Feasibility problems with complementarity constraints ☆

R. Andreani[a], J. J. Júdice[b,1], J. M. Martínez[a], T. Martini[a,*]

[a] *Department of Applied Mathematics, IMECC-UNICAMP, State University of Campinas, Rua Sérgio Buarque de Holanda, 651 Cidade Universitária "Zeferino Vaz", Distrito Barão Geraldo, 13083-859 Campinas SP, Brazil*

[b] *Instituto de Telecomunicações, University of Coimbra, Coimbra, Portugal*

Abstract

A Projected-Gradient Underdetermined Newton-like algorithm will be introduced for finding a solution of a Horizontal Nonlinear Complementarity Problem (HNCP) corresponding to a feasible solution of a Mathematical Programming Problem with Complementarity Constraints (MPCC). The algorithm employs a combination of Interior-Point Newton-like and Projected-Gradient directions with a line-search procedure that guarantees global convergence to a solution of HNCP or, at least, a stationary point of the natural merit function associated to this problem. Fast local convergence will be established under reasonable assumptions. The new algorithm can be applied to the computation of a feasible solution of MPCC with a target objective function value. Computational experience on test problems from well-known sources will illustrate the efficiency of the algorithm to find feasible solutions of MPCC in practice.

Keywords: Global optimization, Nonlinear programming, Nonlinear Systems of Equations, Complementarity Problems, Mathematical Programming with

☆This work was supported by PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.164/2003 - APQ1), FAPESP (Grants 06/53768-0, 2012/10444-0 and CEPID Industrial Mathematics 2011/51305-0) and CNPq.

*Corresponding author

Email addresses: andreani@ime.unicamp.br (R. Andreani), Joaquim.Judice@co.it.pt (J. J. Júdice), martinez@ime.unicamp.br (J. M. Martínez), tiaramartini@gmail.com (T. Martini)

[1] The research of Joaquim J. Júdice was partially supported in the scope of R&D Unit UID/EEA/50008/2013, financed by the applicable financial framework FCT/MEC through national funds and when applicable co-funded by FEDER - PT2020 partnership agreement.

Complementarity Constraints.

1. Introduction

A Mathematical Programming Problem with Complementarity Constraints (MPCC) [34, 36, 38] can be defined in the form

$$\text{Minimize } \varphi(x,y,w) \text{ subject to } H(x,y,w)=0 \text{ and } \min\{x,w\}=0, \tag{1}$$

where $x, w \in I\!R^n$, $y \in I\!R^m$, while $\varphi : I\!R^{2n+m} \to I\!R$, and $H : I\!R^{2n+m} \to I\!R^r$ are continuously differentiable functions. The feasible set of MPCC will be denoted by D and $\min\{x,w\}$ denotes a vector of components $\min\{x_i, w_i\}$, $i = 1, \ldots, n$. For all $i = 1, \ldots, n$, the variables x_i, w_i are said to be complementary and satisfy:

$$x_i \geqslant 0,\ w_i \geqslant 0,\ x_i w_i = 0,\ i = 1, \ldots, n. \tag{2}$$

MPCC has appeared frequently in optimization models and has significant applications in different areas of science, engineering and economics [34, 36, 38]. Many theoretical and application papers in Operations Research, as well as survey papers on related topics [7, 8, 25, 28, 32], have been devoted to this problem in recent years. For example, transport network models were considered in [17, 42, 43], bilevel optimization in [28], variational inequality formulations in [41], multiobjective problems with complementarity constraints in [33, 46], electricity markets in [10, 19, 22, 45], quadratic programming with complementarity constraints in [39], optimality conditions in [37], order-value applications in [1], and oligopolistic equilibrium in [44], among others.

Clearly, MPCC can be seen as a Nonlinear Programming Problem where the n complementarity constraints $\min\{x_i, w_i\} = 0$ are replaced with (2) or even with $x^\top w = 0$, $x \geqslant 0$, $w \geqslant 0$. Attempts for solving MPCC by means of nonlinear programming algorithms present some difficulties, mainly because these algorithms may converge to points from which there exist obvious first-order descent directions. This issue is a consequence of the so-called double zeros or biactive indices, i.e., feasible points satisfying at least a constraint $x_i w_i = 0$ with both variables x_i and w_i equal to zero. These difficulties have motivated much research on weak forms of stationarity [13, 20, 34, 36, 38, 40] and several algorithms have been designed to compute such weak stationary points [4, 5, 6, 11, 14, 15, 16, 20, 23, 24, 26, 30, 34, 36, 38].

In this paper, we will discuss how to compute a feasible solution of the MPCC, that is, a solution of the following Horizontal [18] (possibly nonlinear) Complementarity Problem (HNCP):

$$\begin{bmatrix} H(x,y,w) \\ x_1 w_1 \\ \vdots \\ x_n w_n \end{bmatrix} = 0, \; x \geq 0, \; w \geq 0. \tag{3}$$

We will assume that $r \leq m + n$, so that the number of equations in (3) is smaller than or equal to the number of unknowns. The case in which $r = m+n$ has been studied in [3]. The case of H affine has been thoroughly discussed in the literature (see for instance [25] for a recent survey). The HNCP is NP-hard in this case [35] but there are many MPCCs where finding a single feasible solution can be considered as an easy task [25].

The problem of finding a feasible point of MPCC at which the objective function achieves a target value c_t is naturally formulated as follows:

$$\varphi(x,y,w) \leqslant c_t, \; H(x,y,w) = 0, \; x \geqslant 0, \; w \geqslant 0 \text{ and } x^\top w = 0. \tag{4}$$

Note that the problem (4) can be written as a standard HNCP adding two auxiliary variables v_1 and v_2, as follows:

$$\begin{gathered} \varphi(x,y,w) + v_1 = c_t, H(x,y,w) = 0, v_1 v_2 = 0, x_i w_i = 0, i = 1, \ldots, n, \\ v_1 \geq 0, \; v_2 \geq 0, \; x \geq 0, \; \text{and} \; w \geq 0. \end{gathered} \tag{5}$$

In this paper we will extend the algorithm introduced in [3], which deals with the case $r = n+m$, for the underdetermined HNCP (3) where r may be smaller than $n+m$. The Projected-Gradient Underdetermined Newton-like algorithm (PGUN) combines fast interior-point iterations with projected-gradient steps. A line-search procedure is employed guaranteeing sufficiently reduction of the natural merit function [2] associated to HNCP. This will allow us to establish global convergence of the PGUN algorithm to a solution of HNCP or to a stationary point of the merit function with a positive function value. In this case the algorithm terminates unsuccessfully. Fast local convergence will be established under reasonable hypotheses.

Computational experience with PGUN for solving the HNCP associated to feasible solutions of some MPCC test problems from a well-known collection [29] will show that, for many instances, projected-gradient iterations are seldom used and the algorithm is able to converge very fast to a solution of HNCP. For other instances, PGUN converges slowly using projected-gradient iterations to a stationary point of the merit function that seems not to be a solution of the HNCP. A practical criterion will be introduced to stop prematurely PGUN and avoid many projected-gradient iterations. As the natural merit function is nonconvex, the choice of the starting point is very important for the success of PGUN. Here we will suggest to restart the PGUN algorithm with a new initial point whenever the criterion mentioned before forced the algorithm to stop prematurely. Numerical results with an implementation of PGUN incorporating these two practical procedures (premature stopping criterion and restarting) show that the method is in general efficient to solve the HNCP and seems to perform better than a Projected Levenberg-Marquardt algorithm [27]. We have also tested PGUN for solving (5) associated to a target c_t equal to the best known objective function value of some MPCCs from the collection mentioned before. As discussed in [12], the introduction of the target constraint to HNCP makes this problem more difficult to tackle and PGUN has more difficulties to solve the HNCP in this case. Despite this, PGUN has been able to provide a target feasible solution of MPCC for the large majority of tested instances.

The organization of this paper is as follows. The properties of the merit function for the HNCP are studied in Section 2. The algorithm PGUN will be described and its global convergence will be analyzed in Section 3. Section 4 will be devoted to the local convergence of the PGUN algorithm. Computational experience with the PGUN algorithm will be reported in Section 5 and some conclusions will be presented in the last section of the paper.

Notation: The 2-norm of vectors and matrices will be denoted by $\|\cdot\|$. If there is no risk of confusion we denote $(x, y, w) = (x^\top, y^\top, w^\top)^\top$, as it has been already done in the Introduction. We adopt the usual convention of denoting X the diagonal matrix whose entries are the elements of $x \in I\!R^n$. The Moore-Penrose pseudoinverse of the matrix A will be denoted by $A^\dagger$. The Jacobian matrix of $\Phi : I\!R^n \to I\!R^m$, with components $\varphi_1, \ldots, \varphi_m$, will be defined by

$$\Phi'(z) = \begin{bmatrix} \frac{\partial \varphi_1}{\partial z_1}(z) & \ldots & \frac{\partial \varphi_1}{\partial z_n}(z) \\ \vdots & \ddots & \vdots \\ \frac{\partial \varphi_m}{\partial z_1}(z) & \ldots & \frac{\partial \varphi_m}{\partial z_n}(z) \end{bmatrix}.$$

We define $e = (1, \ldots, 1)^\top$ and

$$\Omega = \{(x, y, w) : x \geqslant 0, w \geqslant 0\}. \tag{6}$$

The Interior of this set will be denoted by $Int(\Omega)$.

2. Stationary points of the sum of squares

The HNCP (3) may be expressed in the form

$$F(x, y, w) = 0, x \geq 0, w \geq 0, \tag{7}$$

where $F : I\!R^{n+m+n} \longrightarrow I\!R^{r+n}$ is given by

$$F(x, y, w) = \begin{bmatrix} H(x, y, w) \\ x_1 w_1 \\ \vdots \\ x_n w_n \end{bmatrix}, \tag{8}$$

and $H : I\!R^{n+m+n} \to I\!R^r$ has continuous first derivatives.

We define the natural merit function:

$$f(x, y, w) = \|F(x, y, w)\|^2 \tag{9}$$

and we consider the problem

$$\text{Minimize } f(x, y, w) \text{ subject to } (x, y, w) \in \Omega, \tag{10}$$

where Ω is defined in (6). From now on we will denote $z = (x, y, w)$.

It is well known that, if z^* is an unconstrained stationary point of "Minimize $\|\Phi(z)\|^2$" and the residual $\Phi(z^*)$ is not null, then the rows of the $\Phi'(z^*)$ are linearly dependent. In general, this property is not true in the presence of bound constraints. In what follows, generalizing a result proved in [2], we prove that the non-full-rank property also holds in the case of problem (10) with the definitions (8) and (9).

Theorem 2.1. *Suppose that $\overline{z} = (\overline{x}, \overline{y}, \overline{w}) \in \Omega$ is a stationary point of (10). Then,*

(a) if $H(\overline{z}) = 0$ or $H'_y(\overline{z})$ is full row-rank, then $\overline{z}$ is solution of (7);

(b) if $\|F(\overline{z})\| \neq 0$, the rows of the Jacobian $F'(\overline{z})$ are linearly dependent.

Proof. If $\overline{z}$ is a stationary point of (10), then

$$\frac{1}{2}\nabla f(\overline{z}) = F'(\overline{z})^\top F(\overline{z}) = \begin{bmatrix} H'_x(\overline{z})^\top & W \\ H'_y(\overline{z})^\top & 0 \\ H'_w(\overline{z})^\top & X \end{bmatrix} \begin{bmatrix} H(\overline{z}) \\ \overline{x_1}\ \ \overline{w_1} \\ \vdots \\ \overline{x_n}\ \ \overline{w_n} \end{bmatrix} = \begin{bmatrix} \gamma \\ 0 \\ \alpha \end{bmatrix}, \tag{11}$$

$$\begin{aligned} \overline{x_i}\ \ \gamma_i = 0, &\quad i = 1, \ldots, n, \\ \overline{w_i}\ \ \alpha_i = 0, &\quad i = 1, \ldots, n, \\ \overline{x} \geq 0,\ \ \gamma \geq 0,\ \ \overline{w} &\geq 0, \text{ and } \alpha \geq 0. \end{aligned} \tag{12}$$

(a) If $H(\overline{z}) = 0$, we deduce that:

$$\begin{bmatrix} W \\ X \end{bmatrix} \begin{bmatrix} \overline{x_1}\ \ \overline{w_1} \\ \vdots \\ \overline{x_n}\ \ \overline{w_n} \end{bmatrix} = \begin{bmatrix} \gamma \\ \alpha \end{bmatrix}.$$

Thus, $\bar{x}_i \bar{w}_i = 0$ for all $i = 1, \ldots, n$ and $\overline{z}$ is a solution of (7).

On the other hand, if $H'_y(\overline{z})$ is full row-rank, then, by (11), $H(\overline{z}) = 0$. Therefore, as proved above, we have that $\overline{z}$ is solution of (7).

(b) Suppose now that $F(\overline{z}) \neq 0$. By (11), if $\overline{x}_i = \overline{w}_i = 0$ for some $i \in \{1, \ldots, n\}$, the column $r + i$ of

$$\begin{bmatrix} H'_x(\overline{z})^\top & W \\ H'_y(\overline{z})^\top & 0 \\ H'_w(\overline{z})^\top & X \end{bmatrix}$$

is null, Then the the rows of $F'(\overline{z})$ are linearly dependent.

Assume that $\overline{x}_{i_k} > 0$ and $\overline{w}_{i_k} > 0$ for q indices i_k, $k = 1, \ldots, q$ belonging to $\{1, \ldots, n\}$. Then there are three possible cases:

Case 1: $q = n$;

Case 2: $q = 0$;

Case 3: $1 \leq q < n$.

In Case 1, the stationarity imposes that the derivatives of f with respect to all the variables must vanish. Therefore,

$$\begin{bmatrix} H'_x(\overline{z})^\top & W \\ H'_y(\overline{z})^\top & 0 \\ H'_w(\overline{z})^\top & X \end{bmatrix} \begin{bmatrix} H(\overline{z}) \\ \overline{x_1}\ \overline{w_1} \\ \vdots \\ \overline{x_n}\ \overline{w_n} \end{bmatrix} = 0,$$

with $\overline{x}_i \overline{w}_i > 0$ for all $i = 1, \dots, n$. Then, the rows of $F'(\overline{z})$ are linearly dependent.

Let us now consider Case 2. Since the case in which there exists i such that $\bar{x}_i = \bar{w}_i = 0$ has already been considered, we have that $\bar{x}_i + \bar{w}_i > 0$ for all $i = 1, \dots, n$. Then, we may assume without loss of generality that $\bar{x}_i = 0$, $\bar{w}_i > 0$ for all $i = 1, \dots, n$. Then, by (11),

$$\begin{bmatrix} H'_x(\overline{z})^\top & \bar{W} \\ H'_y(\overline{z})^\top & 0 \\ H'_w(\overline{z})^\top & 0 \end{bmatrix} \begin{bmatrix} H(\overline{z}) \\ 0 \end{bmatrix} - \begin{bmatrix} \gamma \\ 0 \\ 0 \end{bmatrix} = 0. \tag{13}$$

Thus,

$$\begin{bmatrix} H'_y(\overline{z})^\top & 0 \\ H'_w(\overline{z})^\top & 0 \end{bmatrix} \begin{bmatrix} H(\overline{z}) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

This implies that the matrix

$$\begin{bmatrix} H'_y(\overline{z})^\top & 0 \\ H'_w(\overline{z})^\top & 0 \end{bmatrix}$$

has at most $r - 1$ linearly independent columns. Therefore, the matrix

$$\begin{bmatrix} H'_x(\overline{z})^\top & \bar{W} \\ H'_y(\overline{z})^\top & 0 \\ H'_w(\overline{z})^\top & 0 \end{bmatrix}$$

has at most $n+r-1$ linearly independent columns. Since $\bar{X}=0$, this implies that

$$\begin{bmatrix} H_x'(\overline{z})^\top & \bar{W} \\ H_y'(\overline{z})^\top & 0 \\ H_w'(\overline{z})^\top & \bar{X} \end{bmatrix}$$

has at most $n+r-1$ linearly independent columns. Thus $F'(\bar{z})$ has at most $n+r-1$ linearly independent rows. Since $F'(\bar{z})$ has $n+r$ rows, it turns out that this Jacobian is not full row-rank.

Let us now consider Case 3. Suppose, without loss of generality, that

$$\overline{x}_i, \overline{w}_i > 0 \text{ for } i = 1, \ldots, q < n \tag{14}$$

and

$$\overline{x}_i = 0,\ \overline{w}_i > 0 \text{ for } i = q+1, \ldots, n. \tag{15}$$

Splitting the first block of (11) into two blocks corresponding to its first q and last $n-q$ equations, using (12), (14) and (15), calling $\widehat{H}_x'(\overline{z})$ to the matrix formed by the first q rows of $H_x'(\bar{x}, \bar{y}, \bar{z})^\top$, and calling $\widehat{W}$ to the diagonal $q \times q$ matrices whose entries are $\bar{w}_1, \ldots, \bar{w}_q$, we obtain:

$$\begin{bmatrix} \widehat{H}_x'(\overline{z})^\top & \widehat{W} \\ H_y'(\overline{z})^\top & 0 \\ \bar{H}_w'(\overline{z})^\top & \bar{X} \\ \tilde{H}_w'(\overline{z})^\top & 0 \end{bmatrix} \begin{bmatrix} H(\overline{z}) \\ \overline{x_1}\ \ \overline{w_1} \\ \vdots \\ \overline{x_q}\ \ \overline{w_q} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{16}$$

Therefore, the matrix

$$A = \begin{bmatrix} \widehat{H}_x'(\overline{z})^\top & \widehat{W} \\ H_y'(\overline{z})^\top & 0 \\ \bar{H}_w'(\overline{z})^\top & \bar{X} \\ \tilde{H}_w'(\overline{z})^\top & 0 \end{bmatrix}$$

has at most $r+q-1$ linearly independent columns. Now define $\tilde{H}_x'(\overline{z})^\top$ as the matrix containing the last $n-q$ rows of $H_x'(\overline{z})^\top$, $\tilde{W}$ as the diagonal matrix whose entries are $\bar{w}_{q+1}, \ldots, \bar{w}_n$, and

$$B = \begin{bmatrix} \widehat{H}'_x(\overline{z})^\top & \widehat{W} & 0 \\ \tilde{H}'_x(\overline{z})^\top & 0 & \tilde{W} \\ H'_y(\overline{z})^\top & 0 & 0 \\ \bar{H}'_w(\overline{z})^\top & \bar{X} & 0 \\ \tilde{H}'_w(\overline{z})^\top & 0 & 0 \end{bmatrix}$$

Since B comes from adding $n-q$ rows and columns to A, the matrix B has at most $n+r-1$ linearly independent columns. But, by (11), (14), and (15), we have that $B = F'(\overline{z})^\top$. Therefore, the Jacobian is not a full row-rank matrix, as we wanted to prove. □

3. Projected gradient underdetermined Newton-like algorithm and global convergence

In this section we introduce a Projected Gradient Underdetermined Newton-like (PGUN) Algorithm for the solution of the (possibly) underdetermined system (8). This algorithm is an extension of the method introduced in [3] for the solution of this system when the number of equalities is equal to the number of variables, i.e., when $r = n+m$. PGUN generates iterates lying inside $Int(\Omega)$ and combines interior-point Newton-like and projected-gradient directions with a line-search procedure. The steps of the PGUN method are presented below.

PGUN Algorithm

Step 0: *Initial setup*: Consider $\gamma > 0$ and $\gamma_k > 0$ for all $k \in I\!N$ and such that $\sum_{k=0}^{\infty} \gamma_k = \gamma < \infty$. Let $\tau \in (0,1)$, $\sigma \in (0,1)$, $0 < \overline{\eta}_1 < \overline{\eta}_2$, $\rho > 0$, $\beta \in (0, \frac{1}{2})$, $c_{\text{big}} > c_{\text{small}} > 0$, $c_{\text{small}} < 1$. Let $z^0 = (x^0, y^0, w^0) \in Int(\Omega)$. Assume that $z^k = (x^k, y^k, w^k) \in Int(\Omega)$, $\sigma_k \in [0, 1/6]$, $\tau_k \in [\tau, 1)$, and $\eta_k \in [\overline{\eta}_1, \overline{\eta}_2]$. Then, the steps for obtaining $z^{k+1} = (x^{k+1}, y^{k+1}, w^{k+1}) \in Int(\Omega)$ or declaring finite convergence are the following:

Step 1: *Declare finite convergence if the scaled projected–gradient is zero*: Compute $g(z^k, \eta_k) = P_\Omega(z^k - \eta_k \nabla f(z^k)) - z^k$. If $g(z^k, \eta_k) = 0$, stop. (An approximate stationary point of (10) has been obtained.)

Step 2: *Newton-like direction*: Compute, if possible, $d^k = (d_x^k, d_y^k, d_w^k) \in \mathbb{R}^{n+m+n}$ satisfying

$$H'(z^k)d^k + H(z^k) = 0 \tag{17}$$

and

$$x_i^k w_i^k + x_i^k \left(d_w^k\right)_i + w_i^k \left(d_x^k\right)_i = \mu_i^k, \tag{18}$$

where $\mu^k \geqslant 0$ and

$$\|\mu^k\|_\infty \le \sigma_k \frac{\left(x^k\right)^\top w^k}{n}. \tag{19}$$

If such a direction d^k does not exist or if $\|d^k\| > c_{\text{big}}$, go to Step 4.

Step 3: *Compute the maximum steplength*: Compute

$$\alpha_k^{\text{break}} = \max\{\alpha \geq 0 \mid z^k + \alpha d^k \in \Omega\} \tag{20}$$

and

$$\alpha_k^{\max} = \min\left\{1, \tau_k \alpha_k^{\text{break}}\right\}. \tag{21}$$

If $\alpha_k^{\max} \le c_{\text{small}} \min\left\{1, \|d^k\|\right\}$, go to Step 4. Otherwise, go to Step 5.

Step 4: *Projected gradient direction*: Compute (or re-define) $d^k = g(z^k, \eta_k)$, and set $\alpha_k^{\max} = \tau_k$.

Step 5: *Line–search*: Set $\alpha = \alpha_k^{\max}$.

Step 5.1: If

$$\|F(z^k + \alpha d^k)\| \le \|F(z^k)\| - \rho\|\alpha d^k\|^2 + \gamma_k \tag{22}$$

set $\alpha_k = \alpha$ and go to Step 6.

Step 5.2: Choose $\alpha_{\text{new}} \in [\beta\alpha, (1-\beta)\alpha]$, set $\alpha = \alpha_{\text{new}}$ and go to Step 5.1.

Step 6: *Compute the new iterate*: Choose $z^{k+1} \in \Omega$ such that

$$\|F(z^{k+1})\| \leq \|F(z^k + \alpha_k d^k)\|. \tag{23}$$

End.

Given z^k not satisying the stopping criterion $g(z^k, \eta_k) = 0$, the fact that z^{k+1} is well defined follows trivially from Step 5, using $\gamma_k > 0$. The global convergence of PGUN is established in Theorem 3.1.

Theorem 3.1. *Given $z^k = (x^k, y^k, w^k)$ such that $x^k > 0$, $w^k > 0$ and $g(z^k, \eta_k) \neq 0$), the point $(x^{k+1}, y^{k+1}, w^{k+1}) \in Int(\Omega)$ is always well defined. Moreover, if $\{z^k\}$ is a sequence generated by Algorithm PGUN and z^* is a cluster point such that $\lim_{k \in K_1} z^k = z^*$, where $K_1 \subset \mathbb{N}$ is an infinite subsequence of indices, then:*

1. *z^* is a stationary point of Minimize $f(z)$ subject to $z \in \Omega$.*
2. *If $F'(z^*)$ is a full row-rank matrix, then $F(z^*) = 0$.*
3. *If K_1 contains infinitely many indices k such that d^k is computed (at Step 2) as a Newton-like direction, then $F(z^*) = 0$.*

Proof. The stationarity of z^* and the fact that $F(z^*) = 0$ when K_1 contains infinitely many Newton-like iterations follow exactly as in [3], where the theorem was proved for the (square) case in which $n + m = r$. In the general case considered here the second part of the thesis is a consequence of the stationarity of z^* and Theorem 2.1. $\square$

4. Local convergence

At Step 2 of PGUN one considers the linear system given by (17) and (18). If this linear system is incompatible the algorithm goes to Step 4 where a projected gradient direction is computed. All along this section we will assume that, whenever (17)–(18) is compatible, the computed direction d^k will be the minimum-norm solution of that system. This implies that d^k belongs to the range space of $F'(z^k)^\top$ and

$$d^k = F'(z^k)^\dagger \begin{bmatrix} -H(z^k) \\ -X_k W_k e + \mu^k \end{bmatrix}, \tag{24}$$

where $\mu^k \geqslant 0$ satisfies (19).

Note that the minimum-norm Newtonian direction associated with the system $F(z) = 0$ would be obtained taking $\mu^k = 0$ in (24).

In Theorem 3.1 we proved that limit points of a sequence generated by PGUN are necessarily stationary points of the natural merit function f. Moreover, when the Jacobian of F is full row-rank at a limit point, this point is a solution of the problem. Finally, every limit point of a subsequence of iterates x^k such that d^k is always computed at Step 2 is necessarily a solution of the nonlinear system. These global convergence results will be complemented in this section by local characterizations that tell us something about convergence of the whole sequence and its speed of convergence.

The local results that will be presented in this section are closely related with the local convergence results of Newton's method for underdetermined nonlinear systems. Roughly speaking, we are going to prove that, in a neighborhood of a solution at which the Jacobian has full row-rank, PGUN reduces to something very similar to Newton's method with the minimum norm choice of the solution of the linear system and, as a consequence, enjoys the local convergence properties of that method. However, the identification of the local PGUN and Newton's method in that case is not complete because μ^k may not be zero in (24).

Recall that PGUN does not admit negative components of (x^k, w^k). Therefore, the search direction is multiplied by a factor α^k_{max} that inhibits the possibility of taking a trial point with non-positive components in (x, w). For proving that, eventually, PGUN behaves as a pure Newton-like method, we need to prove that α^k_{max} is as close to 1 as desired. This essentially means that we do not need to truncate the direction computed at (24). We will prove this property in Theorem 4.1. In Theorem 4.2 we will prove that, if the Jacobian has full row-rank at a limit point, the whole sequence converges to that limit point. As a by-product we will prove that, eventually, $\alpha_k = \alpha_k^{max}$, which means that the first trial point at Step 5 of PGUN is accepted because the norm of F decreases as required by (22). The consequence of Theorems 4.1 and 4.2 is that, for k large enough, PGUN is very similar to Newton's method with the Moore-Penrose pseudoinverse choice of linear-system solution. The fact that $\alpha_k = \alpha_k^{max}$, together with Theorem 4.1, implies that $\alpha_k \approx 1$. Therefore, the result of Theorem 4.3 (superlinear and quadratic convergence) is not surprising, since this is the type of result that is typically obtained for Newton's method in the underdetermined and regular case. Here we could invoke well-known results as the ones given by Chen and Yamamoto in [9] but we prefer include the complete proof for the sake of completeness.

4.1. Behaviour of the maximum steplength

In this section we aim to prove that, in a neighbourhood of a solution z^* of (7) such that $F'(z^*)$ is full row-rank, the steplength α_k^{max}, computed at Step 3 of PGUN (formulas (20) and (21)), with d^k computed at Step 2, can be taken as close to 1 as desired. This means that, given an arbitrary $\delta < 1$, if z^k is close enough to the solution, the maximal steplength α_k^{break} is bigger than δ. This result has been proved in the case that $2n + m = r + n$ (square system) in [3]. The proof in the rectangular case is more involved since the solution of the Newtonian linear system is not unique.

Theorem 4.1 *Assume that Algorithm PGUN is applied to problem (7) and that z^* is a solution at which the Jacobian $F'(z^*)$ is full row-rank. Assume that $\delta \in (0, 1)$. Then, there exists $\varepsilon > 0$ such that, whenever $\|z^k - z^*\| \le \varepsilon$ one has that d^k is well defined by (17) and (18) and $\alpha_k^{break} \ge \delta$.*

Proof. Assume that $F'(z^*)$ is full row-rank and $F(z^*) = 0$. Denote $W \in I\!R^{n\times n}$ the diagonal matrix whose entries are $w_1, \dots, w_n$ and X the diagonal matrix whose entries are $x_1, \dots, x_n$. Then,

$$F'(z) = \begin{bmatrix} H'_x(z) & H'_y(z) & H'_w(z) \\ \\ W & 0 & X \end{bmatrix} \in \mathbb{R}^{(r+n)\times(2n+m)}.$$

Since $F'(z^*)$ is full row-rank, x_i^* and w_i^* can not be zero simultaneously. Without loss of generality (perhaps changing the names of some variables x_i and w_i), we may assume that $x_i^* = 0$ and $w_i^* > 0$ for all $i = 1, \dots, n$. So,

$$F'(z^*) = \begin{bmatrix} H'_x(z^*) & H'_y(z^*) & H'_w(z^*) \\ \\ W_* & 0 & 0 \end{bmatrix}.$$

Therefore, by the linear independence of the rows of $F'(z^*)$, the matrix $\begin{bmatrix} H'_y(z^*) & H'_w(z^*) \end{bmatrix}$ is full row-rank.

Let $\varepsilon > 0$ be such that, for all z such that $\|z - z^*\| \le \varepsilon$,

$$F'(z) \text{ and } H'_{yw}(z) \equiv \begin{bmatrix} H'_y(z) & H'_w(z) \end{bmatrix} \text{ are full row-rank.} \tag{25}$$

Since H has continuous first derivatives, (25) implies that $\|F'(z)^\dagger\|$ and $\|H'_{yw}(z)^\dagger\|$ are uniformly bounded for all z such that $\|z - z^*\| \le \varepsilon$.

For a generic $z = (a, b, c)$, $a > 0$, $c > 0$ such that $\|z - z^*\| \le \varepsilon$, and $\mu \geqslant 0 \in \mathbb{R}^n$ we define x, y, and w in such a way that $(x - a, y - b, w - c)$ is the minimum norm solution of:

$$\begin{cases} H'_x(a,b,c)(x-a) + H'_y(a,b,c)(y-b) + H'_w(a,b,c)(w-c) = -H(a,b,c), \\ C(x-a) + A(w-c) = -Ca + \mu. \end{cases} \tag{26}$$

Clearly, x, y, w are functions of a, b, c, and μ but we do not make this dependence explicit in order to simplify the notation.

By the boundedness of $\|F'(a,b,c)^\dagger\|$,

$$\lim_{(z,\mu)\to(z^*,0)} \|x-a\| = \lim_{(z,\mu)\to(z^*,0)} \|w-c\| = \lim_{(z,\mu)\to(z^*,0)} \|y-b\| = 0. \tag{27}$$

So,

$$\lim_{(z,\mu)\to(z^*,0)} (x,w) = (x^*,w^*) = (0,w^*). \tag{28}$$

By (26) and simplifying the notation, we have that:

$$\begin{bmatrix} H'_x & H'_y & H'_w \\ C & 0 & A \end{bmatrix} \begin{bmatrix} x-a \\ y-b \\ w-c \end{bmatrix} = \begin{bmatrix} -H \\ -Ca+\mu \end{bmatrix} \in I\!R^{r+n}. \tag{29}$$

Taking the minimum norm solution of (29), we have that $(x-a, y-b, w-c)^\top$ belongs to the range space of $F'(z^*)^\top$. Therefore, there exist $q \in I\!R^p$ and $t \in I\!R^n$ such that

$$\begin{bmatrix} x-a \\ y-b \\ w-c \end{bmatrix} = \begin{bmatrix} (H'_x)^\top & C \\ (H'_y)^\top & 0 \\ (H'_w)^\top & A \end{bmatrix} \begin{bmatrix} q \\ t \end{bmatrix} \in I\!R^{m+2n}. \tag{30}$$

Therefore,

$$\begin{cases} x-a = (H'_x)^\top q + Ct \\ y-b = (H'_y)^\top q \\ w-c = (H'_w)^\top q + At \end{cases} \tag{31}$$

Thus, by (29) and (31),

$$\begin{bmatrix} H'_x(H'_x)^\top + H'_y(H'_y)^\top + H'_w(H'_w)^\top & H'_xC + H'_wA \\ C(H'_x)^\top + A(H'_w)^\top & C^2+A^2 \end{bmatrix} \begin{bmatrix} q \\ t \end{bmatrix} = \begin{bmatrix} -H \\ -Ca+\mu \end{bmatrix} \tag{32}$$

Therefore,

$$t = -(C^2+A^2)^{-1}(C(H'_x)^\top + A(H'_w)^\top)q - (C^2+A^2)^{-1}(Ca-\mu). \tag{33}$$

By the first equation of (32) and (33) we have that:

$$\begin{aligned} &((H'_x(H'_x)^\top + H'_y(H'_y)^\top + H'_w(H'_w)^\top) \\ &-(H'_xC + H'_wA)(C^2 + A^2)^{-1}(C(H'_x)^\top + A(H'_w)^\top))q \\ &= -H + (H'_xC + H'_wA)(C^2 + A^2)^{-1}(Ca - \mu). \end{aligned} \tag{34}$$

Note that

$$(C^2 + A^2)^{-1} = C^{-1}(I + C^{-1}A^2C^{-1})^{-1}C^{-1}. \tag{35}$$

Let us define $\tilde{H}' = H'_x(H'_x)^\top + H'_y(H'_y)^\top + H'_w(H'_w)^\top - (H'_xC + H'_wA)(C^2 + A^2)^{-1}(C(H'_x)^\top + A(H'_w)^\top)$.

Then, by (35),

$$\begin{aligned} \tilde{H}' \;=\; & H'_x(H'_x)^\top + H'_y(H'_y)^\top + H'_w(H'_w)^\top \\ & -(H'_x + H'_wAC^{-1})(I + C^{-1}A^2C^{-1})^{-1}((H'_x)^\top + C^{-1}A(H'_w)^\top). \end{aligned} \tag{36}$$

By (36), since $A \to 0$, we have that $\tilde{H}' \to H'_y(z^*)H'_y(z^*)^\top + H'_w(z^*)H'_w(z^*)^\top$.

Since the matrix $\begin{bmatrix} H'_y & H'_w \end{bmatrix}$ is full row-rank, we have that, if (a, b, c) is close enough to z^*, $\tilde{H}'$ is nonsingular and its inverse is bounded. Then, recalling that, by (34),

$$q = (\tilde{H}')^{-1}(-H + (H'_xC + H'_wA)(C^2 + A^2)^{-1}(Ca - \mu)), \tag{37}$$

we obtain that q is bounded if (a, b, c) is close enough to the solution and μ is close enough to 0. Moreover, since $Ca - \mu \to 0$, we have that $q = q(a, b, c, \mu)$ tends to zero as (a, b, c) tends to z^* and μ tends to zero.

In other words,

$$\lim_{(z,\mu)\to(z^*,0)} q(a, b, c, \mu) = 0. \tag{38}$$

Analogously, by (33),

$$\lim_{(z,\mu)\to(z^*,0)} t(a, b, c, \mu) = 0. \tag{39}$$

Recall that $x - a = (H_x')^\top q + Ct$. Then, by (33),

$$\begin{aligned}
Ct &= -C((C^2+A^2)^{-1}(C(H_x')^\top + A(H_w')^\top)q - (C^2+A^2)^{-1}(Ca-\mu)) \\
&= -CC^{-1}(I+C^{-1}A^2C^{-1})^{-1}C^{-1}C((H_x')^\top + C^{-1}A(H_w')^\top)q \\
&\quad -CC^{-1}(I+C^{-1}A^2C^{-1})^{-1}C^{-1}C(a-C^{-1}\mu) \\
&= -(I+C^{-1}A^2C^{-1})^{-1}((H_x')^\top + C^{-1}A(H_w')^\top)q \\
&\quad -(I+C^{-1}A^2C^{-1})^{-1}(a-C^{-1}\mu)
\end{aligned} \tag{40}$$

and

$$\begin{aligned}
x &= (H_x')^\top q + Ct + a \\
&= (I-(I+C^{-1}A^2C^{-1})^{-1})a + (I-(I+C^{-1}A^2C^{-1})^{-1})(H_x')^\top q \\
&\quad -(I+C^{-1}A^2C^{-1})^{-1}(C^{-1}A(H_w')^\top)q + (I+C^{-1}A^2C^{-1})^{-1}C^{-1}\mu.
\end{aligned} \tag{41}$$

Observe that

$$\begin{aligned}
I-(I+C^{-1}A^2C^{-1})^{-1} &= I - I - \textstyle\sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j \\
&= C^{-1}A^2C^{-1}(I+\textstyle\sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j).
\end{aligned}$$

Then, by (41),

$$\begin{aligned}
x &= C^{-1}A^2C^{-1}(I+\textstyle\sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)a \\
&\quad +C^{-1}A^2C^{-1}(I+\textstyle\sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)((H_x')^\top q) \\
&\quad -AC^{-1}(I+C^{-1}A^2C^{-1})^{-1}((H_w')^\top)q + (I+C^{-1}A^2C^{-1})^{-1}C^{-1}\mu.
\end{aligned}$$

Therefore, for all $i = 1, \dots, n$ we have that

$$\begin{aligned}
x_i \;\geq\;& (c_i)^{-2}(a_i)^2[(I+\textstyle\sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)a]_i \\
&+(c_i)^{-2}(a_i)^2[(I+\textstyle\sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)((H_x')^\top q)]_i \\
&-(c_i)^{-1}a_i[(I+C^{-1}A^2C^{-1})^{-1}((H_w')^\top)q]_i.
\end{aligned} \tag{42}$$

Our objective now is to investigate the possible values of $\alpha \in [0,1]$ such that

$$\alpha x_i + (1-\alpha)a_i = 0 \tag{43}$$

or

$$\alpha w_i + (1-\alpha)c_i = 0. \tag{44}$$

If (44) takes place, then

$$\alpha = \frac{c_i}{c_i - w_i}. \tag{45}$$

But, by (27) and since $w_i^* > 0$, an $\alpha \in [0,1]$ satisfying (45) cannot exist if ε is small enough.

Therefore, we only need to analyze the values of α that satisfy (43).

By (43), $\alpha = 1 + \alpha \frac{x_i}{a_i}$. Then, by (42),

$$\begin{array}{rcl} \alpha & \geq & 1 + \alpha \frac{(c_i)^{-2}(a_i)^2}{a_i}[(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)a]_i \\ & & + \frac{(c_i)^{-2}(a_i)^2}{a_i}[(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)((H_x')^\top q)]_i \\ & & - \frac{(c_i)^{-1}a_i}{a_i}[(I + C^{-1}A^2C^{-1})^{-1}(H_w')^\top q]_i. \end{array}$$

Thus,

$$\begin{array}{rcl} \alpha & \geq & 1 + \alpha(c_i)^{-2}a_i[(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)a]_i \\ & & + \alpha(c_i)^{-2}a_i[(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)(H_x')^\top q]_i \\ & & - \alpha c_i[(I + C^{-1}A^2C^{-1})^{-1}(H_w')^\top q]_i \end{array} \tag{46}$$

By (38) and (46), given any $\delta \in [0,1)$, and taking ε small enough we obtain that $\alpha = 1$. Consequently, $\alpha_k^{break} \geq \delta$. □

4.2. Convergence of the whole sequence

Assumption L. For all $z, z' \in \Omega$,

$$\|F'(z) - F'(z')\| \leq L\|z' - z\| \ \ \forall \ \ z, z' \in \Omega \subset \mathbb{R}^{m+2n}. \tag{47}$$

As a consequence, for all $z, z' \in \Omega$,

$$\|F(z') - F(z) - F'(z)(z' - z)\| \leq \frac{L}{2}\|z' - z\|^2. \tag{48}$$

Theorem 4.2 *Assume that Assumption L holds, $z^* \in \Omega$ is a cluster point of a sequence generated by Algorithm PGUN, $F'(z^*)$ is full row-rank and, for k large enough, we choose*

$$z^{k+1} = z^k + \alpha_k d^k \tag{49}$$

at Step 6 of the algorithm. Assume, further, that c_{big} (used at Step 2 of Algorithm PGUN) is greater than $4\|F'(z^*)^\dagger\|$ and $\lim_{k\to\infty} \tau_k = 1$. Then, $\lim_{k\longrightarrow\infty} z^k = z^*$ and

$$\alpha_k = \alpha_k^{max} \tag{50}$$

for k large enough.

Proof. Let K_1 be an infinite sequence of indices such that $\lim_{k\in K_1} z^k = z^*$. By Theorem 3.1, z^* is a stationary point of f over Ω.

The choice of d^k at Step 2 of the algorithm gives:

$$H'(z^k)d^k + H(z^k) = 0 \tag{51}$$

and

$$\left(x_i^k[d_w^k]_i + w_i^k[d_x^k]_i + x_i^k w_i^k\right)^2 = \sigma_k^2 \frac{\langle x^k, w^k\rangle^2}{n^2} \le \sigma_k^2 \frac{\sum_{i=1}^n (x_i^k w_i^k)^2}{n}.$$

So,

$$\sum_{i=1}^n (x_i^k[d_w^k]_i + w_i^k[d_x^k]_i + x_i^k w_i^k)^2 \le \sigma_k^2 \sum_{i=1}^n (x_i^k w_i^k)^2 \le \sigma_k^2 \|F(z^k)\|^2.$$

Then, by (51),

$$\|F'(z^k)d^k + F(z^k)\| \le \sigma_k \|F(z^k)\|. \tag{52}$$

Since $F'(z^*)$ is full row-rank, there exists $\varepsilon_1 > 0$ such that $\|F'(z)^\dagger\| \le M_1 \equiv 2\|F'(z^*)^\dagger\|$ and $F'(z)$ is full row rank whenever $\|z - z^*\| \le \varepsilon_1$. Moreover, $F'(z)^\dagger F'(z) F'(z)^\dagger = F'(z)^\dagger$. Therefore, by (52), for $k \in K_1$ large enough and

$\|z^k - z^*\| \leq \varepsilon_1$,

$$
\begin{aligned}
\|d^k\| = & \left\| F'(z^k)^\dagger \begin{bmatrix} -H(z^k) \\ -X_k W_k e + \mu^k \end{bmatrix} \right\| \\
= & \left\| F'(z^k)^\dagger F'(z^k) F'(z^k)^\dagger \begin{bmatrix} -H(z^k) \\ -X_k W_k e + \mu^k \end{bmatrix} \right\| \\
= & \|F'(z^k)^\dagger F'(z^k) d^k\| \leq \|F'(z^k)^\dagger\| \|F'(z^k) d^k + F(z^k) - F(z^k)\| \\
\leq & \|F'(z^k)^\dagger\| (\|F'(z^k) d^k + F(z^k)\| + \|F(z^k)\|) \leq M_1 (1 + \sigma_k) \|F(z^k)\|.
\end{aligned} \tag{53}
$$

By Theorem 3.1, we have that $F(z^*) = 0$. Moreover, since $c_{big} \geq 4\|F'(z^*)^\dagger\|$, if $\|z^k - z^*\| \leq \varepsilon_1$, $k \in K_1$, large enough, we have that $\|F(z^k)\| \leq 1$ and (53) implies that d^k is computed at Step 2.

Define $M_2 = 2\|F'(z^*)\|$. Then, since F and F' are continuous, $F(z^*) = 0$. By (53) and Theorem 4.1 there exists $\varepsilon_2 \in (0, \varepsilon_1]$ such that for all $k \in I\!N$ such that $\|z^k - z^*\| \leq \varepsilon_2$, we have that:

(i) $\|d^k\| \leq M_1(1 + \sigma_k)\|F(z^k)\|$;

(ii) $\alpha_k^{\max} \geq \max\{1 - \frac{1}{12 M_1 M_2}, \frac{11}{12}\}$;

(iii) $\|F'(z^k)\| \leq M_2$;

(iv) $\|F(z^k)\| \leq \frac{1}{12 L M_1^2}$;

(v) $\rho \|\alpha_k^{max} d^k\|^2 \leq \frac{1}{2}\|F(z^k)\|$.

Then, for all $k \in I\!N$ such that $\|z^k - z^*\| \leq \varepsilon_2$,

$$
\begin{aligned}
& \|F(z^k + \alpha_k^{max} d^k)\| \\
& \leq \|F(z^k + \alpha_k^{max} d^k) - F(z^k) - \alpha_k^{max} F'(z^k) d^k\| + \|F(z^k) + \alpha_k^{max} F'(z^k) d^k\| \\
& \leq \tfrac{L}{2} (\alpha_k^{max})^2 \|d^k\|^2 + \|F(z^k) + F'(z^k) d^k\| + (1 - \alpha_k^{max}) \|F'(z^k) d^k\| \\
& \leq \tfrac{L}{2} (\alpha_k^{max})^2 \|d^k\|^2 + \sigma_k \|F(z^k)\| + (1 - \alpha_k^{max}) \|F'(z^k) d^k\| \\
& \leq \tfrac{L}{2} (\alpha_k^{max})^2 M_1^2 (1 + \sigma_k)^2 \|F z^k\|^2 + \sigma_k \|F(z^k)\| \\
& \quad + (1 - \alpha_k^{max}) \|F'(z^k)\| M_1 (1 + \sigma_k) \|F z^k\| \\
& \leq \left(\tfrac{L}{2} (\alpha_k^{max})^2 M_1^2 \|F(z^k)\| + \sigma_k + (1 - \alpha_k^{max})(1 + \sigma_k) M_2 M_1 \right) \|F(z^k)\| \\
& \leq \tfrac{1}{2} \|F(z^k)\| \leq \|F(z^k)\| - \rho \|\alpha_k^{max} d^k\|^2 + \gamma_k.
\end{aligned} \tag{54}
$$

Therefore, by (22), for all $k \in I\!N$ such that $\|z^k - z^*\| \leq \varepsilon_2$, we have that $\alpha_k = \alpha_k^{max}$ (proving (50)),

$$z^{k+1} = z^k + \alpha_k^{max} d^k, \quad \text{and} \quad \|F(z^{k+1})\| \leq \frac{1}{2}\|F(z^k)\|. \tag{55}$$

Since $\lim_{k \in K_1} F(z^k) = F(z^*) = 0$, there exists $k_0 \in K_1$ such that $\|z^{k_0} - z^*\| \leq \frac{\varepsilon_2}{4}$ and $\|F(z^{k_0})\| \leq \frac{\varepsilon_2}{4(4M_1+1)}$. We will prove by induction that $\|z^k - z^*\| \leq \varepsilon_2$ for all $k \geq k_0$, $k \in I\!N$. This is trivial for $k = k_0$.

Assume, by inductive hypothesis, that $\|z^k - z^*\| \leq \varepsilon_2$ for all $k = k_0, k_0 + 1, \ldots, k_0 + j - 1$. Then, by (55), $\|F(z^{k+1})\| \leq \frac{1}{2}\|F(z^k)\|$ for $k = k_0 + 1, \ldots, k_0 + j - 1$.

By (55) and (i)-(v), we can write:

$$\begin{aligned} \|z^{k_0+j} - z^{k_0}\| &= \|\textstyle\sum_{i=0}^{j-1} \alpha_{k_0+i}^{max} d^{k_0+i}\| \leq 2M_1 \sum_{i=0}^{j-1} \left(\frac{1}{2}\right)^i \|F(z^{k_0})\| \\ &\leq 4M_1 \|F(z^{k_0})\| \leq \tfrac{\varepsilon_2}{4}. \end{aligned}$$

Therefore, $\|z^{k_0+j} - z^*\| \leq \|z^{k_0+j} - z^{k_0}\| + \|z^{k_0} - z^*\| \leq \frac{\varepsilon_2}{2}$. Thus, $\|z^{k_0+j} - z^*\| \leq \varepsilon_2$. This completes the inductive proof.

Let us prove now that $\{z^k\}$ is a Cauchy sequence.

Let $j \geq k_0$ and $\ell \geq 1$. Then,

$$\begin{aligned} \|z^{j+\ell} - z^j\| &\leq \textstyle\sum_{i=0}^{\ell-1} \alpha_{j+i}^{max} \|d^{j+i}\| \\ &\leq 2M_1 \textstyle\sum_{i=0}^{\ell-1} (\frac{1}{2})^{i+1} \|F(z^j)\| \\ &\leq 2M_1 \textstyle\sum_{i=0}^{\ell-1} (\frac{1}{2})^{i+1} \|F(z^j)\| \leq 2M_1 \|F(z^j)\|. \end{aligned} \tag{56}$$

Since $\lim_{j \to \infty} \|F(z^j)\| = 0$, (56) implies that $\{z^k\}$ is a Cauchy sequence. Then, since z^* is a limit point, we have that $\lim_{k \longrightarrow \infty} z^k = z^*$. □

4.3. Superlinear and quadratic convergence

In this section we will prove that, under the assumptions of Theorem 4.2 and adequate choices of the parameters σ_k, the algorithm exhibits superlinear or quadratic convergence.

We will consider the following assumption on the parameters σ_k.

Assumption S. Choose σ_k such that

$$\lim_{k\to\infty} \sigma_k = 0. \tag{57}$$

Theorem 4.3. *Assume that $\{z^k\}$ is generated by Algorithm PGUN and converges to z^* such that $F(z^*) = 0$, where $F'(z^*)$ is full row-rank, and for k large enough we choose*

$$z^{k+1} = z^k + \alpha_k d^k \tag{58}$$

at Step 6 of the algorithm. Assume that the hypotheses of Theorem 4.2, and both assumptions L and S hold. Then, z^k converges superlinearly to z^*.

Moreover, if there exists $c_1, c_2 > 0$ such that, for all k large enough,

$$\sigma_k \leq c_1\|F(z^k)\| \text{ and } 1 - \tau_k \leq c_2\|F(z^k)\|, \tag{59}$$

z^k converges quadratically to z^*.

Proof. Since $\tau_k \to 1$ we have that $\lim_{k\to\infty} \alpha_k^{max} = 1$.

By Theorem 4.2, for all k large enough there exists $M > 0$ such that $\|d^k\| \leq M\|F(z^k)\|$, $\|F'(z^k)\| \leq M$ and

$$\begin{aligned}
\|F(z^{k+1})\| \leq\ & \|F(z^{k+1}) - F(z^k) - \alpha_k^{max} F'(z^k)d^k\| + \|F(z^k) + \alpha_k^{max} F'(z^k)d^k\| \\
\leq\ & \tfrac{L}{2}(\alpha_k^{max})^2\|d^k\|^2 + \|F(z^k) + F'(z^k)d^k\| + (1-\alpha_k^{max})\|F'(z^k)d^k\| \\
\leq\ & \tfrac{L}{2}M^2\|F(z^k)\|^2 + \sigma_k\|F(z^k)\| + (1-\alpha_k^{max})M^2\|F(z^k)\| \\
\leq\ & \left(\tfrac{L}{2}M^2\|F(z^k)\| + \sigma_k + (1-\alpha_k^{max})M^2\right)\|F(z^k)\| = R_k\|F(z^k)\|
\end{aligned} \tag{60}$$

where $R_k = \dfrac{L}{2}M^2\|F(z^k)\| + \sigma_k + M^2(1-\alpha_k^{max})$. Moreover,

$$\|z^{k+1} - z^*\| \leq \sum_{j=k+1}^{\infty} \alpha_j^{max}\|d^j\| \leq 2M\sum_{j=1}^{\infty}\left(\frac{1}{2}\right)^j \|F(z^{k+1})\|.$$

By (60) and (48) we have that

$$
\begin{aligned}
\|z^{k+1} - z^*\| \leq & \ 2MR_k \|F(z^k)\| \\
= & \ 2MR_k \|F(z^k) - F(z^*) - F'(z^k)(z^k - z^*) + F'(z^k)(z^k - z^*)\| \\
\leq & \ 2MR_k \|F(z^k) - F(z^*) - F'(z^k)(z^k - z^*)\| + \|F'(z^k)(z^k - z^*)\| \\
\leq & \ 2MR_k \left(\tfrac{L}{2}\|z^k - z^*\| + M\right) \|z^k - z^*\| \\
\leq & \ 2MR_k L \left(\tfrac{L}{2} + M\right) \|z^k - z^*\|.
\end{aligned}
$$

Since $\lim_{k\to\infty} R_k = 0$, z^k converges superlinearly to z^*.

Now, taking $c = \max\{c_1, c_2\}$, since $\max\{\sigma_k, 1 - \alpha_k^{max}\} \leq \max\{\sigma_k, 1 - \tau_k\} \leq c\|F(z_k)\|$, we have that

$$
\begin{aligned}
\|z^{k+1} - z^*\| \ \leq & \ 2M \left(\tfrac{L}{2} M^2 \|F(z^k)\| + \sigma_k + (1 - \alpha_k^{max}) M^2\right) \|F(z^k)\| \\
\leq & \ 2M \left(\tfrac{L}{2} M^2 + (1 + M^2) c\right) \|F(z^k)\|^2 \\
\leq & \ 2M (\tfrac{L}{2} + M)^2 \left(\tfrac{L}{2} M^2 + (1 + M^2) c\right) \|z^k - z^*\|^2.
\end{aligned}
$$

Therefore, quadratic convergence is proved. □

5. Computational Experience

In this section we will report some experiments with the PGUN algorithm for the solution of (3) and (5). In order to have a better idea of the efficiency of PGUN in practice, we have compared the PGUN method with the Projected-Gradient Levenberg-Marquardt (PLM) algorithm [27].

5.1. The Projected Levenberg-Marquardt Algorithm

The Projected Levenberg-Marquardt (PLM) is an algorithm for the solution of constrained nonlinear systems $F(z) = 0,\ z \in Z$, where $Z \in \mathbb{R}^n$ is a nonempty, closed and convex set. For solving this problem the method is applied to a nonlinear program of a form similar to (10) where the merit function is also defined by (9).

The PLM algorithm generates a sequence $\{z^k\}$ by

$$z^{k+1} = P_Z(z^k + d_U^k) \quad k = 0, 1, ...,$$

where d_U^k is the unique solution of the system of linear equations

$$(J_k^\top J_k + \mu_k I)d_U = -J_k^\top F(z^k) \tag{61}$$

and J_k is an approximation to the Jacobian $F'(z^k)$.

We present below, in general terms, the method based on Algorithm 3.12 of [27] with the additional line search step considered in the experimental section of [27].

For more details about the method and its convergence properties see [27].

PLM Algorithm

Step 0: *Initial setup*: Choose $z^0 \in Z$, $\mu > 0$, $\beta, \sigma, \gamma \in (0,1)$, $\rho > 0$ and $p > 1$.

Step 1: *Declare finite convergence*: If $F(z^k) = 0$, stop.

Step 2: *Unconstrained direction*: Choose J_k, set $\mu_k = \mu\|F(z^k)\|^2$ and compute d_U^k as the solution of (61).

Step 3: *Levenberg-Marquardt step*: If

$$\|F(P_Z(z^k + d_U^k))\| \leqslant \gamma\|F(z^k)\|, \tag{62}$$

then set $z^{k+1} = P_Z(z^k + d_U^k)$ and go to Step 1.

Step 4: *Line Search step*: If the search direction $s^k = P_Z(z^k + d_U^k) - z^k$ is a descent direction of f in the sense that $\nabla f(z^k)^\top s^k \leqslant -\rho\|s^k\|^p$, set $\alpha = 1$ and

Step 4.1: If

$$\|F(z^k + ts^k)\|^2 \leqslant \|F(z^k)\|^2 + \gamma\alpha\nabla f(z^k)^\top s^k$$

then set $z^{k+1} = z^k + \alpha s^k$ and go to Step 1.

Step 4.2: Choose $\alpha_{new} \in (0, \alpha)$, set $\alpha = \alpha_{new}$ and go to Step 4.1.

Step 5: *Projected Gradient step*: Compute a stepsize $\alpha_k = \max\{\beta^l \mid l = 0, 1, 2, ...\}$ such that

$$f(z^k(\alpha_k)) \leqslant f(z^k) + \sigma \nabla f(z^k)^\top (z^k(\alpha_k) - z^k),$$

where $z^k(\alpha) = P_Z(z^k - \alpha \nabla f(z^k))$. Set $z^{k+1} = z^k(\alpha_k)$ and go to Step 1.

5.2. Implementation issues and test problems

The codes for the PGUN and PLM algorithms were written in Fortran 77 with double precision and the experiments were performed using gfortarn-4.6 on an Intel CORE I3-2310M@2.10 GHz with 100 Gb of HD and 4Gb of Ram. Furthermore we used the `ma48` routine of the Harwell Subroutine Library [21] for the solution of the linear systems required by the two algorithms.

We considered the following stopping criteria:

SC1: Stop with z^k if $||g(z^k, \eta)|| < 10^{-5}$.

SC2: Stop with z^k when SC1 is satisfied and $||F(z^k)|| < 10^{-6}$.

SC3: Stop at iteration k if $||F(z^k)|| > 10^{-3}$ and $||F(z^{k-1})|| - ||F(z^k)|| < 10^{-4}$.

PGUN stops if **SC1** occurs at a projected gradient iteration. However, if **SC1** takes place at a interior point Newton-like (IP) iteration we continue the execution with the hope of satisfying **SC2**. If, during this process, a projected gradient iteration is required, we stop with the diagnostic **SC1**.

In some cases the PGUN algorithm converges very slowly using projected-gradient (PG) iterations to a stationary point with a positive value of the merit function. In this case, PGUN is not converging to a solution of the HNCP and there is no reason to continue the execution of the algorithm. To avoid this occurrence, we decided to stop prematurely the algorithm by using the third stopping criterion. Moreover, when **SC3** occurs the algorithm is restarted with a new initial point.

PLM employs fast Levenberg-Marquardt (LM) and slow Projected-Gradient (PG) iterations and use the same stopping criteria SCi, i = 1, 2, 3, employed by PGUN with the LM iterations replacing the IP ones.

We limited the number of iterations of both PGUN and PLM by $\max\{100, \min\{r+1, 2n+m\}^3\}$ and the CPU time by 600 seconds. The initial iterate for both methods was given by:

$$x^0 = e,\ y^0 = 0,\ w^0 = e \tag{63}$$

where e is a vector of ones. The following values for the algorithmic parameters of PGUN were used: $\alpha_{min} = 10^{-8}$, $\beta = 0.25$, $c_{big} = 10^4$, $c_{small} = 10^{-10}$, $\eta_k = \eta = 1.0$, $\gamma_k = \frac{1}{k^2}$, $\rho = 10^{-3}$, $\sigma_k = \sigma = \frac{1}{\sqrt{2n+m}}$, $\tau_k = \tau = 0.9995$ and $\theta = 0.5$. For the PLM Method we utilized the default parameters of [27]: $\alpha_{min} = 10^{-12}$, $\beta = 0.9$, $\mu = 10^{-5}$, $\sigma = 10^{-4}$, $\gamma = 0.99995$, $p = 2.1$ and $\rho = 10^{-8}$.

We have made the experiments with both the algorithms on the solution of 48 MPCC test problems of the collection MacMPEC [29]. These problems are presented in Table 1. In this table, m is the dimension of y, n is the dimension of x and w, p is the dimension of $(\varphi(x, y, w), H(x, y, w))^\top$, nz is the number of possible non zero elements of the Jacobian matrix, density is the density of the Jacobian matrix and min is the lower value known for the function.

5.3. Experiment 1: Computing a Simple Feasible Solution of MPCC

In order to compute a simple feasible solution of the MPCC, we considered the HNCP of the form (3). Table 2 shows the number of complementary pairs for each problem. In this table, NCP represents the number of original complementary pairs and NNG is the number of complementary pairs after each nonnegative non-complementary variable x_i is transformed into a pair of complementary variables (x_i, w_i) with w_i an auxiliary variable.

Table 3 reports the performance of the PGUN algorithm for finding a sim-

Table 1: Selected problems of the Mathematical Programs with Equilibrium Constraints collection.

Problem	m	n	p	nz	density	min	Problem	m	n	p	nz	density	min
bard1	0	6	5	29	22%	17.0000	liswet1	52	102	104	760	1%	0.01399
bard2	0	22	18	98	6%	-6598.00	nash1	2	7	7	35	16%	7.8e-30
bard3	0	8	6	38	17%	-12.6787	outrata31	0	7	6	37	20%	3.20770
bilevel1	2	12	12	62	10%	-60.0000	outrata32	0	7	6	38	21%	3.44940
bilevel3	2	8	8	44	15%	-12.6787	outrata33	0	7	6	38	21%	4.60425
bilin	0	10	8	56	16%	18.4000	outrata34	0	7	6	40	22%	6.59268
dempe	2	2	3	12	40%	28.2500	portfl1	1	75	14	1149	9%	1.5e-05
design_cent1	9	7	13	60	13%	1.86065	qpec1	10	21	21	113	5%	80.0000
desilva	2	7	7	33	15%	-1.00000	qpecgen1	5	103	103	11124	26%	0.09900
df1	1	6	6	27	17%	0.00000	ralph2	0	2	1	7	58%	0.00000
ex911	2	7	8	42	18%	-13.0000	ralphmod	0	109	105	10831	23%	-683.033
ex921	0	7	6	34	19%	17.0000	scale1	0	2	1	7	58%	1.00000
ex922	0	9	7	38	13%	100.000	scale2	0	2	1	7	58%	1.00000
ex925	1	6	6	30	19%	5.00000	scale3	0	2	1	7	58%	1.00000
ex928	0	6	5	24	18%	1.50000	scale4	0	2	1	7	58%	1.00000
flp2	0	7	5	33	20%	0.00000	scale5	0	2	1	7	58%	100.000
gauvin	0	5	4	22	24%	20.0000	scholtes1	1	3	2	14	40%	2.00000
gnash1	1	11	11	57	11%	-230.823	scholtes2	1	3	2	14	40%	15.0000
hakonsen	0	9	7	46	16%	24.3668	scholtes3	0	2	1	7	58%	0.50000
jr1	1	2	2	10	50%	0.50000	scholtes4	1	4	3	18	29%	-3.0e-07
jr2	1	2	2	10	50%	0.50000	scholtes5	0	3	2	12	40%	1.00000
kth1	0	2	1	7	58%	0.00000	sl1	2	11	10	49	10%	0.00010
kth2	0	2	1	7	58%	0.00000	stackelberg1	0	4	3	16	29%	-3266.67
kth3	0	2	1	7	58%	0.50000	traffic1	0	739	737	3679	0.17%	45.1500

ple feasible solution of the Mathematical Program with Complementarity Constraints (MPCC). In this table, we use the following notations:

TERM: termination of the algorithm which can be one of the following:

IP-1: algorithm stopped with an interior-point Newton-like (IP) iteration satisfying SC1.

IP-2: algorithm stopped with an IP iteration satisfying SC2.

PG-1: algorithm stopped with a projected-gradient (PG) iteration satisfying SC1.

IP: number of interior-point Newton-like (IP) iterations.

PG: number of projected-gradient (PG) iterations.

Table 2: Number of complementary pairs for Experiment 1

Problem	NCP	NNG	Problem	NCP	NNG	Problem	NCP	NNG
bard1	3	2	gauvin	2	2	qpecgen	100	2
bard2	4	17	gnash1	8	2	ralph2	1	0
bard3	2	5	hakonsen	4	4	ralphmod	100	8
bilevel1	6	5	jr1	1	0	scale1	1	0
bilevel3	4	3	jr2	1	0	scale2	1	0
bilin	6	3	kth1	1	0	scale3	1	0
dempe	1	0	kth2	1	0	scale4	1	0
design-cent1	3	3	kth3	1	0	scale5	1	0
desilva	2	4	liswet1-inv50	50	51	scholtes1	1	1
df1	1	4	nash1	2	4	scholtes2	1	1
ex911	5	1	outrata31	4	2	scholtes3	1	0
ex921	4	2	outrata32	4	2	scholtes4	1	2
ex922	4	4	outrata33	4	2	scholtes5	2	0
ex925	3	2	outrata34	4	2	sl1	3	7
ex928	2	3	portfl1	12	62	stackelberg1	1	2
flp2	2	4	qpec1	10	10	traffic1	244	494

CG: number of times that the algorithm changed from an IP to a PG iteration or conversely.

NE: number of function evaluations.

TIME: CPU time (in seconds), measured with the function `etime`. A time smaller than 1e-4 is considered as zero.

$||F(\overline{z})||$: value of $||F(\overline{z})||$, where $\overline{z}$ is the solution computed by the algorithm.

SPG_norm: norm of the projected-gradient at the solution computed by the algorithm.

Feas: feasibility measure, that is, Feas $= ||h(\overline{z})||$.

Comp: complementarity measure, that is, Comp $= \max_{i=1,n}\{x_i w_i\}$.

* The algorithm computed a feasible solution of MPCC with an initial point different from (63).

$**$ failure: The algorithm was not able to compute a feasible solution of MPCC after 10 trials with different starting points.

Table 3: Performance of the PGUN method for Experiment 1

Problem	TERM	IP	PG	CG	NE	TIME	$\|\|F(\bar{z})\|\|$	SPG_norm	Feas	Comp
bard1	IP-1	6	0	0	7	0.0000	1.17e-08	3.08e-08	1.16e-08	1.19e-09
bard2	IP-2	12	0	0	13	0.0040	5.86e-14	4.23e-13	1.20e-14	5.73e-14
bard3	IP-1	5	0	0	6	0.0000	4.72e-07	1.06e-06	2.88e-07	3.12e-07
bilevel1	IP-2	25	0	0	26	0.0040	9.90e-07	2.97e-06	9.90e-07	1.36e-20
bilevel3	IP-2	51	0	0	52	0.0040	9.64e-07	4.09e-06	9.64e-07	4.19e-23
bilin	IP-1	7	0	0	8	0.0000	2.68e-08	8.96e-09	1.84e-09	2.48e-08
dempe	IP-1	5	0	0	6	0.0000	1.25e-07	5.07e-07	1.25e-07	1.98e-12
design-cent1	IP-2*	8	0	0	9	0.0000	6.80e-08	1.05e-08	4.51e-09	6.79e-08
desilva	IP-1	5	0	0	6	0.0000	2.21e-07	6.11e-07	2.17e-07	2.94e-08
df1	IP-2	10	0	0	11	0.0000	8.97e-08	1.26e-07	8.97e-08	2.72e-23
ex911	IP-1	8	0	0	9	0.0000	6.08e-08	3.80e-07	5.08e-08	3.34e-08
ex921	IP-2	31	0	0	32	0.0000	6.08e-07	1.74e-06	6.08e-07	1.97e-22
ex922	IP-2	14	0	0	15	0.0000	4.84e-07	4.77e-10	1.13e-15	4.84e-07
ex925	IP-1	8	0	0	9	0.0000	4.57e-07	7.93e-08	6.38e-16	4.57e-07
ex928	IP-1	5	0	0	6	0.0000	2.16e-08	6.12e-09	9.91e-17	1.86e-08
flp2	IP-1	7	0	0	8	0.0000	4.55e-13	1.06e-12	1.14e-15	4.54e-13
gauvin	IP-1	8	0	0	9	0.0000	5.13e-07	2.43e-07	4.43e-15	5.13e-07
gnash1	IP-1*	14	0	0	15	0.0000	4.42e-11	4.60e-11	4.42e-011	1.86e-17
hakonsen	IP-1	9	0	0	10	0.0000	3.54e-11	4.26e-09	3.54e-11	3.45e-15
jr1	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	0.0000	9.53e-07
jr2	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	0.0000	9.53e-07
kth1	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
kth2	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
kth3	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
liswet1-inv50	IP-1	26	0	0	45	0.1400	3.21e-08	3.96e-08	2.82e-08	1.18e-08
nash1	IP-1	8	0	0	9	0.0000	6.52e-07	7.94e-08	2.25e-15	6.52e-07
outrata31	IP-1	8	0	0	9	0.0000	1.06e-08	2.03e-08	3.79e-09	9.93e-09
outrata32	IP-1	8	0	0	9	0.0000	1.06e-08	2.03e-08	3.79e-09	9.93e-09
outrata33	IP-1	8	0	0	9	0.0000	1.06e-08	2.03e-08	3.79e-09	9.93e-09
outrata34	IP-1	8	0	0	9	0.0000	1.06e-08	2.03e-08	3.79e-09	9.93e-09
portfl1	IP-2*	2098	0	0	2100	5.6403	2.21e-09	3.19e-09	2.21e-09	4.98e-17
qpec1	IP-2	12	0	0	13	0.0040	2.66e-07	9.20e-11	0.00000	5.96e-08
qpecgen	**									
ralph2	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
ralphmod	IP-1	16	0	0	17	0.8080	7.28e-09	1.64e-07	6.40e-11	6.95e-09
scale1	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
scale2	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
scale3	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
scale4	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
scale5	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
scholtes1	PG-1	13	2	3	16	0.0000	6.40e-09	6.40e-09	6.40e-09	9.31e-15
scholtes2	PG-1	13	2	3	16	0.0000	6.40e-09	6.40e-09	6.40e-09	9.31e-15
scholtes3	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07
scholtes4	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	1.61e-17	9.53e-07
scholtes5	IP-2	11	0	0	12	0.0000	3.37e-07	2.32e-10	0.00000	2.38e-07
sl1	IP-2	13	0	0	14	0.0000	4.09e-07	4.14e-10	1.94e-14	4.09e-07
stackelberg1	IP-1	7	0	0	8	0.0000	3.49e-07	8.48e-06	3.40e-15	3.49e-07
traffic1	**									

The performance of the PGUN algorithm for finding a simple feasible solution of the 48 MPCCs is illustrated in Table 3. These results indicate that in general the algorithm converged fast to a solution of HNCP, as it performed a small number of IP iterations. In fact, there was only one case in which PGUN required too many IP iterations and only 2 instances where the algorithm required 2 slow PG iterations. For 3 instances the stopping criterion SC3 was applied to avoid the slow convergence of PGUN to a stationary point of the merit function that would not be a solution of HNCP. In these 3 cases PGUN converged to a solution of HNCP by using an alternative starting point. Finally, the algorithm was unable to find a feasible solution of the MPCC in two instances.

We also note from the values of Feas and Comp that PGUN is usually able to compute accurate feasible solutions of the MPCC. Furthermore, the use of the stopping criterion SC2 was shown appropriate for such a goal. This is an interesting point as these accurate solutions can be used as initial points for projected and active-set algorithms [11, 16, 26, 38] that have been designed for the computation of stationary points of MPCC.

In order to have a better idea of the performance of PGUN in practice, we also solved the test problems by the PLM algorithm. The results of the performance of this method are displayed in Table 4, where the notations mentioned before were used together with the following additional ones:

TERM: algorithm termination, which can be one of the following:

LM-1: algorithm stopped with a Levenberg-Marquardt (LM) iteration satisfying SC1.

LM-2: algorithm stopped with a LM iteration satisfying SC2.

PG-1: algorithm stopped with a projected gradient (PG) iteration satisfying SC1.

LM: number of LM iterations (steps 2, 3 and 4).

PG: number of PG iterations (step 5).

The numerical results indicate that the PLM algorithm used a small number of fast LM iterations to converge and rarely employs slow PG iterations. As before, the stopping criterion SC3 was used in order to stop prematurely the convergence to points that are not feasible solutions of MPCC. As for the PGUN algorithm the use of the stopping criterion SC2 usually leads to accurate feasible solutions of MPCC (see values in the columns Comp and Feas). Finally, the

Table 4: Performance of the PLM Method for Experiment 1

Problem	TERM	LM	PG	CG	NE	TIME	$\|\|F(\bar{z})\|\|$	SPG_norm	Feas	Comp
bard1	LM-1	3	0	0	4	0.0000	3.35e-09	1.02e-08	8.84e-16	3.35e-09
bard2	LM-1*	11	1	2	23	0.0000	5.76e-08	2.09e-07	2.17e-11	4.44e-08
bard3	LM-2	7	0	0	8	0.0000	7.25e-09	2.44e-08	5.67e-09	4.45e-09
bilevel1	**									
bilevel3	LM-2	8	0	0	9	0.0000	6.74e-09	5.46e-09	1.63e-09	6.54e-09
bilin	LM-1	7	0	0	8	0.0000	8.81e-12	4.40e-12	1.51e-16	8.81e-12
dempe	LM-2	42	0	0	43	0.0000	6.13e-07	1.71e-06	4.33e-07	4.33e-07
designcent1	LM-2*	9	0	0	10	0.0000	1.06e-10	3.25e-10	1.06e-10	7.45e-14
desilva	LM-2	8	0	0	9	0.0000	1.91e-10	3.07e-10	1.83e-10	3.95e-11
df1	LM-1	9	0	0	10	0.0000	2.02e-08	6.36e-08	1.93e-08	5.52e-09
ex911	LM-2	6	0	0	126	0.0000	9.56e-10	7.93e-10	9.44e-16	7.91e-10
ex921	LM-2	5	0	0	113	0.0000	1.61e-10	3.20e-10	6.76e-16	1.57e-10
ex922	LM-2*	7	0	0	8	0.0000	2.81e-10	5.95e-07	4.39e-15	2.17e-10
ex925	LM-2	11	0	0	12	0.0000	5.16e-08	6.35e-09	1.05e-15	5.16e-08
ex928	LM-1	7	0	0	8	0.0000	1.78e-07	4.48e-08	1.00e-16	1.78e-07
flp2	LM-1	5	0	0	6	0.0000	6.75e-10	2.69e-10	5.93e-16	6.73e-10
gauvin	LM-1*	6	0	0	7	0.00000	5.18e-09	7.95e-10	4.97e-13	5.18e-09
gnash1	**									
hakonsen	**									
jr1	LM-2	18	0	0	19	0.0000	7.68e-08	1.37e-10	1.08e-19	7.68e-08
jr2	LM-2	18	0	0	19	0.0000	7.68e-08	1.37e-10	1.08e-19	7.68e-08
kth1	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07
kth2	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07
kth3	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07
liswet1-inv50	**									
nash1	LM-2	7	0	0	8	0.0000	4.93e-08	2.21e-08	1.36e-15	4.93e-08
outrata31	LM-1	8	0	0	18	0.0000	2.54e-07	7.06e-07	1.13e-07	2.27e-07
outrata32	LM-1	8	0	0	18	0.0000	2.54e-07	7.06e-07	1.13e-07	2.27e-07
outrata33	LM-1	8	0	0	18	0.0000	2.54e-07	7.06e-07	1.13e-07	2.27e-07
outrata34	LM-1	8	0	0	18	0.0000	2.54e-07	7.06e-07	1.13e-07	2.27e-07
portfl1	**									
qpec1	LM-2	19	0	0	20	0.0000	5.68e-07	4.49e-10	0.00000	1.32e-07
qpecgen	**									
ralph2	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07
ralphmod	**									
scale1	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07
scale2	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07
scale3	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07
scale4	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07
scale5	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07
scholtes1	LM-1	8	0	0	9	0.0000	3.60e-07	4.32e-08	2.49e-08	3.59e-07
scholtes2	LM-1	8	0	0	9	0.0000	3.60e-07	4.32e-08	2.49e-08	3.59e-07
scholtes3	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07
scholtes4	LM-1	16	0	0	11	0.0000	2.02e-04	8.90e-06	3.04e-14	1.96e-04
scholtes5	LM-1	3	0	0	98	0.0000	9.99e-19	9.99e-19	0.00000	9.99e-19
sl1	**									
stackelberg1	LM-1*	25	0	0	1767	0.0000	2.29e-09	2.61e-09	3.12e-15	2.29e-09
traffic1	**									

PLM method seems to have more failures for finding a feasible solution than the PGUN algorithm. This leads to our recommendation of using PGUN for computing a feasible solution of an MPCC.

5.4. Experiment 2: Computing a Target Feasible Solution of MPCC

Next, we report the experiments with PGUN and PLM for computing a target feasible solution (i.e., a solution of HNCP (5)) of the MPCC test problems mentioned before when the target value c_t is the best value given by the collection. The definition of the test problems used in this experiment and the numerical results on the performance of the algorithms for these instances are displayed in Table 5 and Tables 6 and 7, respectively. In these tables we used the notations mentioned before and the additional one:

SLACK: represents the value of the slack variable associated to the target constraint. If SLACK is greater than a tolerance 10^{-6}, then the algorithm was able to compute a better feasible solution than the one given by the collection.

Table 5: Number of complementary pairs for Experiment 2

Problem	NCP	NNG	Problem	NCP	NNG	Problem	NCP	NNG
bard1	3	3	gauvin	2	3	qpecgen	100	3
bard2	4	18	gnash1	8	3	ralph2	1	1
bard3	2	6	hakonsen	4	5	ralphmod	100	9
bilevel1	6	6	jr1	1	1	scale1	1	1
bilevel3	4	4	jr2	1	1	scale2	1	1
bilin	6	4	kth1	1	1	scale3	1	1
dempe	1	1	kth2	1	1	scale4	1	1
design-cent1	3	4	kth3	1	1	scale5	1	1
desilva	2	5	liswet1-inv50	50	52	scholtes1	1	2
df1	1	5	nash1	2	5	scholtes2	1	2
ex911	5	2	outrata31	4	3	scholtes3	1	1
ex921	4	3	outrata32	4	3	scholtes4	1	3
ex922	4	5	outrata33	4	3	scholtes5	2	1
ex925	3	3	outrata34	4	3	sl1	3	8
ex928	2	4	portfl1	12	63	stackelberg1	1	3
flp2	2	5	qpec1	10	11	traffic1	244	495

The numerical results indicate the same type of performance shown before. However, there is an increase of failures of the algorithms when the objective function constraint is included in the HNCP associated to a target feasible solution. Furthermore PGUN and PLM always computed the feasible solution given by the collection (see values in the column SLACK). These conclusions confirm the conclusions in [12] that computing a target feasible solution is usually more difficult than finding a simple feasible solution.

Table 6: Performance of the PGUN method for Experiment 2

Problem	TERM	IP	PG	CG	NE	TIME	$\|\|F(\bar{z})\|\|$	SPG_norm	Feas	Comp	SLACK
bard1	IP-1	9	0	0	10	0.0000	7.93e-10	6.21e-09	7.72e-10	1.81e-10	-6.68e-18
bard2	IP-1	18	0	0	19	0.0080	1.06e-12	9.27e-07	1.06e-12	8.36e-14	-1.72e-23
bard3	IP-1	13	0	0	14	0.0000	1.54e-08	8.22e-08	1.48e-08	3.98e-09	-1.55e-17
bilevel1	**										
bilvel3	IP-1	13	0	0	14	0.0000	2.74e-08	1.49e-07	2.70e-08	4.91e-09	3.12e-17
bilin	**										
dempe	**										
design-cent1	**										
desilva	IP-2	13	0	0	14	0.0000	5.01e-07	8.52e-07	5.01e-07	1.41e-11	-1.03e-16
df1	IP-2	13	0	0	14	0.0000	7.90e-07	2.02e-06	7.89e-07	4.69e-08	-1.53e-16
ex911	IP-1	10	0	0	11	0.0000	6.26e-13	7.65e-13	2.11e-15	4.52e-13	1.29e-17
ex921	IP-1	9	0	0	10	0.0000	7.74e-09	5.92e-08	7.37e-09	1.68e-09	-2.22e-18
ex922	IP-2	17	0	0	18	0.0040	6.18e-07	3.93e-07	1.96e-08	6.12e-07	-3.15e-22
ex925	IP-2	16	0	0	17	0.0000	4.48e-07	2.00e-06	4.48e-07	1.12e-15	1.04e-16
ex928	IP-2	8	0	0	9	0.0000	2.13e-09	6.00e-09	7.22e-10	2.01e-09	1.15e-13
flp2	IP-2	19	0	0	20	0.0000	3.59e-07	6.72e-10	3.59e-07	2.16e-17	-1.11e-16
gauvin	IP-1	20	0	0	21	0.0000	3.24e-07	2.90e-06	3.24e-07	3.54e-16	-1.06e-16
gnash1	IP-1*	43	0	0	57	0.0040	2.36e-07	8.88e-07	2.36e-07	3.09e-14	-5.87e-17
hakonsen	IP-1*	10	0	0	11	0.0000	8.37e-15	9.94e-13	8.37e-15	5.07e-22	1.44e-05
jr1	IP-2	11	0	0	12	0.0000	3.45e-07	4.88e-07	3.45e-07	3.00e-17	-1.52e-18
jr2	IP-2	12	0	0	13	0.0000	3.65e-07	5.17e-07	3.65e-07	1.86e-17	2.60e-18
kth1	IP-1*	5	0	0	6	0.0000	6.87e-07	4.88e-07	6.87e-07	2.84e-10	1.56e-10
kth2	IP-1	2	0	0	3	0.0000	6.12e-07	3.53e-07	4.99e-07	2.49e-07	2.49e-07
kth3	IP-2	12	0	0	13	0.0000	6.38e-07	6.38e-07	6.38e-07	7.83e-17	-1.93e-17
liswet1-inv50	**										
nash1	IP-2*	14	0	0	15	0.0000	6.86e-07	1.13e-09	6.86e-07	5.00e-19	-1.43e-17
outrata31	IP-1	11	0	0	12	0.0000	5.05e-08	1.23e-07	5.05e-08	1.29e-09	1.24e-14
outrata32	II-IP*	2197	0	0	5607	0.2280	3.70e-06	1.07e-05	3.70e-06	9.13e-15	1.16e-14
outrata33	IP-1	15	0	0	16	0.0000	2.54e-06	7.39e-06	2.54e-06	5.88e-16	-5.96e-15
outrata34	IP-1	14	0	0	15	0.0000	2.61e-06	7.29e-06	2.61e-06	1.63e-16	8.24e-16
portfl1	**										
qpec1	IP-2*	20	0	0	22	0.0080	5.61e-07	8.49e-09	5.61e-07	2.92e-16	1.40e-17
qpecgen	**										
ralph2	IP-2	13	0	0	14	0.0000	4.01e-07	3.58e-07	3.58e-07	1.79e-07	-6.20e-25
ralphmod	**										
scale1	IP-2	11	0	0	12	0.0000	7.88e-07	1.57e-06	7.88e-07	8.23e-17	-4.34e-19
scale2	IP-2	39	0	0	85	0.0000	3.31e-07	6.62e-07	3.31e-07	2.75e-19	3.00e-19
scale3	IP-2	15	0	0	16	0.0000	3.16e-07	6.33e-07	3.16e-07	4.58e-19	1.30e-19
scale4	IP-1*	86	0	0	275	0.0000	4.86e-06	9.35e-06	4.60e-05	8.36e-07	1.04e-06
scale5	IP-1*	14	0	0	15	0.0000	2.01e-08	4.03e-06	2.01e-08	2.35e-17	2.96e-19
scholtes1	IP-2	15	0	0	16	0.0000	7.98e-07	1.42e-09	7.98e-07	4.42e-16	-1.75e-17
scholtes2	IP-2*	6	0	0	7	0.0000	4.52e-08	1.80e-07	4.52e-08	1.49e-10	2.65e-10
scholtes3	IP-2*	8	0	0	9	0.0000	4.13e-07	4.13e-07	4.13e-07	1.04e-16	-1.68e-17
scholtes4	IP-2	11	0	0	12	0.0000	3.52e-07	3.05e-10	1.84e-18	2.39e-07	-2.84e-20
scholtes5	PG-1	19	1	1	32	0.0000	2.22e-05	2.10e-07	2.22e-05	1.09e-19	-2.21e-20
sl1	IP-2	15	0	0	16	0.0000	6.44e-07	5.81e-09	2.88e-07	5.76e-07	-1.10e-16
stackelberg1	IP-1	27	0	0	28	0.0000	7.76e-08	3.73e-06	7.76e-08	2.99e-14	-7.33e-18
traffic1	**										

6. Conclusions

In this paper, we introduced a Projected-Gradient Underdetermined Newton-like (PGUN) algorithm for computing a feasible solution of a Mathematical Programming Problem with Complementarity Constraints (MPCC). The algorithm can also be applied for the computation of a feasible solution of MPCC that satisfies a certain objective function target. In both cases the algorithm searches a solution of an associated Horizontal Complementarity Problem (HNCP). It was shown that PGUN is globally convergent to a solution of HNCP or to a stationary point of an associated natural merit function. Fast local convergence was established under reasonable hypotheses. The PGUN algorithm seems to perform well for the computation of feasible solutions of an MPCC and seems

Table 7: Performance of the PLM method for Experiment 2

Problem	TERM	LM	PG	CG	NE	TIME	$\|\|F(\overline{z})\|\|$	SPG_norm	Feas	Comp	SLACK
bard1	**										
bard2	**										
bard3	LM-2	83104	0	0	7171315	8.0325	6.59e-08	3.64e-07	6.59e-08	1.00e-17	4.22e-18
bilevel1	**										
bilvel3	LM*	4097	0	0	4120	0.5920	2.31e-03	6.48e-03	2.31e-03	8.71e-14	0.00000
bilin	**										
dempe	**										
design-cent1	LM-1*	9	0	0	39	0.0000	3.40e-07	5.08e-07	2.35e-07	1.34e-07	-1.16e-07
desilva	LM-2*	18	1	2	145	0.0040	2.09e-08	1.27e-06	2.66e-09	2.07e-08	-1.94e-07
df1	**										
ex911	**										
ex921	**										
ex922	**										
ex925	**										
ex928	LM-2*	10	1	2	125	0.0000	9.35e-13	5.79e-10	1.16e-15	9.35e-13	1.40e-18
flp2	LM-2*	22	0	0	52	0.0000	9.76e-07	4.58e-09	9.76e-07	2.49e-17	9.42e-18
gauvin	**										
gnash1	**										
hakonsen	LM-1*	13	0	0	26	0.0000	4.52e-08	1.21e-07	5.16e-10	4.52e-08	1.44e-05
jr1	LM-2*	14	0	0	15	0.0000	5.68e-07	8.04e-07	5.68e-07	1.06e-17	4.50e-18
jr2	LM-2*	15	0	0	16	0.0000	7.56e-07	1.06e-06	7.56e-07	1.97e-19	5.24e-19
kth1	LM-1	7	1	2	119	0.0000	7.31e-19	4.08e-10	6.69e-19	2.44e-19	-8.88e-20
kth2	LM-1	1	0	0	2	0.0000	9.00e-10	0.00000	9.00e-10	0.00000	0.00000
kth3	LM-2*	4	0	0	5	0.0000	7.23e-07	7.22e-07	7.23e-07	9.65e-10	6.81e-16
liswet1-inv50	**										
nash1	**										
outrata31	LM-1*	10	0	2	29	0.0000	4.48e-09	1.84e-07	4.48e-09	9.18e-13	1.92e-13
outrata32	LM*	2197	0	0	298916	0.2360	3.72e-06	1.08e-05	3.72e-06	1.13e-15	1.24e-16
outrata33	**										
outrata34	**										
portfl1	**										
qpec1	LM-1*	1	0	0	2	0.0000	2.11e-07	8.68e-08	2.07e-07	1.20e-08	-1.05e-08
qpecgen	**										
ralph2	LM-2	18	0	0	19	0.0000	6.66e-07	5.96e-07	5.96e-07	2.98e-07	5.92e-18
ralphmod	**										
scale1	LM-2	10	0	0	11	0.0000	9.58e-07	1.91e-06	9.58e-07	9.87e-18	7.46e-18
scale2	LM-2*	23	0	0	24	0.0000	5.65e-07	1.13e-06	5.65e-07	3.60e-20	1.80e-20
scale3	LM-1*	1	1	1	14	0.0000	1.64e-08	1.62e-08	1.64e-08	1.39e-10	1.39e-10
scale4	**										
scale5	LM-1*	19	1	2	123	0.0040	3.68e-08	7.37e-06	3.68e-08	7.09e-20	7.01e-17
scholtes1	LM-2*	22	0	0	23	0.0040	5.35e-07	1.17e-09	5.35e-07	2.97e-17	1.29e-16
scholtes2	LM-1*	10	1	2	222	0.0040	5.50e-11	3.23e-09	5.50e-11	4.53e-16	-5.92e-16
scholtes3	LM-1*	1	0	0	2	0.0000	6.38e-09	8.20e-09	5.31e-09	2.50e-09	2.50e-09
scholtes4	PG-1	16	1	1	37	0.0040	2.17e-06	1.93e-08	4.48e-09	1.46e-06	3.40e-18
scholtes5	LM-2	19	0	0	20	0.0000	5.79e-07	1.32e-09	5.79e-07	2.82e-18	4.21e-18
sl1	**										
stackelberg1	**										
traffic1	**										

to be more efficient than a Projected Levenberg-Marquardt (PLM) algorithm designed before for the same goal. The choice of the initial point for the PGUN and PLM algorithms seems to have an important impact on the efficiency of these algorithms. Future research will address the combination of PGUN with algorithms that require feasible initial points for solving MPCC in order to solve practical problems.

7. Acknowledgement

We are indebted to the associate editor and two anonymous referees for helpful remarks that improved a lot the quality of this paper.

References

[1] R. Andreani, C. Dunder, and J. M. Martínez. Nonlinear-programming reformulation of the order-value optimization problem. *Mathematical Methods of Operations Research*, 61(3):365–384, 2005.

[2] R. Andreani, J. J. Júdice, J. M. Martínez, and J. Patrício. On the natural merit function for solving complementarity problems. *Mathematical Programming*, 130(1):211–223, 2011.

[3] R. Andreani, J. J. Júdice, J. M. Martínez, and J. Patrício. A projected-gradient interior-point algorithm for complementarity problems. *Numerical Algorithms*, 57(4):457–485, 2011.

[4] M. Anitescu. On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 15(4):1203–1236, 2005.

[5] M. Anitescu, P. Tseng, and S. J. Wright. Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties. *Mathematical Programming*, 110(2):337–371, 2007.

[6] H. Y. Benson, A. Sen, D. F. Shanno, and J. Vanderbei. Interior-point algorithms, penalty methods and equilibrium problems. *Computational Optimization and Applications*, 34(2):155–182, 2006.

[7] I. M. Bomze. Copositive optimization–recent developments and applications. *European Journal of Operational Research*, 216(3):509–520, 2012.

[8] X. Chen. Smoothing methods for complementarity problems and their applications: a survey. *Journal of the Operations Research Society of Japan*, 43:32–47, 2000.

[9] X. Chen and T. Yamamoto. Newton-like methods for solving underdetermined nonlinear equations with nondifferentiable terms. *Journal of Computational and Applied Mathematics*, 55(3):311–324, 1994.

[10] A. Ehrenmann and K. Neuhoff. A comparison of electricity market designs in networks. *Operations research*, 57(2):274–286, 2009.

[11] H. Fang, S. Leyffer, and T. Munson. A pivoting algorithm for linear programming with linear complementarity constraints. *Optimization Methods and Software*, 27(1):89–114, 2012.

[12] L. Fernandes, A. Friedlander, M. Guedes, and J. J. Júdice. Solution of a general linear complementarity problem using smooth optimization and its application to bilinear programming and lcp. *Applied Mathematics & Optimization*, 43(1):1–19, 2001.

[13] M. C. Ferris and J. S. Pang. Engineering and economic applications of complementarity problems. *Siam Review*, 39(4):669–713, 1997.

[14] R. Fletcher and S. Leyffer. Solving mathematical programs with complementarity constraints as nonlinear programs. *Optimization Methods and Software*, 19(1):15–40, 2004.

[15] M. Fukushima, Z. Q. Luo, and J. S. Pang. A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints. *Computational Optimization and Applications*, 10(1):5–34, 1998.

[16] M. Fukushima and P. Tseng. An implementable active-set algorithm for computing a b-stationary point of a mathematical program with linear complementarity constraints. *SIAM Journal on Optimization*, 12(3):724–739, 2002.

[17] R. García-Rodenas and D. Verastegui-Rayo. A column generation algorithm for the estimation of origin–destination matrices in congested traffic networks. *European Journal of Operational Research*, 184(3):860–878, 2008.

[18] M. S. Gowda. Reducing a monotone horizontal lcp to an lcp. *Applied Mathematics Letters*, 8(1):97–100, 1995.

[19] L. Guo, G-H. Lin, D. Zhang, and D. Zhu. An mpec reformulation of an epec model for electricity markets. *Operations Research Letters*, 43(3):262–267, 2015.

[20] T. Hoheisel, C. Kanzow, and A. Schwartz. Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints. *Mathematical Programming*, 137(1-2):257–288, 2013.

[21] HSL. A collection of fortran codes for large scale scientific computation. *http://www.hsl.rl.ac.uk*, 2013.

[22] X. Hu and D. Ralph. Using epecs to model bilevel games in restructured electricity markets with locational prices. *Operations research*, 55(5):809–827, 2007.

[23] X. M. Hu and D. Ralph. Convergence of a penalty method for mathematical programming with complementarity constraints. *Journal of Optimization Theory and Applications*, 123(2):365–390, 2004.

[24] H. Jiang and D. Ralph. Extension of quasi-newton methods to mathematical programs with complementarity constraints. *Computational Optimization and Applications*, 25(1-3):123–150, 2003.

[25] J. J. Júdice. Optimization with linear complementarity constraints. *Pesquisa Operacional*, 34(3):559–584, 2014.

[26] J. J. Júdice, H. D. Sherali, I. M. Ribeiro, and A. M. Faustino. Complementarity active-set algorithm for mathematical programming problems with equilibrium constraints. *Journal of Optimization Theory and Applications*, 134(3):467–481, 2007.

[27] C. Kanzow, N. Yamashita, and M. Fukushima. Levenberg-marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints. *Journal of Computational and Applied Mathematics*, 173(2):321–343, 2005.

[28] R. M. Kovacevic and Ch. G. Pflug. Electricity swing option pricing by stochastic bilevel optimization: A survey and new approaches. *European Journal of Operational Research*, 237(2):389–403, 2014.

[29] S. Leyffer. Macmpec: Ampl collection of mpecs. *Argonne National Laboratory. Available at http://wiki.mcs.anl.gov/leyffer/index.php/MacMPEC*, 2000.

[30] S. Leyffer, G. López-Calva, and J. Nocedal. Interior methods for mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 17(1):52–77, 2006.

[31] D. H. Li and M. Fukushima. A derivative-free line search and global convergence of broyden-like method for nonlinear equations. *Optimization Methods and Software*, 13(3):181–201, 2000.

[32] G-H. Lin and M. Fukushima. Stochastic equilibrium problems and stochastic mathematical programs with equilibrium constraints: A survey. *Pacific Journal of Optimization*, 6(3):455–482, 2010.

[33] G-H. Lin, D. Zhang, and Y-C Liang. Stochastic multiobjective problems with complementarity constraints and applications in healthcare. *European Journal of Operational Research*, 226(3):461–470, 2013.

[34] Z. Q. Luo, J. S. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints.* Cambridge University Press, 1996.

[35] K. G. Murty. *Linear complementarity.* Linear and Nonlinear Programming, Heldermann, Berlin, 1988.

[36] J. Outrata, M. Kocvara, and J. Zowe. *Nonsmooth approach to optimization problems with equilibrium constraints: theory, applications and numerical results*, volume 28. Springer Science & Business Media, 1998.

[37] J. S. Pang. Partially b-regular optimization and equilibrium problems. *Mathematics of Operations Research*, 32(3):687–699, 2007.

[38] D. Ralph. Nonlinear programming advances in mathematical programming with complementarity constraints. *Royal Society*, 2007.

[39] D. Ralph and O. Stein. The c-index: A new stability concept for quadratic programs with complementarity constraints. *Mathematics of Operations Research*, 36(3):504–526, 2011.

[40] H. Scheel and S. Scholtes. Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity. *Mathematics of Operations Research*, 25(1):1–22, 2000.

[41] F. Toyasaki, P. Daniele, and T. Wakolbinger. A variational inequality formulation of equilibrium models for end-of-life products with nonlinear constraints. *European Journal of Operational Research*, 236(1):340–350, 2014.

[42] J. Walpen, M. E. Mancinelli, and P. A. Lotito. A heuristic for the od matrix adjustment problem in a congested transport network. *European Journal of Operational Research*, 242(3):807–819, 2015.

[43] D. Wu, Y. Yin, and S. Lawphongpanich. Pareto-improving congestion pricing on multimodal transportation networks. *European Journal of Operational Research*, 210(3):660–669, 2011.

[44] J. Yao, I. Adler, and S. S. Oren. Modeling and computing two-settlement oligopolistic equilibrium in a congested electricity network. *Operations Research*, 56(1):34–47, 2008.

[45] J. Yao, S. S. Oren, and I. Adler. Two-settlement electricity markets with price caps and cournot generation firms. *European Journal of Operational Research*, 181(3):1279–1296, 2007.

[46] J. J. Ye. Necessary optimality conditions for multiobjective bilevel programs. *Mathematics of Operations Research*, 36(1):165–184, 2011.