A GLOBALIZED NEWTON METHOD FOR THE COMPUTATION OF NORMALIZED NASH EQUILIBRIA

Axel Dreves[1], Anna von Heusinger[1],
Christian Kanzow[1], and Masao Fukushima[2]

Preprint 303 March 2011

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

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

March 24, 2011

Abstract. The generalized Nash equilibrium is a Nash game, where not only the players' cost functions, but also the constraints of a player depend on the rival players decisions. We present a globally convergent algorithm that is suited for the computation of a normalized Nash equilibrium in the generalized Nash game with jointly convex constraints. The main tool is the regularized Nikaido-Isoda function as a basis for a locally convergent nonsmooth Newton method and, in another way, for the definition of a merit function for globalization. We conclude with some numerical results.

Key Words: Generalized Nash equilibrium problem, Regularized Nikaido-Isoda function, Nonsmooth Newton method, Global convergence, Superlinear convergence.

1 Introduction

The purpose of this paper is to propose a globally convergent method for the computation of a normalized Nash equilibrium in the *generalized Nash equilibrium problem* (GNEP) with jointly convex constraints. The generalized Nash equilibrium has drawn increasing attention over the past years due to the fact that it serves as a useful modeling tool in, for instance, energy markets, telecommunication engineering, climate policy, and machine learning; see [2] for a recent overview on theory, applications and numerical methods for the GNEP.

Our method is based on the characterizations of the normalized Nash equilibrium given in [10, 12]. More precisely, our algorithm consists of a combination of the locally superlinearly convergent Newton method from [12] and a globally convergent gradient method with a continuously differentiable merit function. It draws on the features of the regularized Nikaido-Isoda function, which has earlier been used as a tool to derive globally convergent methods for the computation of normalized Nash equilibria. There have been various attempts to develop numerical methods for computing generalized Nash equilibria with different objectives and problem settings [3, 7, 9, 10, 11, 12, 15, 16, 19, 21, 23, 26]. However, to the authors' knowledge, none of the proposed methods is both globally convergent and locally superlinearly convergent at the same time, at least under reasonable assumptions. For example, projection-type methods have relatively nice global convergence properties, but the local rate of convergence is typically slow; on the other hand, reformulations of the GNEP as a variational inequality allow, in principle, the application of locally fast Newton-type methods which can also be globalized in a suitable way. However, the assumptions to get local fast convergence usually include a linear independence-type constraint qualification which is often violated in the context of GNEPs due to the fact that some joint constraints are active.

The generalized Nash equilibrium problem is defined through a number of players N, cost functions $\theta_\nu : \mathbb{R}^n \to \mathbb{R}, \nu = 1, \dots, N$, and a joint strategy set $X \subseteq \mathbb{R}^n$. The idea of the Nash game is that each player ν controls a strategy vector $x^\nu \in \mathbb{R}^{n_\nu}$, and the strategy vectors of all players together form the vector $x = (x^1, x^2, \dots, x^N) \in \mathbb{R}^n$ with $n := \sum_{\nu=1}^N n_\nu$. This vector x determines the outcome of the game which are the costs $\theta_\nu(x)$ for each player ν. The particular difficulty of the generalized Nash equilibrium problem is that the vector x has to belong to a *joint strategy set* X that is not necessarily a Cartesian product of strategy sets of individual players, but any convex closed set. For notational convenience, we often denote $x^{-\nu} = (x^1, \dots, x^{\nu-1}, x^{\nu+1}, \dots, x^N) \in \mathbb{R}^{n-n_\nu}$ and use the abbreviation $x = (x^\nu, x^{-\nu})$ for the vector $x = (x^1, x^2, \dots, x^N)$, especially in those situations where we want to emphasize the ν-th player's variables within the vector $x \in \mathbb{R}^n$.

A solution to the generalized Nash equilibrium problem, or simply a *generalized Nash equilibrium*, is a vector $\bar{x} \in X$ such that

$$\theta_\nu(\bar{x}^\nu, \bar{x}^{-\nu}) \leq \theta_\nu(x^\nu, \bar{x}^{-\nu}) \quad \text{for all } x^\nu : (x^\nu, \bar{x}^{-\nu}) \in X \tag{1}$$

holds for all $\nu = 1, \dots, N$. In other words, a generalized Nash equilibrium is a vector $\bar{x} \in X$ such that the individual player ν cannot improve his costs $\theta_\nu(\bar{x}^\nu, \bar{x}^{-\nu})$ by changing his own strategy x^ν (taking into account the joint constraint X), as long as all other players retain their strategies. The notion of a *normalized Nash equilibrium* [24], which is a particular generalized Nash equilibrium, will be explained in the next section. Our algorithm is suited for the following setting.

Assumption 1.1 *(i) The cost functions $\theta_\nu, \nu = 1, \dots, N$, are twice continuously differentiable and (player-) convex, that is, each function θ_ν is convex with respect to the variable x^ν.*

(ii) The joint strategy set $X \subseteq \mathbb{R}^n$ is nonempty and defined by

$$X := \{x \in \mathbb{R}^n \mid g(x) \leq 0\},$$

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

The smoothness assumptions on the cost functions and constraints are necessary since we aim to design a locally superlinearly convergent method.

The paper is organized as follows. In the next section, we introduce the regularized Nikaido-Isoda function along with the terminologies and results we need in order to state our algorithm. Then, in Section 3, we present the algorithm and analyse both its global and local convergence. Numerical results are stated in Section 4. We close with some final remarks in Section 5.

Throughout the paper we use the following notation. For a real-valued function $f : \mathbb{R}^n \to \mathbb{R}$, the gradient of f is denoted by ∇f. For a vector-valued function $G : \mathbb{R}^n \to \mathbb{R}^m$, the symbol ∇G refers to the transpose of the Jacobian of G, that is, the columns of G are the gradients of the component functions of G. Finally, if $\Psi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, then $\nabla^2_{yx}\Psi(x, y)$ denotes the matrix of second partial derivatives which is obtained by first differentiating with respect to y only, and then with respect to the partial vector x.

2 Preliminaries

We consider a numerical approach that is based on the Nikaido-Isoda function. This function initially has been introduced in [22] for theoretical analysis of Nash equilibria. However, it is also suited for computational considerations with the addition of a regularization term. The *regularized Nikaido-Isoda function* has been exploited in [10] and subsequent works. We briefly recall the definitions and results required in the subsequent analysis.

Given the player's cost functions θ_ν, $\nu = 1, \dots, N$, and a fixed parameter $\gamma > 0$, the *regularized Nikaido-Isoda function* is defined by

$$\Psi_\gamma(x, y) := \sum_{\nu=1}^{N} \Big[\theta_\nu(x^\nu, x^{-\nu}) - \theta_\nu(y^\nu, x^{-\nu}) - \frac{\gamma}{2}\|x^\nu - y^\nu\|^2\Big] \tag{2}$$

for $x, y \in \mathbb{R}^n$. The Nikaido-Isoda function measures, roughly speaking, how much a player gains if, at a certain stage of the game, the player changes his strategy from x^ν to y^ν, and sums this advantage up for all players. The regularization term $\frac{\gamma}{2}\|x^\nu - y^\nu\|^2$ mainly serves for technical purposes. More specifically, if the cost functions $\theta_\nu, \nu = 1, \dots, N$, are (player-) convex as in Assumption 1.1(i), then Ψ_γ is strongly concave with respect to y. This, together with the assumption that the feasible set X is nonempty, closed and convex (Assumption 1.1(ii)), implies that the optimization problem $\max\limits_{y \in X} \Psi_\gamma(x, y)$ has a unique solution. Thus, its solution defines a function

$$y_\gamma(x) := \underset{y \in X}{\operatorname{argmax}}\ \Psi_\gamma(x, y), \tag{3}$$

which is, in general, not differentiable everywhere. Further, we define the function

$$V_\gamma(x) := \Psi_\gamma\big(x, y_\gamma(x)\big), \tag{4}$$

and, in particular, for two different fixed parameters $\beta > \alpha > 0$ we consider the function build from the difference of two functions V_α and V_β,

$$V_{\alpha\beta}(x) := V_\alpha(x) - V_\beta(x).$$

The functions y_γ and $V_{\alpha\beta}$ have a number of interesting properties that are summarized in the following result, see [10] for the proof.

Theorem 2.1 *Suppose that Assumption 1.1 holds and let $\gamma > 0$, $\beta > \alpha > 0$ be fixed parameters. Then*

(i) the functions y_γ, V_γ and $V_{\alpha\beta}$ are continuous;

(ii) any vector $\bar{x} \in \mathbb{R}^n$ with $y_\gamma(\bar{x}) = \bar{x}$ is a generalized Nash equilibrium;

(iii) $V_{\alpha\beta}(x) \geq 0$ for all $x \in \mathbb{R}^n$;

(iv) any vector $\bar{x} \in \mathbb{R}^n$ with $V_{\alpha\beta}(\bar{x}) = 0$ is a generalized Nash equilibrium;

(v) the function $V_{\alpha\beta}$ is continuously differentiable and its gradient is given by

$$\begin{aligned}
\nabla V_{\alpha\beta}(x) \quad = \quad & \sum_{\nu=1}^{N} \left[\nabla\theta_\nu(y_\beta^\nu(x), x^{-\nu}) - \nabla\theta_\nu(y_\alpha^\nu(x), x^{-\nu})\right] \\
& + \begin{pmatrix} \nabla_{x^1}\theta_1(y_\alpha^1(x), x^{-1}) - \nabla_{x^1}\theta_1(y_\beta^1(x), x^{-1}) \\ \vdots \\ \nabla_{x^N}\theta_N(y_\alpha^N(x), x^{-N}) - \nabla_{x^N}\theta_N(y_\beta^N(x), x^{-N}) \end{pmatrix} \\
& - \alpha\big(x - y_\alpha(x)\big) + \beta\big(x - y_\beta(x)\big).
\end{aligned}$$

Remark 2.2 (Normalized Nash equilibria) The statements (ii) and (iv) do not yield a characterization of the full set of generalized Nash equilibria. In fact, both give a characterization of the set of *normalized Nash equilibria (NoE)* [24], sometimes also called *variational equilibria* [2], since, given that the cost functions and feasible set have some additional properties, the set of normalized Nash equilibria equals the solution set of a certain variational inequality problem, see [4]. For a discussion on further properties and applications of normalized Nash equilibria, see, e.g., [2, 10]. $\Diamond$

Theorem 2.1 essentially provides two characterizations of normalized Nash equilibria. Let $\gamma = \beta > \alpha > 0$. Statement (ii) says that a normalized Nash equilibrium (NoE) is a solution of the nonlinear equation

$$F_\beta(x) := y_\beta(x) - x = 0, \tag{5}$$

which is nonsmooth in general. Furthermore, statements (iii) and (iv) imply that $\bar{x}$ is a NoE if and only if it is a global solution with vanishing function value of the optimization problem

$$\min_{x \in \mathbb{R}^n} V_{\alpha\beta}(x), \tag{6}$$

where the objective function $V_{\alpha\beta}$ is continuously differentiable from (v). We combine both characterizations to design a globally, and locally fast, convergent method for the computation of a NoE. Specifically, we try to use a nonsmooth Newton direction for the nonlinear equation (5) as long as it decreases the merit function $V_{\alpha\beta}$, but we switch to a gradient direction for the function $V_{\alpha\beta}$ if it fails to be a descent direction. The former aims at fast local convergence, while the latter ensures global convergence of the algorithm.

To validate the proposed algorithm, we should find a condition that ensures that a stationary point of the function $V_{\alpha\beta}$ is a solution of problem (6), and hence, a normalized Nash equilibrium. Further, for the nonsmooth Newton method, we have to establish conditions which ensure that the function F_β is at least Lipschitz-continuous, and calculate a suitable approximation for the Jacobian of $F_\beta(x)$, in order to make a nonsmooth Newton method conveniently implementable. Both questions have been answered in [10, 12].

In particular, the proof of the next lemma can be found in [10, Theorem 4.6].

Lemma 2.3 *Assume that, in addition to Assumption 1.1, the following holds: For any $x \in \mathbb{R}^n$ such that $y_\alpha(x) \neq y_\beta(x)$, the inequality*

$$\sum_{\nu=1}^{N} \left[\nabla\theta_\nu(y_\beta^\nu(x), x^{-\nu}) - \nabla\theta_\nu(y_\alpha^\nu(x), x^{-\nu})\right]^T \big(y_\beta(x) - y_\alpha(x)\big) > 0 \tag{7}$$

holds. Then any stationary point of the function $V_{\alpha\beta}$ is a normalized Nash equilibrium.

Sufficient conditions for the condition (7) in Lemma 2.3 and a further discussion may be found in [10, 11] and are somewhat related to the corresponding conditions given in [6, 25], for example.

Moreover, there is an inequality relation among the values of $V_{\alpha\beta}(x)$, $\|y_\alpha(x) - x\|$, and $\|y_\beta(x) - x\|$, which can be found in [10, Lemma 4.1]

Lemma 2.4 *The inequalities*

$$\frac{\beta-\alpha}{2}\|x-y_\beta(x)\|^2 \le V_{\alpha\beta}(x) \le \frac{\beta-\alpha}{2}\|x-y_\alpha(x)\|^2 \tag{8}$$

hold for all $x \in \mathbb{R}^n$.

In our method, this growth condition on the merit function $V_{\alpha\beta}$ is a necessary tool for the transition from the globally convergent gradient method based on $V_{\alpha\beta}$ to the locally convergent Newton method based on $F_\beta(x) = y_\beta(x) - x$. Such a technique has been applied to the numerical solution of nonlinear complementarity problems, see for instance [5] and [18].

For the nonsmooth Newton method, we use the very same approach as in [12]. That is, we compute an element of the *computable generalized Jacobian* of $F_\beta(x)$. By definition (3), to calculate the value of $y_\beta(x)$, we have to solve the maximization problem

$$\max_{y\in X} \Psi_\beta(x,y). \tag{9}$$

By Theorem 2.1 (i), $y_\beta(\cdot)$ is a continuous function. To get further insight into the analytic properties of the function $y_\beta(\cdot)$, we formulate the Karush-Kuhn-Tucker conditions for problem (9). To this end, we need a constraint qualification. For our purposes, the constant rank constraint qualification (CRCQ), which is weaker than the linear independence constraint qualification (LICQ) [13] suffices. Note, in particular, that linear constraints satisfy the CRCQ. For each $x \in X$, let $I(x) := \{i \in \{1,\dots,m\} \mid g_i(x) = 0\}$ denote the index set of active constraints at x.

Assumption 2.5 *The* constant rank constraint qualification (CRCQ) *holds at* x*, that is, there exists a neighbourhood* $U(x)$ *of* x *such that for any subset* $J \subseteq I(x)$*, the set of gradient vectors* $\{\nabla g_i(z) \mid i \in J\}$ *has the same rank for all* $z \in U(x)$*.*

Given $x \in \mathbb{R}^n$ and the unique solution $y_\beta(x)$ of problem (9), if Assumption 2.5 holds at $y_\beta(x)$, then there exists a Lagrange multiplier $\lambda \in \mathbb{R}^m$ such that the following Karush-Kuhn-Tucker conditions hold:

$$\begin{aligned} &-\nabla_y \Psi_\beta(x, y_\beta(x)) + \sum_{i=1}^m \lambda_i \nabla g_i(y_\beta(x)) = 0, \\ &\lambda_i \ge 0, \quad g_i(y_\beta(x)) \le 0, \quad \lambda_i \cdot g_i(y_\beta(x)) = 0 \quad \forall i = 1,\dots,m. \end{aligned} \tag{10}$$

Since we assume only the constant rank constraint qualification to hold, the Lagrange multiplier is not necessarily unique. Let

$$\mathcal{M}(x) := \{\lambda \in \mathbb{R}^m \mid (x, y_\beta(x), \lambda) \text{ satisfies (10)}\} \tag{11}$$

denote the set of Lagrange multipliers. We define a family of certain subsets of the active index set $I(x)$ by

$$\begin{aligned} \mathcal{B}(x) \;:=\; \Big\{ J \subseteq I(x) \mid \exists \lambda \in \mathcal{M}(x) \text{ such that } \lambda_i = 0 \text{ for all } i \in I(x) \setminus J, \\ \text{and } \{\nabla g_i(x)\}_{i\in J} \text{ are linearly independent } \Big\}. \end{aligned} \tag{12}$$

The next theorem is concerned with a formula for the computable generalized Jacobian of F_β. For details regarding its derivation see [12, Lemmata 3.4 and 3.5]. A function f is said to be *piecewise continuously differentiable near* $x \in \mathbb{R}^n$, if f is continuous and there is a neighbourhood $U(x)$ of x and a finite number of continuously differentiable functions $f_i : U(x) \to \mathbb{R}^n$, $i = 1, \dots, l$, such that $f(y) \in \{f_1(y), \dots, f_l(y)\}$ for all $y \in U(x)$.

Lemma 2.6 *Assume that Assumption 1.1 holds and that Assumption 2.5 holds at $y_\beta(x)$. Then the function $F_\beta(x) := y_\beta(x) - x$ is piecewise continuously differentiable near x, and the computable generalized Jacobian of F_β at x is defined by*

$$\partial_C F_\beta(x) := \{\nabla y_\beta^J(x)^T - I_n \mid J \in \mathcal{B}(x)\}, \tag{13}$$

where

$$\nabla y_\beta^J(x)^T = C^{-1}A - C^{-1}D(D^T C^{-1} D)^{-1} D^T C^{-1} A \tag{14}$$

with

$$\begin{array}{rcccl} A & := & A(x) & := & \nabla^2_{yx}\Psi_\beta(x, y_\beta(x)), \\ C & := & C^J(x) & := & -\nabla^2_{yy}\Psi_\beta(x, y_\beta(x)) + \sum_{i \in J} \lambda_i \nabla^2 g_i(y_\beta(x)), \\ D & := & D^J(x) & := & \nabla g_J(y_\beta(x)). \end{array}$$

Here $\nabla g_J(\cdot)$ is a short-hand notation for the matrix with column vectors $\{\nabla g_i(\cdot)\}_{i\in J}$, and $I_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix.

The explicit formulas for $\nabla^2_{yx}\Psi_\beta(x, y)$ and $\nabla^2_{yy}\Psi_\beta(x, y)$ are given as follows (cf. [9]):

$$\begin{aligned} \nabla^2_{yx}\Psi_\beta(x,y) &= -\begin{pmatrix} \nabla^2_{x^1x^1}\theta_1(y^1, x^{-1}) & \cdots & \nabla^2_{x^1x^N}\theta_1(y^1, x^{-1}) \\ \vdots & \ddots & \vdots \\ \nabla^2_{x^Nx^1}\theta_N(y^N, x^{-N}) & \cdots & \nabla^2_{x^Nx^N}\theta_N(y^N, x^{-N}) \end{pmatrix} \\ &\quad + \operatorname{diag}\begin{pmatrix} \nabla^2_{x^1x^1}\theta_1(y^1, x^{-1}) & & \\ & \ddots & \\ & & \nabla^2_{x^Nx^N}\theta_N(y^N, x^{-N}) \end{pmatrix} + \beta I, \\ \nabla^2_{yy}\Psi_\beta(x,y) &= -\operatorname{diag}\begin{pmatrix} \nabla^2_{x^1x^1}\theta_1(y^1, x^{-1}) & & \\ & \ddots & \\ & & \nabla^2_{x^Nx^N}\theta_N(y^N, x^{-N}) \end{pmatrix} - \beta I. \end{aligned}$$

Note that the Lagrange multiplier λ used in the definition of C does depend on x, though not explicitly stated.

3 Globalized Newton Method via $V_{\alpha\beta}$

Using the notation of the previous section, we are now ready to state the algorithm.

Algorithm 3.1 *(Globalized Newton method via $V_{\alpha\beta}$)*

(S.0) *Choose* $x^0 \in \mathbb{R}^n, \varepsilon \geq 0, s > 1, \rho > 0, \tau \in (0,1), \sigma \in (0,1)$*, and set* $k := 0$*.*

(S.1) *If* $\|F_\beta(x^k)\| = \|y_\beta(x^k) - x^k\| \leq \varepsilon$*, STOP.*

(S.2) *Compute an element* $H_k \in \partial_C F_\beta(x^k)$*, and find a solution* $d^k \in \mathbb{R}^n$ *of the linear system*

$$H_k d^k = -F_\beta(x^k), \tag{15}$$

if one exists.

(S.3) *If the system* (15) *was solved and if*

$$V_{\alpha\beta}(x^k + d^k) \leq \tau V_{\alpha\beta}(x^k), \tag{16}$$

set $x^{k+1} := x^k + d^k, k \leftarrow k+1$*, and go to (S.1).*

(S.4) *If the system* (15) *was not solved or if the condition*

$$\nabla V_{\alpha\beta}(x^k)^T d^k \leq -\rho \|d^k\|^s \tag{17}$$

is not satisfied, set $d^k := -\nabla V_{\alpha\beta}(x^k)$*.*

(S.5) *Compute* $t_k := \max\{2^{-l} \mid l = 0,1,2,\ldots\}$ *such that*

$$V_{\alpha\beta}(x^k + t_k d^k) \leq V_{\alpha\beta}(x^k) + \sigma t_k \nabla V_{\alpha\beta}(x^k)^T d^k. \tag{18}$$

Set $x^{k+1} := x^k + t_k d^k, k \leftarrow k+1$*, and go to (S.1).*

This algorithm is reminiscent of the algorithm in [14] for solving the mixed complementarity problem, which combines a nonsmooth Newton method for the natural residual equation with a globalization strategy using the D-gap function.

Our analysis of Algorithm 3.1 starts with its accumulation points.

Theorem 3.2 *Suppose Assumption 1.1 holds, and that Assumption 2.5 holds for all* $x \in X$*. Then Algorithm 3.1 with* $\varepsilon = 0$ *either stops at a normalized Nash equilibrium or every accumulation point* $\bar{x}$ *of a sequence generated by the algorithm is either a stationary point of* $V_{\alpha\beta}$ *or a normalized Nash equilibrium.*

Proof. The proof is similar to those of the corresponding global convergence results in [17, 18], and is given here mainly for the sake of completeness.

If the algorithm stops at (S.1), then we have $F_\beta(x^k) = 0$, and x^k is a normalized Nash equilibrium in view of Theorem 2.1 (ii). Otherwise, consider a subsequence $\{x^k\}$ converging to $\bar{x}$. If for an infinite set of indices in this subsequence, we have $d^k := -\nabla V_{\alpha\beta}(x^k)$, then $\bar{x}$ is a stationary point of $V_{\alpha\beta}$ by standard arguments. If (16) holds infinitely often, we get $V_{\alpha\beta}(x^k) \to 0$ since $V_{\alpha\beta}(x) \geq 0$ and $\tau \in (0,1)$, implying that $\bar{x}$ is a normalized Nash equilibrium by Theorem 2.1 (i), (iv). Therefore we assume, without loss of generality, that the search direction is always obtained from the linear system $H_k d^k = -F_\beta(x^k)$ and that the condition (17) is always satisfied.

From (17), we have $\nabla V_{\alpha\beta}(x^k)^T d^k \leq -\rho \|d^k\|^s$, which together with $s > 1$ and continuity of $\nabla V_{\alpha\beta}$ implies boundedness of the sequence $\{\|d^k\|\}$. Subsequencing if necessary, we can assume $d^k \to \bar{d}$.

Assume $\bar{d} \neq 0$. From (18), we have $V_{\alpha\beta}(x^k + t_k d^k) \leq V_{\alpha\beta}(x^k)$. Since $V_{\alpha\beta}(x) \geq 0$ for all x, it follows that $V_{\alpha\beta}(x^k + t_k d^k) - V_{\alpha\beta}(x^k) \to 0$, which yields

$$t_k \nabla V_{\alpha\beta}(x^k)^T d^k \to 0. \tag{19}$$

Now suppose $t_k \to 0$. Notice that the line search rule implies

$$\frac{V_{\alpha\beta}(x^k + 2t_k d^k) - V_{\alpha\beta}(x^k)}{2t_k} > \sigma \nabla V_{\alpha\beta}(x^k)^T d^k.$$

Passing to the limit and exploiting the continuous differentiability of $V_{\alpha\beta}$, we have

$$\nabla V_{\alpha\beta}(\bar{x})^T \bar{d} \geq \sigma \nabla V_{\alpha\beta}(\bar{x})^T \bar{d}.$$

Since $\sigma \in (0, 1)$, this yields $\nabla V_{\alpha\beta}(\bar{x})^T \bar{d} \geq 0$, contradicting (17) since $\bar{d} \neq 0$. Therefore, $\{t_k\}$ must be bounded away from zero. However, (19) and (17) then imply $d^k \to 0$, contradicting $d^k \to \bar{d} \neq 0$.

Therefore we must have $\bar{d} = 0$. Lemma 2.6 implies that F_β is a piecewise continuously differentiable function, and hence, in a neighbourhood of $\bar{x}$, each H_k is the Jacobian of one of finitely many C^1 functions. Thus, the sequence $\{H_k\}$ is bounded. Since $\|F_\beta(x^k)\| = \|H_k d^k\| \leq \|H_k\| \|d^k\|$ and $d^k \to \bar{d} = 0$, it follows that $F_\beta(\bar{x}) = 0$. Hence, again from Theorem 2.1 (ii), we conclude that $\bar{x}$ is a normalized Nash equilibrium. □

Remark 3.3 Note that Theorem 3.2 requires Assumption 2.5 to hold at every point $x \in X$. This requirement is unnecessarily strong and is used here only for a simple statement of Algorithm 3.1 (since the computable generalized Jacobian may not exist without the CRCQ). However, the result would remain true if the matrix H_k is alternatively chosen from any set $G(x^k)$, where G is a set-valued mapping which is upper semi-continuous and such that $G(x)$ is a nonempty and compact set for all $x \in \mathbb{R}^n$. A possible candidate for this set-valued mapping is, for example, Clarke's generalized Jacobian, cf. [1]. The reason for using the computable generalized Jacobian from the very beginning is essentially due to the observation that this Jacobian allows us to prove a very nice local convergence result. ◊

Lemma 2.3 shows a condition under which any stationary point of the function $V_{\alpha\beta}$ is a normalized Nash equilibrium. Next we examine the local convergence of Algorithm 3.1. To this end, we consider the nonsingularity of matrices H_k and acceptance of the full Newton step in (S.3) of Algorithm 3.1. The nonsingularity of matrices H_k in a neighbourhood of a normalized Nash equilibrium is guaranteed by [12, Lemma 4.2] in conjunction with the following assumption. The result in [12, Lemma 4.2] is stated for $x \in X$. However, taking a closer look at the proof of the lemma reveals that the assertion holds for all $x \in \mathbb{R}^n$.

Assumption 3.4 *For each $J \in \mathcal{B}(x)$ and $\lambda \in \mathcal{M}(x)$, we have*

$$d^T \left(M(x, y_\beta(x)) + \sum_{j \in J} \lambda_j \nabla^2 g_j(y_\beta(x)) \right) d \neq 0 \quad \forall d \in \mathcal{T}^J(x) \setminus \{0\},$$

with $\mathcal{T}^J(x) := \{d \in \mathbb{R}^n \mid \nabla g_j(y_\beta(x))^T d = 0 \, \forall j \in J\}$ and

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

In addition to the nonsingularity of matrices H_k, for local superlinear convergence, we need a superlinear approximation property. From [12, Theorem 4.5], we immediately obtain the following lemma.

Lemma 3.5 *Let $\bar{x}$ be a normalized Nash equilibrium, suppose Assumptions 1.1, 2.5 and 3.4 hold at $\bar{x}$, and let $\{x^k\}$ be any sequence converging to $\bar{x}$. If d^k is a solution of $H_k d^k = -F_\beta(x^k)$, then we have*

$$\|x^k + d^k - \bar{x}\| = o\|x^k - \bar{x}\|$$

Furthermore, if all functions θ_ν and g_i have locally Lipschitz continuous second derivatives, then

$$\|x^k + d^k - \bar{x}\| = O(\|x^k - \bar{x}\|^2).$$

It remains to show that the full Newton step is eventually accepted in (S.3) of Algorithm 3.1. This is done in a way as in [5] and [18] for the nonlinear complementarity problem, using Lemma 3.5 above and the growth condition from Lemma 2.4.

Theorem 3.6 *Let $\bar{x}$ be a normalized Nash equilibrium and suppose that $\bar{x}$ is an accumulation point of a sequence $\{x^k\}$ generated by Algorithm 3.1. If Assumptions 1.1, 2.5 and 3.4 hold at $\bar{x}$, then the entire sequence $\{x^k\}$ converges to $\bar{x}$. Moreover, eventually the linear system* (15) *is solvable and condition* (16) *is satisfied, and $\{x^k\}$ converges superlinearly to $\bar{x}$. If, in addition, all functions θ_ν and g_i have locally Lipschitz continuous second derivatives, the convergence rate is quadratic.*

Proof. (a) We begin with some preliminary observations. From [12, Lemma 4.2], under Assumption 3.4, all matrices $H(\bar{x}) \in \partial_C F_\beta(\bar{x})$ are nonsingular, and so are the matrices $H(x) \in \partial_C F_\beta(x)$ for all x sufficiently close to $\bar{x}$. Hence, the system (15) is solvable for all x near $\bar{x}$, and

$$\|H(x)(x - \bar{x})\| \geq \frac{\|x - \bar{x}\|}{\|H(x)^{-1}\|} \geq c\|x - \bar{x}\|$$

for some constant $c > 0$ and all x in a small neighbourhood of $\bar{x}$. Moreover, we have

$$\|F_\beta(x) - H(x)(x - \bar{x})\| \leq \frac{c}{2}\|x - \bar{x}\|$$

for all x sufficiently close to $\bar{x}$, cf. [12, Lemma 4.4]. Therefore we obtain

$$\begin{aligned} c\|x - \bar{x}\| - \|F_\beta(x)\| &\leq \|H(x)(x - \bar{x})\| - \|F_\beta(x)\| \\ &\leq \|F_\beta(x) - H(x)(x - \bar{x})\| \\ &\leq \frac{c}{2}\|x - \bar{x}\|, \end{aligned}$$

that is,

$$\frac{c}{2}\|x - \bar{x}\| \leq \|F_\beta(x)\|, \tag{20}$$

provided x is sufficiently close to $\bar{x}$.

(b) Next, we show that the entire sequence $\{x^k\}$ converges to $\bar{x}$. By [20, Lemma 4.10], it suffices to show that $\bar{x}$ is a locally unique solution of $F_\beta(x) = 0$ and that $\{\|x^{k+1} - x^k\|\}_K$ converges to 0 for any subsequence $\{x^k\}_K$ converging to $\bar{x}$. The fact that $\bar{x}$ is a locally unique solution follows, under our assumptions, immediately from (20). The updating rules in Algorithm 3.1 imply

$$\|x^{k+1} - x^k\| \leq \|d^k\| \quad \forall k \in K. \tag{21}$$

Furthermore, for all $k \in K$ satisfying the test (17), the Cauchy-Schwarz inequality gives

$$\rho\|d^k\|^s \leq -\nabla V_{\alpha\beta}(x^k)^T d^k \leq \|\nabla V_{\alpha\beta}(x^k)\| \, \|d^k\|,$$

which together with $s > 1$ implies

$$\rho\|d^k\|^{s-1} \leq \|\nabla V_{\alpha\beta}(x^k)\|.$$

It then follows from Theorem 3.2 that $\{\|d^k\|\}_K$ tends to 0. (Recall that $d^k = -\nabla V_{\alpha\beta}(x^k)$ for all $k \in K$ violating (17).) Hence the desired result follows from (21).

(c) Finally, we prove that $\{x^k\}$ converges locally superlinearly/quadratically to $\bar{x}$. This is done by showing that the globalized Newton method from Algorithm 3.1 eventually coincides with the local Newton method and, therefore, inherits the convergence properties from the local method. To this end, we have to show that the linear system (15) is solvable and the corresponding Newton direction d^k satisfies the test (16) for all k sufficiently large.

From Lemma 2.4, we know that

$$\frac{\beta - \alpha}{2}\|F_\beta(x)\|^2 \leq V_{\alpha\beta}(x) \leq \frac{\beta - \alpha}{2}\|F_\alpha(x)\|^2 \tag{22}$$

for all $x \in \mathbb{R}^n$. Let $L > 0$ be the local Lipschitz constant of the function F_α around $\bar{x}$ (which exists since F_α is piecewise continuously differentiable due to Lemma 2.6). Then we have

$$\begin{aligned}\sqrt{V_{\alpha\beta}(x^k+d^k)} &\overset{(22)}{\leq} \sqrt{\frac{\beta-\alpha}{2}}\|F_\alpha(x^k+d^k)-\underbrace{F_\alpha(\bar{x})}_{=0}\| \\ &\leq \sqrt{\frac{\beta-\alpha}{2}}L\|x^k+d^k-\bar{x}\| \\ &\overset{\text{Lem. 3.5}}{=} o\big(\|x^k-\bar{x}\|\big).\end{aligned}$$

Therefore, for k sufficiently large, we have

$$\begin{aligned}\sqrt{V_{\alpha\beta}(x^k+d^k)} &\leq \sqrt{\tau}\sqrt{\frac{\beta-\alpha}{2}}\frac{c}{2}\|x^k-\bar{x}\| \\ &\overset{(20)}{\leq} \sqrt{\tau}\sqrt{\frac{\beta-\alpha}{2}}\|F_\beta(x^k)\| \\ &\overset{(22)}{\leq} \sqrt{\tau}\sqrt{V_{\alpha\beta}(x^k)}.\end{aligned}$$

Hence the test (16) is eventually successful. By Lemma 3.5, we then have superlinear or quadratic convergence of $\{x^k\}$ to $\bar{x}$. □

The above proof shows that the globalized Newton method eventually coincides with the local Newton method. Moreover, another result in [12] indicates that Algorithm 3.1 has the following finite termination property: If the assumptions of Theorem 3.6 hold and the Nash equilibrium problem is a quadratic game, i.e., the cost functions θ_ν are quadratic for all players $\nu = 1, \dots, N$ and the strategy set X is polyhedral, and if x^k is sufficiently close to a normalized Nash equilibrium $\bar{x}$, then the next iterate x^{k+1} coincides with $\bar{x}$.

4 Numerical Results

We implemented the globalized Newton method in MATLAB®. We used the solver SNOPT from the TOMLAB® package to solve the subproblem

$$\max_y \Psi_\gamma(x,y) \quad \text{s.t.} \quad g(y) \leq 0.$$

The algorithm stops if $\|F_\beta(x^k)\| < \varepsilon$, or the maximum number of iterations $kmax$ is reached. As parameters for the algorithm, we used $\varepsilon = 10^{-6}$, $kmax = 100$, $s = 2.1$, $\rho = 10^{-8}$, $\tau = 0.5$, $\sigma = 10^{-2}$, and for the function $V_{\alpha\beta}$, we set $\alpha = 10^{-2}$ and $\beta = 1$. We solved the jointly convex GNEPs named A11–A18, which are taken from the Appendix of the report version of the paper [3], and an electricity market model named Heu, which is taken from [8]. The details of these examples can be found in the references, thus we skip them here.

Table 1 reports the name of the example and the chosen starting point x^0 in the first two columns. All starting points x^0 were chosen so that all of their components were identical. For simplicity of notation, we denote for example $x^0 = 1$ instead of $x^0 = (1, 1, \dots, 1)^T$. The next three columns of the table show the number of iterations (It.), the number of gradient steps (grad), and the function value ($\|F_\beta(x)\|$) at the computed solution for the globalized Newton method. Since our method is a globalization of the local Newton method from [12], we also report numerical results for the local method, using both the parameter α in columns 6 and 7 and the parameter β in columns 8 and 9. The local method has the same stopping criteria as the globalized one. Additionally the algorithm stops if the Newton equation (15) cannot be solved sufficiently accurate in the sense that for the computed solution d^k we have $\|H_k d^k + F_\gamma(x^k)\| > 10^{-2}$. If this happens or if the maximum number of iterations is reached, we report an “F” in Table 1, thus indicating a failure of the method. The results show that the local method is very sensitive to the choice of the parameter.

The results for Examples A11–A17 do not show much difference between the local and the globalized Newton methods. Note, however, that for Examples A16a, A16c and A16d, we had failures for the starting point $x^0 = 1000$ for the local method with the smaller parameter $\alpha = 10^{-2}$. Moreover, Example A18 suffered from the singularity of matrices H_k, and hence the globalized method did not use the Newton direction. Therefore the globalized method was just a gradient method, and so it was slowly convergent. The local methods had the same singularity problem in A18, but nevertheless tried to compute the Newton direction. Although this failed in most cases, the method was successful in one case (for the smaller parameter α and starting point $x^0 = 1$), since the Newton equation was solved with sufficient accuracy without H_k being nonsingular! Example Heu shows the expected behaviour. The globalized method first took some gradient steps until it got close to a solution, and then switched to Newton steps for fast local convergence. The local method (with the small parameter α) was successful and much faster than the global one for the two starting points closer to the solution, but could not find a solution within the maximum number of iterations for the more far away one. For the larger parameter β, however, the local method could not solve the Newton equation with sufficiently high accuracy for any of the starting points.

The results show that the globalized Newton method is, as expected, more reliable than the local one, because it has the additional option to switch to the gradient step. In fact, the globalized Newton method was able to solve all test problems, whereas the local method had six failures for each of the parameter values α and β (on a different set of examples). The results also verify the finite termination property of the Newton method that was mentioned at the end of the previous section. In fact, for most examples, the function value at the last iterate is exactly zero!

5 Final Remarks

This paper describes a globalized Newton method for the computation of normalized Nash equilibria and is based on the local Newton-type method from [12]. The globalized method turns out to be significantly more reliable than the corresponding local variant in our numerical tests.

To the best of our knowledge, the method presented here is currently the only one for which both global and local fast convergence (under fairly mild assumptions) can be shown. In the moment, the globalization is rather simple by switching to a steepest descent direction whenever the Newton-like direction does not seem to work. Alternatively, however, it is possible to use more sophisticated globalization schemes by switching to conjugate gradient or quasi-Newton directions whenever the Newton direction turns out to be inadequate. The corresponding theory can, very likely, be adapted in a suitable way, and the numerical performance might improve considerably by using such a strategy. The details are left as possible future research topics.

References

[1] Clarke, F.H.: Optimization and Nonsmooth Analysis. John Wiley, 1983 (reprinted by SIAM, 1990).

[2] Facchinei, F., Kanzow, C.: Generalized Nash equilibrium problems. 4OR – A Quarterly Journal of Operations Research 5, 173–210 (2007).

[3] Facchinei, F., Kanzow, C.: Penalty methods for the solution of generalized Nash equilibrium problems. SIAM Journal on Optimization 20, 2228–2253 (2010).

[4] Facchinei, H., Fischer, A., Piccialli, V.: On generalized Nash games and variational inequalities. Operations Research Letters 35, 159–164 (2007).

[5] Facchinei, F., Soares, J.: A new merit function for nonlinear complementarity problems and a related algorithm. SIAM Journal on Optimization 7, 225–247 (1997).

[6] Flåm, S.D., Ruszczyński, R.: Noncooperative convex games: Computing equilibrium by partial regularization. IIASA Working Paper 94-42, Laxenburg, Austria, 1994.

[7] Fukushima, M.: Restricted generalized Nash equilibria and controlled penalty algorithm. Computational Management Science, to appear.

[8] von Heusinger, A.: Numerical methods for the solution of the generalized Nash equilibrium problem, PhD Thesis, Institute of Mathematics, University of Würzburg, 2009.

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

[10] von Heusinger, A., Kanzow, C.: Optimization reformulations of the generalized Nash equilibrium problem using Nikaido-Isoda-type functions. Computational Optimization and Applications 43, 353–377 (2009).

[11] von Heusinger, A., Kanzow, C.: Relaxation methods for generalized Nash equilibrium problems with inexact line search. Journal of Optimization Theory and Applications 143, 159–183 (2009).

[12] von Heusinger, A., Kanzow, C., Fukushima, M.: Newton's method for computing a normalized equilibrium in the generalized Nash game through fixed point formulation. Mathematical Programming, to appear.

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

[14] Kanzow, C., Fukushima, M.: Solving box constrained variational inequalities by using the natural residual with D-gap function globalization. Operations Research Letters 23, 45–51 (1998).

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

[16] Kubota, K., Fukushima,M.: Gap function approach to the generalized Nash equilibrium problem. Journal of Optimization Theory and Applications 144, 511–531 (2010).

[17] De Luca, T., Facchinei, F., Kanzow, C.: A semismooth equation approach to the solution of nonlinear complementarity problems. Mathematical Programming 75, 407–439 (1996).

[18] De Luca, T., Facchinei, F., Kanzow, C.: A theoretical and numerical comparison of some semismooth algorithms for complementarity problems. Computational Optimization and Applications 16, 173–205 (2000).

[19] Matioli, L.C., Sosa, W., Yuan, J.: A numerical algorithm for finding solutions of a generalized Nash equilibrium problem. Technical Report, Departamento de Matemática, UFPR, Centro Politénico, Curitiba, Brazil, January 2011.

[20] Moré, J.J., Sorensen, D.C.: Computing a trust region step. SIAM Journal on Scientific and Statistical Computing 4, 553–572 (1983).

[21] Nabetani, K., Tseng, P., Fukushima, M.: Parametrized variational inequality approaches to generalized Nash equilibrium problems with shared constraints. Computational Optimization and Applications 48, 423–452 (2011).

[22] Nikaido, H., Isoda, K.: Note on noncooperative convex games. Pacific Journal of Mathematics 5, 807–815 (1955).

[23] Pang, J.-S., Fukushima, M.: Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games. Computational Management Science 2, 21–56 (2005). [Erratum: ibid. 6, 373–375 (2009).]

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

[25] Uryasev, S., Rubinstein, R.Y.: On relaxation algorithms in computation of noncooperative equilibria. IEEE Transactions on Automatic Control 39, 1263–1267 (1994).

[26] Zhang, J.Z., Qu, B., Xiu, N.H.: Some projection-like methods for the generalized Nash equilibria. Computational Optimization and Applications 45, 89–109 (2010).

		globalized Newton method			local method with α		local method with β	
Example	x^0	It.	grad	$\|F_\beta(x)\|$	It.	$\|F_\alpha(x)\|$	It.	$\|F_\beta(x)\|$
A11	0	2	0	0.0000e+00	1	0.0000e+00	2	0.0000e+00
A11	1	1	0	0.0000e+00	1	0.0000e+00	1	0.0000e+00
A11	100	1	0	0.0000e+00	1	0.0000e+00	1	0.0000e+00
A12	0	1	0	0.0000e+00	1	0.0000e+00	1	0.0000e+00
A12	1	1	0	0.0000e+00	1	0.0000e+00	1	0.0000e+00
A12	100	1	0	0.0000e+00	3	0.0000e+00	1	0.0000e+00
A13	0	2	0	0.0000e+00	1	0.0000e+00	2	0.0000e+00
A13	1	2	0	0.0000e+00	1	0.0000e+00	2	0.0000e+00
A13	100	2	0	0.0000e+00	1	0.0000e+00	2	0.0000e+00
A14	0.01	3	0	0.0000e+00	3	0.0000e+00	3	0.0000e+00
A14	1	3	0	0.0000e+00	4	0.0000e+00	3	0.0000e+00
A14	100	4	1	0.0000e+00	3	0.0000e+00	3	0.0000e+00
A15	0	1	0	0.0000e+00	2	0.0000e+00	1	0.0000e+00
A15	1	1	0	0.0000e+00	2	0.0000e+00	1	0.0000e+00
A15	100	2	0	0.0000e+00	3	0.0000e+00	2	0.0000e+00
A16a	10	3	0	0.0000e+00	2	0.0000e+00	3	0.0000e+00
A16a	100	3	0	0.0000e+00	3	0.0000e+00	3	0.0000e+00
A16a	1000	3	0	0.0000e+00	1	F	3	0.0000e+00
A16b	10	3	0	0.0000e+00	3	0.0000e+00	3	0.0000e+00
A16b	100	3	0	0.0000e+00	3	0.0000e+00	3	0.0000e+00
A16b	1000	3	0	0.0000e+00	3	0.0000e+00	3	0.0000e+00
A16c	10	3	0	0.0000e+00	3	0.0000e+00	3	0.0000e+00
A16c	100	3	0	0.0000e+00	3	0.0000e+00	3	0.0000e+00
A16c	1000	3	0	0.0000e+00	2	F	3	0.0000e+00
A16d	10	4	0	0.0000e+00	2	0.0000e+00	4	0.0000e+00
A16d	100	3	0	0.0000e+00	3	0.0000e+00	3	0.0000e+00
A16d	1000	3	0	0.0000e+00	2	F	3	0.0000e+00
A17	0	2	0	0.0000e+00	3	0.0000e+00	2	0.0000e+00
A17	1	2	0	0.0000e+00	1	0.0000e+00	2	0.0000e+00
A17	100	2	0	0.0000e+00	4	0.0000e+00	2	0.0000e+00
A18	0	17	17	2.9461e-07	4	F	9	F
A18	1	17	17	2.9476e-07	2	0.0000e+00	8	F
A18	100	14	14	3.2129e-07	7	F	1	F
Heu	100	33	30	0.0000e+00	4	0.0000e+00	1	F
Heu	500	17	13	0.0000e+00	6	0.0000e+00	1	F
Heu	1000	19	14	0.0000e+00	101	F	1	F

Table 1: Numerical results for globalized and local Newton methods