NEWTON'S METHOD FOR COMPUTING A NORMALIZED EQUILIBRIUM IN THE GENERALIZED NASH GAME THROUGH FIXED POINT FORMULATION[1]

Anna von Heusinger[2], Christian Kanzow[2], and Masao Fukushima[3]

Preprint 286 January 2009

[3] University of Würzburg
Institute of Mathematics
Am Hubland, 97074 Würzburg, Germany
e-mail: heusinger@mathematik.uni-wuerzburg.de
kanzow@mathematik.uni-wuerzburg.de

[3] Kyoto University
Department of Applied Mathematics and Physics
Graduate School of Informatics
Kyoto 606-8501, Japan
e-mail: fuku@i.kyoto-u.ac.jp

January 24, 2009

[1]This research was supported by a grant from the international doctorate program "Identification, Optimization, and Control with Applications in Modern Technologies" within the Elite-Network of Bavaria. This research was also supported in part by a Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science.

Abstract. We consider the generalized Nash equilibrium problem (GNEP), where not only the players' cost functions but also their strategy spaces depend on the rivals' decision variables. Existence results for GNEPs are typically shown by using a fixed point argument for a certain set-valued function. Here we use a regularization of this set-valued function in order to obtain a single-valued function that is easier to deal with from a numerical point of view. We show that the fixed points of the latter function constitute an important subclass of the generalized equilibria called normalized equilibria. This fixed point formulation is then used to develop a nonsmooth Newton method for computing a normalized equilibrium. The method uses a so-called computable generalized Jacobian that is much easier to compute than Clarke generalized Jacobian or B-subdifferential. We establish local superlinear/quadratic convergence of the method under the constant rank constraint qualification, which is weaker than the frequently used linear independence constraint qualification, and a suitable second-order condition. Some numerical results are presented to illustrate the performance of the method.

Key Words: Generalized Nash equilibrium problem; Normalized equilibrium; Fixed point characterization; Nonsmooth Newton method; Computable generalized Jacobian; Constant rank constraint qualification; Local superlinear/quadratic convergence

1 Introduction

We consider the *generalized Nash game*, or the *generalized Nash equilibrium problem* (GNEP), with N players $\nu = 1, \ldots, N$. We denote each player ν's decision variable as $x^\nu \in \mathbb{R}^{n_\nu}$, and collectively write $x := (x^1, \ldots, x^N) \in \mathbb{R}^n$ with $n := n_1 + \cdots + n_N$. Moreover, when we want to emphasize the special role of player ν's variable within the vector x, we use the short-hand notation $x := (x^\nu, x^{-\nu})$, where $x^{-\nu} := (x^1, \ldots, x^{\nu-1}, x^{\nu+1}, \ldots, x^N) \in \mathbb{R}^{n-n_\nu}$. In the game, each player ν controls the variable $x^\nu \in \mathbb{R}^{n_\nu}$ and tries to minimize a cost function $\theta_\nu : \mathbb{R}^n \to \mathbb{R}$ subject to the constraint $(x^\nu, x^{-\nu}) \in X$ with $x^{-\nu}$ given as exogenous, where $X \subseteq \mathbb{R}^n$ is a common strategy set[2]. Then, a vector $x^* := (x^{*,1}, \ldots, x^{*,N}) \in X$ is called a solution of the GNEP, or a *generalized Nash equilibrium* (GNE), if for each $\nu = 1, \ldots, N$, $x^{*,\nu}$ solves the following optimization problem with $x^{*,-\nu}$ being fixed:

$$\min_{x^\nu} \ \theta_\nu(x^\nu, x^{*,-\nu}) \quad \text{s.t.} \quad (x^\nu, x^{*,-\nu}) \in X. \tag{1}$$

Note that the strategy set in player ν's problem (1) depends on the other players' strategies $x^{-\nu}$. If X is defined as the Cartesian product of certain sets $X_\nu \subseteq \mathbb{R}^{n_\nu}$, i.e., $X = X_1 \times \cdots \times X_N$, then the GNEP reduces to the standard Nash equilibrium problem (NEP). In this case, the constraint in (1) is independent of the other players' strategies and simply written as $x^\nu \in X_\nu$.

The GNEP has a long history originating from Debreu [7][3] and Arrow and Debreu [2], which appeared slightly later than the famous work by Nash [24] on the NEP. Recently, the GNEP has particularly attracted increasing attention of researchers from different fields, due to the fact that it has a number of important applications in, for example, liberalized energy markets, global and regional environments, and various engineering design problems. See [10] and the references cited therein for more details.

It may be worth mentioning at this point that the GNEP usually has multiple solutions, and in fact uniqueness is rarity. In general, different solutions of an equilibrium problem (such as the NEP and the GNEP) may have different implications, unlike an optimization problem in which all optimal solutions can be regarded as having the equal value in terms of the objective

[2]A more general GNEP assumes that the constraint in each player ν's optimization problem (1) is given by $(x^\nu, x^{*,-\nu}) \in X^\nu$, where X^ν is a subset of $\mathbb{R}^n$ and may vary with ν. The GNEP considered in this paper assumes $X^1 = \cdots = X^N =: X$. This class of GNEPs was first studied by Rosen [32] and is sometimes called the GNEP with shared, or common, constraints.

[3]The term 'generalized Nash game' began to be used much later. In fact, the GNEP was called an abstract economy in [7].

function. From this perspective, it may be meaningful to single out a special GNE that has a certain additional property. Rosen [32] proposed the solution concept called a *normalized equilibrium* which is a special GNE characterized by some conditions imposed on Lagrange multipliers associated with the constraints in each player's problem. The uniqueness of a normalized equilibrium can be established under some assumptions [32].

The purpose of this paper is to develop a Newton-type method for computing a normalized equilibrium in the GNEP. Although there have been quite a few attempts to develop numerical methods designed to find a GNE for various purposes [5, 11, 12, 14, 15, 16, 22, 23], many of those methods do not enjoy a fast, i.e., quadratic or superlinear, convergence property. A few exceptions include Newton-type methods applied to a system of equations obtained from a variational inequality reformulation of the GNEP [11] as well as those based on the regularized Nikaido-Isoda function [15]. The former [11] uses a completely different approach from the one used in this paper and does not present any numerical results, while the latter [15] is closer to our approach, but has the disadvantage that certain generalized Jacobians (or generalized Hessians, to be more precise) are difficult (and sometimes impossible) to compute. Moreover, fast local convergence is established in [15] with the linear independence constraint qualification, which will be replaced by the weaker condition called the constant rank constraint qualification in this paper.

Our present approach also uses a Newton-type method but it will be applied to a system of equations obtained from a fixed point formulation of the GNEP. Needless to say, since the very beginning of game theory, fixed point theorems such as Kakutani and Brouwer have played the most fundamental role in establishing the existence of an equilibrium [3, 7, 24, 25, 32] as well as in computing an equilibrium [1]. From the computational viewpoint, however, a fixed point approach is not necessarily efficient because the problem is typically reformulated as the Kakutani fixed point problem involving a set-valued function. To overcome this difficulty, we employ a regularization technique to yield a fixed point problem involving a single-valued function. Under certain conditions, this function admits a so-called computable generalized Jacobian [33], which enables us to design a Newton-type method with a fast local convergence property. Moreover, the use of this computable generalized Jacobian allows us to avoid the common linear independence constraint qualification in order to show local superlinear/quadratic convergence.

The organization of the paper is as follows. In Section 2, we formally state the definition of a normalized equilibrium and present a fixed point characterization by means of the solution map of a parametrized optimization problem. We next introduce and analyze the computable generalized Jacobian of this

solution map in Section 3. Utilizing the computable generalized Jacobian, we present a Newton-type method for computing a normalized equilibrium and prove local superlinear convergence under suitable assumptions in Section 4. We show some numerical results in Section 5, and conclude with some final remarks in Section 6.

A few words about our notation: The gradient $\nabla f(x)$ of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is always regarded as a column vector. Furthermore, we denote by $\nabla F(x)$ the transposed Jacobian of a differentiable function $F : \mathbb{R}^n \to \mathbb{R}^m$ at a given point x, i.e., the columns of $\nabla F(x) \in \mathbb{R}^{n\times m}$ are the gradients of the component functions F_i. For a real-valued function $f(x, y)$ involving variables $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, the partial derivatives with respect to x and y are denoted by $\nabla_x f(x, y) \in \mathbb{R}^n$ and $\nabla_y f(x, y) \in \mathbb{R}^m$, respectively. Second-order partial derivatives obtained by first differentiating with respect to y and then with respect to x are written as the matrix $\nabla^2_{yx} f(x, y) \in \mathbb{R}^{m\times n}$. Similarly, we denote the first- and second-order partial derivatives of a real-valued function $g(x)$ with the variable x comprising several block components $x^1, \ldots, x^N$ by $\nabla_{x^\nu} g(x)$ and $\nabla^2_{x^\nu x^\mu} g(x)$, etc.

2 Normalized Equilibrium and Fixed Point Formulation

Throughout the paper, we make the following blanket assumption.

Assumption 2.1 *(a) The set X is a nonempty convex set and has the representation*

$$X := \{x \in \mathbb{R}^n \mid g_i(x) \le 0,\ i = 1, \ldots, m\}$$

with a twice continuously differentiable function $g : \mathbb{R}^n \to \mathbb{R}^m$ whose component functions g_i are convex.

(b) The cost functions θ_ν, $\nu = 1, \ldots, N$, are twice continuously differentiable and, as a function of x^ν alone, convex.

Convexity is a standard assumption in game theory, while the twice continuous differentiability is a reasonable requirement since we want to develop a locally superlinearly convergent method. Note that we could also add linear equality constraints in the representation of X without changing the theory essentially, but for the sake of notational simplicity, we consider only equality constraints.

Under Assumption 2.1, each player's optimization problem is a smooth convex program. Hence, under a certain constraint qualification (including

the CRCQ given in Section 3), a tuple $x = (x^1, \ldots, x^N)$ is a GNE if and only if it satisfies the following Karush-Kuhn-Tucker (KKT) conditions of problem (1) for all $\nu = 1, \ldots, N$ simultaneously:

$$\begin{aligned} &\nabla_{x^\nu}\theta_\nu(x^\nu, x^{-\nu}) + \sum_{i=1}^{m} \lambda_i^\nu \nabla_{x^\nu} g_i(x^\nu, x^{-\nu}) = 0, \\ &\lambda_i^\nu \geq 0, \quad g_i(x^\nu, x^{-\nu}) \leq 0, \quad \lambda_i^\nu \cdot g_i(x^\nu, x^{-\nu}) = 0, \quad i = 1, \ldots, m, \end{aligned} \tag{2}$$

where $\lambda^\nu = (\lambda_1^\nu, \ldots, \lambda_m^\nu)^T$, $\nu = 1, \ldots, N$, are Lagrange multipliers.

The solution concept called a normalized equilibrium, first introduced by Rosen [32], is a special GNE characterized by the Lagrange multipliers, whose definition is formally stated as follows:

Definition 2.2 *Let $\gamma \in \mathbb{R}^N$ be a vector with positive components γ_ν, $\nu = 1, \ldots, N$. A GNE x is called a* normalized equilibrium (NoE) *with weights $\gamma = (\gamma_1, \ldots, \gamma_N)^T$ if the Lagrange multipliers $\lambda^\nu \in \mathbb{R}^m$, $\nu = 1, \ldots, N$, satisfy, in addition to the KKT conditions* (2)*, the condition*

$$\gamma_1 \lambda^1 = \gamma_2 \lambda^2 = \cdots = \gamma_N \lambda^N. \tag{3}$$

In the remainder of the paper, we restrict ourselves to the special case where the weights of all players are equal, i.e.,

$$\gamma_1 = \gamma_2 = \cdots = \gamma_N. \tag{4}$$

Then, since (3) and (4) imply

$$\lambda^1 = \lambda^2 = \cdots = \lambda^N, \tag{5}$$

a NoE with equal weights is characterized as a GNE x for which there exists a single multiplier vector $\lambda = (\lambda_1, \ldots, \lambda_m)^T \in \mathbb{R}^m$ such that

$$\begin{aligned} &\nabla_{x^\nu}\theta_\nu(x^\nu, x^{-\nu}) + \sum_{i=1}^{m} \lambda_i \nabla_{x^\nu} g_i(x^\nu, x^{-\nu}) = 0, \quad \nu = 1, \ldots, N, \\ &\lambda_i \geq 0, \quad g_i(x) \leq 0, \quad \lambda_i \cdot g_i(x) = 0, \quad i = 1, \ldots, m. \end{aligned} \tag{6}$$

Hereafter, whenever we refer to a NoE, it means a NoE with equal weights.

Remark 2.3 Restriction to the case of equal weights is for simplicity of presentation. It should be kept in mind that the role of weights $\gamma = (\gamma_1, \ldots, \gamma_N)^T$ in the definition of NoE is something beyond the scaling of the players' cost functions. A NoE is not invariant with respect to the choice of γ. That is,

different values of γ usually yield different NoEs [32]. Moreover, the condition ensuring the uniqueness of a NoE depends on the value of γ [32]. More specifically, it can happen that a NoE is unique for some γ, whereas there are multiple NoEs for some $\gamma' \neq \gamma$. Thus the structure of NoEs may be affected by the weights γ. Consequently, the choice of weights is important from the practical and game theoretic viewpoint. The significance of NoE and a related solution concept of GNEP are discussed in [12]. $\Diamond$

To show the existence of a NoE, Rosen [32] considered the following optimization problem[4] in the variable $y = (y^1, \ldots, y^N) \in \mathbb{R}^n$:

$$\begin{array}{ll} \text{minimize}_y & \displaystyle\sum_{\nu=1}^{N} \theta_\nu(y^\nu, x^{-\nu}) \\ \text{subject to} & g_i(y) \leq 0, \quad i = 1, \ldots, m, \end{array} \tag{7}$$

where the vector $x = (x^1, \ldots, x^N) \in \mathbb{R}^n$ is treated as a parameter. Let the set of optimal solutions to (7) be denoted by $Y(x)$. Then it is not difficult to see that a vector $x \in \mathbb{R}^n$ is a NoE if and only if x is a fixed point of the solution map $Y(\cdot)$, i.e., $x \in Y(x)$. This fact was used to establish the existence of a NoE by using the Kakutani fixed point theorem [32]. However, a Kakutani fixed point is not very convenient to deal with numerically as it is based on a set-valued function.

As a remedy for this inconvenience, we consider the following regularized version of problem (7):

$$\begin{array}{ll} \text{minimize}_y & \displaystyle\sum_{\nu=1}^{N} \left[\theta_\nu(y^\nu, x^{-\nu}) + \frac{\alpha}{2}\|y^\nu - x^\nu\|^2\right] \\ \text{subject to} & g_i(y) \leq 0, \quad i = 1, \ldots, m. \end{array} \tag{8}$$

Here and throughout the rest of this paper, $\alpha > 0$ is a fixed regularization parameter. Under a suitable constraint qualification, a necessary and sufficient optimality condition for (the convex) problem (8) is that there exists a vector $\lambda = (\lambda_1, \ldots, \lambda_m)^T \in \mathbb{R}^m$ satisfying the KKT conditions

$$\begin{array}{l} \nabla_{x^\nu}\theta_\nu(y^\nu, x^{-\nu}) + \alpha(y^\nu - x^\nu) + \displaystyle\sum_{i=1}^{m} \lambda_i \nabla_{x^\nu} g_i(y) = 0, \quad \nu = 1, \ldots, N, \\ \lambda_i \geq 0, \quad g_i(y) \leq 0, \quad \lambda_i \cdot g_i(y) = 0, \quad i = 1, \ldots, m. \end{array} \tag{9}$$

[4]Similar parametrized problems are commonly used as a basic tool for analysis not only in the GNEP but also in the NEP (see, e.g., [25]).

Since the objective function is strongly convex in y, problem (8) has a unique optimal solution for every fixed $x \in X$, which we denote $y_\alpha(x)$. The solution map $x \mapsto y_\alpha(x)$ is therefore a single-valued function defined on X. Alternatively, this function may be derived by using the so-called regularized Nikaido-Isoda function, cf. von Heusinger and Kanzow [14]. The basic properties of the function $y_\alpha(\cdot)$ are summarized as follows:

Proposition 2.4 *Suppose that a certain constraint qualification for the set X holds. Then the following statements hold:*

(a) The function $y_\alpha(\cdot) : X \to X$ is continuous.

(b) A vector $x \in X$ is a NoE if and only if it is a fixed point of the function $y_\alpha(\cdot)$, i.e., $x = y_\alpha(x)$.

Proof. (a) This follows from a classical stability result in parametric optimization [17].
(b) Suppose $x = y_\alpha(x)$. Then it follows from (9) that x together with λ satisfies (6), showing that x is a NoE. Conversely, if $x = (x^1, \ldots, x^N)$ is a NoE, then it along with some λ satisfies (6), which in turn implies that (9) is satisfied with $(y, x) = (x, x)$. This means $y_\alpha(x) = x$, since the vector y satisfying (9) is uniquely determined for every x. □

In general, the function $y_\alpha(\cdot)$ is not differentiable. However, in order to apply Newton's method to the fixed point problem $x = y_\alpha(x)$, we need some first-order information about $y_\alpha(\cdot)$. One could use the concept of Clarke's generalized Jacobian [4] or the Bouligand (B-) subdifferential [30]. These two have been thoroughly analyzed in the past years, but in our particular situation, it is practically difficult to compute an element of these generalized differentials of $y_\alpha(\cdot)$. Here, we therefore adopt an approach that makes use of the so-called computable generalized Jacobian, which was suggested by Sun et al. [33] in the context of projection operators.

3 The Computable Generalized Jacobian

In this section we define a kind of replacement for the Jacobian of the function $y_\alpha(\cdot)$, which is related to the B-subdifferential, but easier to compute in practice. The concept is similar to the computable generalized Jacobian introduced in [33] for the projection operator onto a closed convex set. The extension here for the function $y_\alpha(\cdot)$ is actually motivated to a large extent by the ideas from [33].

By definition, $y_\alpha(x)$ is the unique solution of the parameterized optimization problem

$$\begin{array}{l} \text{minimize}_y \ \psi_\alpha(x,y) \\ \text{subject to} \quad g_i(y) \le 0, \quad i = 1,\dots,m, \end{array} \tag{10}$$

where $\psi_\alpha : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is defined by

$$\psi_\alpha(x,y) = \sum_{\nu=1}^{N} \left[\theta_\nu(y^\nu, x^{-\nu}) + \frac{\alpha}{2}\|y^\nu - x^\nu\|^2 \right].$$

Then the KKT conditions for problem (10) can be written as

$$\begin{array}{l} \nabla_y \psi_\alpha(x,y) + \sum_{i=1}^{m} \lambda_i \nabla g_i(y) = 0, \\ \lambda_i \ge 0, \quad g_i(y) \le 0, \quad \lambda_i \cdot g_i(y) = 0, \quad i = 1,\dots,m. \end{array} \tag{11}$$

Note that $\nabla_y \psi_\alpha(x,y)$ is given by

$$\nabla_y \psi_\alpha(x,y) = \begin{pmatrix} \nabla_{x^1}\theta_1(y^1, x^{-1}) + \alpha(y^1 - x^1) \\ \vdots \\ \nabla_{x^N}\theta_N(y^N, x^{-N}) + \alpha(y^N - x^N) \end{pmatrix} \in \mathbb{R}^n. \tag{12}$$

Let

$$I_0(x) := \{ i \mid g_i(y_\alpha(x)) = 0 \} \tag{13}$$

be the index set of active constraints at $y = y_\alpha(x)$.

We adopt one of the main assumptions used in [33] which also appear in the context of piecewise differentiable functions.

Assumption 3.1 *The constant rank constraint qualification (CRCQ) holds at $y_\alpha(x)$, i.e., there exists a neighbourhood $N(y_\alpha(x))$ of $y_\alpha(x)$ such that for every set $J \subseteq I_0(x)$, the set of gradient vectors*

$$\{ \nabla g_i(y) \mid i \in J \}$$

has the same rank (which depends on J) for all $y \in N(y_\alpha(x))$.

The CRCQ is weaker than the linear independence constraint qualification. Moreover, it is always fulfilled in the case of linear constraints. Furthermore, due to a result by Janin [18], it is known that the CRCQ is a suitable constraint qualification in the sense that the satisfaction of CRCQ at

the minimizer $y_\alpha(x)$ of problem (10) guarantees the existence (not necessarily uniqueness) of corresponding Lagrange multipliers λ such that the KKT conditions (11) hold. Hence the set

$$\mathcal{M}(x) := \{\, \lambda \in \mathbb{R}^m \mid (y_\alpha(x), \lambda) \text{ satisfies (11)} \,\} \tag{14}$$

is always nonempty under Assumption 3.1.

There is a family of index sets that will play a crucial role in our analysis. For each $x \in \mathbb{R}^n$, define

$$\begin{aligned} \mathcal{B}(x) \;:=\; \big\{ J \subseteq I_0(x) \mid \nabla g_i(y_\alpha(x))\,(i \in J) \text{ are linearly independent and} \\ \operatorname{supp}(\lambda) \subseteq J \text{ for some } \lambda \in \mathcal{M}(x) \big\}, \end{aligned} \tag{15}$$

where $\operatorname{supp}(\lambda)$ denotes the support of the nonnegative vector $\lambda \in \mathbb{R}^m$, i.e.,

$$\operatorname{supp}(\lambda) := \{i \in \{1, \ldots, m\} \mid \lambda_i > 0\}.$$

We first claim that the family $\mathcal{B}(x)$ is always nonempty.

Lemma 3.2 *Suppose $\mathcal{M}(x) \neq \emptyset$. Then $\mathcal{B}(x) \neq \emptyset$.*

Proof. Let us choose a multiplier $\lambda \in \mathcal{M}(x)$ with minimal support. If $\operatorname{supp}(\lambda) = \emptyset$, we take $J := \emptyset$ and immediately see that $J \in \mathcal{B}(x)$. Now suppose $\operatorname{supp}(\lambda) \neq \emptyset$. We claim that $J := \operatorname{supp}(\lambda)$ belongs to $\mathcal{B}(x)$. Obviously, we have $\operatorname{supp}(\lambda) \subseteq J \subseteq I_0(x)$. Hence it remains to show that $\nabla g_i(y_\alpha(x))$ $(i \in J)$ are linearly independent. Suppose this is not true. Then there is a nonzero vector $\beta_J = (\beta_i)_{i \in J}$ such that

$$\sum_{i \in J} \beta_i \nabla g_i(y_\alpha(x)) = 0.$$

Replacing β_J by $-\beta_J$ if necessary, we may assume without loss of generality that at least one component β_i $(i \in J)$ is positive. Let $\tilde{t} := \min\{\lambda_i/\beta_i \mid \beta_i > 0\}$. Then we have $\lambda_i - \tilde{t}\beta_i \geq 0$ for all $i \in J$ and $\lambda_{i_0} - \tilde{t}\beta_{i_0} = 0$ for at least one index $i_0 \in J$. Now define

$$\tilde{\lambda}_i := \begin{cases} \lambda_i - \tilde{t}\beta_i, & i \in J, \\ \lambda_i, & i \notin J. \end{cases}$$

Then it follows immediately that the vector $\tilde{\lambda} = (\tilde{\lambda}_1, \ldots, \tilde{\lambda}_m)^T$ belongs to $\mathcal{M}(x)$. However, by construction, the support of $\tilde{\lambda}$ is strictly contained in the support of λ, a contradiction to our choice of λ. $\square$

Recall that $\mathcal{M}(x) \neq \emptyset$ under Assumption 3.1, and hence the statement of Lemma 3.2 holds, in particular, in this situation.

For an index set $J \subseteq \{1, \ldots, m\}$ with complement $\hat{J} := \{1, \ldots, m\} \backslash J$, we now consider the function $\phi_\alpha(\cdot, \cdot, \cdot\,; J) : \mathbb{R}^{n+n+m} \to \mathbb{R}^{n+m}$ defined by

$$\phi_\alpha(x, y, \lambda; J) := \begin{pmatrix} \nabla_y \psi_\alpha(x, y) + \sum_{i \in J} \lambda_i \nabla g_i(y) \\ g_J(y) \\ \lambda_{\hat{J}} \end{pmatrix}, \tag{16}$$

where the partition $(J, \hat{J})$ of $\{1, \ldots, m\}$ is used to split the vectors λ and $g(y)$ into $\lambda = (\lambda_J, \lambda_{\hat{J}})$ and $g(y) = (g_J(y), g_{\hat{J}}(y))$, respectively.

Lemma 3.3 *Let $x \in X$ and suppose that Assumption 3.1 holds. Furthermore, let $\mathcal{M}(x)$ be defined by (14). Then, for any $J \in \mathcal{B}(x)$, there exists a unique vector $\lambda \in \mathcal{M}(x)$ such that $\phi_\alpha(x, y_\alpha(x), \lambda; J) = 0$.*

Proof. Let $J \in \mathcal{B}(x)$ and let $\lambda \in \mathcal{M}(x)$ be such that $\mathrm{supp}(\lambda) \subseteq J$. Then we have $\lambda = (\lambda_J, \lambda_{\hat{J}})$ with $\lambda_J \geq 0$ and $\lambda_{\hat{J}} = 0$. Since $(x, y_\alpha(x), \lambda)$ satisfies the KKT conditions (11), we have $\nabla_y \psi_\alpha(x, y_\alpha(x)) + \sum_{i \in J} \lambda_i \nabla g_i(y_\alpha(x)) = 0$ and $g_J(y_\alpha(x)) = 0$ (since $J \subseteq I_0(x)$). Hence $\phi_\alpha(x, y_\alpha(x), \lambda; J) = 0$ holds. Furthermore, the gradients $\nabla g_i(y_\alpha(x))$ $(i \in J)$ are linearly independent, which implies that λ is uniquely determined. □

We next show that, under certain assumptions, for any fixed x and $J \in \mathcal{B}(x)$, the system of equations $\phi_\alpha(x, y, \lambda; J) = 0$ has a locally unique solution $(y(x; J), \lambda(x; J))$.

Lemma 3.4 *Let $\bar{x} \in X$ be given, and suppose that Assumption 3.1 holds at $\bar{y} := y_\alpha(\bar{x})$. Let $J \in \mathcal{B}(\bar{x})$ be a fixed index set and $\bar{\lambda} \in \mathcal{M}(\bar{x})$ be the corresponding unique multiplier from Lemma 3.3 such that $\phi_\alpha(\bar{x}, \bar{y}, \bar{\lambda}; J) = 0$. Then the following statements hold:*

(a) There exist open neighbourhoods $N^J(\bar{x})$ of $\bar{x}$ and $N^J(\bar{y}, \bar{\lambda})$ of $(\bar{y}, \bar{\lambda})$, and a C^1-diffeomorphism $(y(\cdot\,; J), \lambda(\cdot\,; J)) : N^J(\bar{x}) \to N^J(\bar{y}, \bar{\lambda})$ such that $y(\bar{x}; J) = \bar{y}$, $\lambda(\bar{x}; J) = \bar{\lambda}$ and

$$\phi_\alpha(x, y(x; J), \lambda(x; J); J) = 0 \tag{17}$$

holds for all $x \in N^J(\bar{x})$.

(b) The transposed Jacobian of the function $y(\cdot\,; J)$ is given by the formula

$$\nabla y(x; J) = A^T C^{-1} - A^T C^{-1} D (D^T C^{-1} D)^{-1} D^T C^{-1}, \tag{18}$$

where

$$\begin{aligned} A = A(x;J) &:= -\nabla^2_{yx}\psi_\alpha(x, y(x;J)), \\ C = C(x;J) &:= \nabla^2_{yy}\psi_\alpha(x, y(x;J)) + \sum_{i\in J} \lambda_i(x;J)\nabla^2 g_i(y(x;J)), \\ D = D(x;J) &:= \nabla g_J(y(x;J)). \end{aligned}$$

Proof. (a) First note that, by Lemma 3.3, the pair $(\bar{y}, \bar{\lambda})$ is determined uniquely for any given $\bar{x}$ and $J \in \mathcal{B}(\bar{x})$. The Jacobian of $\phi_\alpha(\cdot\,; J)$ with respect to the variables (y, λ) is given by (after some reordering)

$$\nabla_{(y,\lambda)}\phi_\alpha(x, y, \lambda; J) = \begin{pmatrix} \nabla^2_{yy}\psi_\alpha(x, y) + \sum_{i\in J} \lambda_i \nabla^2 g_i(y) & \nabla g_J(y) & 0 \\ \nabla g_J(y)^T & 0 & 0 \\ 0 & 0 & I_{|\hat{J}|} \end{pmatrix}. \tag{19}$$

We claim that this matrix is nonsingular at $(x, y, \lambda) = (\bar{x}, \bar{y}, \bar{\lambda})$. Statement (a) is then an immediate consequence of the standard implicit function theorem. In fact, the nonsingularity follows from the observation that the Jacobian $\nabla g_J(\bar{y})$ has full rank by the choice of $J \in \mathcal{B}(\bar{x})$ together with the observation that the blanket Assumption 2.1 (b) implies the positive definiteness of the matrix $\nabla^2_{yy}\psi_\alpha(\bar{x}, \bar{y})$, whereas Assumption 2.1 (a) guarantees that the terms $\bar{\lambda}_i \nabla^2 g_i(\bar{y})$ are at least positive semidefinite for all $i \in J$.

(b) Differentiating equation (17) with respect to x and using some algebraic manipulations, it is not difficult to obtain the desired formula for the derivatives of the function $y(\cdot\,; J)$. The details are left to the reader. □

Our aim is to give a relation between the functions $y(\cdot\,; J)$ as defined in Lemma 3.4 and the function $y_\alpha(\cdot)$ that is the solution map of the parameterized optimization problem (10). More precisely, we will show that, under the same assumptions as in Lemma 3.4, there is a neighbourhood of the point $\bar{x}$ such that, for every x in this neighbourhood, there is an index set J (depending on the point x) such that $y_\alpha(x) = y(x; J)$ holds. This is made precise in the next lemma.

Lemma 3.5 *Let $\bar{x} \in X$ be given, and suppose that Assumption 3.1 holds at $\bar{y} := y_\alpha(\bar{x})$. Then there exists a neighbourhood $N(\bar{x})$ of $\bar{x}$ such that for all $x \in N(\bar{x})$, the following statements hold:*

(a) The CRCQ holds at $y_\alpha(x)$;

(b) $\mathcal{B}(x) \subseteq \mathcal{B}(\bar{x})$;

(c) at any given point $x \in N(\bar{x})$, the equality $y_\alpha(x) = y(x; J)$ holds for any index set $J \in \mathcal{B}(x)$, where $y(\cdot\,; J)$ is the function defined in Lemma 3.4.

Proof. (a) This follows from the definition of the CRCQ and the continuity of the function $y_\alpha(\cdot)$, cf. Proposition 2.4 (a).

(b) The proof is essentially the same as the one in [29] for the projection operator. Assume there exists no neighbourhood $N(\bar{x})$ of $\bar{x}$ such that $\mathcal{B}(x) \subseteq \mathcal{B}(\bar{x})$ for all $x \in N(\bar{x})$. Then there is a sequence $\{x^k\}$ converging to $\bar{x}$ such that for each k, there is an index set $J_k \in \mathcal{B}(x^k) \setminus \mathcal{B}(\bar{x})$. Since there are only finitely many such index sets, by working with a subsequence if necessary, we may assume that these index sets J_k are the same for all k. Let this common index set be J.

According to the definition of $\mathcal{B}(x^k)$, the vectors $\nabla g_i(y_\alpha(x^k))$ $(i \in J)$ are linearly independent and there exists $\lambda^k \in \mathcal{M}(x^k)$ such that $\mathrm{supp}(\lambda^k) \subseteq J \subseteq I_0(x^k)$, but $J \notin \mathcal{B}(\bar{x})$. Due to the continuity of the functions g_i and y_α, it holds that $I_0(x^k) \subseteq I_0(\bar{x})$, hence we have $J \subseteq I_0(\bar{x})$ for all k sufficiently large. Furthermore, the assumed CRCQ condition guarantees that the vectors $\nabla g_i(y_\alpha(\bar{x}))$ $(i \in J)$ are also linearly independent. Hence we have $J \notin \mathcal{B}(\bar{x})$ only if there is no $\lambda \in \mathcal{M}(\bar{x})$ such that $\mathrm{supp}(\lambda) \subseteq J$. However, the KKT conditions imply that

$$\nabla_y \psi_\alpha(x^k, y_\alpha(x^k)) + \sum_{i \in J} \lambda_i^k \nabla g_i(y_\alpha(x^k)) = 0 \quad \text{for all } k. \tag{20}$$

Since the functions y_α and ∇g_i are continuous, we have $\nabla g_i(y_\alpha(x^k)) \to \nabla g_i(y_\alpha(\bar{x}))$. Taking into account the linear independence of $\{\nabla g_i(y_\alpha(\bar{x}))\}_{i \in J}$, we see that the sequence $\{\lambda^k\}$ is convergent, say $\lambda_i^k \to \check{\lambda}_i$ for all $i \in J$. Taking the limit in (20) and setting $\check{\lambda}_i = 0$ for $i \in \hat{J}$, we can easily verify that the vector $\check{\lambda} := (\check{\lambda}_J, \check{\lambda}_{\hat{J}})$ belongs to $\mathcal{M}(\bar{x})$. Moreover, the definition of $\check{\lambda}$ guarantees that $\mathrm{supp}(\check{\lambda}) \subseteq J$, and hence $J \in \mathcal{B}(\bar{x})$. This contradicts our assumption.

(c) From (a) and (b) it follows that there is a neighbourhood $N(\bar{x})$ such that for any $x \in N(\bar{x})$, the CRCQ holds at $y_\alpha(x)$ and $\mathcal{B}(x) \subseteq \mathcal{B}(\bar{x})$. Furthermore, for each $J \in \mathcal{B}(\bar{x})$, let $N^J(\bar{x})$ and $N^J(\bar{y}, \bar{\lambda})$ be the neighbourhoods defined in Lemma 3.4, where $\bar{\lambda}$ is the vector also defined there. We then define the neighbourhood

$$V(\bar{x}) := \bigcap_{J \in \mathcal{B}(\bar{x})} N^J(\bar{x}) \cap N(\bar{x}),$$

which is open since there are only finitely many J's.

For any given vector $x \in V(\bar{x})$, the optimization problem (10) has a unique solution $y_\alpha(x)$. Moreover, Lemma 3.3 implies that for every fixed

$J \in \mathcal{B}(x)$, there exists a unique Lagrange multiplier $\lambda = \lambda^J(x) \in \mathcal{M}(x)$ such that $(x, y_\alpha(x), \lambda^J(x))$ satisfies

$$\phi_\alpha(x, y_\alpha(x), \lambda^J(x); J) = 0.$$

In particular, for the Lagrange multiplier associated with $\big(\bar{x}, \bar{y}\big)$, we write $\lambda^J(\bar{x})$ as $\bar{\lambda}^J$.

On the other hand, Lemma 3.4 implies that there is a continuously differentiable function $\big(y(\cdot\,; J), \lambda(\cdot\,; J)\big) : N^J(\bar{x}) \to N^J(\bar{y}, \bar{\lambda}^J)$ such that, for every $x \in N^J(\bar{x})$, the pair $\big(y(x; J), \lambda(x; J)\big)$ is the unique solution of

$$\phi_\alpha(x, y, \lambda; J) = 0 \tag{21}$$

in the set $N^J(\bar{y}, \bar{\lambda}^J)$. Hence, if we can show that there exists an open neighbourhood $U(\bar{x}) \subseteq V(\bar{x})$ such that for every $x \in U(\bar{x})$ and every $J \in \mathcal{B}(x)$ we have

$$\big(y_\alpha(x), \lambda^J(x)\big) \in N^J(\bar{y}, \bar{\lambda}^J),$$

then the uniqueness implies that

$$\big(y_\alpha(x), \lambda^J(x)\big) = \big(y(x; J), \lambda(x; J)\big)$$

for all $x \in U(\bar{x})$ and $J \in \mathcal{B}(x)$, and this would conclude the proof.

Suppose there exists no such open neighbourhood $U(\bar{x}) \subseteq V(\bar{x})$. Then there exists a sequence $\{x^k\}$ with $x^k \to \bar{x}$ and $J_k \in \mathcal{B}(x^k)$ such that

$$(y_\alpha(x^k), \lambda^{J_k}(x^k)) \notin N^{J_k}(\bar{y}, \bar{\lambda}^{J_k}) \quad \text{for all } k. \tag{22}$$

By working with a subsequence, we may assume that J_k is the same index set for all k. Denote this index set by J. Furthermore, choose open neighbourhoods $N^J(\bar{y})$ of $\bar{y}$ and $N^J(\bar{\lambda}^J)$ of $\bar{\lambda}^J$ such that $N^J(\bar{y}) \times N^J(\bar{\lambda}^J) \subseteq N^J(\bar{y}, \bar{\lambda}^J)$.

Since the function y_α is continuous, we have $y_\alpha(x^k) \to y_\alpha(\bar{x}) = \bar{y}$. Hence $y_\alpha(x^k) \in N^J(\bar{y})$ for all k sufficiently large. On the other hand, for every x^k with associated $y_\alpha(x^k)$ and $\lambda^J(x^k)$, we have from (16)

$$\begin{gathered} \nabla_y \psi_\alpha(x^k, y_\alpha(x^k)) + \sum_{i \in J} \lambda_i^J(x^k) \nabla g_i(y_\alpha(x^k)) = 0, \\ \lambda_i^J(x^k) = 0, \quad i \in \hat{J} \end{gathered} \tag{23}$$

for all k. The continuity of the functions $\nabla_y \psi_\alpha$, y_α and ∇g_i, together with the linear independence of the vectors $\nabla g_i(\bar{y})$ $(i \in J)$, which is again a consequence of the CRCQ, implies that the sequence $\{\lambda^J(x^k)\}$ is convergent.

Let $\tilde{\lambda}^J$ be the corresponding limit point. Taking the limit in (23) therefore gives

$$\nabla_y \psi_\alpha(\bar{x}, \bar{y}) + \sum_{i \in J} \tilde{\lambda}_i^J \nabla g_i(\bar{y}) = 0$$

as well as $\tilde{\lambda}_i^J = 0$ for all $i \in \hat{J}$. Then the CRCQ implies that $\tilde{\lambda}^J$ is the only vector satisfying these equations. However, by definition, $\bar{\lambda}^J$ also satisfies these equations, so it follows that $\tilde{\lambda}^J = \bar{\lambda}^J$.

Hence $\lambda^J(x^k)$ converges to $\bar{\lambda}^J$, meaning that $\lambda^J(x^k) \in N^J(\bar{\lambda}^J)$ for all k sufficiently large. Therefore we have

$$\big(y_\alpha(x^k), \lambda^J(x^k)\big) \in N^J(\bar{y}) \times N^J(\bar{\lambda}^J) \subseteq N^J(\bar{y}, \bar{\lambda}^J)$$

for all k sufficiently large, a contradiction to (22). □

Since there are only finitely many possible index sets $J \subseteq \{1, \ldots, m\}$, it follows from Lemma 3.5 that, given any point x in a sufficiently small neighbourhood of $\bar{x}$, the function $y_\alpha(\cdot)$ is equal to one of the finitely many functions $y(\cdot\,; J)$ and, therefore, piecewise smooth. However it is not necessarily easy to compute an element of the B-subdifferential of y_α at x, which is defined by

$$\partial_B y_\alpha(x) := \{G \in \mathbb{R}^{n \times n} \mid G = \lim_{x^k \to x} \nabla y_\alpha(x^k)^T,\ \{x^k\} \subseteq \Omega\},$$

where $\Omega := \{x \in \mathbb{R}^n \mid y_\alpha(\cdot) \text{ is differentiable at } x\}$. Lemma 3.5 then suggests to use, in place of the B-subdifferential, the following modification of a generalized Jacobian, which we call the *computable generalized Jacobian* of $y_\alpha(\cdot)$ at x:

$$\partial_C y_\alpha(x) := \big\{\nabla y(x; J)^T \ \mid\ J \in \mathcal{B}(x)\big\}, \tag{24}$$

where $\mathcal{B}(x)$ is defined by (15). Note that computing an element of $\partial_C y_\alpha(x)$ amounts to finding an index set $J \in \mathcal{B}(x)$, which is implementable in practice. While the inclusion $\partial_B y_\alpha(x) \subseteq \partial_C y_\alpha(x)$ holds at any x, the converse is not true in general, see [9, Example 11]. Under additional assumptions, however, it can be shown that these two sets coincide; see, in particular, [28, Corollary 3.2.2] or [8, Theorem 3.3].

To conclude this section, we consider the special case of a GNEP with quadratic cost functions and linear constraints. Here the function $y_\alpha(\cdot)$ turns out to be piecewise linear (or piecewise affine, to be more precise).

Proposition 3.6 *Consider the case where the cost functions θ_ν are quadratic, i.e.,*

$$\theta_\nu(x) = \frac{1}{2}(x^\nu)^T A_{\nu\nu} x^\nu + \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^{N} (x^\nu)^T A_{\nu\mu} x^\mu$$

for $\nu = 1, \ldots, N$. Suppose that the feasible set X is given by linear inequalities, i.e., $X := \{x \in \mathbb{R}^n \mid Bx \leq b\}$ for some matrix $B \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^m$. Let $\bar{x} \in X$ be arbitrarily given. Then there exists a neighbourhood $U(\bar{x})$ of $\bar{x}$ such that for every $x \in U(\bar{x})$ and every $J \in \mathcal{B}(x)$, there exist a matrix $V^J \in \mathbb{R}^{n \times n}$ and a vector $w^J \in \mathbb{R}^n$ such that $y_\alpha(x) = y(x; J) = V^J x + w^J$.

Proof. Since X is polyhedral, the CRCQ holds at every point $x \in X$. By Lemma 3.5, there exists a neighbourhood $N(\bar{x})$ of $\bar{x}$ such that for all $x \in N(\bar{x})$, we have $\mathcal{B}(x) \subseteq \mathcal{B}(\bar{x})$ and $y_\alpha(x) = y(x; J)$ for all $J \in \mathcal{B}(x)$, where $y(\cdot\,; J)$ is the function defined in Lemma 3.4.

Now consider an arbitrary index set $J \in \mathcal{B}(\bar{x})$, and let $y(\cdot\,; J)$ be the corresponding function. Furthermore, let $\bar{A}$ denote the $n \times n$ matrix $\bar{A} = (A_{\nu\mu})_{\nu,\mu=1}^N$ and $\operatorname{diag}(A_{\nu\nu})$ denote the block-diagonal matrix with block component matrices $A_{\nu\nu}$, $\nu = 1, \ldots, N$. From Lemma 3.4, $y(\cdot\,; J)$ is a continuously differentiable function on $N(\bar{x})$ with Jacobian

$$V^J := \nabla y(x; J)^T = C^{-1}A - C^{-1}D(D^T C^{-1} D)^{-1} D^T C^{-1} A,$$

where $A := -\bar{A} + \operatorname{diag}(A_{\nu\nu}) + \alpha I$, $C := \operatorname{diag}(A_{\nu\nu}) + \alpha I$ and $D := B_J^T$. The assumptions on the cost functions θ_ν and the set X imply that the matrix V^J is constant. Consequently, $y(\cdot\,; J)$ is an affine function, i.e., there is a vector w^J such that $y(x; J) = V^J x + w^J$. □

Note that it follows from the above proof that we have

$$y_\alpha(x) \in \{V^J x + w^J \mid J \in \mathcal{B}(\bar{x})\}$$

for all x in a sufficiently small neighbourhood of $\bar{x}$, i.e., $y_\alpha(\cdot)$ is a piecewise affine function.

4 Newton's Method for Computing Normalized Equilibria

For the computation of a NoE, we use the nonsmooth Newton method from [20] and apply it to the system of equations

$$F(x) := y_\alpha(x) - x = 0.$$

From the current iterate x^k, the next iterate x^{k+1} is computed by

$$x^{k+1} = x^k - H_k^{-1} F(x^k), \tag{25}$$

where H_k is an element of the nonempty computable generalized Jacobian

$$\partial_C F(x^k) = \partial_C y_\alpha(x^k) - I = \{\nabla y(x^k; J)^T - I \mid J \in \mathcal{B}(x^k)\}. \tag{26}$$

In this section, we give sufficient conditions for the matrices H_k to be nonsingular and show local superlinear/quadratic convergence of this nonsmooth Newton method.

For convenience, we write

$$M(x,y) := \begin{pmatrix} \nabla^2_{x^1x^1}\theta_1(y^1, x^{-1}) & \cdots & \nabla^2_{x^1x^N}\theta_1(y^1, x^{-1}) \\ \vdots & \ddots & \vdots \\ \nabla^2_{x^Nx^1}\theta_N(y^N, x^{-N}) & \cdots & \nabla^2_{x^Nx^N}\theta_N(y^N, x^{-N}) \end{pmatrix}.$$

This notation also facilitates the comparison with Newton methods from [11] which are based on a variational inequality formulation of the GNEP.

The following assumption will be needed to establish fast local convergence of the nonsmooth Newton method (25).

Assumption 4.1 *For each $J \in \mathcal{B}(x)$ and $\lambda \in \mathcal{M}(x)$, we have*

$$d^T\big(M(x, y_\alpha(x)) + \sum_{i\in J} \lambda_i \nabla^2 g_i(y_\alpha(x))\big)\, d \neq 0 \quad \forall d \in \mathcal{T}^J(x),\ d \neq 0, \tag{27}$$

where $\mathcal{T}^J(x)$ is defined by

$$\mathcal{T}^J(x) := \{d \in \mathbb{R}^n \mid \nabla g_i(y_\alpha(x))^T d = 0\ \forall i \in J\}. \tag{28}$$

The condition (27) is a kind of second-order sufficiency condition. We will revisit this condition after showing the following nonsingularity result.

Lemma 4.2 *Let $\bar{x} \in X$ and $\bar{y} := y_\alpha(\bar{x})$. Suppose that Assumptions 3.1 and 4.1 hold at $\bar{x}$. Then the matrix $\nabla y(\bar{x}; J)^T - I$ is nonsingular for any index set $J \in \mathcal{B}(\bar{x})$.*

Proof. Assume that there exists an index set $J \in \mathcal{B}(\bar{x})$ such that the matrix $\nabla y(\bar{x}; J)^T - I$ is singular. Let $\bar{\lambda} \in \mathcal{M}(\bar{x})$ be the corresponding Lagrange multiplier, which is unique by Lemma 3.3 under Assumption 3.1, such that $\phi_\alpha(\bar{x}, \bar{y}, \bar{\lambda}; J) = 0$ holds. Furthermore, let $y(\cdot\,; J)$ and $\lambda(\cdot\,; J)$ be the functions defined in Lemma 3.4; in particular, recall that we have $y(\bar{x}; J) = \bar{y}$ and $\lambda(\bar{x}; J) = \bar{\lambda}$.

Since $\nabla y(\bar{x}; J)^T - I$ is singular, there exists a nonzero vector $v \in \mathbb{R}^n$ such that $\big(\nabla y(\bar{x}; J)^T - I\big)v = 0$, which is equivalent to saying that $\nabla y(\bar{x}; J)^T$ has

an eigenvalue equal to one with eigenvector v. From Lemma 3.4, along with the fact that $y(\bar{x}; J) = \bar{y}$ and $\lambda(\bar{x}; J) = \bar{\lambda}$, we have the formula

$$\nabla y(\bar{x}; J)^T = C^{-1}A - C^{-1}D(D^T C^{-1} D)^{-1} D^T C^{-1} A, \tag{29}$$

with

$$\begin{aligned} A &= A(\bar{x}; J) := -\nabla^2_{yx}\psi_\alpha(\bar{x}, \bar{y}), \\ C &= C(\bar{x}; J) := \nabla^2_{yy}\psi_\alpha(\bar{x}, \bar{y}) + \sum_{i \in J} \bar{\lambda}_i \nabla^2 g_i(\bar{y}), \\ D &= D(\bar{x}; J) := \nabla g_J(\bar{y}). \end{aligned}$$

This expression of $\nabla y(\bar{x}; J)^T$ reveals immediately that $D^T \nabla y(\bar{x}; J)^T = 0_{m \times n}$, which implies that

$$0 = D^T \nabla y(\bar{x}; J)^T v = D^T v = \nabla g_J(\bar{y})^T v$$

holds, and thus,

$$v \in \mathcal{T}^J(\bar{x}), \tag{30}$$

where $\mathcal{T}^J(\bar{x})$ is given by (28) with $x = \bar{x}$. Therefore, multiplication of equation (29) from the left side with $v^T C$ and from the right side with v gives, using the fact that v is an eigenvector of the matrix $\nabla y(\bar{x}; J)^T$ with eigenvalue 1 once again,

$$v^T C v = v^T A v. \tag{31}$$

Note that the matrices C and A are expressed as

$$\begin{aligned} C &= \nabla^2_{yy}\psi_\alpha(\bar{x}, \bar{y}) + \sum_{i \in J} \bar{\lambda}_i \nabla^2 g_i(\bar{y}) \\ &= \begin{pmatrix} \nabla^2_{x^1 x^1}\theta_1(\bar{y}^1, \bar{x}^{-1}) & & \\ & \ddots & \\ & & \nabla^2_{x^N x^N}\theta_N(\bar{y}^N, \bar{x}^{-N}) \end{pmatrix} \\ &\quad + \alpha I + \sum_{i \in J} \bar{\lambda}_i \nabla^2 g_i(\bar{y}) \end{aligned}$$

and

$$\begin{aligned} A &= -\nabla^2_{yx}\psi_\alpha(\bar{x}, \bar{y}) \\ &= -\begin{pmatrix} \nabla^2_{x^1 x^1}\theta_1(\bar{y}^1, \bar{x}^{-1}) & \cdots & \nabla^2_{x^1 x^N}\theta_1(\bar{y}^1, \bar{x}^{-1}) \\ \vdots & \ddots & \vdots \\ \nabla^2_{x^N x^1}\theta_N(\bar{y}^N, \bar{x}^{-N}) & \cdots & \nabla^2_{x^N x^N}\theta_N(\bar{y}^N, \bar{x}^{-N}) \end{pmatrix} \end{aligned}$$

$$+\begin{pmatrix} \nabla^2_{x^1x^1}\theta_1(\bar{y}^1,\bar{x}^{-1}) & & \\ & \ddots & \\ & & \nabla^2_{x^Nx^N}\theta_N(\bar{y}^N,\bar{x}^{-N}) \end{pmatrix}+\alpha I.$$

Hence we have

$$C - A = M(\bar{x},\bar{y}) + \sum_{i\in J}\bar{\lambda}_i\nabla^2 g_i(\bar{y}). \tag{32}$$

On the other hand, by (27) in Assumption 4.1 and (30), we have

$$v^T\big(M(\bar{x},\bar{y}) + \sum_{i\in J}\bar{\lambda}_i\nabla^2 g_i(\bar{y})\big)\,v \neq 0.$$

This together with (32) contradicts (31). □

Recall [11] that $\bar{x}$ is a NoE if and only if it satisfies the variational inequality

$$\Theta(x^*)^T(x - x^*) \geq 0 \quad \forall x \in X,$$

where $\Theta : \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$\Theta(x) := \begin{pmatrix} \nabla_{x^1}\theta_1(x) \\ \vdots \\ \nabla_{x^N}\theta_N(x) \end{pmatrix}. \tag{33}$$

Since we have $x^* = y^* := y_\alpha(x^*)$ at a NoE by Proposition 2.4, the Jacobian of function Θ evaluated at x^* is precisely the matrix $M(x^*, y^*)$, and the second-order condition (27) for the fixed multiplier $\lambda \in \mathcal{M}(x^*)$ corresponds to the standard second-order sufficiency condition for optimization problems. (Note, however, that we only require $\neq 0$ in (27) since we are not in an optimization setting.)

In the case of quadratic cost functions, there is a simple sufficient condition for Assumption 4.1 to hold.

Corollary 4.3 *Suppose that the cost functions θ_ν are given by*

$$\theta_\nu(x) = \frac{1}{2}(x^\nu)^T A_{\nu\nu}x^\nu + \sum_{\substack{\mu=1\\ \mu\neq\nu}}^{N}(x^\nu)^T A_{\nu\mu}x^\mu$$

for $\nu = 1,\ldots,N$. Then Assumption 4.1 holds provided that the matrix $A := (A_{\nu\mu})_{\nu,\mu=1}^N$ is positive definite.

Next we prove that the matrices H_k provide a superlinear approximation for the function F.

Lemma 4.4 *Let x^* be a NoE. Suppose that Assumption 3.1 holds at x^*. Then we have for any $H \in \partial_C F(x)$*

$$F(x) - F(x^*) - H(x - x^*) = o(\|x - x^*\|). \tag{34}$$

Furthermore if the second derivatives of all θ_ν and all g_i are Lipschitz continuous around x^*, then

$$F(x) - F(x^*) - H(x - x^*) = O(\|x - x^*\|^2). \tag{35}$$

Proof. By Lemma 3.4, for each $J \in \mathcal{B}(x^*)$, there is a neighbourhood $N^J(x^*)$ of x^* and a continuously differentiable function $y(\cdot\,; J)$ defined on $N^J(x^*)$ such that $y(x^*; J) = y_\alpha(x^*) = x^*$. Let $\varepsilon > 0$ be arbitrarily given. Then the continuous differentiability of $y(\cdot\,; J)$ on $N^J(x^*)$ ensures the existence of a $\delta(\varepsilon, J) > 0$ such that

$$\frac{\|y(x; J) - y(x^*; J) - \nabla y(x; J)^T (x - x^*)\|}{\|x - x^*\|} < \varepsilon \tag{36}$$

holds whenever $\|x - x^*\| < \delta(\varepsilon, J)$. Let $\bar{\delta}(\varepsilon) := \min_{J \in \mathcal{B}(x^*)} \delta(\varepsilon, J) > 0$. Then (36) holds for any x such that $\|x - x^*\| < \bar{\delta}(\varepsilon)$ and any $J \in \mathcal{B}(x^*)$.

Now consider an arbitrary sequence $\{x^k\}$ converging to x^* and pick any $H_k \in \partial_C F(x^k)$. By the definition (26) of $\partial_C F(x)$, H_k can be written as $H_k = \nabla y(x^k, J_k)^T - I$ for some $J_k \in \mathcal{B}(x^k) \subseteq \mathcal{B}(x^*)$. Hence, from the preceding argument, we have

$$\begin{aligned}
&\frac{\|F(x^k) - F(x^*) - H_k(x^k - x^*)\|}{\|x^k - x^*\|} \\
&= \frac{\|y_\alpha(x^k) - y_\alpha(x^*) - \nabla y(x^k; J_k)^T (x^k - x^*)\|}{\|x^k - x^*\|} \\
&= \frac{\|y(x^k; J_k) - y(x^*; J_k) - \nabla y(x^k; J_k)^T (x^k - x^*)\|}{\|x^k - x^*\|} < \varepsilon
\end{aligned}$$

for all k such that $\|x^k - x^*\| < \bar{\delta}(\varepsilon)$. Since $\{x^k\}$ and ε are arbitrary, we may conclude that (34) holds.

If all functions θ_ν and g_i have Lipschitz continuous second derivatives, then for all $J \in \mathcal{B}(x^*)$ the function $\nabla y(\cdot\,; J)$ is locally Lipschitz continuous. This follows from formula (18) and the fact that the sum and the product of locally Lipschitz continuous functions again lead to a locally Lipschitz continuous function. Then it is not difficult to derive (35) in a similar manner

to the above. □

Summarizing the above arguments, we are now in a position to state the main local convergence result which shows that our method is locally superlinearly/quadratically convergent. Note that this result holds under the CRCQ condition which is weaker than the linear independence constraint qualification.

Theorem 4.5 *Let x^* be a NoE and suppose that Assumptions 3.1 and 4.1 hold at x^*. Then there is a neighbourhood $N(x^*)$ of x^* such that for an arbitrary initial point $x^0 \in N(x^*)$, the sequence generated by the nonsmooth Newton method (25) converges to x^* superlinearly. Furthermore, if all the functions θ_ν and g_i have Lipschitz continuous second derivatives, then the convergence rate is quadratic.*

Proof. By Lemma 4.2, each $H \in \partial_C F(x^*) = \{\nabla y(x^*; J)^T - I \mid J \in \mathcal{B}(x^*)\}$ is nonsingular. Since the functions $\nabla y(\cdot\,; J)$ $(J \in \mathcal{B}(x^*))$ are continuous, there exists a neighbourhood $N(x^*)$ of x^* such that $\mathcal{B}(x^k) \subseteq \mathcal{B}(x^*)$ and hence the matrices $H_k \in \partial_C F(x^k) = \{\nabla y(x^k; J)^T - I \mid J \in \mathcal{B}(x^k)\}$ are nonsingular for all $x^k \in N(x^*)$. The rest of the proof consists of standard arguments based on Lemma 4.4 and the definition of the nonsmooth Newton method (25). □

Our final result shows that the nonsmooth Newton method enjoys a local one-step convergence property if the GNEP is described by quadratic cost functions and linear constraints.

Proposition 4.6 *Suppose that the cost functions and the constraints are given as in Proposition 3.6 and that the matrix $A := (A_{\mu\nu})_{\mu,\nu=1}^N$ is positive definite. Let x^* be a NoE. Then there is a neighbourhood $N(x^*)$ of x^* such that, once x^k enters $N(x^*)$, the next iterate x^{k+1} coincides with x^*.*

Proof. By Lemma 3.5, there exists a neighbourhood $N(x^*)$ of x^* such that for every $x \in N(x^*)$ and every $J \in \mathcal{B}(x)$, we have $y_\alpha(x) = y(x; J)$ and $\mathcal{B}(x) \subseteq \mathcal{B}(x^*)$. Moreover, from Proposition 3.6, we have $y(x; J) = V^J x + w^J$ for all $x \in N(x^*)$ with V^J and w^J being some constant matrix and vector, respectively. Define the function $F(\cdot\,; J)$ on $N(x^*)$ by $F(x; J) := y(x; J) - x$.

Let $x^k \in N(x^*)$ and $J_k \in \mathcal{B}(x^k)$. Since $y(\cdot\,; J_k)$ is an affine function on $N(x^*)$, Taylor's formula yields

$$F(x^*; J_k) = F(x^k; J_k) + F'(x^k; J_k)(x^* - x^k), \tag{37}$$

where $F'(\cdot\,; J_k) = V^{J_k} - I$ is the Jacobian of $F(\cdot\,; J_k)$, which is nonsingular from Lemma 4.2 and Corollary 4.3. Since $\mathcal{B}(x^k) \subseteq \mathcal{B}(x^*)$, we have $F(x^*; J_k) = y(x^*; J_k) - x^* = y_\alpha(x^*) - x^* = F(x^*) = 0$ by Lemma 3.5 (c) and Proposition 2.4. Exploiting the nonsingularity of $F'(x^k; J)$, we then obtain from (37) that

$$x^* = x^k - F'(x^k; J_k)^{-1} F(x^k; J_k).$$

The right-hand side is precisely the Newton iteration at x^k, and hence x^{k+1} coincides with the NoE x^*. □

5 Numerical Results

In order to compare the performance of the Newton method proposed in this paper with other methods from [22, 21, 14, 15, 16], we run the Newton method on a number of test problems that can be found in the literature.

We implemented the Newton method using MATLAB. In each iteration, the algorithm has to solve a nonlinear optimization problem in order to compute the value of $y_\alpha(x^k)$. We use the solver SNOPT from the TOMLAB package and report the number of iterations the SNOPT solver requires in order to compute the value of $y_\alpha(x^k)$ in the column 'InnerIt'.

The parameter α is set to 10^{-4} for all test runs and the iteration is stopped whenever $\|y_\alpha(x^k) - x^k\| < 10^{-12}$, hence we require a very high precision which cannot be reached by some of the other methods since their local rate of convergence is not sufficiently fast.

To calculate an element of the computable generalized Jacobian of y_α at x^k, we need to find an index set $J \in \mathcal{B}(x^k)$ together with a corresponding multiplier λ^k. This is an easy task if the linear independence constraint qualification (LICQ) holds at the minimum $y_\alpha(x^k)$. In this case, we can take, for example, $J := I_0(x^k)$, where $I_0(x)$ is defined by (13). However, since LICQ is not needed in our convergence theory, we have to find a way to compute J and λ^k under the weaker CRCQ assumption. To this end, consider the linear program

$$\begin{aligned} \min_{\lambda} \; & \sum_{i \in I_0} \lambda_i \\ \text{s.t. } & \nabla g(y_\alpha(x^k))\, \lambda = -\nabla_y \psi_\alpha(x^k, y_\alpha(x^k)), \\ & \lambda_i \geq 0 \quad \forall\, i \in I_0, \\ & \lambda_i = 0 \quad \forall\, i \in \{1, \ldots, m\} \setminus I_0, \end{aligned} \tag{38}$$

where $I_0 := I_0(x^k)$. Since CRCQ holds at $y_\alpha(x^k)$, it follows that $\mathcal{M}(x^k)$ is nonempty, and hence (38) has at least one feasible point. Moreover, the objective function is obviously bounded from below on the feasible set. Standard linear programming theory then shows that (38) is solvable; moreover, at least one of the vertices of the polyhedron defined by the feasible set of (38) is also a solution. Now, let λ^k be such a vertex solution of (38). Then, again by standard results for linear programs, it follows that the gradients $\nabla g_i(y_\alpha(x^k))$ corresponding to the positive components $\lambda_i^k > 0$ are linearly independent. This proves the following result.

Lemma 5.1 *Suppose that the CRCQ (or any other constraint qualification) holds at $y_\alpha(x^k)$. Let λ^k be a vertex solution of the linear program* (38) *and define $J := \{i \in I_0 \mid \lambda_i^k > 0\}$. Then J belongs to $\mathcal{B}(x^k)$.*

Note that, in principle, a vertex solution of the linear program (38) can be calculated by the simplex method. It should be noted, however, that the linear program (38) is not given in standard form since the rows of the constraint matrix may be linearly dependent. Typically, implementations of the simplex method deal with this problem automatically. Alternatively, one could modify (38) like in the Big-M method to get an equivalent linear program which satisfies the full row rank condition.

Example 5.2 This test problem is the river basin pollution game taken from [22]. The cost functions in this game[5] are quadratic with linear constraints and of the structure defined in Corollary 4.3 with positive definite matrix A. In view of Proposition 4.6, we may expect that the algorithm terminates after a finite number of iterations. We actually observe convergence in just one step, cf. Table 1 for the corresponding numerical results.

k	x_1^k	x_2^k	x_3^k	$\|y_\alpha(x^k) - x^k\|$	InnerIt
0	10.000000	10.000000	10.000000	12.0479757781438828	0
1	21.144791	16.027846	2.725969	0.0000000000000000	6

Table 1: Numerical results for Example 5.2

[5]For some of the examples used in our numerical experiments, the players' optimization problems are stated as maximization problems in their original references. Here those problems are converted to minimization problems by negating the utility or payoff functions in the original formulations.

Example 5.3 This test problem is the internet switching model introduced by Kesselman et al. [19] and further analyzed by Facchinei et al. [11]. The cost function of each user is given by

$$\theta_\nu(x) = \frac{x^\nu}{B} - \frac{x^\nu}{\sum_{\nu=1}^N x^\nu},$$

with constraints $x^\nu \geq 0.01$, $\nu = 1, \ldots, N$ and $\sum_{\nu=1}^N x^\nu \leq B$. The constraints have been slightly modified from those in [19] to ensure that the cost functions θ_ν are defined on the whole feasible set. In the solution, all components are equal. This implies that our matrix $M(x^*, y_\alpha(x^*))$ is symmetric (whereas, in general, it is asymmetric at an arbitrary point x). In fact, this matrix turns out to be positive definite. To see this, recall that $M(x^*, y_\alpha(x^*))$ is equal to the Jacobian $\Theta'(x^*)$ (at the solution x^*), where Θ is the function defined by (33). Using [11, Eq. (26)], we obtain the following expression for this matrix:

$$M(x^*, y_\alpha(x^*)) = \Theta'(x^*) = -P/\mathcal{X}^3,$$

where $\mathcal{X} := x^{*,1} + \cdots + x^{*,N}$ and

$$P := -\begin{pmatrix} a & \cdots & a \\ \vdots & \ddots & \vdots \\ a & \cdots & a \end{pmatrix} - \operatorname{diag}(\mathcal{X}, \ldots, \mathcal{X})$$

with $a := (N-1)x^{*,\nu}$ for an arbitrary ν (recall that all components of the solution vector x^* are equal). Thus P is the sum of a negative semi-definite rank-one matrix and a negative definite diagonal matrix. Hence P is negative definite, implying that $M(x^*, y_\alpha(x^*))$ itself is positive definite. This ensures that Assumption 4.1 holds at x^* and we can expect local quadratic convergence of our Newton-type method.

For our numerical tests, we set $N = 10$ (and $B = 1$ in the description of the model in [11]) and use the starting point $x^0 = (0.1, \ldots, 0.1)^T$. We also use the starting point $x^0 = (0.1, 0.11, 0.12, \ldots, 0.19)$, since the relaxation method [16] failed with this starting point. The exact solution of this problem is $x^* = (0.09, \ldots, 0.09)^T$. We only show the first two components of the iteration vectors in Table 2 (all components are equal at each iteration) and the first four components in Table 3 for the second starting point.

Example 5.4 We consider a modification of a simple two-player game originally suggested by Rosen [32]. The solution violates strict complementarity,

k	x_1^k	x_2^k	$\|y_\alpha(x^k) - x^k\|$	InnerIt
0	0.100000	0.100000	0.1622713514699797	0
1	0.090238	0.090238	0.0037589337871505	4
2	0.090000	0.090000	0.0000000000000000	3

Table 2: Numerical results for Example 5.3 using $x^0 = (0.1, 0.1, \ldots, 0.1)^T$

k	x_1^k	x_2^k	x_3^k	x_4^k	$\|y_\alpha(x^k) - x^k\|$	InnerIt
0	0.100000	0.110000	0.120000	0.130000	0.4364630568571630	0
1	0.010000	0.010000	0.010000	0.010000	0.2846049894364130	1
2	0.100000	0.100000	0.100000	0.100000	0.1622713515888371	1
3	0.090238	0.090238	0.090238	0.090238	0.0037589338084981	4
4	0.090000	0.090000	0.090000	0.090000	0.0000000000000000	3

Table 3: Numerical results for Example 5.3 using $x^0 = (0.1, 0.11, \ldots, 0.19)^T$

and with our modification, LICQ does not hold, whereas CRCQ is satisfied. More precisely, this example has the two players' cost functions

$$\theta_1(x_1, x_2) = \frac{1}{2}x_1^2 - x_1x_2 \quad \text{and} \quad \theta_2(x_1, x_2) = x_2^2 + x_1x_2$$

and the common constraints given by

$$x_1 \geq 1,\ x_2 \geq 0,\ x_1 + x_2 \geq 1.$$

The unique normalized Nash equilibrium of this GNEP is $x^* = (1, 0)^T$. Since this game involves quadratic cost functions and linear constraints, we expect (locally) one-step convergence to the exact solution. Indeed, this is clearly observed from our numerical results shown in Table 4.

k	x_1^k	x_2^k	$\|y_\alpha(x^k) - x^k\|$	InnerIt
0	1.000000	1.000000	0.9999999999353941	0
1	1.000000	0.000000	0.0000000000000000	1

Table 4: Numerical results for Example 5.4 using $x^0 = (1, 1)^T$

Example 5.5 This test problem is the Cournot oligopoly problem with shared constraints and nonlinear cost functions as described in Outrata et al. [27, p. 233]. Our results (using different values for the parameter P from [27]) are given in Table 5. Although the objective functions are non-quadratic (in fact, highly nonlinear), our method converges to a solution in 2 or 3 iterations only, and the accuracy of the computed solution, displayed in the second last column, is very high in all instances of this problem.

k	x_1^k	x_2^k	x_3^k	x_4^k	x_5^k	$\|y_\alpha(x^k) - x^k\|$	InnerIt
$P = 75$							
0	10.000000	10.000000	10.000000	10.000000	10.000000	11.5863027186178531	0
1	10.727996	13.099087	15.304209	17.218265	18.650443	0.2667545777621945	7
2	10.403967	13.035818	15.407354	17.381555	18.771306	0.0000000000000000	4
$P = 100$							
0	10.000000	10.000000	10.000000	10.000000	10.000000	22.5856681233344716	0
1	14.742243	17.889842	20.649363	22.776440	23.942112	0.5830566903965523	7
2	14.050339	17.798223	20.907147	23.111451	24.132840	0.0002091129151843	5
3	14.050091	17.798381	20.907187	23.111428	24.132914	0.0000000000000000	2
$P = 150$							
0	10.000000	10.000000	10.000000	10.000000	10.000000	44.8079718213763485	0
1	24.666020	28.638950	31.530397	32.884666	32.279967	0.9504256360932131	9
2	23.588783	28.684250	32.021532	33.287256	32.418178	0.0000000000000000	7
$P = 200$							
0	10.000000	10.000000	10.000000	10.000000	10.000000	67.1154610852267837	0
1	36.770882	40.503658	42.325655	41.769703	38.630101	0.9181893886394852	10
2	35.785304	40.748979	42.802485	41.966390	38.696842	0.0000305348293455	7
3	35.785335	40.748961	42.802484	41.966378	38.696841	0.0000000000000000	2

Table 5: Numerical results for Example 5.5

Example 5.6 We solve the electricity market problem suggested by Contreras et al. [5]. This model involves three power generating companies with one, two, and three power plants, respectively. We consider the game where restriction is only imposed on the power production of each power plant, which corresponds to Case Study 1 in [5]. Note that this is a standard Nash equilibrium problem. This is a game with quadratic cost functions and linear constraints and it was shown in [5] that the game satisfies assumption (a) of Corollary 4.3. Hence we can expect that the algorithm terminates at the exact solution in a finite number of iterations. Table 6 shows that this is true.

k	x_1^k	x_2^k	x_3^k	x_4^k	x_5^k	$\|y_\alpha(x^k) - x^k\|$	InnerIt
0	0.000000	0.000000	0.000000	0.000000	0.000000	123.1607329584486052	0
1	80.000000	23.535935	12.590457	15.420678	10.118619	22.3691107475859674	32
2	46.661602	32.154064	15.003098	22.107038	12.339619	0.0000706604000202	12
3	46.661605	32.154015	15.003146	22.107028	12.339634	0.0000001040070503	4
4	46.661605	32.154015	15.003146	22.107028	12.339634	0.0000000000000000	2

Table 6: Numerical results for Example 5.6

6 Final Remarks

For a generalized Nash game with shared constraints, this paper exploits a (regularized) fixed point characterization of the normalized Nash equilibrium problem and develop a Newton-type method for its solution. This Newton-type method does not use Clarke's generalized Jacobian since this object is very difficult to compute in our case. Instead, we use a computable generalized Jacobian, which has the additional advantage that the constant rank constraint qualification (together with a second-order condition) suffices to prove local quadratic convergence (and even local one-step convergence for a game with quadratic cost functions and linear constraints). The numerical results indicate that the method is able to find a solution with very high precision. We therefore believe that this method is very promising, and an interesting future research topic is a suitable globalization of our locally convergent method.

References

[1] E. Allgower and K. Georg: *Simplicial and continuation methods for approximating fixed points and solutions to systems of equations.* SIAM Review 22, 1980, pp. 28–85.

[2] K.J. Arrow and G. Debreu: *Existence of an equilibrium for a competitive economy.* Econometrica 22, 1954, pp. 265–290.

[3] J.-P. Aubin: *Mathematical Methods of Game and Economic Theory.* North-Holland, Amsterdam, 1979.

[4] F.H. Clarke: *Optimization and Nonsmooth Analysis.* John Wiley, New York, 1983 (reprinted by SIAM, Philadelphia, 1990).

[5] J. Contreras, M. Klusch, and J.B. Krawczyk: *Numerical solutions to Nash-Cournot equilibria in coupled constraint electricity markets.* IEEE Transactions on Power Systems 19, 2004, pp. 195–206.

[6] R.W. Cottle, J.-S. Pang, and R.E. Stone: *The Linear Complementarity Problem.* Academic Press, Boston, 1992.

[7] G. Debreu: *A social equilibrium existence theorem.* Proceedings of the National Academy of Sciences 38 (10), 1952, pp. 886–893.

[8] S. Dempe and D. Pallaschke: *Quasidifferentiability of optimal solutions in parametric nonlinear optimization.* Optimization 40, 1997, pp. 1-24.

[9] S. Dempe and S. Vogel: *The generalized Jacobian of the optimal solution in parametric optimization.* Optimization 50, 2001, pp. 387–405.

[10] F. Facchinei and C. Kanzow: *Generalized Nash equilibrium problems.* 4OR – A Quarterly Journal of Operations Research 5, 2007, pp. 173–210.

[11] F. Facchinei, A. Fischer, and V. Piccialli: *Generalized Nash equilibrium problems and Newton methods.* Mathematical Programming 117, 2009, pp. 163–194.

[12] M. Fukushima: *Restricted generalized Nash equilibria and controlled penalty algorithm.* Technical Report 2008-007, Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, July, 2008.

[13] G. Gürkan and J.-S. Pang: *Approximations of Nash equilibria.* Mathematical Programming 117, 2009, pp. 223–253.

[14] A. von Heusinger and C. Kanzow: *Optimization reformulations of the generalized Nash equilibrium problem using Nikaido-Isoda-type functions.* Computational Optimization and Applications, available online under DOI 10.1007/s10589-007-9145-6.

[15] A. von Heusinger and C. Kanzow: *SC^1 optimization reformulations of the generalized Nash equilibrium problem.* Optimization Methods and Software 23, 2008, pp. 953–973.

[16] A. von Heusinger and C. Kanzow: *Relaxation methods for generalized Nash equilibrium problems with inexact line search.* Journal of Optimization Theory and Applications, to appear.

[17] W.W. Hogan: *Point-to-set maps in mathematical programming.* SIAM Review 15, 1973, pp. 591–603.

[18] R. Janin: *Directional derivative of the marginal function in nonlinear programming.* Mathematical Programming Study 21, 1984, pp. 110–126.

[19] A. Kesselman, S. Leonardi, and V. Bonifaci: *Game-theoretic analysis of internet switching with selfish users.* Lecture Notes in Computer Science 3828, 2005, pp. 236–245.

[20] M. Kojima and S. Shindo: *Extension of Newton and Quasi-Newton methods to systems of PC^1 equations.* Journal of the Operations Research Society of Japan 29, 1986, pp. 352–372.

[21] J.B. Krawczyk: *Coupled constraint Nash equilibria in environmental games.* Resource and Energy Economics 27, 2005, pp. 157–181.

[22] J.B. Krawczyk and S. Uryasev: *Relaxation algorithms to find Nash equilibria with economic applications.* Environmental Modeling and Assessment 5, 2000, pp. 63–73.

[23] K. Nabetani, P. Tseng and M. Fukushima: *Parametrized variational inequality approaches to generalized Nash equilibrium problems with shared constraints.* Technical Report 2008-011, Department of Applied Mathematics and Physics, Kyoto University, October 2008.

[24] J. Nash: *Non-cooperative games.* Annals of Mathematics 54, 1951, pp. 286–295.

[25] H. Nikaido and K. Isoda: *Note on noncooperative convex games.* Pacific Journal of Mathematics 5, 1955, pp. 807–815.

[26] J.M. Ortega and W.C. Rheinboldt: *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, 1970.

[27] J. Outrata, M. Kocvara, and J. Zowe: *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints.* Kluwer Academic Press, Dordrecht, 1998.

[28] J. Outrata and J. Zowe: *A numerical approach to optimization problems with variational inequality constraints.* Mathematical Programming 68, 1995, pp. 105-130.

[29] J.-S. Pang and D. Ralph: *Piecewise smoothness, local invertibility, and parametric analysis of normal maps.* Mathematics of Operations Research 21, 1996, pp. 401–426.

[30] L. Qi: *Convergence analysis of some algorithms for solving nonsmooth equations.* Mathematics of Operations Research 18, 1993, pp. 227–244.

[31] R.T. Rockafellar and R.J.-B. Wets: *Variational Analysis.* Springer, Berlin, 2004.

[32] J.B. Rosen: *Existence and uniqueness of equilibrium points for concave N-person games.* Econometrica 33, 1965, pp. 520–534.

[33] D. Sun, M. Fukushima, and L. Qi: *A computable generalized Hessian of the D-gap function and Newton-type methods for variational inequality problems.* Complementarity and Variational Problems: State of the Art, M.C. Ferris and J.-S. Pang (eds.), SIAM, Philadelphia, 1997, pp. 452-472.