Adv Syst Sci Appl 2024; **02**:19–31
Published online at https://ijassa.ipu.ru.

Piecewise Levenberg–Marquardt Method for Generalized Nash Equilibrium Problems

Alexey Izmailov[1*], Evgeniy Uskov[2], Yan Zhibai[1]

[1] *Lomonosov Moscow State University, Moscow, Russia*
[2] *Derzhavin Tambov State University, Tambov, Russia*

Abstract: We consider the constrained piecewise Levenberg–Marquardt method globalized by linesearch, and apply it to "min" reformulations of the optimality systems for generalized Nash equilibrium problems. Numerical comparison of the performance of this method with some relevant existing alternatives is provided.

Keywords: generalized Nash equilibrium problem, piecewise smooth equation, constrained equation, Levenberg–Marquardt method, LP-Newton method, globalization of convergence

1. INTRODUCTION

This paper aims at application of the globalized constrained piecewise Levenberg–Marquardt method developed in [24] to "min" reformulations of the optimality systems for generalized Nash equilibrium problems (GNEP), and comparison with the existing alternatives.

In Section 2, we recall the statement of a GNEP and the related first-order optimality system, and we discuss some possible equivalent reformulations of the latter as constrained systems of equations. Section 3 discusses the Levenberg–Marquardt method for piecewise smooth equations, its linesearch-based globalization, and the related global convergence and superlinear rate of convergence theory. Finally, in Section 4, we apply this method to the piecewise smooth reformulation of the first-order optimality system for GNEP, and compare this algorithm with the usual Levenberg–Marquardt method for a smooth constrained reformulation, and with the corresponding two versions of the LP-Newton method [7] globalized according to [13].

Some words about our notation and blanket arrangements are in order. Let all the norms be Euclidian, unless something different is explicitly specified. For a given $y \in \mathbb{R}^m$ and an index set $I \subset \{1, \ldots, m\}$, we denote by y_I the subvector of y with the components y_i, $i \in I$. For a sequence $\{u^k\} \subset \mathbb{R}^p$ convergent to some $u^* \in \mathbb{R}^p$, the rate of convergence is referred to as superlinear with Q-order $\nu > 1$ if there exists $c > 0$ such that $\|u^{k+1} - u^*\| \leq c\|u^k - u^*\|^\nu$ for all k large enough.

*Corresponding author: izmaf@cs.msu.ru

2. GENERALIZED NASH EQUILIBRIUM PROBLEM AND RELATED CONSTRAINED EQUATIONS

In a GNEP, N players are involved, and each player indexed by $\nu \in \{1, \ldots, N\}$ controls the variable $x^\nu \in \mathbb{R}^{n_\nu}$, and aims at minimizing a smooth objective function $f_\nu : \mathbb{R}^n \to \mathbb{R}$ subject to constraints given by a smooth mapping $g^\nu : \mathbb{R}^n \to \mathbb{R}^{m_\nu}$, where $n = \sum_{\nu=1}^N n_\nu$. In particular, both f_ν and g^ν may depend not only on x^ν, but also on the variables of rival players, denoted by $x^{-\nu} \in \mathbb{R}^{n-n_\nu}$, thus forming the entire vector of variables $x = (x^\nu, x^{-\nu}) \in \mathbb{R}^n$. With this notation, the optimization problem of the ν-th player is written as follows:

$$\text{minimize}_{\,x^\nu} \quad f_\nu(x^\nu, x^{-\nu}) \quad \text{subject to} \quad g^\nu(x^\nu, x^{-\nu}) \leq 0. \tag{2.1}$$

The specified problem setting goes back to [30]; recent surveys can be found in [10, 15], including various applications. Among other relevant references on theory and numerical methods for GNEP are [5, 8, 9, 11, 19–22, 27–29]. GNEPs form a difficult problem class, in particular, because their solutions are naturally nonisolated.

Considering $x^{-\nu}$ as a parameter, we introduce the Lagrangian $L_\nu : \mathbb{R}^{n_\nu} \times \mathbb{R}^{n-n_\nu} \times \mathbb{R}^{m_\nu} \to \mathbb{R}$ of (2.1) as

$$L_\nu(x^\nu, x^{-\nu}, \lambda^\nu) = f_\nu(x^\nu, x^{-\nu}) + \langle \lambda^\nu, g^\nu(x^\nu, x^{-\nu}) \rangle.$$

Then the Karush–Kuhn–Tucker (KKT) system characterizing stationary points and Lagrange multipliers of the ν-th player's optimization problem (2.1) has the form:

$$\frac{\partial L_\nu}{\partial x^\nu}(x^\nu, x^{-\nu}, \lambda^\nu) = 0, \quad \lambda^\nu \geq 0, \quad g^\nu(x^\nu, x^{-\nu}) \leq 0, \quad \langle \lambda^\nu, g^\nu(x^\nu, x^{-\nu}) \rangle = 0.$$

Concatenating these systems over all players yields the KKT-type system of the GNEP:

$$\mathcal{L}(x, \lambda) = 0, \quad \lambda \geq 0, \quad g(x) \leq 0, \quad \langle \lambda^\nu, g^\nu(x) \rangle = 0, \ \nu = 1, \ldots, N, \tag{2.2}$$

where we define $\lambda = (\lambda^1, \ldots, \lambda^N) \in \mathbb{R}^m$, $g : \mathbb{R}^n \to \mathbb{R}^m$, $g(x) = (g^1(x), \ldots, g^N(x))$, with $m = \sum_{\nu=1}^N m_\nu$, and

$$\mathcal{L}(x, \lambda) = \left(\frac{\partial L_1}{\partial x^1}(x^1, x^{-1}, \lambda^1), \ldots, \frac{\partial L_N}{\partial x^N}(x^N, x^{-N}, \lambda^N) \right).$$

Furthermore, the system (2.2) can be equivalently reformulated as a constrained equation

$$\Phi(u) = 0, \quad u \in P, \tag{2.3}$$

with some mapping $\Phi : \mathbb{R}^p \to \mathbb{R}^q$ and a nonempty closed convex set $P \subset \mathbb{R}^p$. This can be done in many different ways, and we restrict ourselves to the following two adopted, for example, in [13]. In both, $p = q = n + 2m$, $u = (x, \lambda, y)$, with $y = (y^1, \ldots, y^N) \in \mathbb{R}^m$ being a slack variable, and the constraint set is

$$P = \mathbb{R}^n \times \mathbb{R}^m_+ \times \mathbb{R}^m_+. \tag{2.4}$$

The difference is in how Φ is defined. One possibility is to adopt a piecewise smooth "min" reformulation with

$$\Phi(u) = (\mathcal{L}(x, \lambda), g(x) + y, \min\{\lambda, y\}), \tag{2.5}$$

where min is taken componentwise, while another reformulation is smooth, with

$$\Phi(u) = (\mathcal{L}(x, \lambda), g(x) + y, \lambda \circ y), \tag{2.6}$$

Copyright © 2024 ASSA.

employing the Hadamard product $\lambda \circ y = (\lambda_1 y_1, \ldots, \lambda_m y_m)$. Observe that the set P in (2.4) is polyhedral, and moreover, given by simple bounds, and this is the main reason to use slack variables in these reformulations. Note also that for Φ defined in (2.5), one could actually take $P = \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$, and (2.3) would still be equivalent to (2.2). However, the nonnegativity constraints on λ and y are needed to ensure the so-called P-property and condition (3.13) that will be essential for the results presented in Section 3 below.

Reformulation employing (2.5) will be of primary interest for us, and to that end, we next recall some terminology related to piecewise smoothness. A mapping Φ is refereed to as piecewise smooth if it is continuous, and there exists a finite collection of smooth selection mappings $\Phi^1, \ldots, \Phi^s : \mathbb{R}^p \to \mathbb{R}^q$ such that

$$\Phi(u) \in \{\Phi^1(u), \ldots, \Phi^s(u)\} \quad \forall\, u \in \mathbb{R}^p.$$

Considering $s = 1$ recovers the case of a smooth mapping Φ. Needless to say, a collection of smooth selection mappings corresponding to a given piecewise smooth mapping is not uniquely defined, and we will assume that some such collection is fixed, i.e., Φ is defined by a given collection of smooth selection mappings. Obviously, the mapping Φ defined according to (2.5) is piecewise smooth, with a natural collection of smooth selection mappings

$$\Phi^j(u) = \big(\mathcal{L}(x, \lambda), g(x) + y, \lambda_{I(j)}, y_{\{1, \ldots, m\} \setminus I(j)}\big), \tag{2.7}$$

where a one-to-one mapping $j \mapsto I(j)$ from $\{1, \ldots, 2^m\}$ to the set of all different subsets of $\{1, \ldots, m\}$ is supposed to be fixed.

3. PIECEWISE LEVENBERG–MARQUARDT METHOD

Getting back to a general piecewise smooth mapping Φ, for each $u \in \mathbb{R}^p$ we define the set

$$\mathcal{A}(u) = \{j \in \{1, \ldots, s\} \mid \Phi(u) = \Phi^j(u)\} \tag{3.8}$$

of indices of smooth selection mappings active at u. Let $G : \mathbb{R}^p \to \mathbb{R}^{q \times p}$ be any mapping satisfying

$$G(u) \in \{(\Phi^j)'(u) \mid j \in \mathcal{A}(u)\} \quad \forall\, u \in \mathbb{R}^p. \tag{3.9}$$

For a current iterate $u^k \in P$, the (constrained) piecewise Levenberg–Marquardt (LM) method generates the next iterate as $u^k + v^k$, where v^k is a solution of the problem

$$\text{minimize} \quad \frac{1}{2}\|\Phi(u^k) + G(u^k)v\|^2 + \frac{1}{2}\sigma(u^k)\|v\|^2 \quad \text{subject to} \quad u^k + v \in P. \tag{3.10}$$

where $\sigma : P \to \mathbb{R}_+$ defines the values of the regularization parameter. If $\sigma(u^k) > 0$, and if P is polyhedral, then (3.10) is a quadratic programming problem with a strongly convex objective function, and in particular this subproblem is uniquely solvable.

It follows from (3.8) and (3.9) that the subproblem (3.10) can be written in the form

$$\text{minimize} \quad \frac{1}{2}\|\Phi^j(u^k) + (\Phi^j)'(u^k)v\|^2 + \frac{1}{2}\sigma(u^k)\|v\|^2 \quad \text{subject to} \quad u^k + v \in P,$$

with some $j \in \mathcal{A}(u^k)$ (that may be different for different k, of course). This means that the iteration of this method coincides with the iteration of the constrained LM method applied to the smooth constrained equation

$$\Phi^j(u) = 0, \quad u \in P. \tag{3.11}$$

Copyright © 2024 ASSA.

In the smooth case, the constrained LM method was proposed in [26], and sharp results on its local superlinear convergence were obtained in [1]. The piecewise smooth case was studied in [6, 14], and an overview of these developments can be found in [17]. See also the very recent improvements of the local convergence analysis of the piecewise LM method in [25].

We proceed with an exposition of the globalized piecewise LM method and the related convergence theory developed in [24]. The algorithm below can be regarded as a generalization of the proposal in [18], where the smooth case was considered, and $\theta = 2$ was taken in the algorithm.

Algorithm 3.1:
Choose the parameters $\theta > 0$, $\varepsilon \in (0,\, 1)$ and $\varkappa \in (0,\, 1)$. Choose $u^0 \in P$, and set $k = 0$.

1. If $\Phi(u^k) = 0$, stop.
2. Set $\sigma(u^k) = \|\Phi(u^k)\|^\theta$, and compute v^k as the solution of (3.10). If $v^k = 0$, stop.
3. Set $\alpha = 1$. If for $j \in \mathcal{A}(u^k)$ such that $G(u^k) = (\Phi^j)'(u^k)$ (see (3.9)) the inequality

$$\|\Phi(u^k + \alpha v^k)\|^2 \le \|\Phi(u^k)\|^2 - \varepsilon\sigma(u^k)\alpha\|v^k\|^2 \tag{3.12}$$

   holds, set $\alpha_k = \alpha$. Otherwise replace α by $\varkappa\alpha$ and check again the inequality in (3.12), etc., until (3.12) is satisfied, and then set $\alpha_k = \alpha$.
4. Set $u^{k+1} = u^k + \alpha_k v^k$, increase k by 1 and go to Step 1.

The key ingredient of the global convergence analysis is the following assumption [24, (1.5)] that appeared before in [13, (4.8)], [16, (32)], and in [3, (0.6)]:

$$\|\Phi(u)\| \le \|\Phi^j(u)\| \quad \forall\, j \in \{1,\, \ldots,\, s\},\ \forall\, u \in P. \tag{3.13}$$

Theorem 3.1:
Let $\Phi : \mathbb{R}^p \to \mathbb{R}^q$ be a piecewise smooth mapping with continuously differentiable smooth selection mappings $\Phi^1,\, \ldots,\, \Phi^s : \mathbb{R}^p \to \mathbb{R}^q$. Let $P \subset \mathbb{R}^p$ be a nonempty closed convex set, and assume that (3.13) *holds. Let $G : \mathbb{R}^p \to \mathbb{R}^{q\times p}$ be a fixed mapping satisfying* (3.9).

Then Algorithm 3.1 uniquely defines the iterates $u^0,\, u^1,\, \ldots$, and either terminates at a point $u^k \in P$ satisfying

$$\langle((\Phi^j)'(u^k))^\top\Phi(u^k),\, u - u^k\rangle \ge 0 \quad \forall\, u \in P, \tag{3.14}$$

for at least one $j \in \mathcal{A}(u^k)$, or generates an infinite sequence $\{u^k\}$, and any accumulation point $\bar{u}$ of this sequence belongs to P and satisfies

$$\langle((\Phi^j)'(\bar{u}))^\top\Phi(\bar{u}),\, u - \bar{u}\rangle \ge 0 \quad \forall\, u \in P, \tag{3.15}$$

for at least one $j \in \mathcal{A}(\bar{u})$.

We complete this section with a result on asymptotic superlinear convergence rate, and in order to do this, we need the following two assumptions. The P-property at $\bar{u}$, introduced in [14, p. 434], consists of saying that $U_j \subset U$ near $\bar{u}$ for all $j \in \mathcal{A}(\bar{u})$, where U stands for the solution set of (2.3), while U_j stands for the solution set of (3.11). Evidently, (3.13) implies the P-property at any solution.

Another assumption we need is the constrained local Lipschitzian error bound for each active selection, that is,

$$\forall\, j \in \mathcal{A}(\bar{u}) \quad \operatorname{dist}(u,\, U_j) = O(\|\Phi^j(u)\|) \quad \text{as } u \in P \text{ tends to } \bar{u}. \tag{3.16}$$

Theorem 3.2:
Let $\Phi : \mathbb{R}^p \to \mathbb{R}^q$ be a given mapping, $P \subset \mathbb{R}^p$ a closed convex set, and $\bar{u} \in U$. Assume that Φ

Copyright © 2024 ASSA.

is piecewise smooth and the derivatives of its smooth selection mappings $\Phi^1, \ldots, \Phi^s : \mathbb{R}^p \to \mathbb{R}^q$ *are Lipschitz-continuous near* $\bar{u}$*. Let the* P*-property at* $\bar{u}$ *and condition* (3.16) *as* $u \in P$ *tends to* $\bar{u}$ *be satisfied. Let* $G : \mathbb{R}^p \to \mathbb{R}^{q \times p}$ *be a fixed mapping satisfying* (3.9)*.*

Then, if Algorithm 3.1 run with $\theta \in (0, 2]$ *generates an iterate close enough to* $\bar{u}$*, it either terminates with* $u^k \in U$*, or generated an infinite sequence* $\{u^k\}$ *convergent to some* $u^* \in U$*, and the rate of convergence is superlinear with* Q*-order* $\min\{\theta + 1, 2\}$*.*

One can easily see that the assumption (3.13), and hence, the P-property at any solution always hold for the GNEP KKT-type system reformulation using (2.4) and (2.5). As for the assumption (3.16), some natural sufficient conditions for it were derived in [14, Theorems 4, 5], and we adapt them to our problem setting next.

Proposition 3.1:
Let $f_\nu : \mathbb{R}^n \to \mathbb{R}$ *and* $g^\nu : \mathbb{R}^n \to \mathbb{R}^{m_\nu}$*,* $\nu \in \{1, \ldots, N\}$*, be twice differentiable near* $\bar{x} \in \mathbb{R}^n$*, with their second derivatives being Lipschitz-continuous near* $\bar{x}$*. Assume that* $\bar{u} = (\bar{x}, \bar{\lambda})$ *with some* $\bar{\lambda} \in \mathbb{R}^m$ *is a solution of* (2.2)*, where the notation introduces in Section 2 for GNEP is employed. Let* Φ *be defined in* (2.5)*, with its smooth selection mappings* Φ^j*,* $j \in \{1, \ldots 2^m\}$*, defined according to* (2.7)*, and let* P *be defined in* (2.4)*. Furthermore, define the index set*

$$A = \{i \in \{1, \ldots, m\} \mid g_i(\bar{x}) = 0\},$$

and its partitions

$$A_+ = \{i \in A \mid \exists\, \nu \in \{1, \ldots, N\} : \bar{\lambda}_i^\nu > 0\}, \quad A_0 = A \setminus A_+,$$

and

$$A_+^\nu = \{i \in A \mid \bar{\lambda}_i^\nu > 0\}, \quad A_0^\nu = A \setminus A_+^\nu,$$

for every $\nu \in \{1, \ldots, N\}$*.*

If for any $I \subset A_0$ *and any* $I^\nu \subset A_0^\nu$*,* $\nu \in \{1, \ldots, N\}$*, the matrices*

$$\left(\begin{array}{c|cccc} & \left(\dfrac{\partial g_{A_+^1 \cup I^1}}{\partial x^1}(x)\right)^\top & 0 & \ldots & 0 \\ \dfrac{\partial \mathcal{L}}{\partial x}(x,\, \lambda) & 0 & \left(\dfrac{\partial g_{A_+^2 \cup I^2}}{\partial x^2}(x)\right)^\top & \ldots & 0 \\ & \vdots & \vdots & \ddots & \vdots \\ & 0 & \ldots & \ldots & \left(\dfrac{\partial g_{A_+^N \cup I^N}}{\partial x^N}(x)\right)^\top \\ \hline g'_{A_+ \cup I}(x) & & 0 & & \end{array}\right)$$

have the same rank for all $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m$ *near* $(\bar{x}, \bar{\lambda})$*, then the piecewise constrained error bound condition* (3.16) *holds.*

Copyright © 2024 ASSA.

Corollary 3.1:
Under the assumptions of Proposition 3.1, if the matrix

$$\left(\begin{array}{c|cccc} & \left(\dfrac{\partial g_{A_+^1}}{\partial x^1}(\bar{x})\right)^\top & 0 & \dots & 0 \\ \dfrac{\partial \mathcal{L}}{\partial x}(\bar{x},\,\bar{\lambda}) & 0 & \left(\dfrac{\partial g_{A_+^2}}{\partial x^2}(\bar{x})\right)^\top & \dots & 0 \\ & \vdots & \vdots & \ddots & \vdots \\ & 0 & \dots & \dots & \left(\dfrac{\partial g_{A_+^N}}{\partial x^N}(\bar{x})\right)^\top \\ \hline g'_A(\bar{x}) & \multicolumn{4}{c}{0} \end{array}\right)$$

has full row rank, then the piecewise constrained error bound condition (3.16) *holds.*

The sufficient condition for the piecewise constrained error bound in Proposition 3.1 is automatically satisfied if f_ν is quadratic and g^ν is affine, for all $\nu \in \{1, \dots, N\}$. Observe also that any strict complementarity-like conditions are neither involved in Proposition 3.1, nor in its Corollary 3.1.

4. NUMERICAL RESULTS

The test set used for the numerical comparisons presented below is a collection of GNEPs from [5], later also used in [4, 13]. Information about these test problems and references to their sources and detailed descriptions are provided in [5]. For each test problem, 20 random starting points were used (the same for all algorithms involved, with random components of x^0 distributed uniformly within $(0.1,\, 20)$, which guarantees that all the functions appearing in *all* test problems are well-defined at x^0. Furthermore, λ^0 and y^0 were chosen according to [13, Section 5].

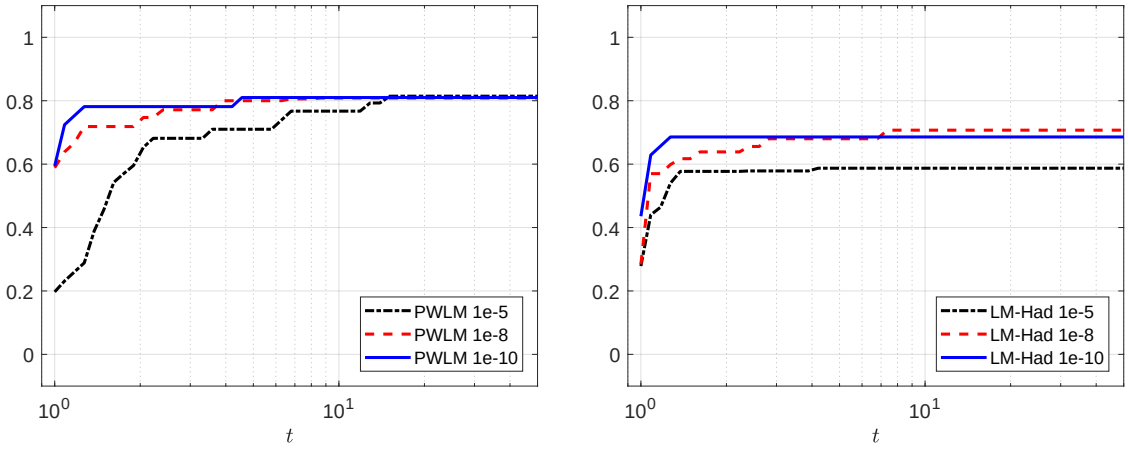
Throughout the rest of this section, P is defined according to (2.4). We compared the performance of Algorithm 3.1 applied to (2.3) with Φ defined in (2.5) (abbreviated below as PWLM) with the following alternatives:

- Globalized LP-Newton method from [13, Algorithm 1] supplied with all the performance-improving modification proposed in [13, Section 5], and applied to the same instances of (2.3) as PWLM (abbreviated as PWLPN).
- The LM method globalized according to Algorithm 3.1, but applied to (2.3) with smooth Φ defined in (2.6) (abbreviated as LM-Had).
- Globalized LP-Newton method, the same as PWLPN, but applied to (2.3) with smooth Φ defined in (2.6) (abbreviated as LPN-Had).

In case of a smooth Φ from (2.6), (3.9) implies that $G(u)$ at any $u = (x,\, \lambda,\, y) \in \mathbb{R}^p$ equals $\Phi'(u)$. For Φ from (2.5), the values of a mapping G satisfying (3.9) were computed according to the following rule: for every $i \in \{1, \dots, m\}$, if $\lambda_i > y_i$, then the $(n+m+i)$-th row of $G(u)$ is $G_{n+m+i}(u) = (0,\, e^i)$, with e^i being the i-th element of the standard basis in $\mathbb{R}^m$; otherwise $G_{n+m+i}(u) = (e^i,\, 0)$; the other rows of $G(u)$ are the gradients of the corresponding (smooth) components of Φ.

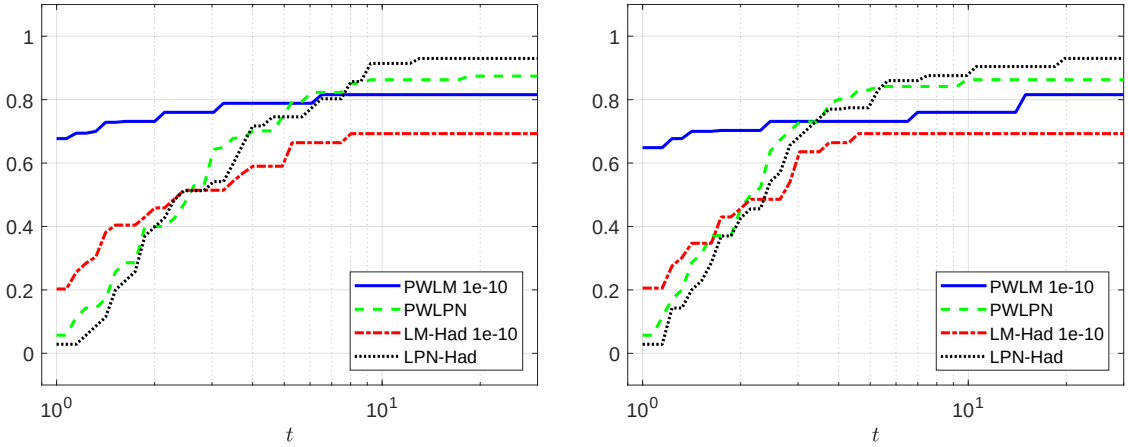
The experiments were performed in Matlab, with its built-in solver quadprog used for quadratic programming subproblems of PWLM and LM-Had, and linprog for

Copyright © 2024 ASSA.



(a) For "min" reformulation (b) For Hadamard product-based reformulation

Fig. 4.1. Comparison by average iteration counts (the same as evaluations of G) for different values of $\bar{\sigma}$.



(a) By average iteration counts (b) By average evaluations of Φ

Fig. 4.2. Comparison of different algorithms.

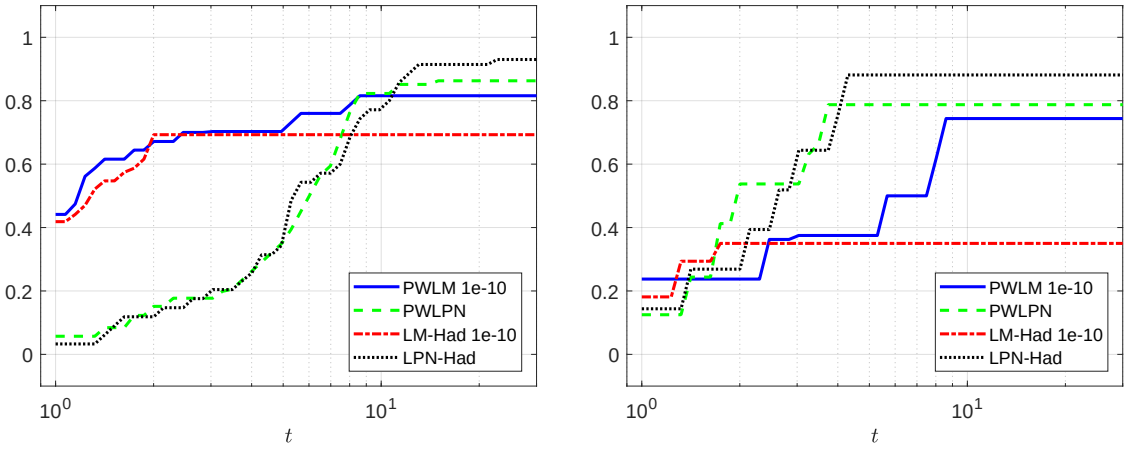
linear programming subproblems of PWLPN and LPN-Had, the former run with option 'OptimalityTolerance', 1e-15, and the latter with options 'OptimalityTolerance', 1e-8, 'ConstraintTolerance', 1e-8.

The parameters in Algorithm 3.1: were chosen as follows: $\theta = 2$, $\varepsilon = 0.001$, $\varkappa = 0.5$. The parameters of the globalized LP-Newton methods were chosen according to [13, Section 5]. In order to avoid large values of the regularization parameter far from solutions, the rule for it in Step 2 of Algorithm 3.1 was replaced by $\sigma(u^k) = \min\{\bar{\sigma}, \|\Phi(u^k)\|^\theta\}$, where several different values of $\bar{\sigma} > 0$ were tied. Runs were declared successful when terminated because of

$$\|\Phi(u^k)\|_\infty \leq 10^{-6},$$

within 1000 iterations, where for all the algorithms involved, Φ defined in (2.5) was used in this stopping test. Otherwise, failure was declared, as well as in the cases when the backtracking procedure in Step 3 of Algorithm 3.1 was producing the trial stepsize parameter

Copyright © 2024 ASSA.



(a) All test problems

(b) Large test problems

Fig. 4.3. Comparison by time.

value $\alpha \le 10^{-13}$, and also when $\|((\Phi^j)'(u^k))^\top \Phi(u^k)\| \le 10^{-12}$ was encountered for $j \in \mathcal{A}(u^k)$ such that $G(u^k) = (\Phi^j)'(u^k)$, and in particular, the stationarity condition (3.14) was approximately satisfied. For the LP-Newton-based algorithms, similar rules were adopted, but the approximate stationarity test was replaced by $|\Delta(u^k)| \le 10^{-12}$, where the quantity $\Delta(u^k)$ estimating from above the directional derivative of the merit function is defined according to [13, (2.3)], where $\gamma(u^k)$ is taken the optimal value of the LP-Newton subproblem.

It turned out that in order to achieve reasonable robustness of the LM-based algorithms, $\bar{\sigma}$ has to be taken quite small: larger values lead to long series of short steps far from solutions, and this often ends up with a failure. This happens for LM-Had when run for problems A7, Heu, Tr1b, Tr1c, and for PWLM when run for A10e, Tr1a. For A10c and A10d, runs of PWLM are generally successful for larger $\bar{\sigma}$ as well, but taking smaller values of this parameter reduces the iteration count significantly.

Our numerical results are presented in the form of performance profiles originally proposed in [2], and later adapted in [23] for the case when multiple starting points are used for every test problem. Specifically, for each algorithm a, we present the plot the function $\pi_a : [1, +\infty) \to [0, 1]$ constructed as follows. Let k_τ^a be the average of some measure of efficiency of algorithm a on problem τ, where the average is taken over successful runs, and let $s_\tau^a \in [0, 1]$ stand for the portion of successful runs of algorithm a on problem τ. Let r_τ be the best (say, smallest) value of k_τ^a among all the algorithms. Then for each $t \in [1, +\infty)$, we set

$$\pi_a(t) = \frac{1}{T} \sum_{\tau \in R_a(t)} s_\tau^a,$$

where T the overall number of problems in the test set, and $R_a(t)$ is the subset of problems for which the performance of algorithm a is no more than t times worse than that of the best algorithm:

$$R_a(t) = \{\tau \in \{1, \ldots, T\} \mid k_\tau^a \le t r_\tau\}.$$

In particular, $\pi_a(t)$ for large t is the average portion of successful runs (averaged over all test problems), while $\pi_a(1)$ is the quantity computed in a similar way but using the portions of successful runs only on those problems for which the average performance of the given algorithm over successful runs is the best among all algorithms being tested. In case of a single run for each test problem, this agrees with the original proposal in [2], and in particular,

Copyright © 2024 ASSA.

Table 4.1. Average elapsed times (seconds) and percentage of tails of last full steps out of the average iteration counts, over successful runs

Test	PWLM		PWLPN		LM-Had		LPN-Had	
	Time	Tail	Time	Tail	Time	Tail	Time	Tail
A1	0.080	100.0	0.068	100.0	0.016	76.3	0.057	100.0
A2	NaN	NaN	0.044	100.0	NaN	NaN	0.067	100.0
A3	0.003	100.0	0.043	100.0	0.006	100.0	0.066	100.0
A4	0.010	82.8	0.064	98.9	0.011	78.4	0.055	97.1
A5	0.005	100.0	0.042	100.0	0.009	100.0	0.053	100.0
A6	0.014	94.6	0.070	99.7	0.013	90.6	0.067	99.3
A7	0.022	100.0	0.092	100.0	0.018	86.5	0.077	100.0
A8	0.003	100.0	0.025	100.0	0.006	100.0	0.033	100.0
A9a	0.064	75.7	2.508	18.0	0.055	97.7	0.277	70.8
A9b	1.349	35.1	0.612	59.9	0.173	100.0	0.737	77.3
A10a	0.031	49.0	0.101	99.2	0.018	82.3	0.091	98.1
A10b	0.248	75.6	0.466	99.7	0.309	87.3	0.520	95.8
A10c	8.627	10.2	1.612	84.3	NaN	NaN	4.097	81.0
A10d	6.777	36.3	9.110	55.1	2.800	92.3	11.134	60.7
A10e	25.550	37.2	42.698	42.6	NaN	NaN	74.579	34.7
A11	0.004	100.0	0.021	100.0	NaN	NaN	0.029	100.0
A12	0.003	100.0	0.022	100.0	0.003	100.0	0.025	100.0
A13	0.004	100.0	0.027	100.0	0.004	100.0	0.026	100.0
A14	0.010	100.0	0.040	98.9	0.012	81.8	0.049	100.0
A15	0.003	100.0	0.034	100.0	0.004	100.0	0.037	100.0
A16a	0.008	100.0	0.030	100.0	0.004	100.0	0.033	97.9
A16b	0.005	100.0	0.030	100.0	0.004	100.0	0.030	100.0
A16c	0.006	100.0	0.031	100.0	0.005	100.0	0.039	84.0
A16d	0.007	100.0	0.032	100.0	0.007	100.0	0.037	100.0
A17	0.004	100.0	0.024	100.0	0.005	100.0	0.036	100.0
A18	0.007	100.0	0.050	100.0	0.011	94.5	0.053	100.0
Harker	0.003	100.0	0.023	100.0	0.006	100.0	0.036	100.0
Heu	0.012	100.0	0.053	100.0	0.011	100.0	0.057	100.0
Lob	0.068	37.6	NaN	NaN	NaN	NaN	NaN	NaN
NTF1	0.004	100.0	0.022	100.0	0.003	100.0	0.031	100.0
NTF2	0.007	90.5	0.036	99.4	0.006	94.8	0.034	100.0
Spam	1380.005	72.7	545.909	95.9	NaN	NaN	166.393	83.7
Tr1a	0.059	71.1	0.111	90.4	0.050	94.4	0.071	96.7
Tr1b	NaN	NaN	1.617	66.8	1.188	43.2	1.636	46.1
Tr1c	3.878	26.2	2.245	71.4	2.231	39.7	1.365	62.6

Copyright © 2024 ASSA.

Table 4.2. Failures

Test	PWLM	PWLPN	LM-Had	LPN-Had
A1	0	0	0	1 SS
A2	20 SS	0	20 (13 SF, 2 IL, 5 SS)	0
A3	0	12 SP	1 SF	9 (7 SP, 2 SS)
A4	4 (1 SF, 3 SS)	0	4 SF	0
A5	0	0	0	0
A6	2 SS	0	1 SF	0
A7	14 SS	8 SP	0	0
A8	0	0	5 SF	0
A9a	8 (2 SF, 6 SS)	12 IL	0	0
A9b	1 IL	0	0	0
A10a	0	0	3 SF	0
A10b	0	0	2 (1 SF, 1 SS)	0
A10c	0	0	20 (2 SF, 18 SS)	0
A10d	0	5 (4 IL, 1 SP)	19 (8 SF, 11 SS)	2 IL
A10e	2 (1 SF, 1 SS)	9 IL	20 (13 SF, 7 SS)	0
A11	0	0	20 SF	0
A12	0	0	5 SF	0
A13	5 SF	0	0	0
A14	0	0	0	0
A15	0	0	2 SF	0
A16a	1 SF	0	2 SF	0
A16b	2 SF	0	2 SF	0
A16c	1 SF	0	6 SF	0
A16d	0	0	2 SF	0
A17	0	0	2 SF	0
A18	0	0	1 SF	0
Harker	0	0	0	0
Heu	15 SS	0	6 SF	0
Lob	0	20 IL	20 SS	20 IL
NTF1	0	0	4 SF	0
NTF2	0	0	4 SF	0
Spam	0	15 SF	20 SF	17 (14 SF, 1 SP, 2 SS)
Tr1a	16 (9 SF, 7 SS)	2 SP	1 SF	0
Tr1b	20 (14 SF, 1 IL, 5 SS)	1 SP	12 (9 SF, 3 SS)	0
Tr1c	18 (5 SF, 3 IL, 10 SS)	4 (3 IL, 1 SP)	11 (5 SF, 1 IL, 5 SS)	0

$\pi_a(t)$ for large t is the portion of test problems the runs of algorithm a on which are successful, while $\pi_a(1)$ is the portion of problems on which the runs of algorithm a are successful and its efficiency is the best.

Copyright © 2024 ASSA.

Figure 4.1 presents the comparison for three different values of $\bar{\sigma}$. Figure 4.1a demonstrates the advantages of running PWLM with $\bar{\sigma} = 10^{-10}$, by iteration count, while robustness of all compared variants is nearly the same. In Figure 4.1b, the comparison by iteration count is similar, while robustness for this choice of $\bar{\sigma}$ is only slightly lower than for the value $\bar{\sigma} = 10^{-8}$. That is why $\bar{\sigma} = 10^{-10}$ was adopted in further experiments.

At this point we mention that we have also experimented with different values of the exponent θ in the regularization parameter employed by PWLM. Specifically, we tried $\theta = 1.5$ and $\theta = 1$, along with the basic choice $\theta = 2$, but much difference in performance was detected neither by iteration counts not by evaluations of Φ. The variant with $\theta = 1$ seriously outperforms the alternatives by the elapsed time, and especially the variant with $\theta = 2$. This phenomenon is explained by the difference in solution times for PWLM subproblems formed with different θ, which shows up mostly for relatively small problems. For large problems (as selected below), the difference in behavior is negligible, by all the criteria, including times, and we kept $\theta = 2$ for the rest of the experiments.

In Figure 4.2a, we compare the specified versions of PWLM and LM-Had with each other, and with PWLPN and LPN-Had, with average counts of iterations and evaluations of Φ adopted as efficiency measures. PWLM demonstrates robustness somehow lower than that of LP-Newton-based methods (thus providing an extra evidence of the use of the latter), but definitely higher than that of LM-Had. Moreover, the performance of PWLPN and LPN-Had is quite similar, while PWLM by far outperforms all the alternatives by efficiency.

Furthermore, in Figure 4.3, we provide the results of comparison by average elapsed time as a measure of efficiency. Figure 4.3a is for the entire set of test problems, while in Figure 4.3b, only selected large problems were left for comparison, namely, those with more that 100 variables and/or constraints: A9b, A10b, A10c, A10d, A10e, Spam, Tr1b, Tr1c. PWLM outperforms both PWLPN and LPN-Had by efficiency on the full test set, although it is somewhat less robust. The picture with robustness remains the same for large problems (and robustness of LM-Had appears very low in this case), but now the performance of PWLPN and LPN-Had is much improved by efficiency as well, apparently because large linear programming subproblems are solved faster than the quadratic programming subproblems. The detailed information on average times is collected in Table 4.1, where the identifiers of large problems are boldfaced. In addition, in that table, we also report on the percentage of tails of last full (with $\alpha_k = 1$) steps of the algorithms in question before successful termination, out of the average iteration counts. The issue of ultimate acceptance of the full step is crucial for superlinear convergence guaranties, like those established for Algorithm 3.1 in Theorem 3.2. In our experiments, the only cases where the last step was not full were encountered only for LM-Had, in 5 runs out of 16 successful for NTF2, and for LPN-Had, in 1 run out of 20 successful for A16c.

The largest problem in the test set is Spam, in which $N = 101$, $n = 2020$, $m = 4040$. It is successfully solved by PWLM, and in a modest number of iterations, while all the other algorithms involved typically fail on it. Many failures are caused by inability of quadprog or linprog to solve the corresponding subproblems; for example, this is the case for PWLPN, LM-Had, and LPN-Had, when run for Spam. LM-Had also systematically fails for the specified reason for problems A2, A10d, A10e, A11, Tr1c. Typical failures of other kind are caused by the stepzize parameter becoming too small.

Problem A2 is also troublesome for PWLM in a different way, with all runs ending up with failures because of a too small stepsize generated, but that was actually caused by convergence to stationary points for an active smooth selection being used, that are not solutions. Observe that such scenario is of course not ruled out by Theorem 3.1.

Information about failures and their reasons is summarized in Table 4.2 making use of the following abbreviations for failures' reasons:

- SF: solving subproblem failed.
- IL: failure by iteration limit.
- SP: stationary point that is not a solution.

Copyright © 2024 ASSA.

- SS: stepsize too small.

At the end, we briefly mention some possibilities to further improve the performance of PWLM for GNEP. Tuning $\bar{\sigma}$ for each problem sometimes gives some positive effect. Moreover, alternative rules for $\sigma(\cdot)$ might be tried, such as $\sigma(u) = \bar{\sigma}\|\Phi(u)\|^\theta/(1 + \|\Phi(u)\|^\theta)$, or $\sigma(u) = \bar{\sigma}\|(\Phi'(u))^\top\Phi(u)\|^\theta$ as proposed in [12], perhaps involving some extra parameters, or some combinations of these rules. The cases when a short step is generated can be handled in a more sophisticated way rather than just declaring failure, like trying some safeguarding steps in such cases. Finally, the use of more advanced QP-solvers for subproblems should certainly be helpful, e.g., those applying some pre-processing/scaling. All that will be the subject of future research.

ACKNOWLEDGEMENTS

The authors thank Axel Dreves and Markus Herrich for their assistance with a Matlab implementation of the GNEP test collection and of the LP-Newton method, used in Section 4 of this work. This work was funded by the Russian Science Foundation Grant 24-21-00015 (https://rscf.ru/en/project/24-21-00015/).

REFERENCES

1. Behling, R. & Fischer, A. (2012) A unified local convergence analysis of inexact constrained Levenberg–Marquardt methods, *Optim. Lett.*, **6**, 927–940.
2. Dolan, E.D. & More, J.J. (2002) Benchmarking optimization software with performance profiles, *Math. Program.*, **91**, 201–213.
3. Dorovskikh, D.I., Izmailov, A.F. & Uskov, E.I. (2024) Globalizing convergence of piecewise Newton methods, *Russian Universities Reports. Mathematics*, **29**, 149–163.
4. Dreves, A., Facchinei, F., Fischer, A. & Herrich, M. (2014) A new error bound result for Generalized Nash Equilibrium Problems and its algorithmic application, *Comput. Optim. Appl.*, **59**, 63–84.
5. Dreves, A., Facchinei, F., Kanzow, C. & Sagratella, S. (2011) On the solution of the KKT conditions of generalized Nash equilibrium problems, *SIAM J. Optim.*, **21**, 1082–1108.
6. Facchinei, F., Fischer, A. & Herrich, M. (2013) A family of Newton methods for nonsmooth constrained systems with nonisolated solutions, *Math. Methods Oper. Res.*, **77**, 433–443.
7. Facchinei, F., Fischer, A. & Herrich, M. (2014) An LP-Newton method: Nonsmooth equations, KKT systems, and nonisolated solutions, *Math. Program.*, **146**, 1–36.
8. Facchinei, F., Fischer, A. & Piccialli, V. (2007) On generalized Nash games and variational inequalities, *Oper. Res. Lett.*, **35**, 159–164.
9. Facchinei, F., Fischer, A. & Piccialli, V. (2009) Generalized Nash equilibrium problems and Newton methods, *Math. Program.*, **117**, 163–194.
10. Facchinei, F. & Kanzow, C. (2010) Generalized Nash equilibrium problems, *Ann. Oper. Res.*, **175**, 177–211.
11. Facchinei, F. & Kanzow, C. (2010) Penalty methods for the solution of generalized Nash equilibrium problems, *SIAM J. Optim.*, **20**, 2228–2253.
12. Fischer, A. (2002) Local behavior of an iterative framework for generalized equations with nonisolated solutions, *Math. Program.*, **94**, 91–124.
13. Fischer, A., Herrich, M., Izmailov, A.F. & Solodov, M.V. (2016) A globally convergent LP-Newton method, *SIAM J. Optim.*, **26**, 2012–2033.

Copyright © 2024 ASSA.

14. Fischer, A., Herrich, M., Izmailov, A.F. & Solodov, M.V. (2016) Convergence conditions for Newton-type methods applied to complementarity systems with nonisolated solutions, *Comput. Optim. Appl.*, **63**, 425–459.
15. Fischer, A., Herrich, M. & Schönefeld, K. (2014) Generalized Nash equilibrium problems – recent advances and challenges, *Pesquisa Operacional*, **34**, 521–558.
16. Fischer, A., Izmailov, A.F. & Jelitte, M. (2021) Newton-type methods near critical solutions of piecewise smooth nonlinear equations, *Comput. Optim. Appl.*, **80**, 587–615.
17. Fischer, A., Izmailov, A.F. & Solodov, M.V. (2023) The Levenberg–Marquardt method: an overview of modern convergence theories and more, *Comput. Optim. Appl.*, 2024.
18. Fischer, A. & Shukla, P.K. (2008) A Levenberg–Marquardt algorithm for unconstrained multicriteria optimization, *Oper. Res. Lett.*, **36**, 643–646.
19. Fukushima, M. & Pang, J.-S. (2005) Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games, *Comput. Manag. Science*, **2**, 21–56.
20. Harker, P.T. (1991) Generalized Nash games and quasi-variational inequalities, *Eur. J. Oper. Res.*, **54**, 81–94.
21. von Heusinger, A., Kanzow, C. & Fukushima, M. (2012) Newton's method for computing a normalized equilibrium in the generalized Nash game through fixed point formulation, *Math. Program.*, **132**, 99–123.
22. Izmailov, A.F. & Solodov, M.V. (2014) On error bounds and Newton-type methods for generalized Nash equilibrium problems, *Comput. Optim. Appl.*, **59**, 201–218.
23. Izmailov, A.F., Solodov, M.V. & Uskov, E.I. (2015) Combining stabilized SQP with the augmented Lagrangian algorithm, *Comput. Optim. Appl.*, **62**, 405–429.
24. Izmailov, A.F., Uskov, E.I. & Yan Zhibai (2024) Globalization of convergence of the constrained piecewise Levenberg–Marquardt method, Submitted.
25. Izmailov, A.F., Uskov, E.I. & Yan Zhibai (2024) The piecewise Levenberg–Marquardt method, *Adv. Syst. Sci. Appl.*, **24**, 29–39.
26. Kanzow, C., Yamashita, N. & Fukushima, M. (2004) Levenberg–Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints, *J. Comput. Appl. Math.*, **172**, 375–397.
27. Kubotam, K. & Fukushima, M. (2010) Gap function approach to the generalized Nash equilibrium problem, *J. Optim. Theory Appl.*, **144**, 511–531.
28. Kulkarni, A.A. & Shanbhag, U.V. (2012) Revisiting generalized Nash games and variational inequalities, *J. Optim. Theory Appl.*, **154**, 175–186.
29. Nabetani, K., Tseng, P. & Fukushima, M. (2011) Parametrized variational inequality approaches to generalized Nash equilibrium problems with shared constraints, *Comput. Optim. Appl.*, **48**, 423–452.
30. Rosen, J.B. (1965) Existence and uniqueness of equilibrium points for concave N-person games, *Econometrica*, **33**, 52–534.

Copyright © 2024 ASSA.