
Volume 35
Issue 5
May 2008
ISSN 0305-0548
ELSEVIER
computers & operations research
Part Special Issue:
Algorithms and Computational Methods in Feasibility and Infeasibility
Guest Editor:
John W. Chinneck
Editor: Stefan Nickel
Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 1494 – 1509
computers & operations research
www.elsevier.com/locate/cor

A reformulation–linearization–convexification algorithm for optimal correction of an inconsistent system of linear constraints

P. Amaral[a,*], J. Júdice[b,c], H.D. Sherali[d]

[a]Departamento de Matemática, Universidade Nova de Lisboa, Monte da Caparica, 2829-516 Caparica, Portugal
[b]Departamento de Matemática, Universidade de Coimbra, Apartado 3008, 3001-454 Coimbra, Portugal
[c]Department of Electrical and Computer Engineering, Instituto Telecomunicações, University of Coimbra, Pole II, P-3030-290 Coimbra, Portugal
[d]Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, 201 Durham Hall, Blacksburg, VA 24061, USA

Available online 23 October 2006

Abstract

In this paper, an algorithm is introduced to find an optimal solution for an optimization problem that arises in total least squares with inequality constraints, and in the correction of infeasible linear systems of inequalities. The stated problem is a nonconvex program with a special structure that allows the use of a reformulation–linearization–convexification technique for its solution. A branch-and-bound method for finding a global optimum for this problem is introduced based on this technique. Some computational experiments are included to highlight the efficacy of the proposed methodology.

Inconsistent systems play a major role on the reformulation of models and are a consequence of lack of communication between decision makers. The problem of finding an optimal correction for some measure is of crucial importance in this context. The use of the Frobenius norm as a measure seems to be quite natural and leads to a nonconvex fractional programming problem. Despite being a difficult global optimization, it is possible to process it by using a branch-and-bound algorithm incorporating a local nonlinear programming method.

Keywords: Infeasible systems; Nonlinear programming; Global optimization

1. Introduction

The problem we address in this paper arises in the context of correcting infeasible linear systems of inequalities, such as $Ax \leqslant b$, $x \in \mathbb{R}^n$, where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

In linear programming and in constraint satisfaction, one is often confronted with an empty set of solutions due to many causes, such as the lack of communication between different decision makers, update of old models, or integration of partial models. In post-infeasibility analysis, several attempts are made in order to retrieve valuable information regarding the inconsistency of a given model [1–5], such as the identification of conflicting sets of constraints [6–10] and irreducible inconsistent systems (IIS) of constraints, for both continuous [11–14] and mixed-integer [15] problems. This information can be used to reformulate the model, either by removing constraints or slightly changing the coefficients of the constraints. In [16], the authors proposed a method based on an hierarchical classification of constraints to remove

* Corresponding author. Fax: +351 21 2948391.
E-mail address: paca@fct.unl.pt (P. Amaral).

0305-0548/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2006.08.007

constraints in order to obtain a feasible set. This procedure is, however, inadequate in cases where the physical events that the constraints seek to prevent cannot be ignored, or when we are working with only approximate data. In the context of linear programming, some theoretical results regarding the distance to infeasibility of a linear system are presented in [17,18], and techniques for deciding about the existence of solutions for approximate data are explored in [19].

Although the problem of inconsistency in linear models has attracted some attention, not much has been done concerning the development of exact algorithms for finding minimal corrections (according to some criteria) of the coefficients of an infeasible linear system of inequalities. The perturbation of the vector b alone is a less difficult problem and has been considered in [20]. The correction of both A and b is a more challenging task, due to the introduction of nonlinearity, but is more adequate and advisable in practice.

Let us consider, for instance, a production model where x represents the production of some products, the matrix A represents both the consumption of raw material per unit and certain cost and sale price coefficients, and b is the amount of available raw material together with a desired level of profit. In such a situation, it is plausible in the case of inconsistency to not only rectify the amount of available raw material and the target profit (right-hand side values), but also the specifications of the production process and the sale price. Even if our primary goal is not to derive a modification of the model, it can be useful to find a feasible model that is closest to the infeasible one in some sense. Furthermore, if the original system describes some ideal properties of a new product or the behavior of some structure, in case of inconsistency, of the model, it would be of importance to understand how the paradigms we are seeking must be minimally adapted for the purpose of implementation. The solution of a feasible perturbed model can be insightful. The smaller the difference between the infeasible and the modified feasible model, the greater the possibility that the suggested changes can be realistically implemented. The knowledge of a feasible solution for a corrected model can provide insights into the nature of the infeasibility and the way to overcome it. We can even analyze only a subset of constraints, as for example, an IIS or a set of soft constraints in such an analysis to either focus attention on different subsets of inconsistent constraints or on manipulating parameters of flexible relationships. In such contexts, a useful piece of information (among others) to provide the decision-maker would be an optimally perturbed feasible model, in the sense of minimizing a measure of distance between the set of parameters of the infeasible and corrected constraints.

These considerations lead to a general formulation of the following optimization problem that seeks the optimal correction p and H, respectively, of the vector b and the matrix A of the given linear system of inequalities $Ax \leqslant b$:

$$\text{Minimize} \quad \varphi(H, p) \tag{1}$$

$$\text{subject to} \quad (A+H)x \leqslant b + p, \tag{2}$$

$$x \in \mathbf{X}, \quad H \in \mathbb{R}^{m\times n}, \quad p \in \mathbb{R}^m, \tag{3}$$

where $\mathbf{X} \subseteq \mathbb{R}^n$ is a convex set and φ is an appropriate matrix norm. For $\varphi = \|\cdot\|_{l_1}$ and $\varphi = \|\cdot\|_{l_\infty}$ (the generalization for matrices of the respective norms[1] l_1 and l_∞), Vatolin [21] proved that it is possible to find an optimal correction by solving a set of linear programming problems. Later in [22], it was shown that this approach is also applicable for the ∞ norm,[2] $\varphi = \|\cdot\|_\infty$, and that the number of linear programming problems to be solved is $2n+1$ for the l_1 and ∞ norms, and 2^n for the l_∞ norm. Furthermore, it was required that an optimal correction should involve changes in only one column of (A, b) in the case of norms l_1 or ∞, while for the norm l_∞, the perturbation of the coefficients of every row should only differ in sign. This introduced a fixed pattern for the correction matrix, which turns out to be quite unnatural in practical situations where a free pattern is more suitable. These conclusions have motivated us to study the case of finding an optimal correction with respect to the Frobenius norm $\|\cdot\|_F$, that is, to consider the following

1

$$\|A\|_{l_1} = \sum_{ij} |a_{ij}|, \quad \|A\|_{l_\infty} = \max_{ij} |a_{ij}|.$$

2

$$\|A\|_\infty = \max_i \sum_j |a_{ij}|.$$

optimization problem:

$$
\begin{aligned}
(P)\ \ \text{Minimize} \quad & \|[H,\, p]\|_{\mathrm{F}}^2 \\
\text{subject to} \quad & (A+H)x \leqslant b+p, \\
& H \in \mathbb{R}^{m\times n}, \quad p \in \mathbb{R}^m, \quad x \in \mathbf{X}.
\end{aligned} \tag{4}
$$

The interest here is not as much to directly diagnose what was the underlying problem in formulating the infeasible model, but rather, to provide this insight indirectly through the solution of problem (4), which yields the least Frobenius norm correction to the constraint matrix and right-hand side in order to attain feasibility.

It is interesting to note that for $\mathbf{X} = \mathbb{R}^n$, problem (4) may fail to have a solution, and that a local minimizer exists iff the correction corresponds to an application of the total least squares (TLS) method to the set of active constraints [23]. Some algorithms for finding such a local minimizer are discussed in [24]. In [25], a tree search procedure based on the enumeration of the active set of constraints was proposed, where some reduction tests were implemented in order to reduce the tree search. Although problems of small size have been efficiently solved, the overall effort required for finding a global minimum is usually too high for medium-scale problems. In practice, it is important to define the set $\mathbf{X}$ such that the existence of an optimal solution is guaranteed, and that the solution of the corrected system is in a certain domain of interest. The most general choice for $\mathbf{X}$ corresponds to

$$\mathbf{X} \subseteq \{x \in \mathbb{R}^n : l \leqslant x \leqslant u\}, \tag{5}$$

where l and u are fixed vectors. Then, $\mathbf{X}$ is compact and the optimization problem (P) has a global minimum. This is the choice we adopt in this paper.

This paper is organized as follows. In the next section, we present some important results that lead to the development of the main algorithm to be introduced in Section 3. Section 4 includes the report of some experiments with the algorithm for a number of test instances. Some conclusions and recommendations for future research are provided in the closing section of the paper.

2. Preliminary results

In [23] it was shown that for $\mathbf{X} = \mathbb{R}^n$, problem (4) is equivalent to the unconstrained nonlinear and nonconvex problem:

$$(P)\ \ \min_{x\in\mathbf{X}} \frac{\|(Ax-b)^+\|^2}{1+\|x\|^2}, \tag{6}$$

where $(\cdot)^+$ denotes a component-wise application of the operator $\max\{0, \cdot\}$ and $\|\cdot\|$ represents the Euclidian norm. This equivalence still holds for

$$\mathbf{X} \subseteq \{x : l \leqslant x \leqslant u\}.$$

Once problem (6) is solved, we can directly obtain an optimal correction matrix $[H, p]$ for problem (4) via

$$[H,\, p] = -\lambda^*[x^{*\mathrm{T}}, 1], \tag{7}$$

where x^* is the optimal solution found for problem (6) and

$$\lambda^* = \frac{1}{1+\|x^*\|^2}(Ax^*-b)^+. \tag{8}$$

The proof can be found in [23]. The next theorem presents another formulation for the original problem (4) that is exploited in this paper.

Propoisition 2.1. *Formulation* (6) *is equivalent to*

$$(P1)\ \textit{Minimize} \quad \frac{\beta}{1+\|x\|^2} \tag{9}$$

$$\textit{subject to} \quad \beta \geqslant \|v\|^2, \tag{10}$$

$$v \geqslant Ax - b, \tag{11}$$

$$v \geqslant 0, \tag{12}$$

$$x \in \mathbf{X}. \tag{13}$$

Proof. It is sufficient to show that for any feasible solution to one problem, there exists a feasible solution to the other problem having at least as good an objective value.

Given x feasible to (P), it is easy to see that there exists (x, v, β) feasible to $(P1)$ with the same objective value. In fact, for $v = (Ax - b)^+$ and $\beta = \|v\|^2$, we get

$$\frac{\beta}{1+\|x\|^2} = \frac{\|(Ax-b)^+\|^2}{1+\|x\|^2}.$$

Now, for (x, v, β) feasible to $(P1)$, let $\overline{v} = (Ax - b)^+$ and $\overline{\beta} = \|\overline{v}\|^2$. Then $(x, \overline{v}, \overline{\beta})$ is feasible to $(P1)$ and

$$\frac{\overline{\beta}}{1+\|x\|^2} = \frac{\|\overline{v}\|^2}{1+\|x\|^2} = \frac{\|(Ax-b)^+\|^2}{1+\|x\|^2} \leqslant \frac{\|v\|^2}{1+\|x\|^2} \leqslant \frac{\beta}{1+\|x\|^2}. \tag{14}$$

The first inequality in (14) follows because $Ax - b \leqslant v$ and $v \geqslant 0$ imply that $0 \leqslant (Ax - b)^+ \leqslant v$ and so, $\|(Ax - b)^+\|^2 \leqslant \|v\|^2$. Also, x is feasible to (P) and (14) shows that it has at least as good an objective value in problem (P) as (x, v, β) does in problem $(P1)$. This completes the proof. □

In $(P1)$, upon using the substitution

$$\|x\|^2 = \alpha,$$

and considering that $\mathbf{X} \subseteq \{x : 0 \leqslant l \leqslant x \leqslant u\}$, we immediately obtain the following result.

Corollary 2.2. *If*

$$\alpha_l = \|l\|^2 \quad \textit{and} \quad \alpha_u = \|u\|^2, \tag{15}$$

then the problem

$$(P2)\ \textit{Minimize} \quad \frac{\beta}{1+\alpha} \tag{16}$$

$$\textit{subject to} \quad \beta \geqslant \|v\|^2, \tag{17}$$

$$v \geqslant Ax - b, \tag{18}$$

$$\|x\|^2 = \alpha, \tag{19}$$

$$x \in \mathbf{X} \subseteq \{x : 0 \leqslant l \leqslant x \leqslant u\}, \tag{20}$$

$$v \geqslant 0, \tag{21}$$

$$\alpha_l \leqslant \alpha \leqslant \alpha_u \tag{22}$$

is equivalent to $(P1)$ *and, consequently, to problem* (4).

Now, consider the nonlinear programming relaxation to problem $(P2)$:

$$(RP2)\ \text{Minimize}\quad \frac{\beta}{1+\alpha} \tag{23}$$

$$\text{subject to}\quad \beta \geqslant \|v\|^2, \tag{24}$$

$$v \geqslant Ax - b, \tag{25}$$

$$\|x\|^2 \leqslant \alpha, \tag{26}$$

$$x \in \mathbf{X} \subseteq \{x : 0 \leqslant l \leqslant x \leqslant u\}, \tag{27}$$

$$v \geqslant 0, \tag{28}$$

$$\alpha_l \leqslant \alpha \leqslant \alpha_u. \tag{29}$$

Then the feasible region of $(RP2)$ is convex and the objective function is pseudoconvex over this set [26]. Therefore, any Karush–Kuhn–Tucker (KKT) solution to this problem is also a global minimum of $(RP2)$. Furthermore, any such solution is a global minimum of problem $(P2)$ if and only if $\|x\|^2 = \alpha$. It might therefore seem that there could exist some cases where the solution to $(P2)$ can be recovered by simply finding a global minimum for $(RP2)$. Unfortunately, this is not usually the case, as the inequality $\|x\|^2 \leqslant \alpha$ is in general inactive at such a global minimum to $(RP2)$. This is quite understandable, as α tends to increase as much as possible in order to minimize the objective function of $(RP2)$. So $(P2)$ needs to be processed by a global optimization algorithm. In this paper, we propose a branch-and-bound algorithm for $(P2)$ that is based on the idea of partitioning the set

$$\Omega = \{x : 0 \leqslant l \leqslant x \leqslant u\}.$$

Each node k of the enumeration tree in this process is associated with a proper subset of Ω identified as

$$\Omega^k = \{x : l_i^k \leqslant x_i \leqslant u_i^k,\ \text{for } i = 1, \ldots, n\}, \tag{30}$$

along with the following corresponding node subproblem:

$$\begin{aligned}
(P2^k)\ \text{Minimize}\quad & \frac{\beta}{1+\alpha} \\
\text{subject to}\quad & \beta \geqslant \|v\|^2, \\
& v \geqslant Ax - b, \\
& \|x\|^2 = \alpha, \\
& x \in \Omega^k, \\
& v \geqslant 0, \\
& \alpha_l^k \leqslant \alpha \leqslant \alpha_u^k,
\end{aligned} \tag{31}$$

where $\alpha_l^k = \|l^k\|^2$ and $\alpha_u^k = \|u^k\|^2$. At each node, instead of solving $(P2^k)$ directly, we obtain a lower bound for the optimal value of $(P2^k)$ by solving a special convex problem. To construct such a program, we can simply replace the equality $\|x\|^2 = \alpha$ by the inequality constraint (26). This nonlinear relaxation is denoted by $(RP2^k)$ and is obtained from $(RP2)$ by requiring x to belong to the set Ω^k and by constraining α accordingly (as in (31)). Alternatively, we can exploit the so-called reformulation–linearization–convexification technique (RLT), as described in [27], and consider the following relaxation:

$$LB(P2^k) \text{ Minimize } \quad \frac{\beta}{1+\alpha} \tag{32}$$

$$\text{subject to} \quad \beta \geqslant \|v\|^2, \tag{33}$$

$$v \geqslant Ax - b, \tag{34}$$

$$\sum_{i=1}^{n} y_i = \alpha, \tag{35}$$

$$x_i^2 \leqslant y_i \quad \forall i = 1, \ldots, n, \tag{36}$$

$$[(x_i - l_i^k)(u_i^k - x_i)]_{\mathrm{L}} \geqslant 0 \quad \forall i = 1, \ldots, n, \tag{37}$$

$$x \in \mathbf{X}, \tag{38}$$

$$x \in \Omega^k, \tag{39}$$

$$\alpha_l^k \leqslant \alpha \leqslant \alpha_u^k, \tag{40}$$

where $[\cdot]_{\mathrm{L}}$ denotes the linearization of the product term $[\cdot]$ under the substitution

$$y_i = x_i^2 \quad \forall i = 1, \ldots, n. \tag{41}$$

Note that in (37), known as *bound-factor-constraints* [27],

$$\begin{aligned}[(x_i - l_i^k)(u_i^k - x_i)]_{\mathrm{L}} &= [x_i u_i^k - l_i^k u_i^k + l_i^k x_i - x_i^2]_{\mathrm{L}} \\ &= x_i u_i^k - l_i^k u_i^k + l_i^k x_i - y_i.\end{aligned}$$

Problem $LB(P2^k)$ is a convex nonlinear program with a very special structure that can be efficiently solved by a nonlinear programming algorithm. It should be added that the constraint $\alpha = \|x\|^2$ has been convexified by introducing new variables y_i and additional constraints (35)–(37), where (36) and (37) approximate the nonconvex relationship $y_i = x_i^2, \forall i = 1, \ldots, n$. The following result holds in regard to $LB(P2^k)$.

Propoisition 2.3. *If $(\overline{x}, \overline{v}, \overline{\beta}, \overline{\alpha}, \overline{y})$ solves problem $LB(P2^k)$ with objective value $v(LB(P2^k))$, then $v(LB(P2^k))$ is a lower bound for the optimal value of $(P2^k)$. Moreover, if $\overline{x}_i = l_i^k$ or $\overline{x}_i = u_i^k$, for each $i = 1, \ldots, n$, then $\overline{y}_i = \overline{x}_i^2$, $\forall i = 1, \ldots, n$.*

Proof. Follows from Sherali and Tuncbilek [27]. □

It is also important to add that the relaxation $LB(P2^k)$ provides tighter lower bounds than the previous one $(RP2^k)$. In fact, from (35) and (36), we get that for any feasible solution to $LB(P2^k)$,

$$\|x\|^2 = \sum_{i=1}^{n} x_i^2 \leqslant \sum_{i=1}^{n} y_i = \alpha.$$

Therefore, the feasible region of problem $LB(P2^k)$ projected onto the (x, v, β, α)-space is included in that for $(RP2^k)$. For this reason, in this paper, we use relaxation $LB(P2^k)$ instead of $(RP2^k)$.

3. Overall algorithm

At each node of the proposed branch-and-bound procedure, starting with $\Omega^0 = \Omega$, problem $LB(P2^k)$ is solved to derive a lower bound on the node subproblem. Let $(\overline{x}, \overline{v}, \overline{\beta}, \overline{\alpha}, \overline{y})$ be the optimal solution obtained and let $v(LB(P2^k))$

be its optimal value. If

$$v(LB(P2^k)) \geqslant UB(1-\varepsilon) \tag{42}$$

for some tolerance $\varepsilon \geqslant 0$, we fathom this node. Otherwise, denoting K as the total number of nodes enumerated thus far, we partition this node into subproblems $(P2^{K+1})$ and $(P2^{K+2})$, based on the corresponding partition of Ω^k into Ω^{K+1} and Ω^{K+2} as follows:

$$\Omega^{K+1} = \{x : l_i^{K+1} \leqslant x_i \leqslant u_i^{K+1},\ \text{for } i = 1, \ldots, n\}$$

and

$$\Omega^{K+2} = \{x : l_i^{K+2} \leqslant x_i \leqslant u_i^{K+2},\ \text{for } i = 1, \ldots, n\},$$

where the bounds describing Ω^{K+1} and Ω^{K+2} are discussed next, within the following proposed branching strategy.

3.1. Branching variable selection scheme

Let

$$p \in \arg \max_{i=1,\ldots,n} \{\theta_i\} \quad \text{where } \theta_i = \overline{y}_i - \overline{x}_i^2 \ \text{ for } i = 1, \ldots, n. \tag{43}$$

If $\theta_p > 0$, then

$$\Omega^{K+1} = \{x : l_i^k \leqslant x_i \leqslant u_i^k, i = 1, \ldots, n, i \neq p, l_p^k \leqslant x_p \leqslant \overline{x}_p\}, \tag{44}$$

$$\Omega^{K+2} = \{x : l_i^k \leqslant x_i \leqslant u_i^k, i = 1, \ldots, n, i \neq p, \overline{x}_p \leqslant x_p \leqslant u_p^k\}. \tag{45}$$

If $\max_{i=1,\ldots,n} \{\theta_i\} = 0$, then by Proposition 2.3, $(\overline{x}, \overline{v}, \overline{\beta}, \overline{\alpha})$ is a feasible solution for $(P2^k)$, whose value equals $v(LB(P2^k))$, a lower bound that is achieved, and hence, is an optimal solution to $(P2^k)$. Therefore, we update the incumbent solution (along with UB), if necessary, and fathom the node in this case.

3.2. Computing upper bounds

Given the solution $(\overline{x}, \overline{v}, \overline{\beta}, \overline{\alpha}, \overline{y})$ to $LB(P2^k)$, we compute an upper bound UB by setting

$$UB \leftarrow \min\{UB, f(\overline{x}), \phi(\overline{x}, \overline{\beta}, \overline{v})\}, \tag{46}$$

where

$$f(\overline{x}) = \frac{\|(A\overline{x} - b)^+\|^2}{1 + \|\overline{x}\|^2}$$

and

$$\phi(\overline{x}, \overline{\beta}, \overline{v}) = \frac{\widehat{\beta}}{1 + \|\widehat{x}\|^2},$$

where $(\widehat{x}, \widehat{\beta}, \widehat{v})$ is a stationary point to $(P1)$ that is obtained by starting at the initial point $(\overline{x}, \overline{\beta}, \overline{v})$ and applying a nonlinear programming algorithm (we used GAMS–MINOS [28,29] for this purpose).

Remark 3.1. As reported in Section 4, it is further advisable to compute an upperbound at the initial root node by obtaining a stationary point to the following equivalent representation of the original problem:

$$(P3)\ \text{Minimize} \quad \frac{\|v\|^2}{1+\|x\|^2} \tag{47}$$

$$\text{subject to} \quad v \geqslant Ax - b, \tag{48}$$

$$v \geqslant 0, \tag{49}$$

$$x \in \mathbf{X}. \tag{50}$$

3.3. Computing an initial feasible solution

In order to induce a faster convergence toward optimality for each subproblem $LB(P2^k)$, it is important to start with an initial point, $(\hat{x}, \hat{v}, \hat{\beta}, \hat{\alpha}, \hat{y})$, say, that is feasible for $LB(P2^k)$. To do this, we solve the following program:

$$\hat{x} = \arg \min_{x \in \mathbf{X} \cap \{x : l^k \leqslant x \leqslant u^k\}} \|x - x^*\|_{l_1}, \tag{51}$$

where x^* is the current best known solution. Note that for $\mathbf{X}$ polyhedral (including $\mathbf{X} = \mathbb{R}^n$), (51) can be solved via the LP:

$$\text{Minimize} \left\{ \sum_{i=1}^{n} z_i : z_i \geqslant x_i - x_i^* \text{ and } z_i \geqslant x_i^* - x_i, \quad \forall i = 1, \ldots, n,\ x \in \mathbf{X}, l^k \leqslant x \leqslant u^k \right\}. \tag{52}$$

After obtaining $\hat{x}$, we then compute the remainder of the initial solution as

$$\hat{v} = \max\{0, A\hat{x} - b\},$$

$$\hat{\beta} = \|\hat{v}\|^2,$$

$$\hat{y}_i = \hat{x}_i^2 \quad \forall i = 1, \ldots, n,$$

$$\hat{\alpha} = \|\hat{x}\|^2. \tag{53}$$

The following result holds.

Propoisition 3.2. *The solution* $(\hat{x}, \hat{v}, \hat{\beta}, \hat{\alpha}, \hat{y})$ *as given by* (51)–(53) *is feasible to* $LB(P2^k)$.

Proof. Feasibility to (33)–(36) and (38)–(40) is evident, by the construction of the solution (51)–(53) and the bounds derived based on $l^k \leqslant x \leqslant u^k$. Moreover, feasibility to the RLT constraints (37) follows by construction, since the product relationships in (41) are satisfied via (53). This completes the proof. □

3.4. Algorithm and convergence theorem

In this subsection, we describe the main steps of the algorithm. To do this, we start by describing the following parameters:

k index for the current subproblem under analysis,
K total number of nodes enumerated (in addition to the root node),
UB best known upper bound,
x_{inc} incumbent solution,
L queue of indices of subproblems created but not expanded,
$LB(P2^k)$ lower-bounding problem as described by (32)–(40),
$(x^k, v^k, \beta^k, \alpha^k, y^k)$ optimal solution obtained for $LB(P2^k)$,

Ω^k as defined in (30),
ε optimality tolerance,
$v(.)$ optimal value of problem (.).

Algorithm RLT-BB.

(0) (**Initialization**) Let $K = k = 0$, $L = \emptyset$, $UB = \infty$, and $\varepsilon \geqslant 0$ (we chose $\varepsilon = 10^{-6}$). Solve Problem $LB(P2^k)$. Update UB through (46) and set x_{inc} equal to the corresponding best x solution found.
(1) (**Pick next node**) If $L = \emptyset$ then stop; otherwise, find $k \in \arg\min\{v(LB(P2^t)) : t \in L\}$.
(2) (**Dequeue**) Set $L \leftarrow L - \{k\}$.
(3) (**Branching rule**) Find a branching index p via (43). If $\theta_p > 0$, go to Step (4). Otherwise, update UB and x_{inc} using the solution to $LB(P2^k)$, remove any indices t from L for which $v(LB(P2^t)) \geqslant UB(1 - \varepsilon)$, and go to Step (1).
(4) (**Solve**, **Update**, **and Queue**) Set $i = 1$.
   (4.1) Define Ω^{K+i} according to (44)–(45). Solve problem $LB(P2^{K+i})$.
   (4.2) If $v(LB(P2^{K+i})) < UB(1 - \varepsilon)$ then go to Step (4.3); otherwise, go to Step (4.5).
   (4.3) Update UB according to (46). If UB was updated remove all indices $t \in L$ for which$v(LB(P2^t)) \geqslant UB(1 - \varepsilon)$ and put x_{inc} equal to the revised incumbent x solution found.
   (4.4) Set $L \leftarrow L \cup \{K + i\}$.
   (4.5) If $i = 2$, set $K \leftarrow K + 2$ and go to Step (1); otherwise, let $i = 2$ and go to Step (4.1).

To complete this section we present the convergence theorem for this algorithm.

Propoisition 3.3. *The algorithm RLT-BB, when run with* $\varepsilon \equiv 0$, *either terminates finitely with a global optimum to the problem, or else an infinite branch-and-bound tree is generated, such that any accumulation point of the relaxation lower-bounding problem solution along any infinite branch of the enumeration tree is a global optimum for problem* $(P2)$.

Proof. The case of finite convergence is obvious from the validity of the bounds derived by the algorithm. Now, suppose that an infinite branch-and-bound tree is generated. Then there exists a branching index p that is selected infinitely often along an infinite branch of this tree. Let S_1 be an infinite sequence of nested nodes generated by the sequence of such branchings over p, and let $LB(P2^k)$ and $(x^k, \upsilon^k, \beta^k, \alpha^k, y^k)$ for $k \in S_1$ be the corresponding lower-bounding problems and optimal solutions, respectively. Over some convergent subsequence ($k \in S_2 \subseteq S_1$) suppose that

$$(x^k, \upsilon^k, \beta^k, \alpha^k, y^k, l^k, u^k) \underset{k \in S_2}{\longrightarrow} (x^*, \upsilon^*, \beta^*, \alpha^*, y^*, l^*, u^*).$$

Then, using the proof in [27], we see that in the limit $x_p^* = l_p^*$ or $x_p^* = u_p^*$. Moreover, by Proposition 2.3, this yields $\theta_p = 0$, in the limit, where θ_i is defined in (43). Furthermore, again by (43), this gives $\theta_i = 0, \forall i = 1, \ldots, n$, in the limit. Hence, since the limiting solution $(x^*, \upsilon^*, \beta^*, \alpha^*)$ is feasible to problem (P2) with objective value V^*, we get

$$V^* \geqslant v(P2). \tag{54}$$

However, the least lower bound node selection criterion ensures that $v(LB(P2^k)) \leqslant v(P2), \forall k \in S_2$. In the limit, this yields $V^* \leqslant v(P2)$. This, together with (54), yields $V^* = v(P2)$ and so $(x^*, \upsilon^*, \beta^*, \alpha^*)$ solves problem $(P2)$. This completes the proof. $\square$

3.5. An example

Consider the following inconsistent system of inequalities, illustrated in Fig. 1:

$$\begin{cases} -x_1 - x_2 & \leqslant -7 \\ x_2 & \leqslant 3 \\ 2x_1 - x_2 & \leqslant -2. \end{cases}$$

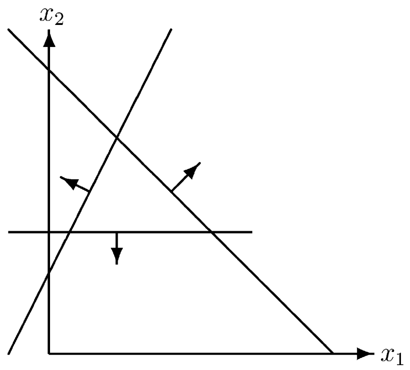

Fig. 1. Inconsistent system for the illustrative example.

Suppose that we seek an optimal correction on the domain **X** defined by

$$1 \leqslant x_i \leqslant 5 \quad \text{for } i = 1, 2.$$

To apply the algorithm, we have

$$l = [1, 1] \quad \text{and} \quad u = [5, 5].$$

Hence,

$$\Omega^0 = \{x : 1 \leqslant x_1 \leqslant 5, 1 \leqslant x_2 \leqslant 5\},$$

$$\alpha_l^0 = 2.0000, \quad \alpha_u^0 = 50.0000.$$

The solution to $LB(P2^0)$ is as follows:

$$x^0 = (1.6079, 4.6618), \quad y^0 = (4.6473, 22.9709),$$

$$\alpha^0 = 27.6182, \quad \beta^0 = 3.6018,$$

$$v^0 = (0.7303, 1.6618, 0.5539),$$

and the lower bound is given by

$$v(LB(P2^0)) = 0.1259.$$

The upper bound can be updated to 0.1423 according to (46), and the corresponding incumbent solution is $x_{\text{inc}} = x^0$. In order to apply the branching rule to partition Ω^0 we get:

$$\theta_1 = \overline{y}_1 - \overline{x}_1^2 = 2.0620,$$

$$\theta_2 = \overline{y}_2 - \overline{x}_2^2 = 1.2385,$$

$$\max_{i=1,2}\{\theta_i\} = 2.0620 \quad \text{and} \quad p = \arg\max_{i=1,2}\{\theta_i\} = 1.$$

We thus obtain Ω^1 and Ω^2 based on the partition of $[l_1, u_1] = [1, 5]$ into $[l_1^1, u_1^1] = [1, 1.6079]$ and $[l_1^2, u_1^2] = [1.6079, 5]$:

$$\Omega^1 = \{x : 1 \leqslant x_1 \leqslant 1.6079, 1 \leqslant x_2 \leqslant 5\},$$

$$\Omega^2 = \{x : 1.6079 \leqslant x_1 \leqslant 5, 1 \leqslant x_2 \leqslant 5\}.$$

Now, solving $LB(P2^1)$, we get:

$$x^1 = (1.5668, 4.6576), \quad y^1 = (2.4782, 22.9457),$$

$$\alpha^1 = 25.4239, \quad \beta^1 = 3.5758,$$

$$v^1 = (0.7756, 1.6576, 0.4760), \quad v(LB(P2^1)) = 0.1353$$

and the upper bound UB is updated to 0.1422, with $x_{\text{inc}} = x^1$.

Likewise, $LB(P2^2)$ leads to the following solution:

$$x^2 = (1.6249, 4.6790), \quad y^2 = (2.6979, 23.0738),$$

$$\alpha^2 = 25.7718, \quad \beta^2 = 3.6294,$$

$$v^2 = (0.6961, 1.6790, 0.5709), \quad v(LB(P2^2)) = 0.1356.$$

The upper bound is updated to 0.1421, with $x_{\text{inc}} = x^2$. Now, $K = 2$, $L = \{1, 2\}$ and since

$$k \in \arg\min\{v(LB(P2^t)) : t \in L\} = 1,$$

the first subproblem is chosen to apply the branching rule and to partition the corresponding hyperrectangle Ω^1. This yields:

$$\max_{i=1,2}\{\theta_i\} = 1.2523 \quad \text{and} \quad p = \arg\max_{i=1,2}\{\theta_i\} = 2.$$

Accordingly, Ω^3 and Ω^4 are constructed based on the partition of $[l_2^1, u_2^1] = [1, 5]$ into $[l_2^3, u_2^3] = [1, 4.6576]$ and $[l_2^4, u_2^4] = [4.6576, 5]$, respectively:

$$\Omega^3 = \{x : 1 \leqslant x_1 \leqslant 1.6079,\ 1 \leqslant x_2 \leqslant 4.6576\},$$

$$\Omega^4 = \{x : 1 \leqslant x_1 \leqslant 1.6079,\ 4.6576 \leqslant x_2 \leqslant 5\}.$$

The solution to $LB(P2^3)$ is as follows:

$$x^3 = (1.5686, 4.6575), \quad y^3 = (2.4830, 21.6929),$$

$$\alpha^3 = 24.1759, \quad \beta^3 = 3.5764,$$

$$v^3 = (0.7738, 1.6575, 0.4797), \quad v(LB(P2^3)) = 0.1421 = UB.$$

Consequently, the node corresponding to subproblem $LB(P2^3)$ is fathomed. Hence, L is updated to $\{2\}$ and $LB(P2^4)$ is solved, producing the following solution:

$$x^4 = (1.5880, 4.7563), \quad y^4 = (2.5335, 22.6460),$$

$$\alpha^4 = 25.1795, \quad \beta^4 = 3.6906,$$

$$v^4 = (0.6557, 1.7563, 0.4198), \quad v(LB(P2^4)) = 0.1410.$$

The upper bound is updated to 0.1412 with $x_{\text{inc}} = x^4$. Thus, $K = 4$, $L = \{2, 4\}$, and $LB(P2^2)$ is the next subproblem that is selected to continue the search, because

$$\begin{aligned} v(LB(P2^2)) &= \min\{v(LB(P2^2)), v(LB(P2^4))\} \\ &= \min\{0.1356, 0.1410\} \\ &= 0.1356. \end{aligned}$$

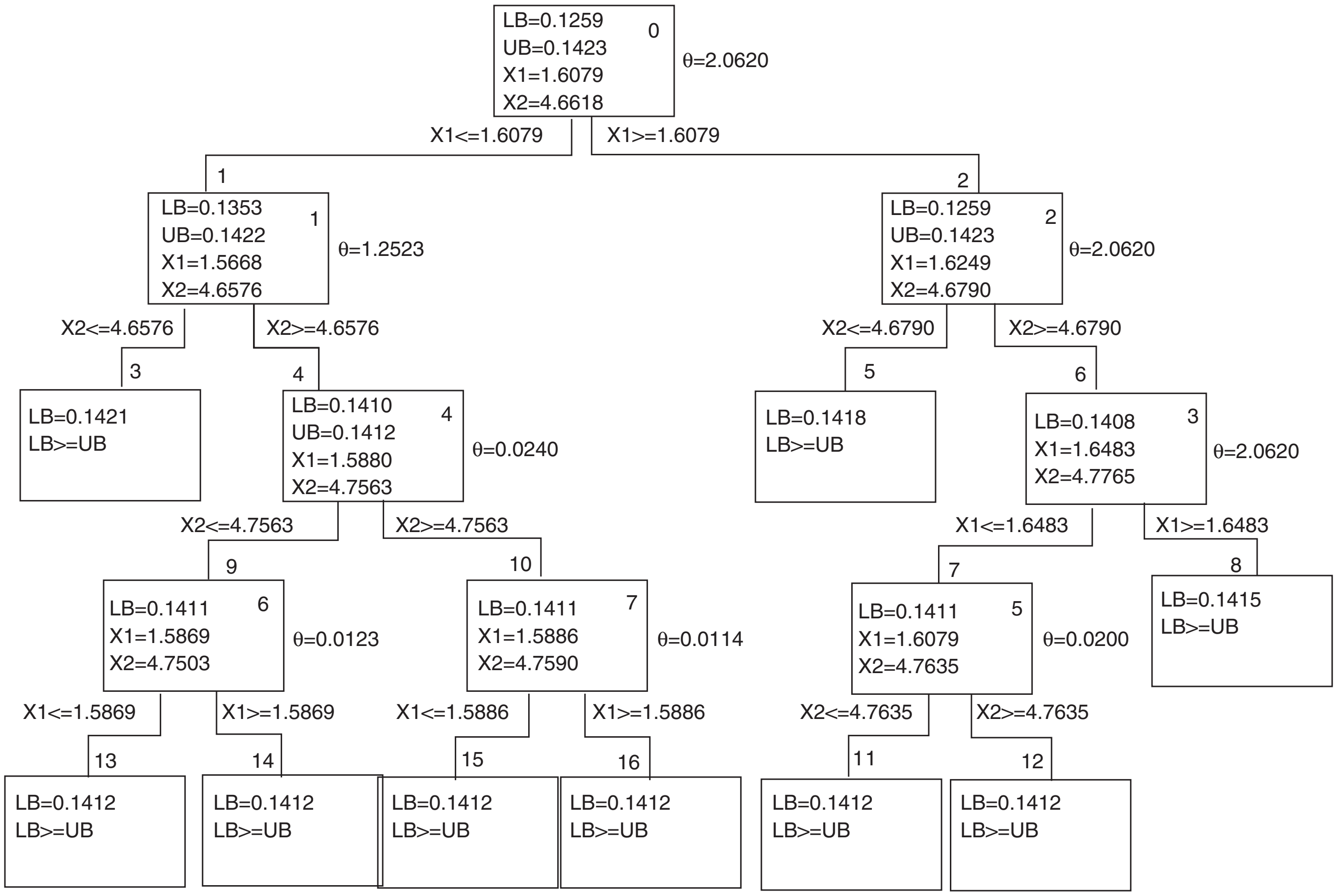

Fig. 2. Graphical representation for the solution of the illustrative example.

Now, $\max_{i=1,2}\{\theta_i\} = 1.1810$ and $p = 2$, and proceeding as above, the partitioning of Ω^2 results in

$$\Omega^5 = \{x : 1.6079 \leqslant x_1 \leqslant 5, 1 \leqslant x_2 \leqslant 4.6790\},$$

$$\Omega^6 = \{x : 1.6079 \leqslant x_1 \leqslant 5, 4.6790 \leqslant x_2 \leqslant 5\}.$$

Since $v(LB(P2^5)) = 0.1418$ is greater than the upper bound, this node is fathomed. The optimal value to $LB(P2^6)$ is $v(LB(P2^6)) = 0.1408$. Hence, $K = 6$, L is updated to $L = \{4, 6\}$. The next subproblem to be picked for branching is $LB(P2^6)$. Now $\max\{\theta_i\} = 0.1356$, $p = 1$ and from the partition of Ω^6, we obtain

$$\Omega^7 = \{x : 1.6079 \leqslant x_1 \leqslant 1.6483, 4.6790 \leqslant x_2 \leqslant 5\},$$

$$\Omega^8 = \{x : 1.6483 \leqslant x_1 \leqslant 5, 4.6790 \leqslant x_2 \leqslant 5\}.$$

The optimal values of $v(LB(P2^7))$ and $v(LB(P2^8))$ are, respectively, 0.1411 and 0.1415. Since this last value is greater than the upper bound, we only include node 7 in L, which yields $L = \{4, 7\}$ with $K = 8$. This process continues until a global minimum is found. The search tree in Fig. 2 indicates how the algorithm has performed in order to find such a global minimum. For each node, the optimal value of the lower bound problem (LB), the upper bound (UB), when it is updated at that node, and the optimal solution $x = (x_1, x_2)$ of the corresponding relaxation are shown. In the right-upper corner of each box (node), we indicate the order in which each node is selected from the queue L. The number appearing above each box gives the node number in the order in which it is generated. The value of $\theta = \max_{i=1,2}\{\theta_i\}$ is given to the right of each box.

The search inspects 16 nodes, but only seven are introduced in the queue L for branching. The variable that induced the partition of Ω^k for each branch is also depicted in Fig. 2. The optimal solution obtained is the incumbent solution corresponding to the last update of the upper bound (0.1412), which is given by $x^4 = (1.5880, 4.7563)$. The optimal correction of the matrix $[A, -b]$, as given by (7) and (8), is

$$[H, p] = -\frac{1}{26.1441}\begin{bmatrix} 0.0251 \\ 0.0672 \\ 0.0161 \end{bmatrix}[1.5880 \;\; 4.75633 \;\; 1] = \begin{bmatrix} -0.03983 & -0.1193 & 0.0251 \\ -0.1067 & -0.3195 & 0.0672 \\ -0.0255 & -0.0764 & 0.0161 \end{bmatrix}.$$

Thus, the corrected linear system is

$$\begin{cases} -1.0398x_1 & -1.1193x_2 & \leqslant -6.9749, \\ -0.1067x_1 & +0.6805x_2 & \leqslant \quad 3.0672, \\ \;\;1.9745x_1 & -1.0764x_2 & \leqslant -1.9839. \end{cases}$$

As expected, upon the substitution of $(x_1, x_2) = (1.5880, 4.7563)$ the inequalities are verified to be satisfied as equalities.

4. Computational experience

In order to test the performance of the algorithm we report some computational results for a set of infeasible linear systems of the type $\{x \in \mathbb{R}^n : Ax \leqslant b, l \leqslant x \leqslant u\}$, where A is a real matrix of order $m \times n$ and b is a real vector of size m.

We consider two sets of problems. The first group is taken from a set of infeasible linear programming problems[3] selected from Netlib.[4] For the application of the algorithm in its current version, we have added lower and upper bounds ($l_i = 1$ and $u_i = 5$, for all i) and transformed each equality into two inequalities in each problem. Table 1 includes the number of constraints m and variables n of the chosen Netlib problems.

The second set of test problems consists of matrices A and vectors b, such that $A^{\mathrm{T}} = [A_1^{\mathrm{T}}, A_2^{\mathrm{T}}]$, $b^{\mathrm{T}} = [b_1^{\mathrm{T}}, b_2^{\mathrm{T}}]$, so that $Ax \leqslant b$ is infeasible and has an optimal correction given by the TLS solution vector x_{TLS} [30], of the system $A_1 x = b_1$. To construct these problems [22,23] a nonsingular matrix B and a vector $\hat{b}$ are first considered such that $Bx \leqslant \hat{b}$ has at least a feasible solution. Then a new constraint $\gamma^{\mathrm{T}} x \leqslant \delta$ is added in order to render the following system infeasible:

$$A_1 x \leqslant b_1 \quad \Leftrightarrow \quad \begin{bmatrix} B \\ \gamma^{\mathrm{T}} \end{bmatrix} x \leqslant \begin{bmatrix} \hat{b} \\ \delta \end{bmatrix}.$$

A simple choice for this vector γ and scalar δ is as follows:

$$\begin{cases} \gamma = B^{\mathrm{T}}\psi, \\ \delta < \psi^{\mathrm{T}}\hat{b}, \end{cases}$$

Table 1
Characteristics of test problems from Netlib

Name	m	n
Galenet	10	8
Itest2	9	4
Itest6	13	8
Bgprtr	34	34
Forest6	96	95
Klein1	54	54
Woodinfe	70	89

[3] Test problems collected by John W. Chinneck.

[4] The Netlib repository contains freely available software, documents, and databases. The repository is maintained by AT&T Bell Laboratories, the University of Tennessee, and Oak Ridge National Laboratory among individual contributions.

Table 2
Dimension of test problems

Name	m	n
Prob6–Prob10	20	10
Prob11–Prob15	30	15
Prob16–Prob20	40	20

Table 3
Computational results

	RLT-BB						
Problems	ND	CPU	ITER	INITUB	VALOPT	NDOPT	NUPDUB
Galenet	1	0	133	3.7313	3.7313	1	0
Itest2	27	0	751	0.4257	0.4257	1	0
Itest6	1	0	37	82 654 535.9118	82 654 535.9118	1	0
Bgprtr	1	0.01	194	1264.5915	1264.5915	1	0
Forest6	1	0.08	471	3458.7896	3458.7896	1	0
Klein1	221	4.62	23 919	34.6664	34.6664	1	0
Woodinfe	1	0	8193	0.0019	0.0019	1	0
Prob4	5	0	217	157.6815	157.6815	1	0
Prob5	4	0	187	262.8295	262.8295	1	0
Prob6	5	0.01	211	315.1673	315.1673	1	0
Prob7	68	0	1196	214.8726	214.8726	1	0
Prob8	1	0	103	2187.0132	2187.0132	1	0
Prob9	20	0.01	522	149.3484	149.3484	1	0
Prob10	29	0	604	250.8560	250.8560	1	0
Prob11	26	0.01	500	396.0405	396.0405	1	0
Prob12	3	0.01	184	1130.5302	1130.5302	1	0
Prob13	12	0.02	489	679.4526	679.4526	1	0
Prob14	6	0.02	382	756.9513	756.9513	1	0
Prob15	476	1.08	23 552	365.3541	364.4484	424	13
Prob16	26	0.12	2052	1121.5359	1121.5359	1	0
Prob17	27	0.13	1980	1092.9802	1092.9802	1	0
Prob18	9	0.04	623	1750.9055	1738.2852	8	4
Prob19	25	0.17	1999	1104.5614	1104.5614	1	0
Prob20	199	1.26	18 410	944.7827	944.7827	1	0

where ψ is a negative random vector. It is now possible to find the TLS solution, x_{TLS}, of the system $A_1x = b_1$. The matrix A and the vector b are then constructed by augmenting A_1 and b_1, respectively, by a matrix $A_2 \in \Re^{(m-(n+1))\times n}$ and a vector $b_2 \in \Re^{m-(n+1)}$, such that x_{TLS} is feasible for the set of constraints $A_2x \leqslant b_2$. Again for the application of the current version of the algorithm, we have introduced for each problem a set of finite lower and upper bounds ($l_i = 1$ and $u_i = 5$, for all i). Table 2 summarizes the dimensions of these problems.

All the tests have been performed on a Pentium IV (Intel) with hyperthreading, CPU 3.0 hz, 2GB RAM, and operating system Linux. The method was implemented in the General Algebraic Modeling System (GAMS) language (Rev 118 Linux/Intel) and the NLP solver MINOS (version 5.5) has been used to compute the lower bounds required at each node and for computing ϕ in (46), as well as for determining a stationary point to the nonlinear program (47)–(50) that provides an upper bound at the root node.

Table 3 reports the following information for each test problem:

ND— total number of nodes in the tree,
CPU— total CPU time in seconds,
ITER— total number of MINOS iterations,

INITUB— initial upper bound (obtained at the root node),
VALOPT— optimal value,
NDOPT— node at which the optimal solution was obtained,
NUPDUB— number of upper bound updates.

In all the runs, we have used the stopping criteria (42), with $\varepsilon = 10^{-6}$. We have successfully found an optimal solution for all the test problems by using this tolerance, with the exception of the problems Klein1, Woodinfe, and Prob15. For these three problems, the fathoming criterion used was

$$UB - v(LB(P2^k)) \leqslant \tilde{\varepsilon} \max\{1, UB\}.$$

The smallest values of $\tilde{\varepsilon}$ that led to a successful termination of the algorithm (in fewer than 500 nodes) are given below:

- $\tilde{\varepsilon} = 10^{-2}$ for problem Klein1.
- $\tilde{\varepsilon} = 10^{-3}$ for problem Woodinfe.
- $\tilde{\varepsilon} = 10^{-6}$ for problem Prob15.

The results show that the optimal global solution is almost always (except for two of the 24 instances) the upper bound computed at the root node. This is a key feature of the tree search and substantially reduces the computational time. There are three problems for which the algorithm has been unable to terminate with the normal stopping criterion as discussed above. These three test problems indicate the need for computing better lower bounds for reducing the search. Also, the algorithm appears to solve medium-scale instances with a reasonable computational effort. Although we address in this paper a problem that is different from the one solved in [31], because no lower and upper bounds were included in the set **X** in [31], and we have used a different computational platform, we can still make some comparisons with the enumerative procedure discussed there: the algorithm introduced in this paper performs much better in terms of the computational effort required to find a global optimum.

5. Conclusions and future work

In this paper, we have proposed a method for obtaining an optimal solution for a nonlinear nonconvex program that arises in a TLS approach for finding a correction of an infeasible linear system of inequalities. Using an equivalent reformulation, we solved this problem to global optimality via a new branch-and-bound algorithm. This procedure exploits the reformulation–linearization convexification technique (RLT) [27] to convexify a relaxation for deriving lower bounds on the optimal value of the original problem. Together with a framework to obtain upper bounds, we developed a tree search procedure based on a partitioning of the domain of the original variables, and established global convergence of the proposed algorithm. Computational experience reveals that the approach is suitable for handling problems having m and n ranging to about 100. This is not too restrictive, especially considering the application of this theory in the context of infeasible problem corrections. In many of these cases, we are required to maintain some constraints unchanged, and so the nonlinear problem we formulate is defined only over a particular subset of constraints chosen by the user or by an expert system [9,11]. Also, this formulation could be applied to irreducible inconsistent systems (IIS) as identified in post-optimality analyses [13,14], where each IIS might involve only a relatively small subset of the original LP constraints and variables.

We would also like to point out the importance of finding global optimal corrections, for instance, in the framework of constraint satisfaction techniques, as in other contexts. When dealing with real models, it is essential to make as small changes as possible, in order to mitigate the risk of invalidating the corrected model. Our approach provides such a facility of determining a minimallyperturbed feasible model in a least squares sense.

The solution of problems of this type with other definitions for the compact set **X** is a useful topic for future investigation. We also recommend considering a partitioning of the constraints into two groups, namely, soft and hard constraints, where the set of hard constraints is assumed to be invariant and cannot be corrected. This situation is typical in the analysis of problems that arise in real-life applications.

Acknowledgments

This research has been partially supported by the National Science Foundation, under Grant number DMI 0094462, by project FCT-POCTI/35059/MAT/2000, Portugal, and by the Centro de Matemática e Aplicações (CMA), FCT UNL, Portugal. We would also like to acknowledge Luís M. Fernandes for his help on the computational experience and the referees for their valuable comments and suggestions.

References

[1] Chinneck JW. MINOS: Infeasibility analysis using MINOS. Computers and Operations Research 1994;21:1–9.
[2] Greenberg HJ. Computer-assisted analysis for diagnosing infeasible or unbounded linear programs. Mathematical Programming 1987;31: 79–97.
[3] Greenberg HJ. Consistency, redundancy and implied equalities in linear systems. Annals of Mathematics and Artificial Intelligence 1993;17: 37–83.
[4] Greenberg HJ. Enhancements of ANALYSE: a computer-assisted analysis system for mathematical programming models and solutions. ACM Transactions on Mathematical Software 1993;19:233–56.
[5] Greenberg HJ, Murphy FH. Approaches to diagnosing infeasible linear programs. ORSA Journal on Computing 1991;31:79–97.
[6] Amaldi E, Kann V. The complexity and approximability of finding maximum feasible subsystems of linear relations. Theoretical Computer Science 1995;147:181–210.
[7] Amaldi E, Pfetsch ME, Trotter JLE. On the maximum feasible subsystems, IISs and IIS-hypergraphs. Mathematical Series A 2003;95:533–54.
[8] Chakravarti N. Some results concerning post-infeasibility analysis. European Journal of Operational Research 1994;73:139–43.
[9] Chinneck JW. Finding a useful subset of constraints for analysis in an infeasible linear program. INFORMS Journal on Computing 1997;9: 164–74.
[10] Chinneck JW, Saunders MA. MINOS(iis) version 4.2: analysing infeasibilities in linear programs. European Journal of Operational Research 1995;81:217–8.
[11] Chinneck JW, Dravnieks EW. Locating minimal infeasible constraint sets in linear programs. ORSA Journal on Computing 1991;3:157–68.
[12] Greenberg HJ. How to analyze the results of linear programs—part 3: Infeasibility diagnoses. Interfaces 1993;23:120–39.
[13] Loon JNM. Irreducibly inconsistent systems of linear inequalities. European Journal of Operational Research 1981;8:282–8.
[14] Wang HF, Huang CS. Inconsistent structures of linear systems. International Journal of General Systems 1992;21:65–81.
[15] Guieu O, Chinneck JW. Analyzing infeasible mixed-integer and integer linear programs. INFORMS Journal on Computing 1999;11:63–77.
[16] Holzbaur C, Menezes F, Barahona P. Defeasibility in CLP(Q) through generalised slack variables. In: Freuder EC, editor. Proceedings of CP'96 second international conference on principles and practice of constraint programming. Lecture notes in computer science, vol. 1118. Berlin: Springer; 1996. p. 209–23.
[17] Pena J. Understanding the geometry of infeasible perturbations of a conic linear system. SIAM Journal on Optimization 2000;10:534–50.
[18] Renegar J. Some perturbation theory for linear programming. Mathematical Programming 1994;65:73–91.
[19] Vera JR. Ill-posedness and the complexity of deciding existence of solutions to linear programs. SIAM Journal on Optimization 1996;6: 549–69.
[20] Roodman GM. Post-infeasibility analysis in linear programming. Management Science 1979;25:916–22.
[21] Vatolin AA. An lp-based algorithm for the correction of inconsistent linear equation and inequality systems. Optimization 1992;24:157–64.
[22] Amaral P. Contribuições para o estudo de sistemas lineares inconsistentes. PhD dissertation, Faculty of Science and Technology, UNL, Lisbon, Portugal, 2001 (in Portuguese).
[23] Amaral P, Barahona P. Connections between the total least squares and the correction of an infeasible system of linear inequalities. Linear Algebra and its Applications 2005;395:191–210.
[24] Amaral P, Barahona P. About infeasibility in the constraints of a linear model. Ricerca Operativa 1999;92:49–67.
[25] Amaral P, Barahona P. A framework for optimal correction of inconsistent linear constraints. Constraints 2005;10:67–86.
[26] Bazaraa MS, Sherali HD, Shetty CM. Nonlinear programming: theory and algorithms. third ed., New York, NY: Wiley; 2006.
[27] Sherali HD, Tuncbilek CH. A global optimization algorithm for polynomial programming problems using a reformulation–linearization technique. Journal of Global Optimization 1992;2:101–12.
[28] Rosenthal RE. GAMS—a user's guide ⟨http://www.gams.com/docs/document.htm⟩.
[29] Murtagh BA, Saunders MA, Murray W, Gill PE, Raman R, Kalvelagen E. MINOS—NLP solver from Stanford University. ⟨http://www.gams.com/docs/document.htm⟩.
[30] Huffel SV. The total least squares problem: computational aspects and analysis. Frontiers in applied mathematics. Philadelphia, PA: SIAM; 1991.
[31] Amaral P, Barahona P. On optimal correction of inconsistent linear constraints. In: Hentenryck PV, editor. Principles and practice of constraint programming, CP'2002 (Proceedings), Lecture notes in computer science, vol. 2470; 2002. pp. 33–46.