spectral projected gradient methods on convex sets. SIAM J Optim 10:1196–1211

Cottle RW, Pang JS, Stone RE (1992) The linear complementarity problem. Academic Press, New York

DeCarlo R (2001) Linear circuit analysis, 2nd edn. Pen-Lin Min, Oxford

Dennis JE, Schnabel RB (1996) Numerical methods for unconstrained optimization and nonlinear equations. SIAM, Philadelphia

Dussault JP, Ferland JA, Lemaire B (1986) Convex quadratic programming with one constraint and bounded variables. Math Program 36:90–104

Feynman RP, Leighton RB, Sands M (1963) The Feynman lectures on physics. Addison-Wesley, New York

Hehl ME (1987) Linguagem de programação estruturada FORTRAN 77. McGraw-Hill, São Paulo

Júdice JJ, Pires FM (1992) Solution of large-scale separable strictly convex quadratic programs on the simplex. Linear Algebra Appl 170:214–220

Martos B (1975) Nonlinear programming: theory and methods. North-Holland, New York

Murtagh BA, Saunders MA (1987) MINOS 5.1 user guide. Technical report, Department of Operations Research, Stanford University

Pardalos PM, Kovoor N (1990) An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. Math Program 46:321–328