Decision Making in Manufacturing and Services
Vol. 9 • 2015 • No. 1 • pp. 63–78

Interval Methods for Computing Strong Nash Equilibria of Continuous Games

Bartłomiej Jacek Kubica*, Adam Woźniak*

Abstract. The problem of seeking strong Nash equilibria of a continuous game is considered. For some games, these points cannot be found analytically, only numerically. Interval methods provide us with an approach to rigorously verify the existence of equilibria in certain points. A proper algorithm is presented. We formulate and prove propositions, that give us features which have to be used by the algorithm (to the best knowledge of the authors, these propositions and properties are original). Parallelization of the algorithm is also considered, and numerical results are presented. As a particular example, we consider the game of "misanthropic individuals", a game, invented by the first author, that may have several strong Nash equilibria depending on the number of players. Our algorithm is able to localize and verify these equilibria.

Keywords: strong Nash equilibria, continuous games, interval computations, numerical game solving

Mathematics Subject Classification: 65G40, 65K99, 91A06, 91A35, 91B50

Revised: April 22, 2015

1. INTRODUCTION

Game theory tries to predict decisions and/or advise the decision makers on how to behave in a situation when several players (sometimes called "agents") have to choose their behavior (strategy; the i-th player chooses the strategy $x^i \in X_i$) that will also influence the others. Usually, we assume that each player tends to minimize his cost function (or maximize his utility) $q_i(x^1, \ldots, x^n)$.

So, each of the decision makers solves the following problem:

$$\begin{aligned} &\min q_i(x^1, \ldots, x^n) \\ &\text{s.t.} \\ &x^i \in X_i \end{aligned} \tag{1}$$

* Warsaw University of Technology, Institute of Control and Computation Engineering, Warsaw, Poland, e-mail: bkubica@elka.pw.edu.pl

DOI: http://dx.doi.org/10.7494/dmms.2015.9.1.63

What solution are they going to choose?

One of the oldest, most famous, and still widely-used concepts is the Nash equilibrium (Nash, 1950). It can be defined as a situation (an assignment of strategies to all players), when each player's strategy is optimal against those of the others.

Formally, the tuple $x^* = (x^{1*}, \ldots, x^{n*})$ is a Nash equilibrium, if:

$$(\forall i = 1, \ldots, n)\ (\forall x^i \in X_i)\ q_i(x^{1*}, \ldots, x^{i-1*}, x^i, x^{i+1*}, \ldots, x^{n*}) \geq q_i(x^{1*}, \ldots, x^{n*}) \quad (2)$$

We shall use a shorter notation, also: $\forall i = 1, \ldots, n \quad \forall x^i \quad q_i(x^{\setminus i*}, x^i) \geq q_i(x^{\setminus i*}, x^{i*})$.

Such points, however, have several drawbacks – both theoretical (rather strong assumptions about the players' knowledge and rationality) and practical (they can be Pareto-inefficient; i.e., it is possible to improve the outcome of one player without worsening the others' results (Miettinen, 1999)).

Hence, several "refinements" to the notion have been introduced, including the strong Nash equilibrium (SNE, for short), in particular; see (Aumann, 1959). For such points, not only none of the players can improve their performance by changing strategy, but also no *coalition* of players can improve the performance of all of its members by mutually deviating from the SNE. Formally:

$$(\forall I \subseteq \{1, \ldots, n\}) \quad (\forall x^I \in \times_{i \in I} X_i) \quad (\exists i \in I) \quad q_i(x^{\setminus I*}, x^I) \geq q_i(x^{\setminus I*}, x^{I*}) \quad (3)$$

Also, the notion of a k-SNE (or k-equilibrium) is sometimes encountered. Its definition is similar to ordinary SNE, but the coalition I in (3) can consist of k members at most. Obviously, a $(k+l)$-SNE is also a k-SNE (if $l > 0$) and, in particular, a SNE is also a k-SNE for any $k = 1, 2, \ldots, n$.

Strong Nash equilibria have been long thought to be too restrictive to be useful in practical situations, but they have received increased interest in recent years. Apparently, there exist some important games having SNE; e.g., some congestion games (Rosenthal, 1973), as pointed out in (Holzman and Law-Yone, 1997), or economies with multilateral environmental externalities; e.g., (Nessah and Tian, 2014). Existence of such a "strong" equilibrium may result in great stability of the system, as virtually no group of players will intend to change the *status quo*. Verifying the existence (or non-existence) of such a point and locating it may be very important. Consequently, the interest in computing SNE grows, also – see; e.g., (Gatti *et al.*, 2013; Nessah and Tian, 2014).

In this paper, we consider continuous single-stage games; i.e., the case, when the player's strategy is a tuple of numbers (vector) they choose from the given set, i.e. $x^i = \left(x^i_1, \ldots, x^i_{k_i}\right) \in X_i \subseteq \mathbb{R}^{k_i}$. Let us denote K_i – the set of components of the i-th player decision variable x^i, k_i – its size, K_I – the union of all K_i for $i \in I$ and $x = (x^1, \ldots, x^n) = (x^1_1, \ldots, x^1_{k_1}, x^2_1, \ldots, x^2_{k_2}, \ldots x^n_1, \ldots, x^n_{k_n})$. Also, we call Nash points (equilibria) that are not strong, "plain" Nash equilibria, to distinguish them from SNE.

Computing Nash equilibria – plain or strong ones – of such games is a hard task in general. We are going to present an approach based on interval analysis, extending our earlier algorithm for plain Nash points; see (Kubica and Woźniak, 2010, 2012).

2. BASICS OF INTERVAL COMPUTATIONS

Now, we shall define some basic notions of intervals and their arithmetic. The idea can be found in several textbooks; e.g., (Hansen and Walster, 2004; Jaulin *et al.*, 2001; Kearfott, 1996; Moore *et al.*, 2009; Shary, 2013).

We define the (closed) interval $[\underline{x}, \overline{x}]$ as a set $\{x \in \mathbb{R} \mid \underline{x} \leq x \leq \overline{x}\}$. Following (Kearfott *et al.*, 2010), we use boldface lowercase letters to denote interval variables; e.g., $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, and $\mathbb{IR}$ denotes the set of all real intervals.

We design arithmetic operations on intervals so that the following condition was fulfilled: if we have $\odot \in \{+, -, \cdot, /\}$, $a \in \mathbf{a}$, $b \in \boldsymbol{b}$, then $a \odot b \in \mathbf{a} \odot \boldsymbol{b}$. The actual formulae for arithmetic operations – see; e.g., (Hansen and Walster, 2004; Jaulin *et al.*, 2001; Kearfott, 1996) – are as follows:

$$
\begin{aligned}
[\underline{a}, \overline{a}] + [\underline{b}, \overline{b}] &= [\underline{a} + \underline{b}, \overline{a} + \overline{b}] \\
[\underline{a}, \overline{a}] - [\underline{b}, \overline{b}] &= [\underline{a} - \overline{b}, \overline{a} - \underline{b}] \\
[\underline{a}, \overline{a}] \cdot [\underline{b}, \overline{b}] &= [\min(\underline{ab}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{ab}), \max(\underline{ab}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{ab})] \\
[\underline{a}, \overline{a}] \;/\; [\underline{b}, \overline{b}] &= [\underline{a}, \overline{a}] \cdot \left[1 \,/\, \overline{b}, 1 \,/\, \underline{b}\right] , \qquad 0 \notin [\underline{b}, \overline{b}]
\end{aligned}
$$

The definition of interval vector $\mathbf{x}$, a subset of $\mathbb{R}^n$ is straightforward: $\mathbb{R}^n \supset \mathbf{x} = \mathbf{x}_1 \times \cdots \times \mathbf{x}_n$. Traditionally, interval vectors are called *boxes*.

Links between real and interval functions are set by the notion of an *inclusion function*: see; e.g., (Jaulin *et al.*, 2001); also called an *interval extension*; e.g., (Kearfott, 1996).

Definition 2.1. *A function* $\mathsf{f} \colon \mathbb{IR} \to \mathbb{IR}$ *is an* inclusion function *of* $f \colon \mathbb{R} \to \mathbb{R}$*, if for each interval* $\mathbf{x}$ *within the domain of* f *the following condition is satisfied:*

$$\{f(x) \mid x \in \mathbf{x}\} \subseteq \mathsf{f}(\mathbf{x})$$

The definition is analogous for functions $f \colon \mathbb{R}^n \to \mathbb{R}^m$*.*

When computing interval operations – either the ones above or computing the enclosure for a transcendental function – we can round the lower bound downward and the upper bound upward. This will result in an interval that will be overestimated, but will be *guaranteed to contain the true result of the real-number operation.*

3. NECESSARY CONDITIONS FOR A SNE

In interval global optimization, we use the Fritz–John conditions (Kearfott, 1996) to discard boxes that do not contain critical points. For unconstrained problems, we can discard all boxes, where the gradient of the objective cannot be equal to zero (unless *bound constraints* are active – see (Kearfott, 1996); i.e., when x belongs to the boundary of X). We are going to denote part of the gradient of a function f with respect to some of the variables $x_j = (x_1^j, \ldots, x_{k_j}^j)$ as:

$$\frac{\partial f}{\partial x^j} = \Big(\frac{\partial f}{\partial x_1^j}, \ldots, \frac{\partial f}{\partial x_{k_j}^j}\Big)$$

The vector equality $(y_1, \ldots, y_k) = 0$ is understood componentwise.

In (Kubica and Woźniak, 2010), we considered the necessary conditions of a Nash equilibrium and realized that – if no constraints are active – the point x has to satisfy the following conditions to be a Nash equilibrium:

$$\frac{\partial q_i(x)}{\partial x^i} = 0, i = 1, \ldots, n$$

What other conditions should a point satisfy to be a SNE?

Proposition 3.1. *(Necessary conditions for a 2-SNE) Consider a strategy profile x, such that no constraints are active for x (i.e., $x \in \operatorname{int} X$). Suppose, for two players i and j, we have $\frac{\partial q_i(x)}{\partial x^j} \neq 0$ and $\frac{\partial q_j(x)}{\partial x^i} \neq 0$. Then, x is not a 2-SNE.*

Interpretation

In a 2-SNE point, for no pair of players, it is possible for them to mutually improve each other's cost value – at least for one of them, their cost is minimized for the other's decision for x.

Proof. Suppose x is a 2-SNE of the game. From the definition, for each pair of players (i, j) the pair of their cost functions $\big(q_i(x), q_j(x)\big)$ has to be weakly non-dominated – see; e.g., (Nessah and Tian, 2014):

$$\nexists x' \in X \quad \big(q_i(x') < q_i(x) \text{ and } q_j(x') < q_j(x)\big)$$

Necessary conditions for the weak Pareto-optimality can be formulated as follows – see; e.g., (Miettinen, 1999) – there exist $u_1 \in [0, 1]$ and $u_2 \in [0, 1]$ such tha:

$$\begin{aligned} u_1 \cdot \frac{\partial q_i}{\partial x^i} + u_2 \cdot \frac{\partial q_j}{\partial x^i} &= 0 \\ u_1 \cdot \frac{\partial q_i}{\partial x^j} + u_2 \cdot \frac{\partial q_j}{\partial x^j} &= 0 \\ u_1 + u_2 &= 1 \end{aligned}$$

From the necessary conditions of any Nash equilibrium – see; e.g., (Kubica and Woźniak, 2010) – we know that $\frac{\partial q_i}{\partial x^i} = \frac{\partial q_j}{\partial x^j} = 0$. Thus, we obtain:

$$\begin{aligned} u_2 \cdot \frac{\partial q_j}{\partial x^i} &= 0 \\ u_1 \cdot \frac{\partial q_i}{\partial x^j} &= 0 \\ u_1 + u_2 &= 1 \end{aligned}$$

As $u_1 + u_2 = 1$, we cannot have $u_1 = u_2 = 0$. So, at most, one of the partial derivatives – $\frac{\partial q_j}{\partial x^i}$ or $\frac{\partial q_i}{\partial x^j}$ – has to be equal to zero. ■

Proposition 3.2. *(Necessary conditions for a 3-SNE) Consider a strategy profile x, such that no constraints are active for x (i.e., $x \in \operatorname{int} X$). Consider three players: i, j and k. A necessary condition for x to be a 3-SNE is that the following two conditions are satisfied:*

$$\begin{aligned} &\frac{\partial q_i(x)}{\partial x^j} = 0 \text{ or } \frac{\partial q_j(x)}{\partial x^k} = 0 \text{ or } \frac{\partial q_k(x)}{\partial x^i} = 0 \\ &\frac{\partial q_i(x)}{\partial x^k} = 0 \text{ or } \frac{\partial q_k(x)}{\partial x^j} = 0 \text{ or } \frac{\partial q_j(x)}{\partial x^i} = 0 \end{aligned} \tag{4}$$

Interpretation

For no trio of players, it is possible for them to mutually improve each other's cost value.

Proof. Suppose x is a 3-SNE of the game. Analogously to the previous proof, for each trio of players (i, j, k) the pair of their cost functions $\big(q_i(x), q_j(x), q_k(x)\big)$ has to be weakly non-dominated. Necessary conditions for weak Pareto-optimality can be formulated as follows in this case:

$$\begin{aligned} &u_1 \cdot \frac{\partial q_i}{\partial x^i} + u_2 \cdot \frac{\partial q_j}{\partial x^i} + u_3 \cdot \frac{\partial q_k}{\partial x^i} = 0 \\ &u_1 \cdot \frac{\partial q_i}{\partial x^j} + u_2 \cdot \frac{\partial q_j}{\partial x^j} + u_3 \cdot \frac{\partial q_k}{\partial x^j} = 0 \\ &u_1 \cdot \frac{\partial q_i}{\partial x^k} + u_2 \cdot \frac{\partial q_j}{\partial x^k} + u_3 \cdot \frac{\partial q_k}{\partial x^k} = 0 \\ &u_1 + u_2 + u_3 = 1 \end{aligned}$$

As earlier, we have $\dfrac{\partial q_i}{\partial x^i} = \dfrac{\partial q_j}{\partial x^j} = \dfrac{\partial q_k}{\partial x^k} = 0$, which reduces the above equations to:

$$\begin{aligned} &u_2 \cdot \frac{\partial q_j}{\partial x^i} + u_3 \cdot \frac{\partial q_k}{\partial x^i} = 0 \\ &u_1 \cdot \frac{\partial q_i}{\partial x^j} + u_3 \cdot \frac{\partial q_k}{\partial x^j} = 0 \\ &u_1 \cdot \frac{\partial q_i}{\partial x^k} + u_2 \cdot \frac{\partial q_j}{\partial x^k} = 0 \\ &u_1 + u_2 + u_3 = 1 \end{aligned} \tag{5}$$

while – from the 2-SNE's necessary conditions for each pair of players, we have:

$$\begin{aligned} &\frac{\partial q_i(x)}{\partial x^j} = 0 \text{ or } \frac{\partial q_j(x)}{\partial x^i} = 0 \\ &\frac{\partial q_i(x)}{\partial x^k} = 0 \text{ or } \frac{\partial q_k(x)}{\partial x^i} = 0 \\ &\frac{\partial q_j(x)}{\partial x^k} = 0 \text{ or } \frac{\partial q_k(x)}{\partial x^j} = 0 \end{aligned} \tag{6}$$

Conditions (6) themselves do not assure (4) – we can choose all three partial derivatives to be equal to zero from the same line of (4). But together with (5), we can imply the following:

- We can either choose all three pairs from various equations of (5) or two of them from the same equation.
- If two pairs in the same equations of (5) are equal to zero, they are from both equations of (4).
- If all three partial derivatives are chosen from separate equations of (5), the system (5) transforms into the system of the following three equations: $u_1 \cdot a = 0$, $u_2 \cdot b = 0$, $u_3 \cdot c = 0$, where a, b and c are *different* partial derivatives. As, at most one u_i can be equal to zero, it makes at least two of the derivatives a, b and c to be equal to zero.

In all cases, derivatives from both lines of (4) are equal to zero. ■

4. THE PROPOSED APPROACH

The general schema is going to be a specific variant of the branch-and-bound type (b&b-type) method described by the author in (Kubica, 2012, 2015). The algorithm is going to seek points satisfying the logical conditions defined by (3).

The input of the algorithms is the game; i.e., the number of players, formulae for cost functions of each of them, and domains of their control variables. The program results in two sets of boxes containing "verified" and "possible" strong Nash equilibria of the game.

To process boxes in the b&b-type algorithm, we have to use the necessary conditions investigated in Section 3. Please note that these conditions form an overdetermined system. There are methods to solve overdetermined systems – e.g., (Horacek and Hladik, 2013, 2014) – but in our case, another approach seems more appropriate. We have a system of N equations in N variables (necessary conditions for a Nash point) plus additional conditions that are *alternatives of equations*.

It seems reasonable to consider the first system separately. We use the following tools to solve it:

- a variant of the monotonicity test – Algorithm 2; see also (Kubica and Woźniak, 2010),
- a variant of the "concavity" test – Algorithm 4; e.g., (Kearfott, 1996),
- an interval Newton operator (see below).

Hence, the 2-SNE necessary conditions investigated in Proposition 3.1 are used in Algorithm 3. Conditions for k-SNE, $k \geq 3$ are not checked in the current implementation – it seems to be a costly procedure and unlikely to be very useful.

Conditions from the definition of SNE – equation (3) – are directly verified in the second phase of Algorithm 1.

The "concavity" test – Algorithm 4 – could more precisely be called the "non-convexity" test. It verifies whether the function can be convex on the box $\mathbf{x}$; i.e., if no component of the Hesse matrix is negative. If bound constraints can be active, the check is not performed, as even a function that is concave with respect to some of its variables can still have a minimum on the boundaries.

The general b&b-type algorithm is implemented by Algorithm 1.

Algorithm 1 The branch-and-bound-type method for seeking SNE

Require: $x^0, \mathsf{q}(\cdot), \varepsilon$
$L_{ver} = L_{pos} = L_{check} = L_{small} = \emptyset$
$\mathbf{x} = x^{(0)}$
loop
  $\mathbf{x}^{old} = \mathbf{x}$
  perform the monotonicity test (Algorithm 2) on ($\mathbf{x}$, $\mathbf{x}^{(0)}$, $\mathbf{x}^{old}$, q)
  perform the 2-SNE-monotonicity test (Algorithm 3) on ($\mathbf{x}$, $\mathbf{x}^{(0)}$, $\mathbf{x}^{old}$, q)
  perform the "concavity" test (Algorithm 4) on ($\mathbf{x}$, $\mathbf{x}^{(0)}$, $\mathbf{x}^{old}$, q)
  perform the Newton operator on ($\mathbf{x}$, $\mathbf{x}^{(0)}$, q)
  if ($\mathbf{x}$ was discarded, but not all q_i's are monotonous on it) **then**
    push (L_{check}, $\mathbf{x}^{old}$)
    discard $\mathbf{x}$
  else if (the tests resulted in two subboxes of $\mathbf{x}$: $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$) **then**
    $\mathbf{x} = \mathbf{x}^{(1)}$
    push (L, $\mathbf{x}^{(2)}$)
    cycle loop
  else if (wid $(\mathbf{x}) < \varepsilon$) **then**
    push (L_{small}, $\mathbf{x}$)
  end if
  if ($\mathbf{x}$ was discarded **or** $\mathbf{x}$ was stored) **then**
    $\mathbf{x}$ = pop (L)
    if (L was empty) **then**
      break
    end if
  else
    bisect ($\mathbf{x}$), obtaining $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$
    $\mathbf{x} = \mathbf{x}^{(1)}$
    push (L, $\mathbf{x}^{(2)}$)
  end if
end loop
{Second phase – verification}
for all ($\mathbf{x} \in L_{small}$) **do**
  check if another solution from L_{small} does not invalidate $\mathbf{x}$ (see Subsection 4.1)
  verify if no box from L_{check} contains a point that would invalidate $\mathbf{x}$
  put $\mathbf{x}$ to L_{ver}, L_{pos} or discard it, according to the results
end for
return L_{ver}, L_{pos}

Algorithm 2 The monotonicity test

Require: $\mathbf{x}, \mathbf{x}^{(0)}, \mathbf{x}^{old}, \mathbf{q}(\cdot)$
$n_{mon} = 0$
for $(i = 1, \ldots, n)$ **do**
  for $(k = 1, \ldots, k_i)$ **do**
    if $(\frac{\partial \mathbf{q}_i(\mathbf{x})}{\partial x_k^i} > 0)$ **then**
      if $(\underline{x}_k^i > \underline{x}_k^{i(0)})$ **then**
        increment n_{mon}
        break {the inner loop}
      else
        set $\overline{x}_k^i = \underline{x}_k^{i(0)}$
      end if
    else if $(\frac{\partial \mathbf{q}_i(\mathbf{x})}{\partial x_k^i} < 0)$ **then**
      if $(\overline{x}_k^i < \overline{x}_k^{i(0)})$ **then**
        increment n_{mon}
        break {the inner loop}
      else
        set $\underline{x}_k^i = \overline{x}_k^{i(0)}$
      end if
    end if
  end for
end for
if $(n_{mon} > 0)$ **then**
  if $(n_{mon} < n)$ **then**
    push $(L_{check}, \mathbf{x}^{old})$
  end if
  discard $\mathbf{x}$
end if

Algorithm 3 The 2-SNE monotonicity test

Require: $\mathbf{x}, \mathbf{x}^{(0)}, \mathbf{x}^{old}, \mathbf{q}(\cdot)$
for $(i = 1, \ldots, n)$ **do**
  for $(j = i + 1, \ldots, n)$ **do**
    ith_has_no_zero = jth_has_no_zero = **false**
    for $(k = 1, \ldots, k_j)$ **do**
      if $(\overline{x}_k^j < \overline{x}_k^{j(0)}$ **and** $\frac{\partial \mathbf{q}_i(\mathbf{x})}{\partial x_k^j} < 0)$ **or** $(\underline{x}_k^j > \underline{x}_k^{j(0)}$ **and** $\frac{\partial \mathbf{q}_i(\mathbf{x})}{\partial x_k^j} > 0)$ **then**
        ith_has_no_zero = **true**
      end if
    end for
    for $(k = 1, \ldots, k_i)$ **do**
      if $(\overline{x}_k^i < \overline{x}_k^{i(0)}$ **and** $\frac{\partial \mathbf{q}_j(\mathbf{x})}{\partial x_k^i} < 0)$ **or** $(\underline{x}_k^i > \underline{x}_k^{i(0)}$ **and** $\frac{\partial \mathbf{q}_j(\mathbf{x})}{\partial x_k^i} > 0)$ **then**
        jth_has_no_zero = **true**
      end if
    end for
    if (ith_has_no_zero **and** jth_has_no_zero) **then**
      discard $\mathbf{x}$
      return
    end if
  end for
end for

Algorithm 4 The "concavity" test

Require: $\mathbf{x}, \mathbf{x}^{(0)}, \mathbf{x}^{old}, \mathbf{q}(\cdot)$
1: $n_{conc} = 0$
2: **if** (**not** $\mathbf{x} \subset \operatorname{int} \mathbf{x}^{(0)}$) **then**
3: **return**
4: **end if**
5: **for** $(i = 1, \ldots, n)$ **do**
6: {check the Hesse matrix of $\mathbf{q}_i(\mathbf{x})$ with respect to x^i}
7: **if** $(\frac{\partial^2 \mathbf{q}_i(\mathbf{x})}{\partial (x_k^i)^2} < 0$ for some $k = 1, \ldots, k_i)$ **then**
8: increment n_{conc}
9: **end if**
10: **end for**
11: **if** $(n_{conc} > 0)$ **then**
12: **if** $(n_{conc} < n)$ **then**
13: push $(L_{check}, \mathbf{x}^{old})$
14: **end if**
15: discard $\mathbf{x}$
16: **end if**

As the Newton operator, we use the interval Gauss-Seidel operator with the inverse-midpoint preconditioner. We shall not present the code, as it is available in several textbooks; e.g., (Hansen and Walster, 2004; Kearfott, 1996; Moore *et al.*, 2009; Shary, 2013).

4.1. THE SECOND PHASE – VERIFICATION

Verification of the solutions obtained in the b&b-type algorithm is based on the following property:

Property 4.1. The point $x^* = (x^{1*}, \ldots, x^{n*})$ is a SNE, if $\forall x = (x^1, \ldots, x^n) \in X$:

$$\begin{aligned} &\Big((\exists i = 1, \ldots, n) \quad \big(q_i(x) \geq q_i(x^*)\big) \text{ and } (x^i \neq x^{i*})\Big) \text{ or} \\ &\Big((\forall i = 1, \ldots, n)\, (x^i = x^{i*})\Big) \end{aligned} \tag{7}$$

Proof. From (3), we infer that no coalition $I \subseteq \{1, \ldots, n\}$ can improve the objectives of all of its members. If x is the strategy profile when players cooperating in coalition I deviated from x^*, it means that, at least for one $i \in I$, the value of q_i has not improved, and for $i \notin I$, the players did not change their strategy; i.e., $x^i = x^{i*}$. ■

Proposition 4.1 means that, to invalidate x^* as a SNE, we have to find the strategy profile x such that:

$$\begin{aligned} &\Big((\forall i = 1, \ldots, n) \quad \big(q_i(x) < q_i(x^*)\big) \text{ or } (x^i = x^{i*})\Big) \text{ and} \\ &\Big((\exists i = 1, \ldots, n)\, (x^i \neq x^{i*})\Big) \end{aligned} \tag{8}$$

This condition can easilly be checked for all other points in the list L_{small}. Boxes in L_{check} are larger, and – in general – we need to bisect them, performing a "nested" b&b--type procedure to verify if they contain a point invalidating a specific solution or not.

4.2. PARALLELIZATION

The algorithm is parallelized using threads.

In the first phase, we have a shared queue L – guarded by two mutexes, as described in (Kubica and Woźniak, 2010) – and several threads processing boxes in parallel.

In the second phase, we verify different boxes from L_{small} in parallel; i.e., we parallelize the loop in line 31. The verification procedure in line 33 (that is a nested b&b-type algorithm) is not parallelized, as – being a recursive procedure – it would require more sophisticated parallelization methods; e.g., using Intel Threading Building Blocks; see (Kubica, 2012, 2015).

5. EXAMPLES OF GAMES TO SOLVE

We are going to present results from a few test problems. The first three have been discussed in (Ślepowrońska, 1996) and then considered in (Jauernig *et al.*, 2006; Kołodziej *et al.*, 2006; Kubica and Woźniak, 2010).

The first game has two players; each of them controls one real-valued decision variable.

$$\begin{aligned} &\min_{x_1} \left(q_1(x_1, x_2) = (x_1 - x_2 + 1)^2\right) \\ &\min_{x_2} \left(q_2(x_1, x_2) = (x_2 - x_1^2)^2 + (x_1 - 1)^2\right) \\ &x_1 \in [-1, 2.5],\ x_2 \in [-1, 3] \end{aligned} \tag{9}$$

This game has three Nash equilibria: $(2, 3)$ on the boundary and two in the interior of the feasible set: $(-0.618034, 0.381966)$ and $(1.618033, 2.618033)$. It is not known *a priori* if they are strong or not; our solver indicates that they are. Accuracy $\varepsilon = 10^{-7}$ is used for this game.

The second game is also a game of two players, but now each of them has 9 decision variables.

$$\begin{aligned} &\min_{x_1,\ldots,x_9} \big(q_1(x) = (x_1 - 1)^2 + (x_2 - 1)^2 + x_3^2 + (x_4 - 1)^2 + x_5^2 + (x_6 - 1)^2 \\ &+(x_7 - 1)^2 + x_8^2 + x_9^2 + x_{11}^2 + (x_{12} - 0.5)^2 + x_{13}^2 + (x_{16} + 0.5)^2 + (x_{18} - 1)^2\big) \\ &\min_{x_{10},\ldots,x_{18}} \big(q_2(x) = (x_{10} + 1)^2 + x_{11}^2 + (x_{12} - 1)^2 + x_{13}^2 + x_{14}^2 + (x_{15} + 1)^2 \\ &+(x_{17} - 1)^2 + x_{16}^2 + (x_{18} - 1)^2 + (x_2 - 0.5)^2 + x_3^2 + (x_4 - 0.5)^2 + (x_8 - 0.5)^2\big) \\ &x_i \in [-2, 2.4]^9 \qquad i = 1, 2 \end{aligned} \tag{10}$$

The game has one Nash equilibrium: $(1, 1, 0, 1, 0, 1, 1, 0, 0, -1, 0, 1, 0, 0, -1, 0, 1, 1)$. According to our results, this point seems to be a SNE. Accuracy $\varepsilon = 10^{-4}$ is used for our solver.

In the third game, we have three players with two decision variables each.

$$\begin{aligned}
&\min_{x_1,x_2}\left(q_1(x) = (x_1+1)^2(x_1-1)^2 + (x_2+1)^2(x_2-1)^2 + x_3x_4 + x_5x_6\right) \\
&\min_{x_3,x_4}\left(q_2(x) = (x_4-0.5)^2(x_4+1)^2 + (x_3+1)^2 + x_1x_2 + x_5x_6\right) \\
&\min_{x_5,x_6}\left(q_3(x) = (x_5+0.5)^2(x_5-1)^2 + (x_6-1)^2 + x_1x_2 + x_3x_4\right) \\
&x_i \in [-2, 2.4]^6 \qquad i = 1,2,3
\end{aligned} \tag{11}$$

This game has 16 Nash equilibria (they are listed in (Kołodziej *et al.*, 2006; Ślepowrońska, 1996); none of them is a SNE. We use accuracy parameter $\varepsilon = 10^{-7}$.

The fourth test problem is a game of two players; both have a single real-valued decision variable:

$$\begin{aligned}
&\min_{x_1}\left(q_1(x_1,x_2) = x_1^2 \cdot (x_1^2 - 3.75 \cdot x_1 + 3.25) + 1 + x_2^2\right) \\
&\min_{x_2}\left(q_2(x_1,x_2) = x_2^2 \cdot (x_2^2 - 3.75 \cdot x_2 + 3.25) + 1 + x_1^2\right) \\
&x_i \in [-3, 3.2], \quad i = 1,2
\end{aligned} \tag{12}$$

The game has a single Nash point at $(2,2)$, but it is not a SNE – mutually deviating from 2 to 0 is beneficial for both players (but the point $(0,0)$ is not a Nash point, at all!). Accuracy is set to $\varepsilon = 10^{-7}$.

5.1. THE GAME OF MISANTHROPIC INDIVIDUALS

This game has been proposed by the first author. Inspirations for it were congestion games (Rosenthal, 1973) and the game of dog and rabbit by Hugo Steinhaus (Steinhaus, 1960).

Consider n players, choosing their positions on a compact board – a two-dimensional domain for which we choose rectangle $D = [-3,3] \times [-2,2]$. Their objective is to be as far from the others as possible. Specifically, we assume that each of the players (let us give him the number $i = 1, \ldots, n$) maximizes, by choosing position $(x_i, y_i) \in D$, the following function:

$$q_i(x_i, y_i) = \sum_{j=1, j\neq i}^{n} \left((x_i - x_j)^2 + (y_i - y_j)^2\right) \tag{13}$$

Solutions of the game

Depending on n, the game can have different numbers of Nash equilibria – all or none of them being strong.

For two players, we have 4 Nash equilibrium points, each of them are strong. Their structures are obvious: one of the individuals is located in one of the four corners and the other one – diagonally opposite to him. It is clear that all of them are SNE – cooperation of both players cannot increase their distance in any way. This case is a "degenerate" case of a game, as both players maximize the same function – the (square of) the distance between them.

For three players we have 36 Nash equilibria: 24 with all three individuals located in different corners ($4 \times 3 \times 2$) and 12 with one of the three individuals in a corner and both others diagonally opposite to him (one of the 3 individuals $\times$ 4 corners). In all cases, one of the individuals has a better position than the two others. And actually, none of these solutions is strong – the two players with worse values can always collude to change their positions and improve their payoffs at the expense of the third player.

For four players, we have 36 Nash equilibria: 24 solutions with each individual in his own corner ($4 \times 3 \times 2$) and 12 solutions with two pairs of players in opposite corners. Counter-intuitively, formula (13) makes their values identical for both types of solutions. All of these 36 solutions are strong Nash equilibria.

For larger number of players, it is very difficult to analyze all possible solutions and their structures. In Section 6, we present; i.a., computational results for such situations (Tables 2 and 3).

6. NUMERICAL EXPERIMENTS

Numerical experiments were performed on a computer with four cores (allowing hyper-threading), namely, an Intel Core i7-3632QM with 2.2GHz clock. The machine ran under control of a 64-bit Manjaro 0.8.8 GNU/Linux operating system with the GCC 4.8.2, glibc 2.18 and the Linux kernel 3.10.22-1-MANJARO.

The solver is written in C++ and compiled using the GCC compiler. The C-XSC library (version 2.5.3) (C-XSC, 2013) was used for interval computations.

The parallelization was done using the threads of the C++11 standard. OpenBLAS 0.2.8 (OpenBLAS, 2013) was linked for BLAS operations.

6.1. RESULTS FOR PROBLEMS (9)–(11)

Computational results for these problems can be found in Table 1.

Table 1. *Computational results for the solver, with a single thread*

problem	(9)	(10)	(11)	(12)
cost fun. evals	26557	356776	0	238
gradient evals	6875	93914	0	134
Hesse matrix evals	204	182	609	162
bisections	49	45	101	29
deleted monot. test.	35	45	67	21
deleted strong mon.	0	0	35	1
deleted "conc."	0	0	0	3
deleted Newton	0	0	0	7
boxes after 1^{st} ph.	3	1	0	3
possibly dominating	41	45	102	32
deleted 2^{nd} phase	0	0	0	3
possible solutions	3	1	0	0
verified solutions	0	0	0	0
time (milisec.)	491	2358	459	461

6.2. RESULTS FOR THE GAME OF MISANTHROPIC INDIVIDUALS

We present results for computing SNE and plain Nash equilibria – Tables 2 and 3, respectively. Accuracy $\varepsilon = 10^{-8}$ is set in all cases.

Table 2. *Computational results for the solver, with four threads*

players number	2	3	4	5	6	7
cost fun. evals	7616	1164	4853056	70235	5576803158	1519735
gradient evals	0	0	728800	0	1210016856	0
Hesse matrix evals	3774	18141	71164	300555	1136634	4677113
bisections	943	3023	8895	30055	94719	334079
deleted monot. test.	0	0	220	168	256	1536
deleted strong mon.	0	0	0	0	0	0
deleted "conc."	928	2960	8640	28864	90368	316160
deleted Newton	0	0	0	0	0	0
boxes after 1^{st} ph.	16	64	256	1024	4096	16384
possibly dominating	944	3280	10304	41972	133120	516864
deleted 2^{nd} phase	12	64	220	1024	3696	16384
possible solutions	0	0	36	0	400	0
verified solutions	4	0	0	0	0	0
time (sec.)	0.452	0.555	4.442	5.577	5221	189

Table 3. *Computational results for computing plain Nash equilibria, using four threads*

players number	2	3	4	5	6	7
cost fun. evals	5196	47335	47602	685225	1111178	14443406
gradient evals	0	0	0	0	0	0
Hesse matrix evals	3774	18141	71164	300555	1136634	4677113
bisections	943	3023	8895	30055	94719	334079
deleted monot. test.	0	0	0	168	256	1536
deleted "conc."	928	2960	8640	28864	90368	316160
deleted Newton	0	0	0	0	0	0
boxes after 1^{st} ph.	16	64	256	1024	4096	16384
possibly dominating	944	3280	10304	41972	133120	516864
deleted 2^{nd} phase	12	28	220	624	3696	11484
possible solutions	0	32	0	336	0	4000
verified solutions	4	4	36	64	400	900
time (sec.)	0.474	0.581	1.220	7.296	37	483

7. ANALYSIS OF THE RESULTS

The algorithm finds the SNE in all cases, but it is rarely able to verify them. The conditions are a bit complicated to be verified rigorously – actually, the verification was successful in one case only – and a very specific one (the game of misanthropic individuals, $n = 2$).

The solver finds the solution for problem (11) very quickly. This is because, for this game, all 16 Nash equilibrium points can be verified *not* to be SNE early – in the first phase (see the row "boxes after 1st ph."), using the *2-SNE monotonicity test* (Algorithm 3). See Table 1 for specific results.

In the game of misanthropic individuals – problem (13) – for some n's, the number of gradient evaluations is equal to zero. This usually happens when the number of points to verify in the second phase is equal to zero.

Gradients are computed in two cases:

- in the interval Newton operator, verifying first-order conditions for the Nash equilibria,
- in the second phase – also, in the Newton operator, but now verifying the inequality that $q_i(\mathbf{x})$ is lower than the verified value.

For the game of misanthropic individuals, cost functions q_i are concave, so we never apply the Newton operator in the first phase. Nor we do in the second phase, if there are no solutions to verify (in the case, we simply do nothing in the second phase).

It is worth noting how the computational effort changes with n for the game of misanthropic individuals. For odd numbers of players ($n = 3, 5, 7$), the effort of finding all strong Nash equilibria is particularly low. The reason is simple – there is no SNE (this hypothesis has been verified by numerical experiments for $n = 3, 5, 7$; we haven't proven it for other values of n, but it seems plausible) and all possible solutions are quickly discarded by comparisons with other possible solutions (see Subsection 4.1). The time-consuming nested branch-and-bound type procedure does not have to be executed at all. Because of this, the solver for SNE is more efficient than for plain Nash equilibria for these points; see Tables 2 and 3.

For $n = 6$, the solver finds 400 points that are strong Nash equilibria, probably. Isolating so many solutions of the game is possible thanks to the virtues of the interval calculus: see; e.g., (Kubica, 2015; Shary, 2013).

8. CONCLUSIONS

We presented an interval solver able to compute strong Nash equilibrium points of continuous games. We tested it on a few test problems, showing its usefulness. Parallelization using threads allows us to handle relatively difficult problems. For one of the examples, it allowed us to isolate 400 equilibrium points.

Also, a specific test problem has been proposed – the game of misanthropic individuals; a continuous game with an arbitrary number of players, having various numbers of plain and strong Nash equilibria, depending on the number of players. It seems an interesting benchmark, due to its complex and counter-intuitive properties.

ACKNOWLEDGMENTS

The authors are grateful to the reviewers who helped to improve this presentation.

REFERENCES

Aumann, R.J., 1959. Acceptable points in general cooperative games. In: A.W. Tuckar, R.D. Luce (eds), *Contributions to the Theory of Games IV*, Princeton University Press.

C-XSC, 2013. C++ eXtended Scientific Computing library, `http://www.xsc.de`.

Gatti, N., Rocco, M., Sandholm, T., 2013. On the verification and computation of strong Nash equilibrium. *Proceedings of 2013 international conference on Autonomous agents and multi-user systems*, International Foundation for Autonomous Agents and Multiagent Systems.

Hansen, E., Walster, W., 2004. *Global Optimization Using Interval Analysis*, Marcel Dekker, New York.

Holzman, R., Law-Yone, N., 1997. Strong equilibrium in congestion games. *Games and Economic Behavior*, **21**(1), pp. 85–101.

Horacek, J., Hladik, M., 2013. Computing enclosures of overdetermined interval linear systems. *Reliable Computing*, **19**(3), pp. 142–155.

Horacek, J., Hladik, M., 2014. Subsquares approach – a simple scheme for solving overdetermined interval linear systems. *Lecture Notes in Computer Science*, **8385**, pp. 613–622. PPAM 2013 (10^{th} International Conference on Parallel Processing and Applied Mathematics) Proceedings.

Jauernig, K., Kołodziej, J., Stysło, M., 2006. HGS-Nash evolutionary strategy as an effective method of detecting the Nash equilibria in n-person non-cooperative games. *Proceedings of KAEiOG'06, Murzasichle*, pp. 171–178.

Jaulin, L., Kieffer, M., Didrit, O., Walter, E., 2001. *Applied Interval Analysis*. Springer, London.

Kearfott, R.B., 1996. *Rigorous Global Search: Continuous Problems*. Kluwer, Dordrecht.

Kearfott, R.B., Nakao, M.T., Neumaier, A., Rump, S.M., Shary, S.P., van Hentenryck, P., 2010. Standardized notation in interval analysis. *Vychislennyie tiehnologii (Computational technologies)*, **15**(1) pp. 7–13.

Kołodziej, J., Jauernig, K., Cieślar, A., 2006. HGS-Nash strategy as the decision-making method for water resource systems with external disagreement of interests. *Proceedings of PARELEC'06*, Wrocław, pp. 313–318.

Kubica, B.J., 2012. A class of problems that can be solved using interval algorithms. *Computing*, **94**, pp. 271–280. SCAN 2010 (14^{th} GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics) Proceedings.

Kubica, B.J., 2015. Interval methods for solving various kinds of quantified nonlinear problems. *Lecture Notes in Computer Science*. SCAN 2014 Proceedings, submitted.

Kubica, B.J., Woźniak, A., 2010. An interval method for seeking the Nash equilibria of non-cooperative games. *Lecture Notes in Computer Science*, **6068**, pp. 446–455. PPAM 2009 Proceedings.

Kubica, B.J., Woźniak, A., 2012. Applying an interval method for a four agent economy analysis. *Lecture Notes in Computer Science*, **7204**, pp. 477–483. PPAM 2011 (9^{th} International Conference on Parallel Processing and Applied Mathematics) Proceedings.

Miettinen, K., 1999. *Nonlinear multiobjective optimization*, Vol. 12. Kluwer Academic Publishers, Dordrecht.

Moore, R.E., Kearfott, R.B., Cloud, M.J., 2009. *Introduction to Interval Analysis*. SIAM, Philadelphia.

Nash, J.F., 1950. Equilibrium points in n-person games. *Proceedings of National Association of Science*, **36**, pp. 48–49.

Nessah, R., Tian, G., 2014. On the existence of strong Nash equilibria. *Journal of Mathematical Analysis and Applications*, **414**(2), pp. 871–885.

OpenBLAS, 2013. OpenBLAS library, `http://xianyi.github.com/OpenBLAS/`.

Rosenthal, R.W., 1973. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, **2**(1), pp. 65–67.

Sharyy, S.P., 2015. Konechnomernyy interval'nyy analiz [Finite-dimensional Interval Analysis], Izdatel'stvo XYZ, Novosibirsk. Electronic book: `http://www.nsc.ru/interval/Library/InteBooks/SharyBook.pdf` (accessed 2014.05.15)

Ślepowrońska, K., 1996. A parallel algorithm for finding Nash equilibria. Master's thesis (in Polish) under supervision of A. Woźniak, ICCE WUT.

Steinhaus, H., 1960. Definitions for a theory of games and pursuit. *Naval Research Logistics Quarterly*, **7**, pp. 105–107.