2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Nash Equilibria in Normal Games via Optimization Methods

Jens Buttler and Natalia Akchurina

Abstract— **This paper is devoted to Nash equilibria of normal-form games. We give a survey on computing Nash equilibria via optimization methods. Further on we prove, that the Nash equilibria of a game coincide with the zeros of a nonlinear, almost everywhere smooth system of equations with the same dimension as the strategy space. Thus, we can apply tools for solving nonlinear systems of equations. We present an algorithm for computing Nash equilibria, which has shown very satisfactory results.**

I. INTRODUCTION

In 1951 John Nash [20] introduced a concept of equilibrium that was destined to become a central idea in game theory and bear his name. Briefly, a Nash equilibrium of a game is a state, where no player has an incentive to change its strategy. Other concepts of solving games are less intuitive. Either they do not ensure the existence of the solutions in every game or do not have an economic interpretation.

For a long time no algorithm had been known to compute Nash equilibria. The Lemke-Howson algorithm [13] dating back to 1964 can be seen as the outset of research in this area. The algorithm was developed for 2-player games, but later extended for n-player games. It belongs to the class of path following algorithms.

For decades the trend has remained to compute Nash equilibria via path following methods. All path following methods are based on the homotopy idea — transfer a solution of an easier problem via a continuous path to a solution of the problem of interest. For example, in the works by Rosenmuller [23] and Wilson [27] the existence of a nonlinear path leading to a Nash equilibrium is proven.

Path following algorithms are often mathematically sophisticated. The drawbacks of path following algorithms are that they refer to tools deep in the theory of homotopy or dynamical systems and are therefore hard to implement. For example, the methods from Rosenmuller [23] or Wilson [27] have not been implemented yet. The algorithm by Govindan and Wilson [8] has shown great results in an implementation, however using very heavy tools from homotopy and dynamical systems. Another disadvantage is, that the algorithms may not find Nash equilibria in all games. For an interested reader, further information on path following algorithms can be found in the surveys [10] or [17].

In addition to the path following algorithms, Nash equilibria may be computed as fixed points, based on Scarf [24], as solutions of complementary problems [25], [15], via polynomial algebra [5] or as solutions of optimization problems. Apart from analytic algorithms, algebraic methods have been recently investigated. For instance, in the survey by Datta [5] computational methods for finding Nash equilibria based on computer algebra are presented. The algebraic approaches use a result from Datta [4] — any semi-algebraic set can be encoded as the set of Nash equilibria of a suitable game. This algebraic property shows, that computer algebra is also a reasonable approach to calculate Nash equilibria, but also illustrates the complexity of the problem.

The above approaches constitute the whole palette of methods each having their own advantages and drawbacks. Since the computation of Nash equilibria is PPAD-complete [22], it is unlikely that Nash equilibria may be computed in polynomial time. Therefore, methods are often developed for certain games, or the algorithms look for refinements of Nash equilibria. In this case, the existence of a solution is not guaranteed. Moreover, the research was rather focused on 2-player games than on games with larger number of players.

Existing algorithms often build up an optimization problem, where the global minima coincide with the Nash equilibria of the game. Other methods provide nonlinear systems of equations, which vanish in the Nash equilibria. The first method of this kind was presented by McKelvey [17] in 1992. Since then, only a few more approaches have been developed. The main problems of the methods are, that the optimization problems are not convex or smooth, and therefore the algorithms are not bound to terminate at a Nash equilibrium. Another crucial point is, that the methods do increase the dimension of the problem. On the other hand optimization problems and nonlinear systems of equations are very well studied and there are a number of efficient optimization algorithms. So reducing the Nash equilibria to optimization problems is more than justified. To the best of our knowledge there is no survey on optimization methods for Nash equilibria.

In this work we will give a survey on optimization methods for computing Nash equilibria and present a new approach, avoiding problems arising in other optimization formulations. As far as we know, none of the earlier algorithms uses a nonlinear, almost everywhere smooth system of equations, whose zeros form the set of Nash equilibria, without increasing the dimension of the problem.

The paper is organized as follows. In section II we introduce n-player games and Nash equilibrium problem. The next section III states the theorems that serve as a base for the existing optimization approaches we introduce in section IV. Section V gives the details of implementation

Jens Buttler is with DB Systel GmbH, Germany (e-mail: Jens.Buttler@deutschebahn.com)

Natalia Akchurina is with Department of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Germany (e-mail: anatalia@rtr.tu-darmstadt.de)

of the approaches from section IV. At last, in the sections VI, VII, VIII and II our approach is proposed, including a short introduction to complementary functions, the obtained system of nonlinear equations, the algorithm itself and the computational results.

II. PRELIMINARIES FROM GAME THEORY

We would like first to recall some fundamental concepts of game theory[1] [21], [19].

Definition 1: A pair of matrices (M^1, M^2) constitute a bimatrix (2-player normal-form) game G, where M^1 and M^2 are of the same size. The rows of M^k correspond to actions of player 1, $a^1 \in A^1$. The columns of M^k correspond to actions of player 2, $a^2 \in A^2$. $A^1 = \left\{a^1_1, a^1_2, \dots, a^1_{m^1}\right\}$ and $A^2 = \left\{a^2_1, a^2_2, \dots, a^2_{m^2}\right\}$ are the finite sets of discrete actions of players 1 and 2 respectively. The payoff $r^k(a^1, a^2)$ to player k can be found in the corresponding entry of the matrix M^k, $k = 1, 2$.

Definition 2: A pure ε-equilibrium of bimatrix game G is a pair of actions (a^1_*, a^2_*) such that

$$r^1(a^1_*, a^2_*) \geq r^1(a^1, a^2_*) - \varepsilon \quad \text{for all} \quad a^1 \in A^1$$
$$r^2(a^1_*, a^2_*) \geq r^2(a^1_*, a^2) - \varepsilon \quad \text{for all} \quad a^2 \in A^2$$

Definition 3: A mixed ε-equilibrium of bimatrix game G is a pair of vectors (ρ^1_*, ρ^2_*) of probability distributions over action spaces A^1 and A^2, such that

$$\rho^1_* M^1 \rho^2_* \geq \rho^1 M^1 \rho^2_* - \varepsilon \quad \text{for all} \quad \rho^1 \in \sigma(A^1)$$
$$\rho^1_* M^2 \rho^2_* \geq \rho^1_* M^2 \rho^2 - \varepsilon \quad \text{for all} \quad \rho^2 \in \sigma(A^2)$$

where $\sigma(A^k)$ is the set of probability distributions over action space A^k, such that for any $\rho^k \in \sigma(A^k)$, $\sum_{a \in A^k} \rho^k_a = 1$.

$$\begin{aligned} \rho^1 M^k \rho^2 &= \sum_{a^1 \in A^1} \sum_{a^2 \in A^2} \rho^1_{a^1} r^k(a^1, a^2) \rho^2_{a^2} = \\ &= \sum_{a^1 \in A^1} \sum_{a^2 \in A^2} r^k(a^1, a^2) \prod_{i=1}^{2} \rho^i_{a^i} \end{aligned}$$

is the expected reward of agent k induced by (ρ^1, ρ^2).

Let us denote the expected reward of agent k induced by (ρ^1, ρ^2) by $r^k(\rho^1, \rho^2)$:

$$r^k(\rho^1, \rho^2) = \sum_{a^1 \in A^1} \sum_{a^2 \in A^2} r^k(a^1, a^2) \prod_{i=1}^{2} \rho^i_{a^i}$$

Definition 4: Nash equilibrium of bimatrix game G is ε-equilibrium with $\varepsilon = 0$.

Theorem 1: [20] There exists a mixed Nash equilibrium for any bimatrix game.

Example 1: In table I a bimatrix game is presented in a short form (the first payoffs of each entry correspond to payoff matrix M^1 of player 1 and the second ones — to M^2).

This game possesses two Nash equilibria in pure strategies $\left(a^1_1, a^2_1\right)$ and $\left(a^1_2, a^2_2\right)$ and one Nash equilibrium in mixed strategies $\left(\left(\frac{5}{6}, \frac{1}{6}\right), \left(\frac{1}{6}, \frac{5}{6}\right)\right)$. Apparently, no agent will gain from unilateral deviation.

Definition 5: An n-player matrix (n-player normal-form) game is a tuple $\langle K, A^1, \dots, A^n, r^1, \dots, r^n \rangle$, where $K = \{1, 2, \dots, n\}$ is the player set, $A^k = \left\{a^k_1, a^k_2, \dots, a^k_{m^k}\right\}$ is the finite discrete action space of player k for $k \in K$ ($|A^k| = m^k$) and $r^k : A^1 \times A^2 \times \ldots \times A^n \to \mathbb{R}$ is the reward function for player k.

Definitions 2, 3, 4 and theorem 1 can be generalized for arbitrary number of players.

Policy of agent $k = 1, 2$ is a vector $x^k = \left(x^k_{a^k_1}, x^k_{a^k_2}, \dots, x^k_{a^k_{m^k}}\right)$, $x^k_h \in \mathbb{R}$ being the probability assigned by agent k to its action $h \in A^k$. Since all probabilities are nonnegative and their sum is equal to one, the vector $x^k \in \mathbb{R}^{m^k}$ belongs to the unit simplex Δ^k:

$$\Delta^k = \left\{ x^k \in \mathbb{R}^{m^k}_+ : \sum_{a^k \in A^k} x^k_{a^k} = 1 \right\}$$

Each player k ($k = 1, 2$) strives to maximize its expected discounted cumulative reward:

$$r^k(x^1, x^2) = \sum_{a^1 \in A^1} \sum_{a^2 \in A^2} r^k(a^1, a^2) \prod_{i=1}^{2} x^i_{a^i}$$

where x^1 and x^2 are the policies of players 1 and 2 respectively.

Definition 6: A profile is a vector $x = (x^1, x^2, \dots, x^n)$, where each component $x^k \in \Delta^k$ is a policy for player $k \in K$. The space of all profiles $\Phi = \times_{k \in K} \Delta^k$.

Let r^k_h be a partial derivative of r^k with respect to x^k_h, $k \in K$ and $h \in A^k$:

$$r^k_h(x) = \frac{\partial r^k(x)}{\partial x^k_h}$$

Let us note that

$$r^k_h(x) = r^k(x^1, \dots, x^{k-1}, e^k_h, x^{k+1}, \dots, x^n)$$

where e^k_h is a pure strategy of agent $k \in K$, $h \in A^k$:

$$e^k_h = \left(e^k_{ha^k_1}, e^k_{ha^k_2}, \dots, e^k_{ha^k_{m^k}}\right)$$

$$e^k_{ha} = \begin{cases} 0 & a \neq h \\ 1 & a = h \end{cases}$$

III. NECESSARY AND EQUIVALENT CONDITIONS

To compute Nash equilibria via optimization methods, one needs necessary or sufficient conditions for a Nash equilibrium. We can get the following necessary conditions for Nash equilibrium:

Theorem 2: [1] If x is a Nash equilibrium, then x is a zero of the system of nonlinear equations

$$\left[r^k_h(x) - r^k(x)\right] x^k_h = 0 \qquad \text{(III.1)}$$

The system is smooth and consists of multi-linear functions. Even though solving this system is no easy task, it is less complex than the systems we will present in the

---

[1] Further on, we will use concepts *player* and *agent*, terms *strategy* and *policy* and *reward* and *payoff* interchangeably.

TABLE I

A BIMATRIX GAME

	a_1^2	a_2^2
a_1^1	5,1	0,0
a_2^1	0,0	1,5

next sections. This system alone is not practicable, since not all solutions are Nash equilibria. But it can be of benefit when we have good starting points. This is in general an important approach for speeding up algorithms — use a reliable algorithm to get an approximate solution, and then use a fast local algorithm.

Since theorem 2 only provides a necessary condition, most of the existing optimization methods are based on the following theorem:

Theorem 3: [16] x is a Nash equilibrium if and only if x satisfies the following conditions:

1) $x \in \Phi$
2) $r_h^k(x) \leq r^k(x)\ \forall k \in K$ and $h \in A^k$

The theorem is very intuitive. Since x is a Nash equilibrium iff no agent has an incentive to change its strategy, there is no strategy whose payoff is higher than of the strategy played.

IV. EXISTING OPTIMIZATION APPROACHES

All existing optimization approaches are based on the theorems referred above. Especially theorem 3 can be seen as a starting point for optimization methods. In this section we will introduce essential concepts and methods for computing Nash equilibria in normal-form games by optimization. Further, we take a look at the drawbacks of the distinct methods, to obtain a more reliable and fast algorithm which addresses these problems.

In the work of McKelvey [16] theorem 3 is almost directly transfered into an optimization problem. This work from 1992 is the earliest, that transfers the Nash equilibrium problem to a standard minimization problem. Additionally to theorem 3 it only uses the equivalence:

$$a \in \mathbb{R}: \ \min a^2 = 0 \iff a = 0$$

So one gets the objective function:

$$w(x) \equiv \sum_{k \in K, h \in A^k} \left(\max\left[r_h^k(x) - r^k(x), 0\right]\right)^2 + \\ + \sum_{k \in K, h \in A^k} \left(\min\left[x_h^k, 0\right]\right)^2 + \sum_{k \in K}\left(1 - \sum_{h \in A^k} x_h^k\right)^2$$

Theorem 4: [16] x is a Nash equilibrium if and only if x is a global minimum of the optimization problem:

$$\min w(x)$$

The disadvantages of the approach are, that only global minima are Nash equilibria, that the functions are not smooth, especially they are not even continuously differentiable.

Another related work by Chatterjee [2] extends the optimization problem by n new variables, but assures that the function stays smooth.

Theorem 5: [2] x is a Nash equilibrium if and only if x is a global minimum of the optimization problem:

$$\min\left[\sum_{k \in K} \beta^k(x) - r^k(x)\right] \ s.t.$$

$$r_h^k(x) \leq \beta^k \ \ \forall k \in K, \ h \in A^k$$

$$\sum_{h \in A^k} x_h^k = 1 \ \ \forall k \in K$$

$$x_h^k \geq 0 \ \ \forall k \in K, \ h \in A^k$$

The problem has the optimal value 0. Further the value of β^k at the optimal point is the expected reward for player k.

Again, only the global minima are Nash equilibria. On the other hand we have everywhere smooth and multi-linear functions. But this formulation increases the dimension of the problem by n variables.

A straightforward approach to transforming the problem from theorem 3 is mentioned in the work of Lipton and Markakis [14]. The inequalities of theorem 3 are transformed by slack variables to equalities:

$$B_{kh} \equiv x_h^k - \beta_{kh}^2 = 0$$

$$\Gamma_k \equiv \sum_{h \in A^k} x_h^k - 1 = 0 \ \ \forall k \in K$$

$$\Lambda_{kh} \equiv r^k(x) - r_h^k(x) - \delta_{kh}^2 = 0 \ \ \forall k \in K, \ h \in A^k$$

Now one has the objective function:

$$v(x) \equiv \sum_{k \in K, h \in A^k} B_{kh}^2 + \sum_{k \in K} \Gamma_k^2 + \sum_{k \in K, h \in A^k} \Lambda_{kh}^2$$

The solutions of the system constitute the Nash equilibria, but the system has a higher dimension, due to the slack variables.

The mentioned methods give basic ideas how to handle the problem. All three attempts have in common, that the global solutions build the set of Nash equilibria. This is a requirement, that an optimization algorithm for computing Nash equilibria should satisfy. Another important issue is of course the dimension of the problem. As seen in the existing approaches, methods often have to increase the dimension. The speed of convergence is the third main characteristic, that depends on the smoothness and convexity of the system.

V. OPTIMIZATION METHODS FOR CONSTRAINED PROBLEMS

Penalty methods solve the optimization problem

$$\min f(x)\ s.t.$$

$$c(x) \leq 0$$

$$h(x) = 0$$

by approximating it with a sequence of unrestricted optimization problems [6]:

$$\min\left[f(x) + pS(x)\right]$$

When the conditions are violated, the penalty function $f(x) + pS(x)$ punishes by high function values of $S(x)$. The weight of $S(x)$ increases iteratively through the penalty parameter p. One choice of $S(x)$ could be

$$S(x) = \frac{1}{2}(\|(c(x))_+\| + \|h(x)\|^2)$$

We know that the functions $f(x)$, $c(x)$, $h(x)$ are always of the same type. They only depend on the chosen formulation of the Nash equilibrium problem. Thus, we can compute upper bounds for the penalty parameter p. So we only have to solve a single optimization problem, rather than a sequence. Formulas for these upper bounds can be found quickly, since

$$\min_{a^1 \in A^1, \ldots, a^n \in A^n} \{r^k(a^1, \ldots, a^n)\} \leq$$

$$\leq r^k(x), r^k_h(x) \leq$$

$$\leq \max_{a^1 \in A^1, \ldots, a^n \in A^n} \{r^k(a^1, \ldots, a^n)\}$$

and

$$0 \leq x^k_h \leq 1$$

$$\sum_{h \in A^k} x^k_h = 1$$

Sequential quadratic programming [7] (SQP) solves the optimization problem

$$\min f(x)\ s.t.$$

$$c(x) \leq 0$$

iteratively through quadratic approximations. Starting in x one tries to find a point $x + s$ with $f(x + s) = 0$. $f(x+s)$ is therefore approximated by the second order Taylor polynomial.

The algorithm of Chatterjee [2] uses SQP to solve the problem from theorem 5. Here we want to discuss further advantages of SQP.

For linear constraints the Hessian matrix vanishes, especially if one takes $0 \leq x^k_h$, $\sum_{h \in A^k} x^k_h = 1$. So if we start in a point $x \in \Phi$, we never become unfeasible, since the linear functions are exactly approximated. This fact is of high interest, since the values in optimization formulations like the penalty function, tend to become unfeasible.

VI. COMPLEMENTARY FUNCTIONS

The equation III.1 shows the complementarity of the Nash equilibrium problem. We want to use this property. For that, we will give a short introduction to complementary functions.

Definition 7: [9] A function $\phi : \mathbb{R}^2 \mapsto \mathbb{R}$ is called complementary, if for all $x, y \in \mathbb{R}$:

$$\phi(x, y) = 0 \iff x \geq 0,\ y \geq 0,\ xy = 0$$

One of the most famous complementary functions is the Fischer-Burmeister function

$$\phi_{FB}(x, y) \equiv (x + y) - (x^2 + y^2)^{\frac{1}{2}}$$

$$\Phi_{FB}(x, y) \equiv |\phi_{FB}(x, y)|^2$$

Lemma 1: [3] ϕ_{FB} and $\Phi_{FB}(x, y)$ are complementary.

The aim of the paper is to show that the Nash equilibrium problem can be represented by complementary functions as a system of nonlinear equations.

The Fischer-Burmeister function is not smooth at zero. If we want to make it smooth at zeros, we can take the following version of Fischer-Burmeister function:

Definition 8: [3]

$$\theta(x, y, \epsilon) \equiv (x + y) - (x^2 + y^2 + 2\epsilon^2)^{\frac{1}{2}}$$

is called the smooth Fischer-Burmeister function.

Lemma 2: [3] The smooth Fischer-Burmeister function $\theta(x, y, \epsilon)$ is continuously differentiable in $\mathbb{R} \times \mathbb{R} \times \mathbb{R}_+$

If $\epsilon = 0$, then the smooth Fischer-Burmeister function coincides with the Fischer-Burmeister function.

VII. COMPLEMENTARY FUNCTION APPROACH

Regarding theorems 2 and 3, we get with

$$0 = x^k_h(r^k_h(x) - r^k(x)) = x^k_h(r^k(x) - r^k_h(x))$$

and

$$x^k_h \geq 0,\ (r^k(x) - r^k_h(x)) \geq 0$$

a nonlinear complementary problem:

$$\phi(x^k_h, r^k(x) - r^k_h(x)) = 0$$

Especially the equation $x^k_h(r^k_h(x) - r^k(x)) = 0$ is now also, because of the complementary function ϕ, fulfilled. With $\sum_{h \in A^k} x^k_h = 1$ for all $k \in K$ we get:

$$F_{cf-RD}(x) \equiv \begin{pmatrix} \phi(x^1_1, r^1(x) - r^1_1(x)) \\ \phi(x^1_2, r^1(x) - r^1_2(x)) \\ \ldots \\ \phi(x^n_{m^n}, r^n(x) - r^n_{m^n}(x)) \\ 1 - \sum\limits_{h \in A_1} x^1_h \\ \ldots \\ 1 - \sum\limits_{h \in A_n} x^n_h \end{pmatrix} = 0 \tag{VII.1}$$

Theorem 6: x is a Nash equilibrium if and only if x satisfies VII.1 with a complementary function ϕ.

$$x \text{ is NE} \iff F_{cf-RD}(x) = 0$$

Proof: "$\Longrightarrow$"
Let x be a Nash equilibrium. Let $k \in K$ and $h \in A^k$. Since $x \in \Phi$, we have $x_h^k \geq 0$. According to theorem 3 $r^k(x) - r_h^k(x) \geq 0$ holds, and thus $(r^k(x) - r_h^k(x))x_h^k = 0$. So the complementary function $\phi(x_h^k, r^k(x) - r_h^k(x))$ vanishes for all $k \in K$ and $h \in A^k$. On the other hand $1 - \sum\limits_{h \in A^k} x_h^k = 0$ since $x \in \Phi$. So the equation is fulfilled.
"$\Longleftarrow$"
Suppose the system of equation $F_{cf-RD}(x)$ vanishes at x. Then we have:

$$\phi\left(x_h^k, r^k(x) - r_h^k(x)\right) = 0 \text{ and } \left(1 - \sum_{h \in A^k} x_h^k\right) = 0$$

for each player k and any $h \in A^k$.
Thus we have $x_h^k \geq 0$, $r^k(x) - r_h^k(x) \geq 0$ for all k, h and $(1 - \sum\limits_{h \in A^k} x_h^k) = 0$ for all players k. Then $x \in \Phi$ and $r^k(x) - r_h^k(x) \geq 0$. According to theorem 3 x forms a Nash equilibrium. ■

The proof is not based on heavy mathematical tools, which can be seen as an advantage for implementation. Still the statement of theorem 6 is quite strong, since it shows that the Nash equilibrium problem can be easily formulated as a nonlinear system of equations in $\sum\limits_{k \in K} m^k$ variables.

VIII. ALGORITHM

The state of the art is to use path following, Newton-Raphson or Levenberg-Marquardt algorithms for solving systems of nonlinear equations. For system VII.1 we have chosen the Levenberg-Marquardt algorithm. It is known to be one of the fastest and most robust algorithms for solving nonlinear equations. We include some heuristics in the algorithm.

The first useful observation is, that system VII.1 leads us more reliably to a Nash equilibrium, but not as fast as III.1. Because the functions in III.1 are smooth and less complicated than in VII.1. So we first take VII.1 to get an approximative solution with $tol_1 < \|F_{cf-RD}(x)\|^2$. Then we choose a smaller tolerance $tol_2 < tol_1$ and get a more precise solution by III.1 with tol_2.

The algorithm still needs starting points. Here we take the uniform distributed strategy with $x_k = \left(\frac{1}{m^k}, \dots, \frac{1}{m^k}\right)$ $\forall k \in K$ and all pure strategies, i.e., $x_k = e_h^k, \forall k \in K, \forall h \in A^k$. The algorithm will stop when the first equilibrium is reached. These points deliver satisfactory results, but also cover a proper area of the set of strategies.

Since Levenberg-Marquardt minimizes the sum of squares, high deviations have great effects. This may lead to the result, that the condition $r_h^k - r^k \leq 0$ is minimized far faster than the other conditions. That means that x becomes easily unfeasible: $x \notin \Phi$. Because of this observation we add factors c_r, c_x, c_s, to keep the balance between the conditions:

$$\begin{pmatrix} \phi(c_x x_1^1, c_r[r^1(x) - r_1^1(x)]) \\ \phi(c_x x_2^1, c_r[r^1(x) - r_2^1(x)]) \\ \dots \\ \phi(c_x x_{m^n}^n, c_r[r^n(x) - r_{m^n}^n(x)]) \\ c_s[1 - \sum\limits_{h \in A^1} x_h^1] \\ \dots \\ c_s[1 - \sum\limits_{h \in A^n} x_h^n] \end{pmatrix} = 0 \qquad \text{(VIII.1)}$$

Now we are capable of stating the algorithm (see algorithm 1).

Algorithm 1 *cf-RD*

```
Input: $c_r, c_x, c_s \in \mathbb{R}_+$, tolerances $tol_1 > tol_2 > 0$, $\epsilon > 0$
and a complementary function $\phi$.
for all $l = 0, \dots, \prod_{k \in K} m^k + 1$ do
    If $l = 0$, choose $x_0$ with $x_h^k = \frac{1}{m^k}$ $\forall k \in K, h \in A^k$.
    If $l > 0$, choose $x_l$ as a pure strategy, with $x_l \neq x_m$ $\forall m < l$
    Solve VII.1 with starting point $x_l$ and tolerance $tol_1$ by
    Levenberg-Marquardt. Set $x_l$ equal the solution.
    Solve III.1 with starting point $x_l$ and tolerance $tol_2$ by
    Levenberg-Marquardt. Set $x_l$ equal the solution.
    if $x_l$ is $\epsilon$-equilibrium then
        break
    end if
end for
```

The algorithm is implemented in Matlab®. We assume, that an implementation in C or C++ would be faster. The implementation in Matlab® was not optimized over and over again, especially the computation of the payoff function leaves room for improvement.

IX. COMPUTATIONAL RESULTS

The percentage of games for which we managed to find Nash equilibria with the use of the above approach with given accuracy $\epsilon = 0.001$ is presented in the corresponding columns of table II. The percentage is calculated for 100 games of each class that differs in the number of agents and actions. The games are generated with the use of Gamut [11] with uniformly distributed payoffs from interval $[-100, 100]$.

We compare our algorithm with a number of methods available through Gambit [18]. Gambit is a collection of software tools for solving finite, non cooperative games. We chose global Newton method [8] and simplicial subdivision [26] for comparison. In addition we compare the Matlab®-implementation *npg* [12] of [2]. *npg* solves the optimization problem from theorem 5, by sequential quadratic programming. We test *cf-RD* with all starting points, as listed in algorithm 1 and just with one starting point $x_h^k = \frac{1}{m^k}$, $\forall k \in K, h \in A^k$[2] The first value of the table shows the percentage of the games solved. The second value gives the

---

[2] The running times have not been compared, because of the platform dependent implementation.

TABLE II

RESULTS OF EXPERIMENTS

n	m^k	*gambit-gnm*	*gambit-simpdiv*	*cf-RD*	*cf-RD one start*	*npg*
2	2	(100,1.20)	(100,1)	(100,1)	(91,1)	(80,1)
2	3	(100,1.36)	(100,1)	(100,1)	(64,1)	(62,1)
2	5	(100,1.72)	(100,1)	(100,1)	(34,1)	(29,1)
2	7	(100,2.74)	(95,1)	(99,1)	(21,1)	(20,1)
2	10	(100,3.74)	(93,1)	(95,1)	(14,1)	(10,1)
3	2	(100,1.56)	(100,1)	(100,1)	(75,1)	(70,1)
3	3	(99,3.40)	(98,1)	(100,1)	(51,1)	(54,1)
3	5	(86,6.34)	(87,1)	(100,1)	(39,1)	(43,1)
3	7	(89,11.07)	(67,1)	(100,1)	(21,1)	(32,1)
5	2	(93,3.52)	(94,1)	(100,1)	(68,1)	(57,1)
5	3	(75,6.68)	(67,1)	(100,1)	(47,1)	(49,1)
7	2	(82,6.35)	(68,1)	(100,1)	(62,1)	(48,1)
7	3	(43,4.21)	(<10,1)	(100,1)	(52,1)	(40,1)

average number of the found Nash equilibria for the solved games.

The method developed in this work shows quite satisfactory results. It is worth noting that almost all tested games could be solved using different starting points.

In comparison to *gambit-simpdiv* and *npg* our method has the same or stronger properties. Especially the comparison to *npg* [2] is important, since both methods are based on theorem 3. But our approach is capable of formulating the problem as a system of equations, while *npg* only states the optimization problem of theorem 5 and needs n additional variables. It would be interesting to test the performance of SQP implementation (like in *npg*) for VII.1.

To the best of our knowledge, there are no algorithms, that compute Nash equilibria via optimization problems and show better results than our implementation of VII.1. We must also keep in mind, that there is still room for improvement in our implementation. An idea to improve VII.1, is to modify the system to get stronger properties. For example, we can take a smooth Fischer-Burmeister function and solve the corresponding system of equations by a homotopy with $t = \epsilon$. Starting in $t_{start} = 1$ and ending in $t_{end} = 0$. For a small enough t, the solutions would constitute ϵ-equilibria. However we assume, that an arbitrary starting point might end up in a local minimum instead of a Nash equilibrium.

CONCLUSION

In this paper we give a survey on optimization methods. We formulate necessary and sufficient conditions for Nash equilibria in normal games without increasing the dimension of the problem. We propose an algorithm based on the developed conditions for calculating Nash equilibria that showed quite satisfactory computational results.

REFERENCES

[1] N. Akchurina. *Multi-agent reinforcement learning algorithms*. PhD thesis, 2010.

[2] B. Chatterjee. An optimization formulation to compute Nash equilibrium in finite games. In *ICM2CS 2009. Proceeding of International Conference on Methods and Models in Computer Science*, pages 1–5, 2009.

[3] J.-S. Chen and S. Pan. A survey on SOC complementarity functions and solution methods for SOCPs and SOCCPs. *Pacific Journal of Optimization*, 8(1):33–74, 2012.

[4] R. S. Datta. Universality of Nash equilibria. *Math. Oper. Res.*, 28(3):424–432, July 2003.

[5] R. S. Datta. Finding all Nash equilibria of a finite game using polynomial algebra. *Economic Theory*, 42(1):55–96, 2010.

[6] A. Fiacco and G. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1990.

[7] R. Fletcher. *Practical methods of optimization*. Wiley-Interscience, New York, NY, USA, 2 edition, 1987.

[8] S. Govindan and R. Wilson. A global Newton method to compute Nash equilibria. *J. Economic Theory*, 110(1):65–86, 2003.

[9] P. T. Harker and J.-S. Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Math. Program.*, 48:161–220, 1990.

[10] P. Herings and R. Peeters. Homotopy methods to compute equilibria in game theory. *Economic Theory*, 42(1):119–156, January 2010.

[11] http://gamut.stanford.edu/.

[12] http://www.mathworks.de/matlabcentral/fileexchange/27837.

[13] C. E. Lemke and J. T. Howson. Equilibrium Points of Bimatrix Games. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):413–423, 1964.

[14] R. J. Lipton and E. Markakis. Nash equilibria via polynomial equations. In M. Farach-Colton, editor, *LATIN*, volume 2976 of *Lecture Notes in Computer Science*, pages 413–422. Springer, 2004.

[15] L. Mathiesen. An algorithm based on a sequence of linear complementarity problems applied to a Walrasian equilibrium model: An example. *Mathematical Programming*, 37:1–18, 1987.

[16] R. McKelvey. *A Liapunov Function for Nash Equilibria*. Social science working paper. Division of the Humanities and Social Sciences, California Institute of Technology, 1998.

[17] R. D. McKelvey and A. McLennan. Computation of equilibria in finite games. In H. M. Amman, D. A. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, volume 1, chapter 2, pages 87–142. Elsevier, 1996.

[18] R. D. Mckelvey, A. M. Mclennan, and T. L. Turocy. Gambit: Software Tools for Game Theory. Technical report, 2006.

[19] R. B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1997.

[20] J. Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, 1951.

[21] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[22] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994.

[23] J. Rosenmuller. On a generalization of the Lemke-Howson algorithm to noncooperative n-person games. *SIAM Journal on Applied Mathematics*, 21(1):73–79, 1971.

[24] H. Scarf and T. Hansen. *The Computation of Economic Equilibria*. Cowles Foundation for Research in Economics at Yale University. Monograph 24. Yale University Press, 1973.

[25] A. van den Elzen and D. Talman. Finding a Nash equilibrium in noncooperative n-person games by solving a sequence of linear stationary point problems. *Math. Meth. of OR*, 39(3):365–375, 1994.

[26] G. Van Der Laan, A. J. J. Talman, and L. Van Der Heyden. Simplicial Variable Dimension Algorithms for Solving the Nonlinear Complementarity Problem on a Product of Unit Simplices Using a General Labelling. *Mathematics of Operations Research*, 12(3):377–397, 1987.

[27] R. Wilson. Computing Equilibria of n-person Games. *Siam Journal on Applied Mathematics*, 21, 1971.