Comput Optim Appl (2010) 47: 401–429
DOI 10.1007/s10589-008-9232-3

Exact penalties for variational inequalities with applications to nonlinear complementarity problems

Thiago A. de André · Paulo J.S. Silva

Received: 24 March 2007 / Revised: 8 December 2008 / Published online: 15 January 2009

Abstract In this paper, we present a new reformulation of the KKT system associated to a variational inequality as a semismooth equation. The reformulation is derived from the concept of differentiable exact penalties for nonlinear programming. The best theoretical results are presented for nonlinear complementarity problems, where simple, verifiable, conditions ensure that the penalty is exact. We close the paper with some preliminary computational tests on the use of a semismooth Newton method to solve the equation derived from the new reformulation. We also compare its performance with the Newton method applied to classical reformulations based on the Fischer-Burmeister function and on the minimum. The new reformulation combines the best features of the classical ones, being as easy to solve as the reformulation that uses the Fischer-Burmeister function while requiring as few Newton steps as the one that is based on the minimum.

Keywords Variational inequality · Semismooth reformulation · Exact penalty · Nonlinear complementarity

1 Introduction

Consider a constrained nonlinear programming problem

$$\begin{aligned} \min\ & f(x) \\ \text{s.t.}\ & g(x) \le 0 \\ & h(x) = 0, \end{aligned} \tag{NLP}$$

T.A. de André was supported by FAPESP, grant 02/10942-9.
P.J.S. Silva was partially supported by CNPq, grants PQ 304133/2004-3, 477083/2006-4, and PRONEX—Optimization.

T.A. de André · P.J.S. Silva (✉)
Instituto de Matemática e Estatística, University of São Paulo, São Paulo, Brazil
e-mail: pjssilva@ime.usp.br

where x lies in $\mathbb{R}^n$ and $f:\mathbb{R}^n \to \mathbb{R}$, $g:\mathbb{R}^n \to \mathbb{R}^m$, and $h:\mathbb{R}^n \to \mathbb{R}^p$ are C^2 functions.

Penalty methods are a very popular framework to solve such problems. In these methods, the constrained problem is replaced by a sequence of unconstrained ones. A good example is the augmented Lagrangian algorithm, that can be derived from a proximal point method applied to the Lagrangian dual problem [3, 28].

Another possibility is the exact penalty approach, where a special penalty function is used to transform (NLP) into a single unconstrained problem. For example, it is easy to see that, under reasonable assumptions, the solutions to (NLP) are exactly the unconstrained minima of

$$\min_{x\in\mathbb{R}^n} \phi(x,\mu)$$
$$\stackrel{\text{def}}{=} f(x) + \mu \max\bigl\{0, g_1(x), g_2(x), \dots, g_m(x), |h_1(x)|, |h_2(x)|, \dots, |h_p(x)|\bigr\},$$

if μ is chosen big enough [4]. However, this unconstrained problem is naturally nonsmooth and special methods should be used to solve it. Moreover, it is not easy to estimate how large μ must be to ensure the equivalence of the minima.

To overcome the lack of differentiability of the maximum function, many authors proposed differentiable formulas for exact penalties. The history of differentiable exact penalties starts with Fletcher in 1970, when he published a series of three articles proposing automatic updates for the multipliers in the augmented Lagrangian method for equality constrained problems [16, 17, 19]. The idea was to estimate the multipliers as a function of the primal variables, denoted by $\lambda(x)$, followed by the minimization of the associated augmented Lagrangian

$$f(x) + \langle \lambda(x), h(x)\rangle + c_k\|h(x)\|^2.$$

However, the multiplier function was not easy to compute and it was not clear how to choose good values for the penalty parameter c_k. Later on, in 1975, Mukai and Polak proposed a new formula for $\lambda(x)$ and showed that there is a threshold for c_k that once achieved would allow the modified augmented Lagrangian to recover the solutions of the original problem after a single minimization [24].

In 1979, Di Pillo and Grippo presented a new formulation for exact penalties that simplified the analysis of the associated problems [6]. In this work, they propose to further extend the augmented Lagrangian function, penalizing deviations from the first order conditions:

$$f(x) + \langle \lambda, h(x)\rangle + c_k\|h(x)\|^2 + \bigl\|M(x)\bigl(\nabla f(x) + Jh(x)'\lambda\bigr)\bigr\|^2,$$

where $Jh(x)$ denotes the Jacobian of h at x. Special choices for $M(x)$ resulted in modified augmented Lagrangians that are quadratic in λ. In this case, it is possible to isolate the dual variable in terms of x. One of such choices for $M(x)$ recovered the method proposed by Fletcher and the results from Mukai and Polak.

This last formulation is also important because it is able to deal with inequality constraints using slack variables s_i and replacing the inequalities by equalities in the form $h_{p+i}(x) \stackrel{\text{def}}{=} g_i(x) + s_i^2 = 0,\ i = 1,\dots,m$. With an appropriate choice for $M(x)$, one obtains a quadratic problem in the slacks. Then, the slacks can be written as an explicit function of the original variables x. However, in this case it is not known how to isolate the multipliers λ as a function of x.

In 1973, Fletcher had already extended his ideas to deal with inequality constraints [18], but the proposed function lacked good differentiability properties. In 1979, Glad and Polak proposed a new formula for $\lambda(x)$ in inequality constrained problems and showed how to control the parameter c_k [20].

Finally, in 1985 and 1989, Di Pillo and Grippo reworked the results from Glad and Polak and created a differentiable exact penalty for inequality constrained problems that depends only on the primal variables [7, 8]. These papers are the base of our work. In particular, from now on we focus exclusively on inequality constrained problems.

In this paper we extend the ideas of Di Pillo and Grippo to variational inequalities with functional constraints and the related KKT system. The remaining of the paper is organized as follows: Sect. 2 presents the formula for the penalty, Sect. 3 derives the exactness results, Sect. 4 specializes the results for Nonlinear Complementarity Problems (NCP), and Sect. 5 uses the proposed penalty to develop a semismooth Newton method for complementarity. This last section is closed with some preliminary computational results comparing the new penalty with classical NCP functions.

2 Extending exact penalties

As described above, it is possible to build differentiable exact penalties for constrained optimization problems using an augmented Lagrangian function coupled with a multiplier estimate computed from the primal point. A natural multiplier estimate for inequality constrained problems was given by Glad and Polak. It is computed solving, in the least-squares sense, the equations involving the multipliers in the KKT conditions

$$\min_{\lambda\in\mathbb{R}^m} \|\nabla_x L(x,\lambda)\|^2 + \zeta^2\|G(x)\lambda\|^2, \tag{1}$$

where L is the usual Lagrangian function, $\zeta > 0$, and $G(x) \in \mathbb{R}^{m\times m}$ is a diagonal matrix with $G(x)_{ii} = g_i(x)$. The first term tries to find a multiplier for which the fixed primal point is a minimum of the Lagrangian function.[1] The second term tries to enforce the complementarity conditions.

This problem is convex and quadratic in λ and can be easily solved if the point x conforms to the Linear Independence Constraint Qualification (LICQ), that is, if the gradients of the constraints that are active at x are linearly independent. The results concerning (1) are summarized in the following proposition.

Proposition 2.1 [20, *Proposition* 1] *Assume that* $x \in \mathbb{R}^n$ *conforms to LICQ and define the matrix* $N(x) \in \mathbb{R}^{m\times m}$ *by*

$$N(x) \overset{\text{def}}{=} Jg(x)Jg(x)' + \zeta^2 G(x)^2.$$

[1] Actually, a first order stationary point.

Then,

1. $N(x)$ *is positive definite.*
2. *The solution* $\lambda(x)$ *of* (1) *is*

$$\lambda(x) = -N^{-1}(x) J g(x) \nabla f(x).$$

3. *If* $(\bar{x}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^m$ *is a KKT pair where LICQ holds then* $\bar{\lambda} = \lambda(\bar{x})$, *i.e.* $\bar{\lambda}$ *solves* (1) *for* $x = \bar{x}$.
4. [7, *Proposition* 4] *If LICQ holds in a neighborhood of* x, *then* $\lambda(\cdot)$ *is continuously differentiable at* x *and its Jacobian is given by*

$$J\lambda(x) = -N^{-1}(x) \Bigg[Jg(x) \nabla^2_{xx} L\big(x, \lambda(x)\big) + \sum_{i=1}^{m} e^i \nabla_x L(x, \lambda(x))' \nabla^2 g_i(x) + 2\zeta^2 \Lambda(x) G(x) J g(x) \Bigg], \tag{2}$$

where e^i *is the* i*-th element of the canonical base of* $\mathbb{R}^m$ *and* $\Lambda(x) \in \mathbb{R}^{m\times m}$ *is a diagonal matrix with* $\Lambda(x)_{ii} = \lambda(x)_i$.

Using such estimate, one can build a differentiable exact penalty from the standard augmented Lagrangian function,

$$\begin{aligned} L_c(x, \lambda) &\stackrel{\text{def}}{=} f(x) + \frac{1}{2c} \sum_{i=1}^{m} \big(\max\{0, \lambda_i + c g_i(x)\}^2 - \lambda_i^2\big) \\ &= f(x) + \langle \lambda, g(x) \rangle + \frac{c}{2} \|g(x)\|^2 - \frac{1}{2c} \sum_{i=1}^{m} \max\{0, -\lambda_i - c g_i(x)\}^2. \end{aligned}$$

The resulting exact penalty function, that we call $w_c(\cdot)$, is obtained plugging the multiplier estimate in the augmented Lagrangian

$$w_c(x) \stackrel{\text{def}}{=} L_c(x, \lambda(x)). \tag{3}$$

Our aim is to extend the definition of w_c to the context of variational inequalities. Let $F: \mathbb{R}^n \to \mathbb{R}^n$ be a continuous function and $\mathcal{F} \subset \mathbb{R}^n$ a non-empty closed set. The variational inequality problem is to find $x \in \mathcal{F}$ such that

$$\forall y \in \mathcal{F}, \quad \langle F(x), y - x \rangle \geq 0. \tag{VIP}$$

If $\mathcal{F}$ is convex it is easy to see that (VIP) is an extension of the geometrical optimality conditions for (NLP) where the gradient of the objective function is replaced by a general continuous function. In this paper we focus on problems where the feasible set can be described as a system of twice differentiable convex inequalities $\mathcal{F} = \{x \mid g(x) \leq 0\}$. We also assume that it is non-empty.

In 1999, Eckstein and Ferris proposed an extension of multiplier methods to nonlinear mixed complementarity problems [11], variational inequalities where $\mathcal{F}$ is a box. Afterwards, Auslender and Teboulle proposed an extension of the notion of augmented Lagrangians to (VIP) with general functional constraints [1, 2]. Their results are related to the rich duality theory for generalized equations developed by Pennanen [25].

Following these developments, a natural way to extend the notion of exact penalty to (VIP) is to use the gradient of $w_c(\cdot)$, replacing ∇f by F. However, such gradient involves the Jacobian of $\lambda(\cdot)$ given in Proposition 2.1. This first choice for exact penalty would have a very complicated formula, depending on the Hessians of the constraints and the Jacobian of F which may not be easily available.

To overcome such difficulty, we start with the classical augmented Lagrangian for variational inequality, which is the gradient of $L_c(\cdot,\cdot)$ with respect to the first variable. We then plug into it the multiplier estimate $\lambda(\cdot)$:

$$\lambda(x) \stackrel{\text{def}}{=} -N^{-1}(x)Jg(x)F(x), \tag{4}$$

$$\begin{aligned} W_c(x) &\stackrel{\text{def}}{=} F(x) + Jg(x)'\lambda(x) + cJg(x)'g(x) \\ &\quad + cJg(x)'\max\{0, -\lambda(x)/c - g(x)\} && (5) \\ &= F(x) + Jg(x)'\lambda(x) + cJg(x)'\max\{g(x), -\lambda(x)/c\}. && (6) \end{aligned}$$

In the next sections we will show the relation between the zeros of W_c, for c large enough, and the KKT system associated to (VIP).

Definition 2.2 The Karush-Kuhn-Tucker (KKT) system associated to (VIP) is

$$\begin{aligned} F(x) + Jg(x)'\lambda &= 0, && \text{(Zero Condition)} \\ g(x) &\le 0, && \text{(Primal Feasibility)} \\ \lambda &\ge 0, && \text{(Dual Feasibility)} \\ \forall i = 1,\ldots,m, \quad \lambda_i g_i(x) &= 0. && \text{(Complementarity)} \end{aligned}$$

A pair $(x,\lambda) \in \mathbb{R}^{n+m}$ that conforms to these equations is called a KKT pair. The primal variable x is called a KKT (stationary) point.

This system is known to be equivalent to (VIP) whenever the feasible set $\mathcal{F}$ is defined by convex inequalities and conforms to a constraint qualification [12].

Some comments must be made before presenting the exactness properties for W_c. Note that since W_c is not the gradient of w_c, its zeros are not clearly related to the solutions of an unconstrained optimization problem. In this sense, the proposed exact penalty approach is not equivalent to the penalties usually proposed in the optimization literature. *In particular, it has the major advantage of not depending on the Jacobian of F and on second order information of the constraints.*

As for the differentiability properties of W_c, the maximum function in its definition clearly make it nonsmooth. This is a direct heritage of the classical augmented Lagrangian used to derive it. Even though, it is (strongly) semismooth if F is (LC^1) C^1 and g is (LC^2) C^2. Therefore, its zeros can be found by an extension of the Newton method to semismooth equations [26, 27]. We present such a method in Sect. 5.

3 Exactness results

Let us present the exactness results for W_c. Here we follow closely the results presented in the nonlinear programming case by Di Pillo and Grippo [7, 8]. First, we show that the proposed penalty has zeros whenever the original KKT system has solutions.

In order to define W_c in the whole space we will need the following assumption, that we assume valid throughout this section:

Assumption 3.1 *LICQ holds on the whole* $\mathbb{R}^n$, *so that* $\lambda(\cdot)$ *and, hence,* W_c *is well-defined everywhere.*

This assumption is restrictive, but was present already in the original papers on (differentiable) exact penalties [6–8, 20]. Fortunately, in many cases it is easily verifiable. For example, it holds trivially in nonlinear and mixed complementarity problems.

Proposition 3.2 *Let* (x, λ) *be a KKT pair. Then, for all* $c > 0$, $W_c(x) = 0$.

Proof The LICQ assumptions ensures that $\lambda = \lambda(x)$. Then,

$$\begin{aligned} W_c(x) &= F(x) + Jg(x)'\lambda(x) + cJg(x)' \max\{g(x), -\lambda(x)/c\} \\ &= 0 + cJg(x)' \max\{g(x), -\lambda/c\} \\ &= 0, \end{aligned}$$

where the last equality follows from primal and dual feasibility and the complementary condition. □

Next, we show that for c large enough the zeroes of W_c are nearly feasible. Then, we show that if a zero is nearly feasible it will be a KKT point associated to (VIP).

Proposition 3.3 *Let* $\{x^k\} \subset \mathbb{R}^n$ *and* $\{c_k\} \subset \mathbb{R}_+$ *be sequences such that* $x^k \to \bar{x}$, $c_k \to \infty$, *and* $W_{c_k}(x^k) = 0$. *Then,* $\bar{x} \in \mathcal{F}$.

Proof We have,

$$0 = W_{c_k}(x^k) = F(x^k) + Jg(x^k)'\lambda(x^k) + c_k Jg(x^k)' \max\{g(x^k), -\lambda(x^k)/c_k\}.$$

Now recall that, under LICQ, $\lambda(\cdot)$ is continuous. Moreover, F is assumed continuous and g continuously differentiable. Hence, as $x^k \to \bar{x}$ and $c_k \to \infty$, we may divide the equation above by c_k and take limits to conclude that

$$0 = \sum_{i=1}^{m} \max\{g_i(\bar{x}), 0\} \nabla g_i(\bar{x}). \tag{7}$$

Now, define

$$\mathcal{G}(x) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{i=1}^{m} \max\{0, g_i(x)\}^2,$$

a function that measures the "infeasibility" of x. Observe that $\mathcal{G}$ is convex, as the function $\max\{0, \cdot\}^2$ is convex and non-decreasing, and the constraints are convex. Moreover, it is differentiable and

$$\nabla \mathcal{G}(x) = \sum_{i=1}^{m} \max\{g_i(x), 0\} \nabla g_i(x).$$

Hence, (7) shows that $\bar{x}$ minimizes $\mathcal{G}$. Therefore, it must be feasible as the feasible set is supposed to be nonempty. □

Proposition 3.4 *Let $\bar{x} \in \mathcal{F}$. Then, there are $c_{\bar{x}}, \delta_{\bar{x}} > 0$ such that if $\|x - \bar{x}\| \le \delta_{\bar{x}}$, $c > c_{\bar{x}}$, and $W_c(x) = 0$ imply that $(x, \lambda(x))$ is a KKT pair associated to (VIP).*

Proof Let us introduce some notation

$$y(x) \stackrel{\text{def}}{=} \max\{0, -\lambda(x)/c - g(x)\}.$$

We will use capital letters to denote the usual diagonal matrix build from vectors. For example, $Y(x)$ denotes the diagonal matrix with $y(x)$ in the diagonal.

It is easy o show that,

$$Y(x)\lambda(x) = -cY(x)\big(g(x) + y(x)\big).$$

Hence,

$$\begin{aligned} Jg(x)\big(F(x) + Jg(x)'\lambda(x)\big) &= Jg(x)F(x) + Jg(x)Jg(x)'\lambda(x) \\ &= -N(x)\lambda(x) + Jg(x)Jg(x)'\lambda(x) \\ &= -\zeta^2 G(x)^2 \lambda(x) \\ &= -\zeta^2 G(x)\big(G(x) + Y(x)\big)\lambda(x) + \zeta^2 G(x)Y(x)\lambda(x) \\ &= -\zeta^2 G(x)\Lambda(x)\big(g(x) + y(x)\big) + \zeta^2 G(x)Y(x)\lambda(x). \end{aligned} \tag{8}$$

We combine the last results to get

$$\frac{1}{c} Jg(x)\big(F(x) + Jg(x)'\lambda(x)\big) = -\zeta^2 G(x)\left(\frac{1}{c}\Lambda(x) + Y(x)\right)\big(g(x) + y(x)\big).$$

Recalling the definition of W_c, we have

$$\begin{aligned} \frac{1}{c} Jg(x)W_c(x) &= \frac{1}{c} Jg(x)\big(F(x) + Jg(x)'\lambda(x)\big) + Jg(x)Jg(x)'\big(g(x) + y(x)\big) \qquad (9) \\ &= K(x, c)\big(g(x) + y(x)\big), \qquad (10) \end{aligned}$$

where $K(x, c) \stackrel{\text{def}}{=} Jg(x)Jg(x)' - \zeta^2 G(x)Y(x) - \frac{1}{c}\zeta^2 G(x)\Lambda(x)$.

For $x = \bar{x}$, recalling that it is feasible, if $c \to \infty$ then $y(\bar{x}) \to -g(\bar{x})$ and therefore $K(\bar{x}, c) \to N(\bar{x})$. As $N(\bar{x})$ is nonsingular due to LICQ, we can conclude that there must be $c_{\bar{x}}, \delta_{\bar{x}} > 0$ such that if $\|x - \bar{x}\| \le \delta_{\bar{x}}$, $c > c_{\bar{x}}$ then $K(x, c)$ is also nonsingular.

Let x, c be such that $\|x - \bar{x}\| \le \delta_{\bar{x}}$, $c > c_{\bar{x}}$ and $W_c(x) = 0$. Then, (10) imply that $g(x) + y(x) = 0$. This is equivalent to

$$\max\{g(x), -\lambda(x)/c\} = 0 \quad \Longleftrightarrow$$
$$g(x) \le 0, \qquad \lambda(x) \ge 0, \qquad \lambda_i(x) g_i(x) = 0, \quad i = 1, \ldots, m.$$

That is, the pair $(x, \lambda(x))$ is primal and dual feasible and it is complementary. Finally, plugging the first equation above in (6), we can see that the zero condition is also valid. □

These two results may be combined in the following exactness theorem:

Theorem 3.5 *Let $\{x^k, c_k\} \subset \mathbb{R}^{n+1}$ be a sequence such that $W_{c_k}(x^k) = 0$, $c_k \to \infty$, and $\{x^k\}$ is bounded. Then, there is a finite index K such that for $k > K$, $(x^k, \lambda(x^k))$ is a KKT solution associated to* (VIP).

Proof Suppose, by contradiction, that we can extract a sub-sequence $\{x^{k_j}\}$ of points that are not KKT. Since $\{x^k\}$ is bounded, we can assume without loss of generality that $\{x^{k_j}\}$ converges to some $\bar{x}$. Using Proposition 3.3 we conclude that $\bar{x}$ is feasible. Then, Proposition 3.4 ensures that when x^{k_j} is close enough to $\bar{x}$, $(x^{k_j}, \lambda(x^{k_j}))$ will be a solution to the KKT system. □

Corollary 3.6 *If there is a $\bar{c} \in \mathbb{R}$ such that the set $\{x \mid W_c(x) = 0,\ c > \bar{c}\}$ is bounded, then there is a $\tilde{c} > 0$ such that $W_c(x) = 0,\ c > \tilde{c}$, implies that $(x, \lambda(x))$ is a KKT solution associated to* (VIP).

Proof Suppose by contradiction that the result is false. Then, there must be $c_k \to \infty$ and a sequence $\{x^k\} \subset \mathbb{R}^n$ such that $W_{c_k}(x^k) = 0$, and such that $(x^k, \lambda(x^k))$ is not a KKT solution. But for $c_k > \bar{c}$ we have that x^k belongs to the bounded set $\{x \mid W_c(x) = 0,\ c > \bar{c}\}$ and then $\{x^k\}$ is bounded. This is not possible, as Theorem 3.5 ensures that for big enough k, $(x^k, \lambda(x^k))$ is a KKT solution. □

A drawback in the exactness results presented above is that they do not show how to choose the penalty parameter c. On the other hand, Glad and Polak presented in [20] a simple algorithm to update iteratively c in search for the right penalty parameter. To achieve this, they introduce a test function $t(x, c)$ that measures the risk of finding a zero of the penalty function W_c that is not associated to a KKT solution. Using our notation, the test function becomes

$$\begin{aligned} a(x, c) &\stackrel{\text{def}}{=} \max\{g(x), -\lambda(x)/c\}, \\ t(x, c) &\stackrel{\text{def}}{=} -\|W_c(x)\|^2 + \|a(x, c)\|^2 / c^\gamma, \end{aligned} \tag{11}$$

where $\gamma > 0$. In [20], γ is chosen to be 3, but we show below that smaller values are possible.

The function $a(\cdot,\cdot)$ has many interesting properties. First, it is easy to see that it is continuous and, for c fixed, it has a zero at x if, and only if, $g(x) \le 0$, $\lambda(x) \ge 0$, and complementarity holds. Moreover, from the definition of the exact penalty in (6), it follows that

$$W_c(x) = F(x) + Jg(x)'\lambda(x) + cJg(x)'a(x,c).$$

Hence, $(x, \lambda(x))$ is a KKT point if, and only if, $W_c(x) = 0$ and $a(x,c) = 0$ or, equivalently, if $W_c(x) = 0$ and $t(x,c) \le 0$.

To adapt Glad and Polak results to (VIP), we need to prove that for any $\bar{x} \in \mathbb{R}^n$, we can choose $c_{\bar{x}}$ big enough such that $t(x,c) \le 0$ for all $c > c_{\bar{x}}$ and x in a neighborhood of $\bar{x}$. We prove this fact in the next two results.

Lemma 3.7 *Let S be a compact subset of $\mathbb{R}^n$ with no KKT points. Then, there are $c_S, \epsilon_S > 0$ such that, for all $x \in S$ and all $c > c_S$, $\|W_c(x)\| \ge \epsilon_S$.*

Proof We start as in the proof of [20, Lemma 3]. Suppose by contradiction that there are two sequences, $\{x^k\} \subset S$ converging to some $x \in S$ and $c_k \to \infty$, such that $\|W_c(x^k)\| \to 0$. Recalling (6) and using the continuity of the functions involved we have

$$\underbrace{F(x^k) + Jg(x^k)'\lambda(x^k)}_{\text{bounded}} + \underbrace{c_k}_{\to\infty} \underbrace{Jg(x^k)' \max\left\{g(x^k), -\lambda(x^k)/c_k\right\}}_{\text{converges to } 0} \to 0. \tag{12}$$

We can now proceed as in the proof of Proposition 3.3 to see that x is feasible.

Define now,

$$\begin{aligned}\tilde{\lambda}^k &\stackrel{\text{def}}{=} \lambda(x^k) + c_k \max\left\{g(x^k), -\lambda(x^k)/c_k\right\}\\ &= \max\left\{\lambda(x^k) + c_k g(x^k), 0\right\}.\end{aligned}$$

It follows from (12) that $F(x^k) + Jg(x^k)^T\tilde{\lambda}^k \to 0$. The definition of $\tilde{\lambda}^k$ shows that if $g_i(x) < 0$, then $\tilde{\lambda}_i^k = 0$ for k big enough. Hence, we can use LICQ to conclude that $\tilde{\lambda}^k$ converges to some $\tilde{\lambda} \ge 0$, such that $F(x) + Jg(x)^T\tilde{\lambda} = 0$ and which is complementary to $g(x)$. Therefore $(x, \tilde{\lambda})$ would be a KKT pair, contradicting the assumption that S does not have KKT points. □

Proposition 3.8 *For all $\bar{x} \in \mathbb{R}^n$, there are $c_{\bar{x}}, \delta_{\bar{x}} > 0$ such that if $c \ge c_{\bar{x}}$ and $\|x - \bar{x}\| \le \delta_{\bar{x}}$ then $t(x,c) \le 0$.*

Proof Suppose by contradiction that there are two sequences $x_k \to \bar{x}$ and $c_k \to \infty$ such that $t(x^k, c_k) > 0$. Let us consider two cases:

1. $\bar{x}$ is not a KKT point. We can apply Lemma 3.7 to the compact set $S \stackrel{\text{def}}{=} \{x^k\} \cup \{\bar{x}\}$. Then, for k big enough,

$$t(x^k, c_k) \le -\epsilon_S^2 + \underbrace{\|a(x^k, c_k)\|^2 / c_k^\gamma}_{\to 0}.$$

A contradiction.

2. $\bar{x}$ is a KKT point. Using (10) and the discussion that follows it, we see that

$$K(x^k, c_k)a(x^k, c_k) = \frac{1}{c_k} Jg(x_k) W_{c_k}(x^k),$$

where $K(x^k, c_k)$ converges to the non-singular matrix $N(\bar{x})$ and $Jg(x^k)$ converges to $Jg(\bar{x})$. Hence, for k big enough

$$\|a(x^k, c_k)\| \le \frac{2}{c_k} \|N(\bar{x})^{-1}\| \|Jg(\bar{x})\| \|W_{c_k}(x^k)\|.$$

Then,

$$\begin{aligned} t(x^k, c_k) &= -\|W_{c_k}(x^k)\|^2 + \frac{1}{c_k^\gamma} \|a(x^k, c_k)\|^2 \\ &\le -\|W_{c_k}(x^k)\|^2 + \frac{2}{c_k^{\gamma+2}} \|N(\bar{x})^{-1}\|^2 \|Jg(\bar{x})\|^2 \|W_{c_k}(x^k)\|^2, \end{aligned}$$

which becomes negative as $c_k \to \infty$, a contradiction. □

Now, following [20], we can modify any algorithm that finds zeroes of W_c to update the c parameter whenever it approaches a point that is not a solution of the original (VIP).

Exact Penalty with dynamical update of the penalty parameter (EPDU) *Let $A(x, c)$ denote the iteration function of an algorithm that computes a zero of W_c. Choose $x^0 \in \mathbb{R}^n$, $c_0 > 0$, and $\xi > 1$ and set $k = 0$.*

1. *If x^k is an approximate solution to* (VIP), *then stop.*
2. *If $t(x^k, c_k) > 0$, make $c_k = \xi c_k$, go to step* 2.
3. *Compute $x^{k+1} = A(x^k, c_k)$, set $k = k + 1$ and go to step* 1.

Theorem 3.9 *If the sequence $\{x^k\}$ computed by the EPDU method is infinite and bounded, then all of its accumulations points solve* (VIP).

Proof If $\{x^k\}$ is bounded, if follows from Proposition 3.8 that the Step 2 can only increase c_k a finite number of times, in particular $t(x^k, c_k) \le 0$ for k big enough. Let c denote the largest c_k value and x be an accumulation point of $\{x^k\}$. Since x is an accumulation point of the algorithm described by the map A, it is a zero of W_c.

Moreover, as $t(\cdot, c)$ is continuous, $t(x, c) \leq 0$. It follows that x is a KKT point and therefore a solution to (VIP). See the discussion following the definition of the test function. □

We should stress that in [20], Glad and Polak assumed strict complementarity and a second order conditions on the solutions computed by EPDU in order to prove a convergence similar to Theorem 3.9. Our approach is able to avoid such stringent assumptions.

Finally, observe that, even though the EPDU method show us how to dynamically search for a good penalty parameter, the exactness results still depend on boundedness assumptions that may not be easily verifiable. To avoid such difficulty, the optimization literature uses an extraneous compact set that should contain the feasible region or at least a minimum [7, 8]. However it is not clear how to choose such compact. Another approach is to exploit coerciveness or monotonicity properties that may be already present in the (VIP). Usually such assumptions are already necessary to ensure that the problem has solutions. We present in the next section some results in this direction for complementarity problems, an important instance of (VIP).

4 Exact Penalties for nonlinear complementarity problems

We specialize the proposed exact penalty to nonlinear complementarity problems:

$$F(x) \geq 0, \quad x \geq 0, \quad \langle F(x), x\rangle = 0. \tag{NCP}$$

It is easy to see that (NCP) is a Variational Inequality with $\mathcal{F} = \mathbb{R}^n_+$ and, as stated before, that LICQ holds everywhere.

After some algebra, we may see that the proposed exact penalty $W_c(\cdot)$, simplifies to

$$W_c(x)_i = \min\left\{\frac{\zeta^2 x_i^2}{1+\zeta^2 x_i^2} F(x)_i + c x_i,\, F(x)_i\right\}, \quad i = 1, \ldots, n. \tag{13}$$

In particular, the multiplier estimate can be computed explicitly.

In this case we can derive a reasonable assumption that ensures that, for large c, the zeros of $W_c(\cdot)$ are solutions to (NCP).

Theorem 4.1 *Assume that there are ρ, $M > 0$ such that $\langle F(x), x\rangle \geq -M$ for $\|x\| > \rho$ or that F is monotone and* (NCP) *has a solution. Then, there is a $\bar{c} > 0$ such that $W_c(\cdot)$ is exact for $c > \bar{c}$, i.e. any zero of $W_c(x)$ for $c > \bar{c}$ is a solution to* (NCP).

Proof Suppose, by contradiction, that the result does not hold. Then there are $c_k \to \infty$, and a sequence $\{x^k\}$ such that $W_{c_k}(x^k) = 0$ and x^k is not a solution to (NCP). Theorem 3.5 asserts that $\|x^k\| \to \infty$. Proposition 3.4 says that x^k is never feasible, as otherwise $(x^k, \lambda(x^k))$ would be a KKT pair associated to (NCP) contradicting the assumption that x^k is not a solution.

For each x^k and each of its coordinates, (13) allows only three possibilities:

1. If $x_i^k > 0$, then $F(x^k)_i = 0$.
   Observe that (13) implies $F(x^k)_i \geq W_{c_k}(x^k)_i = 0$. If $F(x^k)_i > 0$, $W_{c_k}(x^k)_i$ would be the minimum of two strictly positive numbers, which contradicts the fact that it is zero.
2. If $x_i^k = 0$, then $F(x^k)_i \geq 0$.
   It follows from $W_{c_k}(x^k)_i = 0$, that $\min\{0, F(x^k)_i\} = 0$. This is equivalent to $F(x^k)_i \geq 0$.
3. If $x_i^k < 0$, then $F(x^k)_i = -c_k \frac{1+\zeta^2(x_i^k)^2}{\zeta^2 x_i^k}$.
   First, if $F(x^k)_i \leq 0$, $W_{c_k}(x^k)_i$ would be the minimum of a strictly negative number and a negative number. This contradicts $W_{c_k}(x^k)_i = 0$. Now, as $F(x^k)_i > 0$, it is clear that the minimum is achieved in the first term, leading to

$$0 = \frac{\zeta^2(x_i^k)^2}{1+\zeta^2(x_i^k)^2} F(x^k)_i + c_k x_i^k,$$

   which gives the desired result.

Note that the cases above show that $F(x^k) \geq 0$.

Now consider that there are $\rho,\ M > 0$ such that $\langle F(x), x\rangle \geq -M$ for $\|x\| > \rho$. On the other hand, we have just proved that

$$\begin{aligned}\langle F(x^k), x^k\rangle &= \sum_{x_i^k<0} -c_k \frac{1+\zeta^2(x_i^k)^2}{\zeta^2} \qquad (14)\\ &\to -\infty, \quad [x^k \text{ is not feasible}]\end{aligned}$$

a contradiction.

Finally consider the case where F is monotone and where (NCP) has a solution $\bar{x}$. We have

$$\begin{aligned} 0 &\leq \langle F(x^k) - F(\bar{x}), x^k - \bar{x}\rangle \\ &= \langle F(x^k), x^k\rangle - \langle F(x^k), \bar{x}\rangle - \langle F(\bar{x}), x^k\rangle + \langle F(\bar{x}), \bar{x}\rangle \\ &\leq \langle F(x^k), x^k\rangle - \langle F(\bar{x}), x^k\rangle \\ &\leq \sum_{x_i^k<0} -c_k \frac{1+\zeta^2(x_i^k)^2}{\zeta^2} - F(\bar{x})_i x_i^k \\ &= \sum_{x_i^k<0} \frac{-c_k - \zeta^2 x_i^k(c_k x_i^k + F(\bar{x})_i)}{\zeta^2}, \end{aligned}$$

where the second inequality follows from the fact that $F(x^k), \bar{x} \geq 0$ and $\langle F(\bar{x}), \bar{x}\rangle = 0$, and the third follows from (14) and $F(\bar{x}) \geq 0$.

If, for some x^k, $c_k x_i^k + F(\bar{x})_i \leq 0$ whenever $x_i^k < 0$, the last equation already shows a contradiction as it must be strictly smaller than 0. Hence we conclude that for at least one coordinate, $c_k x_i^k + F(\bar{x})_i > 0$ and we can write

$$\begin{aligned} 0 &\le \sum_{x_i^k<0,\ c_k x_i^k+F(\bar{x})_i>0} \frac{-c_k - \zeta^2 x_i^k (c_k x_i^k + F(\bar{x})_i)}{\zeta^2} \\ &\le \sum_{x_i^k<0,\ c_k x_i^k+F(\bar{x})_i>0} \frac{-c_k - \zeta^2 x_i^k F(\bar{x})_i}{\zeta^2} \\ &\le \sum_{x_i^k<0,\ c_k x_i^k+F(\bar{x})_i>0} \frac{-c_k + \zeta^2 F(\bar{x})_i^2 / c_k}{\zeta^2} \quad [c_k x_i^k + F(\bar{x})_i > 0] \\ &\to -\infty, \end{aligned}$$

a contradiction. □

The coerciveness assumption on $F(\cdot)$ that appears in Theorem 4.1 is not very restrictive. In particular, it is related to a weak coercive property associated to the compactness of the solution set of an NCP [12, Proposition 2.2.7]:

Proposition 4.2 *Let F conform to the following coerciveness property*:

$$\liminf_{\|x\|\to\infty} \frac{\langle F(x), x\rangle}{\|x\|^\eta} > 0,$$

for some $\eta \ge 0$. Then, there is a $\rho > 0$ such that $\langle F(x), x\rangle \ge 0$ for $\|x\| > \rho$. In particular, the coercive assumption of Theorem 4.1 holds for any $M > 0$.

Proof There must be an $\epsilon > 0$ such that

$$\liminf_{\|x\|\to\infty} \frac{\langle F(x), x\rangle}{\|x\|^\eta} > 2\epsilon.$$

This implies that there is a $\rho > 0$ such that if $\|x\| > \rho$, $\frac{\langle F(x), x\rangle}{\|x\|^\eta} \ge \epsilon$, which implies that $\langle F(x), x\rangle \ge 0$. □

We should end this section with two remarks. First, observe that a similar approach may lead to an explicit formula for W_c for a mixed complementarity problem (MCP). In an MCP the variables are subject to general box constraints and hence LICQ also holds trivially. For example, if the box is $[0, 1] \times \cdots \times [0, 1]$, the exact penalty would be

$$\begin{aligned} W_c(x)_i = F_i(x) &+ \min\left\{ \frac{(x_i - 1)^2}{x_i^2 + \zeta^2 x_i^2 (1 - x_i)^2 + (1 - x_i)^2} F_i(x) + c x_i, 0 \right\} \\ &+ \max\left\{ \frac{x_i^2}{x_i^2 + \zeta^2 x_i^2 (1 - x_i)^2 + (1 - x_i)^2} F_i(x) + c(1 - x_i), 0 \right\}, \\ &\forall i = 1, \dots, n. \end{aligned}$$

Note that an exactness result similar to Theorem 4.1 can also be proved for MCP.

Second, it is not possible do derive a closed formula for the exact penalty associated to a variational inequality with general constraints. Hence, in this case an algorithm to solve the semismooth equation associated to the exact penalty would involve, typically, the solution of two linear systems at each step. One with m equations, to compute the multiplier estimates used to define the penalty, and the next with n equations to compute a Newton direction. Note that this complexity still compares favorably with the complexity of more classical reformulations, like the ones based on the Fischer-Burmeister or other NCP functions, where usually a linear system of $m+n$ equations is solved at each step [12, 13].

5 Numerical methods for nonlinear complementarity problems

Let us build upon the results of the last two sections and develop a semismooth Newton method for nonlinear complementarity problems using W_c. We will focus on nonlinear complementarity problems due to its simple constraint structure. Such structure was explored in the last section leading to a simple formula for the exact penalty and to our best exactness results described in Theorem 4.1. Note that NCP is a very special class of variational inequalities. It has a huge selection of applications and it has been extensively studied in the literature, see the recent Pang and Facchinei books and the numerous references therein [12, 13]. Another advantage of focusing on NCP is the existence of a large selection of test problems available in the MCPLIB collection [9, 10], making it specially well suited for a preliminary numerical experience with exact penalties.

The method we present below is based on the General Line Search Algorithm from [5] and its Levenberg-Marquardt variation presented in [22]. The idea is to use the exact penalty to compute the Newton direction in a semismooth Newton method that will be globalized using the Fischer-Burmeister function $\varphi_{FB}(a,b) \stackrel{\text{def}}{=} \sqrt{a^2+b^2} - a - b$ [15]. This function has the important property that, whenever $\varphi_{FB}(a,b) = 0$, both a and b are positive and complementary. Such functions are called NCP functions. Hence, the nonlinear complementarity problem can be rewritten as

$$\Phi_{FB}(x) \stackrel{\text{def}}{=} \begin{bmatrix} \varphi_{FB}\big(x_1, F(x)_1\big) \\ \vdots \\ \varphi_{FB}\big(x_n, F(x)_n\big) \end{bmatrix} = 0.$$

Under reasonable assumptions, the above system of equations is semismooth and it can be solved using a semismooth Newton algorithm [26]. Moreover, $\Psi_{FB}(x) \stackrel{\text{def}}{=} 1/2\|\Phi_{FB}(x)\|^2$ is differentiable and can be used to globalize the Newton method.

However, there are other important NCP functions whose least square reformulation is not differentiable. They do not have a natural globalization function. In this case, it is usual to build hybrid methods, where the local fast convergence is obtained by a Newton algorithm based on the desired NCP function, but the globalization is achieved using a differentiable merit function like Ψ_{FB}. Such globalization ideas appeared first in [5] and are also described in [12, 13]. A typical choice is the combination of the NCP function based on the minimum, $\Phi_{\min}(x) \stackrel{\text{def}}{=} \min(x, F(x))$, with a merit function based on Fischer-Burmeister. Such combination gives rise to many practical algorithms [5].

Before presenting the variant of the semismooth Newton method used in this paper, it is natural to search for regularity conditions that can ensure fast local convergence. The semismooth Newton method can be shown to converge superlinearly if all the elements of the B-subdifferential at the desired zero x^* are nonsingular [26]. Such zeroes are called BD-regular.

As shown in [5, Sect. 2.2], in complementarity problems the BD-regularity of the zeroes of a reformulation is usually connected to the concepts of b- and R-regularity of the solutions. Note that b-regularity is weaker than R-regularity. However, both conditions are equivalent in important cases, like when x^* is a nondegenerate solution or if F is a P_0 function, in particular if it is monotone.

The next result shows that the penalty W_c presents the same regularity properties as the $\Phi_{\min}$ NCP function:

Proposition 5.1 *Let F be a $\mathcal{C}^1$ function and suppose that x^* is a b-regular solution of* (NCP). *Then, x^* is a BD-regular solution of the system $W_c(x) = 0$, where W_c is defined in* (13) *and $c > 0$.*

Proof The regularity is actually inherited from the minimum function used to define W_c. The proof is analogous to the proof of [5, Proposition 2.10]. □

This proposition guarantees the fast local convergence of a semismooth Newton method that starts in a neighborhood of a b-regular solution to (NCP), see Theorem 5.3.

We can now present a variant of the Newton method based on W_c and globalized by Ψ_{FB}. We use a Levenberg-Marquardt method, as we based our code on the LMMCP implementation of Kanzow and Petra [21, 22]. In order to make the test more interesting, we also consider the hybrid algorithm when the Newton direction is computed using $\Phi_{\min}$ instead of W_c. More formally, we follow the General Line Search Algorithm from [5], and propose the following modification to Algorithm 3.1 in [22]:

Semismooth Levenberg-Marquardt method with alternative search directions (LMAD) *Let $\Psi_{FB}(x) \stackrel{\text{def}}{=} 1/2\|\Phi_{FB}(x)\|^2$, and let G denote either W_c for a fixed $c > 0$ or $\Phi_{\min}$. Choose $x^0 \in \mathbb{R}^n$, $\epsilon_1, \epsilon_2 \geq 0$, $\bar{\mu} > 0$, $\alpha_1 > 0$. Choose $\alpha_2, \beta, \sigma_1 \in (0, 1)$, $\sigma_2 \in (0, \frac{1}{2})$. Set $k = 0$.*

1. *If $\Psi_{FB}(x^k) \leq \epsilon_1$ or $\|\nabla\Psi_{FB}(x^k)\| \leq \epsilon_2$, stop.*
2. *Compute the search direction:*
   (a) *Compute $G(x^k)$, $H_k \in \partial_B G(x^k)$, and choose the Levenberg-Marquardt parameter $\mu_k \in (0, \bar{\mu}]$.*
   (b) *Find d^k such that*

$$(H_k' H_k + \mu_k I)d^k = -H_k' G(x^k).$$

(c) *If*

$$\Psi_{FB}(x^k + d^k) \le \sigma_1 \Psi_{FB}(x^k),$$

set $x^{k+1} = x^k + d^k$, $k = k + 1$, *and go to Step 1.*

(d) *If*

$$\|d^k\| < \alpha_1 \|\nabla\Psi_{FB}(x^k)\|$$

or if

$$\langle d^k, \nabla\Psi_{FB}(x^k)\rangle > -\alpha_2 \|d^k\| \|\nabla\Psi_{FB}(x^k)\|,$$

change d^k *to* $-\nabla\Psi_{FB}(x^k)/\|\nabla\Psi_{FB}(x^k)\|$.

3. *Find the largest value* t_k *in* $\{\beta^l \mid l = 0, 1, 2, \dots\}$ *such that*

$$\Psi_{FB}(x^k + t_k d^k) \le \Psi_{FB}(x^k) + \sigma_2 t_k \langle\nabla\Psi_{FB}(x^k), d^k\rangle.$$

Set $x^{k+1} = x^k + t_k d^k$, $k = k + 1$ *and go to Step* 1.

Note that the conditions in Step 2d ensure that the Armijo search in Step 3 is well defined and will stop in a finite number of steps.

We should stress that the original General Line Search Algorithm from [5] does not need the conditions in Step 2d. In contrast, it is only defined for NCP functions, whose zeroes are exactly the solutions to (NCP). This is not always the case for the exact penalty W_c. In particular, the exact penalty may not conform to Assumption 1 in [5]. Hence the convergence results of [5] can not be used directly to LMAD in this case.

To prove the global convergence of LMAD we use the spacer steps result presented in [4, Proposition 1.2.6]. Note that the conditions in Step 2d were chosen to ensure that the LMAD directions are gradient related, as needed by this result.

Theorem 5.2 *Let* $\{x^k\}$ *be a sequence computed by the LMAD method. Then, every accumulation point is a stationary point of* Ψ_{FB}.

Proof First, let us recall that Ψ_{FB} is continuously differentiable [14, Proposition 3.4].

Now, let $\mathcal{K}$ be the set of indexes where the condition of Step 2c failed. That is, the set of indexes where the Armijo line search took place. It is not difficult to see that $\{d^k\}_{k\in\mathcal{K}}$ is gradient related and it follows from [4, Proposition 1.2.6], that every limit point of $\{x^k\}_{x\in\mathcal{K}}$ is stationary.

Finally, consider an arbitrary convergent subsequence $x^{k_j} \to x^*$, where it is not always true that $k_j \in \mathcal{K}$. If there is still an infinite subset of the indexes k_j that belong to $\mathcal{K}$, we can easily reduce the subsequence to this indexes to see that x^* is stationary. On the other hand, if $k_j \notin \mathcal{K}$ for all big enough k_j, we use the definition of $\mathcal{K}$ to see that for these indexes

$$0 \le \Psi_{FB}(x^{k_{j+1}}) \le \Psi_{FB}(x^{k_j+1}) = \Psi_{FB}(x^{k_j} + d^{k_j}) \le \sigma_1 \Psi_{FB}(x^{k_j}),$$

where the first inequality follows from the monotonicity of $\Psi_{FB}(x^k)$. Hence, it is trivial to see that $\Psi_{FB}(x^{k_j}) \to 0$, and then the monotonicity of $\Psi_{FB}(x^k)$ ensures that the whole sequence goes to zero. In particular, x^* minimizes Ψ_{FB} and $\nabla \Psi_{FB}(x^*) = 0$. □

We can also present a standard result for local convergence rate.

Theorem 5.3 *Let $\{x^k\}$ be a sequence computed by the LMAD method. Assume that it converges to a b-regular solution to* (NCP). *If $\mu_k \to 0$, then eventually the condition in Step* 2c *will be satisfied and $\{x^k\}$ will converge Q-superlinearly to x^*. Moreover, if F is a LC^1-function and $\mu_k = O(\|H_k' G(x^k)\|)$, we have that the convergence is Q-quadratic.*

Proof Remember that G in LMAD is either $\Phi_{\min}$ or W_c for some $c > 0$. Using the BD-regularity at x^* of these functions, given by [5, Proposition 2.10] and Proposition 5.1 above, it follows that x^* is an isolated solution of the equation $G(x) = 0$. Moreover, as x^* is b-regular, it is an isolated solution to (NCP) [12, Corollary 3.3.9]. Hence it is also an isolated solution to the equation $\Psi_{FB}(x) = 0$.

As G is continuous, there is a neighborhood of x^* and constants $c_1 > 0$ and $c_2 > 0$ such that

$$c_1 \|G(x)\|^2 \le \Psi_{FB}(x) \le c_2 \|G(x)\|^2.$$

Note that this is a local version of Lemma 3.4 in [5].

The result now follows as in the proof of Theorems 3.2 and 4.4(b) in [5]. □

5.1 Computational tests

We have implemented the LMAD method in MATLAB starting from the LMMCP code from Kanzow and Petra [21]. In the current version of the code, the authors have incorporated a filter trust region method as a preprocessor before starting the main algorithm [23]. Since we do not want to use the preprocessor in our tests, we have turn it off. Moreover, the code uses a combination of the Fischer-Burmeister function with an extra term to force complementarity. As we wanted to use the pure Fischer-Burmeister function we have adapted the code. This can be achieved, basically, setting a parameter to 1. All the linear algebra is carried out using the sparse matrix implementation from MATLAB.

Let us describe the choice of the parameters for LMAD. Following the LMMCP we used $\epsilon_1 = 10^{-10}$, $\epsilon_2 = 0$, $\beta = 0.55$, and $\sigma_2 = 10^{-4}$. The Levenberg-Marquardt parameter is chosen to be 10^{-16} if the estimated condition number for H_k is less than $1/\epsilon_{mac}$, where ϵ_{mac} denotes the machine epsilon, that stands approximately for 2.2204×10^{-16}. Otherwise, we use $\mu_k = 10^{-2}/(k+1)$.

As for the constants that control the choice of the alternative direction we have $\sigma_1 = 0.5$, $\alpha_1 = \alpha_2 = \sqrt{\epsilon_{mac}}$. In the case that the alternative direction is based on the exact penalty, that is $G = W_c$, we have used the EPDU, see Sect. 3, to update dynamically the penalty parameter c. The initial parameter for the exact penalty were $c_0 = 5$ and $\zeta = 0.2$. The parameter γ, that define the test function t, and the increasing factor ξ were both chosen to be 2. We observe that the original penalty parameter is usually big enough to avoid any updates. Moreover the final c value is always small,

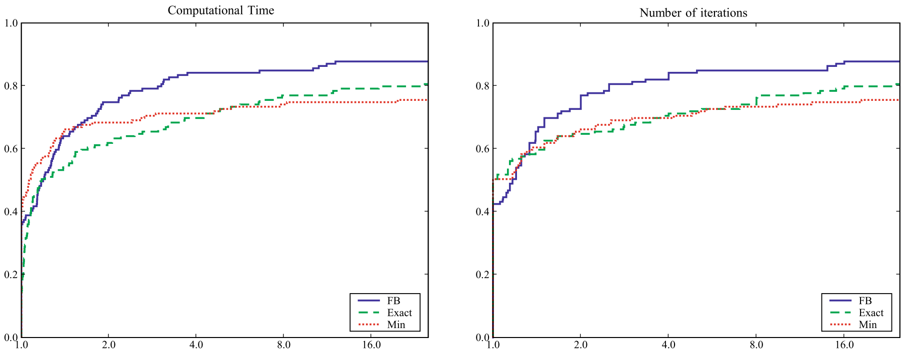

Fig. 1 Performance profile of LMAD variations and the pure Fischer-Burmeister method

never going beyond 160 in our tests. Note that the parameter is increased only in very difficult problems that often lead to failures. The final penalty parameter for all the tests described here can be found in the table with the complete computational results in the Appendix.

The test set is composed by all the nonlinear complementarity problems in the MATLAB version of the MCPLIB test suite. It has 40 different problems, and many of them have different starting points. We have chosen to allow at most 6 different starting points for each problem, summing a total of 139 tests. If a problem had more than 6 starting points, 6 instances were selected randomly. The full list of the selected problems is presented in the Appendix.

Figure 1 presents the performance profile of this first test. The label "FB" stands for the pure Fischer-Burmeister method, while "Exact" represents EPDU+LMAD with $G = W_c$, and "Min" is LMAD with $G = \Phi_{\min}$. We present two profiles. The first uses the total computational time as performance metric. The second is based on the total number iterations, that correspond to the number linear systems solved while calculating the Newton steps. This computation is likely to dominate the computation carried out at each step.

We note that both variations of the LMAD seem to be faster than the pure Fischer-Burmeister method. However the FB method is more reliable. Moreover, the LMAD variation based on the exact penalty seems to be a little more reliable than LMAD based on the $\Phi_{\min}$ function, however it uses a little more computational time at each iteration.

If we analyse the reason for the failures of the LMAD variants in more problems than the FB method, we identify that in some cases the direction computed is not a good descent direction for the merit function based on the Fischer-Burmeister reformulation. This force the LMAD to use the Cauchy step for the merit function as search direction, resulting in a very small improvement.

Hence, it is natural to ask if it is possible to predict, *before the solution of the linear equation described in the Step 2b*, that the resulting direction may not be a good descent direction. In such case, we could try to use a Newton step based on the original merit function instead. With this objective, we propose the following modification of Step 2b:

Modified Alternative Direction *Let* $\theta \in (0, \pi/2)$.
If the angle between $H_k' G(x^k)$ *and* $\nabla \Psi_{FB}(x^k)$ *is smaller that* θ, *find* d^k *such that*

$$(H_k' H_k + \mu_k I) d^k = -H_k' G(x^k). \tag{15}$$

Otherwise, compute $\tilde{H}_k \in \partial_B \Psi_{FB}(x^k)$ *and find* d^k *solving*

$$(\tilde{H}_k' \tilde{H}_k + \mu_k I) d^k = -\nabla \Psi_{FB}(x^k). \tag{16}$$

The idea behind the angle criterion is simple to explain. The solution of (15) can bend its right hand side, $-H_k' G(x^k)$ by a maximum angle of $\pi/2$. Hence, if $-H_k' G(x^k)$ makes a small angle with $-\nabla \Psi_{FB}(x^k)$ the direction computed by the first linear system will be likely a good search direction. On the other hand, if this angle is large, the direction computed by the first linear system can only be a good descent direction if it is bent by the system towards $-\nabla \Psi_{FB}(x^k)$. But there is no guarantee that this will happen. To avoid taking chances, we use the direction based on the merit function itself, given by (16).

The convergence of the modified algorithm can be proved following the same lines of the proofs of Theorems 5.2 and 5.3. In particular, the inequalities that ensure that the search directions are gradient related remains untouched. As for the rate of convergence result, it would require R-regularity of the solution, instead of b-regularity, like in [5, Theorem 4.3]. This is a consequence of the fact that the Newton steps can be taken with respect to Φ_{FB} and not G.

Figure 2 presents the performance profiles of the variations of LMAD when we change Step 2b by the Modified Alternative Direction presented above. The parameter θ was set to $\pi/6$. Both variations, based on the exact penalty W_c and on $\Phi_{\min}$, clearly benefit from the new directions. Both methods became more robust and a little faster.

Figure 3 shows the profile of the three methods together. Here, we can see that the method based on the exact penalty practically dominates the others. It is basically as fast as the method based on $\Phi_{\min}$ but has better robustness, very close to the FB version. However, once again, we can see that each iteration of the method based on the exact penalty takes a little more computational effort than the code based on $\Phi_{\min}$. The reason for this fact is probably a better sparsity structure induced by the minimum function, as discussed in [5].

Note that the main advantage displayed by the exact penalty variation of the LMAD when compared to the version based on $\Phi_{\min}$ is better robustness. However, the globalization based on the Fischer-Burmeister function plays a central role in determining the robustness of LMAD. Hence, we can not be sure if the difference in reliability is only related to the choice of the Newton direction, or if for some reason the exact penalty directions are better suited to the globalization scheme.

To isolate the role of the globalization strategy, we have decided to run one more test where the globalization is dropped entirely. That is, we have run pure Levenberg-Marquardt Newton methods without a merit function or line searches. The result is presented in Fig. 4. Note that in this case the Newton method based on the exact penalty is clearly more robust than the one based on the minimum function. Actually its robustness is already very close to the one achieved by the Fischer-Burmeister

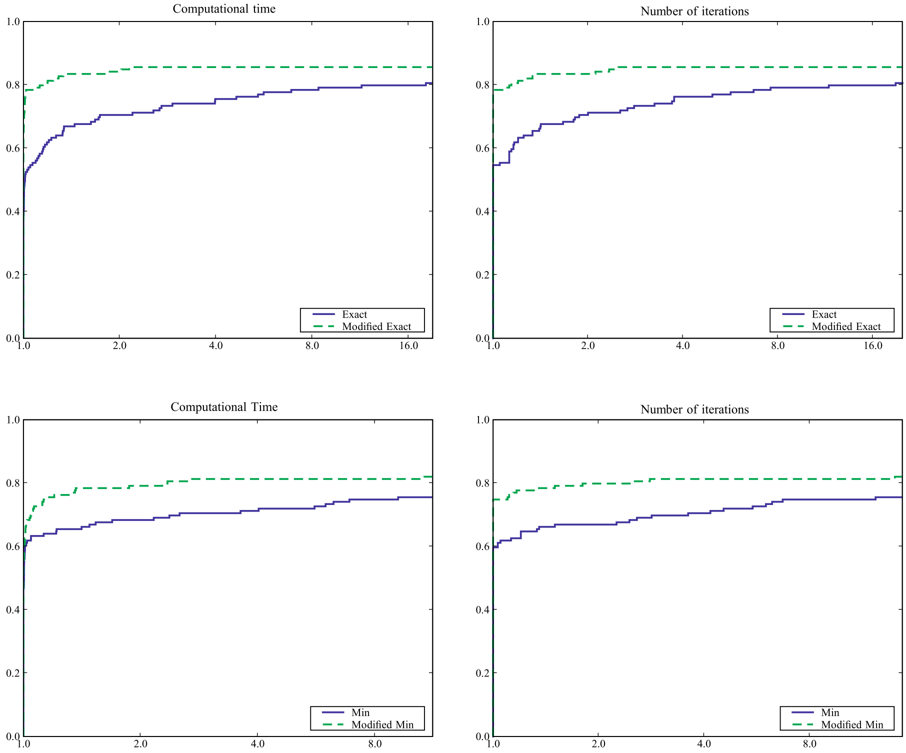

Fig. 2 Performance profiles of the LMAD variations with and without the Modified Alternative Direction

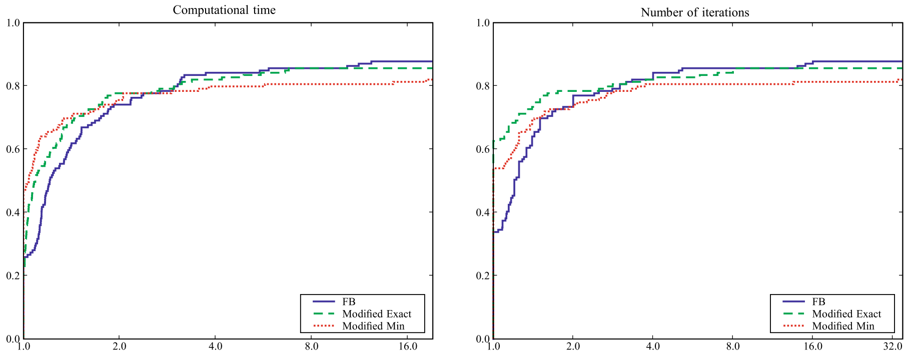

Fig. 3 Performance profile of the LMAD variations using the Modified Alternative Direction and the pure Fischer-Burmeister method

function. The Newton method based on the exact penalty is also the fattest method in this test.

Acknowledgements We would like to thank Ellen Hidemi Fukuda for carefully reading the manuscript, Roberto Andreani for spotting a flaw in the original proof of Proposition 3.3, and the anonymous referees for many valuable comments.

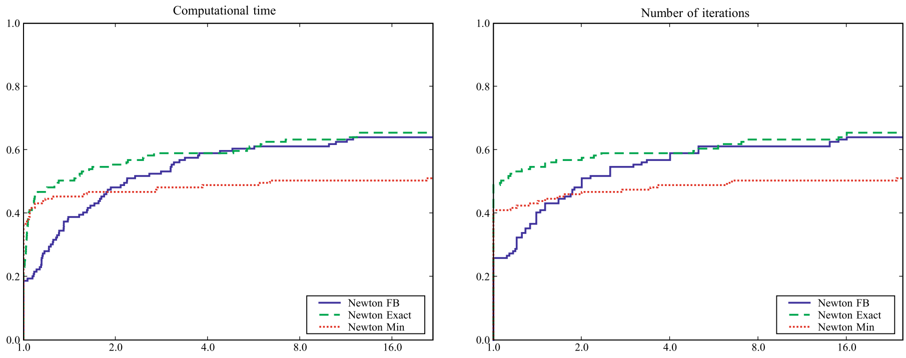

Fig. 4 Performance profile of the pure semismooth Newton method using the Fischer-Burmeister, the exact penalty and the $\Phi_{\min}$ reformulations

Appendix: Tables with numerical results

We present here the full table that was used to draw the performance profiles. The first column shows the problem name, the next eight columns present the performance information of the original LMMCP method based on the Fischer-Burmeister function, of LMAD using the exact penalty W_c and $\Phi_{\min}$, of LMAD with the Modified Alternative Direction based on W_c and $\Phi_{\min}$, and, finally, of the pure Newton method using Fischer Burmeister, W_c, and $\Phi_{\min}$ respectively. Each column is composed of two rows. The first row has the computation time in seconds and the second one presents the number Newton systems solved by each method. If the penalty parameter was increased by the EPDU strategy in a method based on W_c, then the final c value appears between parenthesis in the second row.

Problem	FB	Exact	Min	Mod Ex	Mod Min	New FB	New Ex	New Min
baihaung1	8.6E+00	6.0E+00	4.0E+00	5.1E+00	2.7E+00	8.6E+00	6.1E+00	4.0E+00
	5	3	2	3	3	5	3	2
bertsekas1	6.9E−02	8.8E−02	FAIL	6.7E−02	5.7E−02	2.8E−02	FAIL	FAIL
	28	29		25	39	15		
bertsekas2	6.4E−02	6.7E−02	FAIL	6.1E−02	4.6E−02	2.5E−02	1.8E−02	FAIL
	27	25		25	35	13	9	
bertsekas3	3.4E−02	5.2E−02	FAIL	4.5E−02	6.0E−02	3.0E−02	FAIL	FAIL
	16	24		17	40	16		
bertsekas4	6.9E−02	8.8E−02	FAIL	6.6E−02	5.7E−02	2.8E−02	FAIL	FAIL
	28	29		25	39	15		
bertsekas5	2.6E−02	2.0E−02	1.6E−01	2.1E−02	1.7E−02	2.3E−02	FAIL	FAIL
	12	9	111	9	9	12		
bertsekas6	9.5E−02	1.1E−01	FAIL	6.5E−02	5.2E−02	2.6E−02	FAIL	FAIL
	37	35		25	31	14		

Problem	FB	Exact	Min	Mod Ex	Mod Min	New FB	New Ex	New Min
billups1	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
billups2	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
billups3	2.6E−02 30	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
bishop1	FAIL	FAIL (40)	FAIL	FAIL (10)	FAIL	FAIL	FAIL	FAIL
colvdual1	3.1E−02 14	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
colvdual2	FAIL	FAIL (40)	FAIL	5.2E−01 223	FAIL	FAIL	FAIL	FAIL
colvdual3	2.4E−03 1	2.4E−03 1	2.4E−03 1	2.5E−03 1	2.4E−03 1	2.4E−03 1	2.4E−03 1	2.4E−03 1
colvdual4	3.1E−02 14	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
colvnlp1	2.8E−02 14	2.9E−02 14	2.8E−02 23	3.2E−02 16	2.3E−02 17	4.1E−02 22	FAIL	FAIL
colvnlp2	2.7E−02 14	8.7E−02 43	7.8E−02 71	3.4E−02 17	4.6E−02 29	FAIL	1.0E−01 57	FAIL
colvnlp3	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.1E−03 1	2.2E−03 1	2.2E−03 1
colvnlp4	2.6E−02 13	9.4E−02 45	3.6E−02 33	2.4E−02 12	4.8E−02 32	FAIL	FAIL	FAIL
colvnlp5	2.6E−02 13	9.4E−02 45	3.6E−02 33	2.4E−02 12	4.8E−02 32	FAIL	FAIL	FAIL
colvnlp6	1.8E−02 9	1.8E−02 8	FAIL	1.9E−02 9	1.6E−02 8	FAIL	FAIL	FAIL
cycle1	2.2E−03 3	2.5E−03 3	2.4E−03 3	2.5E−03 3	2.4E−03 3	FAIL	4.6E−02 69	7.3E−02 114
danny11	6.3E−03 9	5.9E−03 8	5.4E−03 7	5.9E−03 8	5.4E−03 7	FAIL	FAIL	FAIL
danny31	2.9E−03 4	9.7E−04 1	9.5E−04 1	9.8E−04 1	1.0E−03 1	2.8E−03 4	9.5E−04 1	9.2E−04 1
danny41	FAIL	7.5E−03 9	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
danny61	2.2E−02 30	5.9E−02 67	5.7E−02 67	8.5E−03 10	8.3E−03 10	FAIL	8.1E−03 12	7.9E−03 12
danny71	3.5E−03 5	2.4E−03 3	1.6E−03 2	2.4E−03 3	1.6E−03 2	3.4E−03 5	2.3E−03 3	1.6E−03 2
danny91	2.9E−03 4	9.6E−04 1	9.3E−04 1	9.7E−04 1	9.4E−04 1	2.8E−03 4	9.4E−04 1	9.1E−04 1

Problem	FB	Exact	Min	Mod Ex	Mod Min	New FB	New Ex	New Min
degen1	3.5E−03 5	1.4E−02 26	3.0E−03 5	5.1E−03 8	3.0E−03 5	3.4E−03 5	FAIL	FAIL
dirkse12	1.0E−02 14	1.9E−02 21	1.8E−02 21	1.8E−02 20	1.8E−02 20	FAIL	FAIL	FAIL
duopoly1	FAIL	FAIL	3.9E−01 262	FAIL	FAIL	FAIL	FAIL	FAIL
explcp1	3.4E−02 19	1.8E−02 9	2.7E−02 17	3.6E−02 19	1.1E−02 6	3.3E−02 19	3.1E−02 17	2.4E−02 16
ferralph11	3.5E−03 5	5.0E−03 8	9.5E−04 1	2.9E−03 4	9.5E−04 1	3.4E−03 5	FAIL	9.1E−04 1
ferralph21	2.2E−03 3	2.4E−03 3	2.3E−03 3	2.4E−03 3	2.3E−03 3	2.2E−03 3	2.3E−03 3	2.2E−03 3
hanskoop2	2.5E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.5E−03 1	2.2E−03 1	2.2E−03 1
hanskoop4	2.5E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.5E−03 1	2.2E−03 1	2.2E−03 1
hanskoop6	2.5E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.5E−03 1	2.2E−03 1	2.2E−03 1
hanskoop8	2.5E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.5E−03 1	2.2E−03 1	2.2E−03 1
hanskoop10	2.5E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.2E−03 1	2.5E−03 1	2.2E−03 1	2.2E−03 1
hansmcf1	5.2E−02 13	3.9E−01 91	5.8E−02 25	4.7E−02 12	1.4E−01 45	FAIL	FAIL	FAIL
jiangqi1	1.6E−03 2	1.0E−03 1	1.1E−03 2	1.0E−03 1	1.1E−03 2	1.6E−03 2	1.0E−03 1	1.1E−03 2
jiangqi2	3.0E−03 4	6.5E−03 8	9.8E−04 1	6.5E−03 8	9.8E−04 1	2.9E−03 4	FAIL (80)	9.7E−04 1
jiangqi3	3.1E−03 4	FAIL	1.7E−03 2	4.3E−03 5	1.7E−03 2	FAIL	3.2E−03 4	1.6E−03 2
josephy1	4.5E−03 6	6.2E−03 8	4.5E−03 6	4.7E−03 6	4.5E−03 6	FAIL	9.4E−03 13	FAIL
josephy2	4.5E−03 6	3.3E−03 4	4.2E−03 5	4.0E−03 5	4.2E−03 5	FAIL	3.2E−03 4	FAIL
josephy4	3.7E−03 5	4.1E−03 5	2.4E−03 3	3.2E−03 4	2.4E−03 3	3.6E−03 5	FAIL (40)	2.4E−03 3
josephy6	4.4E−03 6	4.8E−03 6	5.7E−03 7	4.0E−03 5	5.7E−03 7	FAIL	4.6E−03 6	FAIL
josephy7	3.8E−03 5	1.2E−02 14	2.0E−02 24	4.1E−03 5	3.4E−03 4	FAIL	3.9E−03 5	FAIL
josephy8	2.3E−03 3	1.8E−03 2	1.7E−03 2	1.7E−03 2	1.7E−03 2	2.3E−03 3	1.7E−03 2	1.7E−03 2

Problem	FB	Exact	Min	Mod Ex	Mod Min	New FB	New Ex	New Min
kojshin1	6.0E−03	1.2E−02	1.4E−02	7.6E−03	6.6E−03	8.9E−03	5.0E−03	FAIL
	8	15	18	9	8	13	7	
kojshin3	4.5E−03	1.9E−02	8.9E−02	7.2E−03	1.6E−02	8.3E−03	2.0E−02	FAIL
	6	24(10)	110	9	20	12	28 (20)	
kojshin4	3.0E−03	2.5E−03	2.4E−03	2.5E−03	2.4E−03	2.9E−03	2.4E−03	2.4E−03
	4	3	3	3	3	4	3	3
kojshin6	3.7E−03	4.0E−03	4.9E−03	4.0E−03	4.0E−03	4.9E−03	3.9E−03	FAIL
	5	5	6	5	5	7	5	
kojshin7	4.6E−03	FAIL	2.1E−02	4.1E−03	3.4E−03	1.7E−02	3.9E−03	FAIL
	6		25	5	4	25	5	
kojshin8	2.3E−03	1.8E−03	1.7E−03	1.8E−03	1.7E−03	2.3E−03	1.7E−03	1.7E−03
	3	2	2	2	2	3	2	2
mathinum1	3.0E−03	1.3E−02	4.2E−03	1.2E−02	1.1E−02	2.9E−03	2.1E−02	1.7E−02
	4	15	5	15	14	4	29	25
mathinum2	3.6E−03	3.2E−03	3.1E−03	3.2E−03	3.1E−03	3.5E−03	3.1E−03	3.0E−03
	5	4	4	4	4	5	4	4
mathinum3	9.8E−03	5.7E−03	4.1E−03	5.7E−03	4.1E−03	FAIL	5.2E−03	FAIL
	12	7	5	7	5		7	
mathinum4	5.0E−03	3.9E−03	3.8E−03	3.9E−03	3.8E−03	4.8E−03	3.8E−03	3.7E−03
	7	5	5	5	5	7	5	5
mathinum5	7.0E−03	8.0E−03	7.7E−03	8.0E−03	7.7E−03	6.8E−03	6.6E−03	FAIL
	10	10	10	10	10	10	9	
mathinum6	5.0E−03	5.4E−03	5.3E−03	5.3E−03	5.2E−03	6.1E−03	5.2E−03	5.1E−03
	7	7	7	7	7	9	7	7
mathisum1	3.8E−03	3.3E−03	3.3E−03	3.3E−03	3.3E−03	6.3E−03	4.6E−03	4.5E−03
	5	4	4	4	4	9	6	6
mathisum3	4.4E−03	2.7E−03	3.2E−03	5.8E−03	3.1E−03	4.3E−03	2.4E−03	3.0E−03
	6	3	4	7	4	6	3	4
mathisum4	4.4E−03	4.0E−03	3.9E−03	4.0E−03	3.9E−03	4.3E−03	3.9E−03	3.8E−03
	6	5	5	5	5	6	5	5
mathisum5	2.5E−04	2.7E−04	2.6E−04	2.7E−04	2.6E−04	2.6E−04	2.6E−04	2.4E−04
	1	1	1	1	1	1	1	1
mathisum6	5.1E−03	4.7E−03	4.6E−03	4.7E−03	4.6E−03	5.0E−03	4.6E−03	4.5E−03
	7	6	6	6	6	7	6	6
mathisum7	7.4E−03	2.5E−03	2.5E−03	3.3E−03	2.5E−03	FAIL	2.4E−03	5.1E−02
	10	3	3	4	3		3	71
mr5mcf1	1.7E+00	1.4E+01	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
	35	282						
munson31	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
nash1	1.5E−02	1.1E−02	1.1E−02	1.1E−02	1.1E−02	1.4E−02	1.1E−02	1.1E−02
	8	6	6	6	6	8	6	6

Problem	FB	Exact	Min	Mod Ex	Mod Min	New FB	New Ex	New Min
nash2	1.9E−02 10	1.8E−02 9	2.2E−02 11	1.6E−02 8	2.2E−02 11	FAIL	1.5E−02 8	FAIL
nash3	1.3E−02 7	9.5E−03 5	1.1E−02 6	9.6E−03 5	1.1E−02 6	1.3E−02 7	9.4E−03 5	1.1E−02 6
nash4	9.4E−03 5	7.8E−03 4	1.0E−02 5	8.0E−03 4	1.0E−02 5	1.3E−02 7	7.6E−03 4	FAIL
pgvon1054	7.0E−01 41	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
pgvon1055	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
pgvon1056	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
pgvon1064	FAIL	4.9E+00 184	3.2E+00 222	5.9E+00 197	FAIL	FAIL	2.4E+00 98	FAIL
pgvon1065	FAIL	FAIL	FAIL	7.9E+00 259	FAIL	FAIL	FAIL	FAIL
pgvon1066	FAIL	5.0E+00 184	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
powell1	1.7E−02 8	1.2E−02 7	1.5E−02 9	1.2E−02 7	1.7E−02 8	1.9E−02 9	1.2E−02 8	1.1E−02 7
powell2	1.6E−02 8	1.4E−02 9	1.5E−02 12	1.4E−02 8	1.6E−02 10	1.6E−02 8	FAIL	1.4E−02 12
powell3	2.0E−02 10	8.7E−03 9	8.5E−03 9	8.7E−03 9	8.5E−03 9	2.0E−02 10	8.3E−03 9	7.0E−03 8
powell4	2.2E−02 11	7.1E−03 6	6.9E−03 6	7.1E−03 6	9.2E−03 7	2.2E−02 11	7.0E−03 6	6.7E−03 6
powell5	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	1.5E−02 20	1.2E−02 19
powell6	1.1E−02 5	4.3E−03 3	4.1E−03 3	7.7E−03 4	7.6E−03 4	1.0E−02 5	4.2E−03 3	4.1E−03 3
runge1	9.9E−03 14	5.3E−03 7	9.4E−04 1	5.3E−03 7	9.6E−04 1	9.4E−03 14	4.9E−03 7	9.0E−04 1
runge2	1.1E−02 15	6.2E−03 8	9.4E−04 1	4.6E−03 6	9.4E−04 1	1.0E−02 15	1.1E−02 16	8.8E−04 1
runge3	1.1E−02 16	1.1E−02 15	9.4E−04 1	6.2E−03 8	9.2E−04 1	1.1E−02 16	1.0E−02 15	9.0E−04 1
runge4	9.3E−03 14	1.2E−02 16	9.2E−04 1	1.1E−02 16	9.9E−03 14	8.7E−03 14	1.1E−02 16	8.8E−04 1
runge5	9.2E−03 14	1.0E−02 14	9.8E−03 14	1.0E−02 14	9.9E−03 14	8.8E−03 14	9.5E−03 14	9.2E−03 14
runge7	6.0E−03 9	6.6E−03 9	6.4E−03 9	6.6E−03 9	6.5E−03 9	5.7E−03 9	6.2E−03 9	6.0E−03 9

Problem	FB	Exact	Min	Mod Ex	Mod Min	New FB	New Ex	New Min
scarfanum1	2.2E−02 11	1.8E−02 9	1.7E−02 9	1.6E−02 8	1.8E−02 10	2.2E−02 11	3.4E−02 18	2.0E−02 11
scarfanum2	2.4E−02 12	1.4E−02 8	1.4E−02 7	1.4E−02 8	1.4E−02 7	3.3E−02 17	1.9E−02 10	1.7E−02 9
scarfanum3	2.0E−02 10	FAIL	1.9E−02 10	2.3E−02 11	2.2E−02 11	2.9E−02 15	1.4E−02 7	3.8E−02 24
scarfanum4	8.0E−03 4	4.3E−03 2	4.6E−03 2	4.3E−03 2	4.6E−03 2	7.9E−03 4	4.3E−03 2	4.5E−03 2
scarfasum1	1.6E−02 8	8.6E−03 4	1.1E−02 5	8.5E−03 4	1.1E−02 5	3.2E−02 16	2.3E−02 11	1.0E−02 5
scarfasum2	2.8E−02 13	5.7E−02 20	2.9E−02 12	2.6E−02 11	2.9E−02 12	4.0E−02 20	1.6E−02 8	1.1E−02 6
scarfasum3	2.1E−02 10	2.8E−02 18	1.4E−02 7	1.9E−02 10	1.4E−02 7	2.6E−02 13	1.4E−02 7	1.5E−02 8
scarfasum4	8.2E−03 4	4.3E−03 2	4.7E−03 2	4.4E−03 2	4.7E−03 2	8.1E−03 4	4.3E−03 2	4.7E−03 2
scarfbnum1	9.0E−02 24	FAIL	FAIL	9.7E−02 23	5.2E−01 311	7.3E−02 20	FAIL	FAIL
scarfbnum2	6.2E−02 18	1.1E+00 238	FAIL	2.0E−01 48	FAIL	FAIL	FAIL	FAIL
scarfbsum1	6.2E−02 15	FAIL	FAIL	FAIL	1.2E−01 29	FAIL	FAIL	FAIL
scarfbsum2	3.1E−01 67	2.5E−01 48	FAIL	5.3E−02 13	9.7E−01 432	FAIL	FAIL (10)	FAIL
shansim1	9.5E−03 6	8.4E−03 5	5.6E−03 4	1.0E−02 6	5.7E−03 4	1.5E−02 10	6.5E−03 4	1.0E−02 7
spillmcp1	FAIL	FAIL	FAIL	2.1E+00 191	FAIL	FAIL	FAIL	FAIL
sppe1	1.8E−02 8	2.1E−02 9	4.6E−02 50	1.9E−02 8	2.0E−02 11	1.7E−02 8	2.2E−02 10	FAIL
sppe2	1.4E−02 6	1.6E−02 7	1.9E−02 18	1.4E−02 6	1.2E−02 7	1.3E−02 6	1.4E−02 6	FAIL
sppe3	8.1E−03 4	8.1E−03 4	3.7E−03 4	6.6E−03 4	3.7E−03 4	7.7E−03 4	7.8E−03 4	3.6E−03 4
taji1	1.3E−02 8	1.2E−02 7	1.1E−02 7	1.1E−02 7	1.1E−02 7	4.7E−02 30	9.7E−03 6	FAIL
taji2	1.1E−02 7	1.1E−02 6	1.2E−02 7	1.0E−02 6	1.1E−02 7	2.6E−02 17	2.4E−02 15	FAIL
taji7	1.6E−02 10	1.5E−02 9	1.3E−02 8	1.4E−02 8	1.3E−02 8	FAIL	1.1E−02 7	FAIL
taji8	1.3E−02 8	1.1E−02 6	1.2E−02 7	1.1E−02 6	1.4E−02 8	FAIL	1.3E−02 8	FAIL

Problem	FB	Exact	Min	Mod Ex	Mod Min	New FB	New Ex	New Min
taji9	1.3E−02 8	1.4E−02 8	1.2E−02 7	1.2E−02 7	1.2E−02 7	FAIL	4.5E−02 28	FAIL
taji11	1.3E−02 8	1.0E−02 6	9.6E−03 6	8.6E−03 5	8.5E−03 5	FAIL	9.9E−03 6	FAIL
tiebout11	3.1E+00 146	FAIL	FAIL	1.4E+00 52	FAIL	FAIL	FAIL	FAIL
tiebout12	1.4E+00 66	FAIL	FAIL	1.4E+00 57	FAIL	FAIL	FAIL	FAIL
tiebout21	4.9E+01 107	FAIL	FAIL	8.6E+01 82	9.5E+01 100	FAIL	FAIL	FAIL
tiebout22	4.0E+01 113	FAIL	FAIL	FAIL	5.8E+02 321	FAIL	FAIL	FAIL
tiebout31	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
tiebout32	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL
tinloi2	5.4E−02 7	4.6E−02 5	4.2E−02 5	4.6E−02 5	4.4E−02 5	5.4E−02 7	4.6E−02 5	4.4E−02 5
tinloi41	6.0E−02 6	5.8E−02 5	4.4E−02 5	5.8E−02 5	4.4E−02 5	6.0E−02 6	5.8E−02 5	4.6E−02 5
tinloi45	7.2E−02 6	7.0E−02 5	4.6E−02 5	7.0E−02 5	4.4E−02 5	7.2E−02 6	6.8E−02 5	4.6E−02 5
tinloi55	7.6E−02 5	5.4E−02 2	2.2E−02 2	7.2E−02 5	5.0E−02 5	7.4E−02 5	5.4E−02 2	2.2E−02 2
tinloi58	8.4E−02 7	7.4E−02 5	4.6E−02 5	7.4E−02 5	4.8E−02 5	8.2E−02 7	7.2E−02 5	4.4E−02 5
tinloi63	6.6E−02 3	3.2E−02 1	1.0E−02 1	3.2E−02 1	1.2E−02 1	6.8E−02 3	3.2E−02 1	1.2E−02 1
tinsmall5	2.0E−02 6	1.7E−02 5	1.6E−02 5	1.7E−02 5	1.6E−02 5	2.0E−02 6	1.7E−02 5	1.6E−02 5
tinsmall16	2.0E−02 6	1.8E−02 5	1.6E−02 5	1.7E−02 5	1.6E−02 5	2.0E−02 6	1.7E−02 5	1.6E−02 5
tinsmall28	3.8E−02 11	1.1E−01 31	6.1E−02 18	1.1E−01 31	6.2E−02 18	3.5E−02 10	1.7E−01 48	2.2E−01 65
tinsmall31	2.1E−02 6	2.2E−02 6	2.0E−02 6	2.2E−02 6	2.0E−02 6	2.1E−02 6	1.2E−01 35	FAIL
tinsmall49	2.3E−02 7	2.4E−02 7	2.3E−02 7	2.4E−02 7	2.3E−02 7	2.6E−02 8	1.4E−01 40	9.9E−02 29
tinsmall60	3.5E−02 10	2.5E−02 7	2.3E−02 7	2.5E−02 7	2.3E−02 7	4.1E−02 12	2.4E−01 70	1.1E−01 33
tobin1	3.4E−02 11	1.6E−01 51(20)	4.3E−02 15	2.9E−02 9	3.0E−02 10	3.7E−02 12	FAIL	FAIL

Problem	FB	Exact	Min	Mod Ex	Mod Min	New FB	New Ex	New Min
tobin2	2.7E−02	6.4E−01	2.2E−01	3.5E−02	5.4E−02	4.5E−02	FAIL	FAIL
	8	189(80)	75	10	18	14	(40)	
tobin3	3.3E−02	3.9E−01	1.6E−01	3.4E−02	4.4E−02	4.4E−02	FAIL	FAIL
	10	116(80)	54	10	15	14	(20)	
tobin4	6.0E−03	5.9E−03	6.0E−03	6.0E−03	6.0E−03	5.8E−03	5.9E−03	6.0E−03
	2	2	2	2	2	2	2	2
tqbilat1	3.2E−04	3.5E−04	3.4E−04	3.4E−04	3.5E−04	3.4E−04	3.5E−04	3.9E−04
	1	1	1	1	1	1	1	1
tqbilat2	3.9E−02	3.5E−02	3.0E−02	3.5E−02	3.3E−02	3.9E−02	2.6E−02	2.4E−02
	12	17	20	12	15	12	14	17
vonthmcf1	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL

References

1. Auslender, A., Teboulle, M.: Lagrangian duality and related multiplier methods for variational inequality problems. SIAM J. Optim. **10**(4), 1097–1115 (2000)
2. Auslender, A., Teboulle, M.: Asymptotic Cones and Functions in Optimization and Variational Inequalities. Springer Monographs in Mathematics. Springer, New York (2003)
3. Bertsekas, D.: Constrained Optimization and Lagrange Multipliers. Academic Press, New York (1982)
4. Bertsekas, D.: Nonlinear Programming. Athena Scientific, Nashua (1995)
5. De Luca, T., Facchinei, F., Kanzow, C.: A theoretical and numerical comparison of some semismooth algorithms for complementarity problems. Comput. Optim. Appl. **16**, 173–205 (2000)
6. Di Pillo, G., Grippo, L.: A new class of augmented Lagrangians in nonlinear programming. SIAM J. Control Optim. **17**(5), 618–628 (1979)
7. Di Pillo, G., Grippo, L.: A continuously differentiable exact penalty function for nonlinear programming problems with inequality constraints. SIAM J. Control Optim. **23**(1), 72–84 (1985)
8. Di Pillo, G., Grippo, L.: Exact penalty functions in constrained optimization. SIAM J. Control Optim. **27**(6), 1333–1360 (1989)
9. Dirkse, S.P., Ferris, M.C.: Mcplib: A collection on nonlinear mixed complementarity problems. Optim. Methods Softw. **2**, 319–345 (1995)
10. Dirkse, S.P., Ferris, M.C.: Modeling and solution environments for MPEC: Gams & Matlab. In: Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing, Lausanne, 1997. Applied Optimization, vol. 22, pp. 127–147. Kluwer Academic, Norwell (1999)
11. Eckstein, J., Ferris, M.C.: Smooth methods of multipliers for complementarity problems. Math. Program. Ser. A **86**(1), 65–90 (1999)
12. Facchinei, F., Pang, J.-S.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer Series in Operations Research, vol. I. Springer, New York (2003)
13. Facchinei, F., Pang, J.-S.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer Series in Operations Research, vol. II. Springer, New York (2003)
14. Facchinei, F., Soares, J.: A new merit function for nonlinear complementarity problems and a related algorithm. SIAM J. Optim. **7**(1), 225–247 (1997)
15. Fischer, A.: A special newton-type optimization method. Optimization **24**, 269–284 (1992)
16. Fletcher, R.: A class of methods for nonlinear programming with termination and convergence properties. In: Integer and Nonlinear Programming, pp. 157–175. North-Holland, Amsterdam (1970)
17. Fletcher, R.: A class of methods for nonlinear programming. III. Rates of convergence. In: Numerical Methods for Nonlinear Optimization, Conf. Dundee, 1971, pp. 371–381. Academic Press, London (1972)
18. Fletcher, R.: An exact penalty function for nonlinear programming with inequalities. Math. Program. **5**, 129–150 (1973)

19. Fletcher, R., Lill, S.A.: A class of methods for nonlinear programming. II. Computational experience. In: Nonlinear Programming, Proc. Sympos., Univ. of Wisconsin, Madison, Wis., 1970, pp. 67–92. Academic Press, New York (1970)
20. Glad, T., Polak, E.: A multiplier method with automatic limitation of penalty growth. Math. Program. **17**(2), 140–155 (1979)
21. Kanzow, C.: C. Kanzow web site. http://www.mathematik.uni-wuerzburg.de/~kanzow/. Accessed in January 2007
22. Kanzow, C., Petra, S.: On a semismooth least squares formulation of complementarity problems with gap reduction. Optim. Methods Softw. **19**(5), 507–525 (2004)
23. Kanzow, C., Petra, S.: Projected filter trust region methods for a semismooth least squares formulation of mixed complementarity problems. Optim. Methods Softw. **22**(5), 713–735 (2007)
24. Mukai, H., Polak, E.: A quadratically convergent primal-dual algorithm with global convergence properties for solving optimization problems with equality constraints. Math. Program. **9**(3), 336–349 (1975)
25. Pennanen, T.: Dualization of generalized equations of maximal monotone type. SIAM J. Optim. **10**(3), 809–835 (2000)
26. Qi, L.: Convergence analysis of some algorithms for solving nonsmooth equations. Math. Oper. Res. **18**(1), 227–244 (1993)
27. Qi, L., Sun, J.: A nonsmooth version of Newton's method. Math. Program. Ser. A **58**(3), 353–367 (1993)
28. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res. **1**, 97–116 (1976)