Relaxation algorithms to find Nash equilibria with economic applications

Jacek B. Krawczyk [a,*] and Stanislav Uryasev [b]

[a] *Commerce and Administration Faculty, Victoria University of Wellington, PO Box 600, Wellington, New Zealand*
E-mail: jacek.krawczyk@vuw.ac.nz

[b] *Department of Industrial and Systems Engineering, University of Florida, 474 Weil Hall, Gainesville, FL 32611-6595, USA*
E-mail: uryasev@ise.ufl.edu

Received July 1999; revised September 1999

Recent theoretical studies have shown that a relaxation algorithm can be used to find noncooperative equilibria of synchronous infinite games with nonlinear payoff functions and coupled constraints. In this study, we introduce an improvement to the algorithm, such as the steepest-descent step-size control, for which the convergence of the algorithm is proved. The algorithm is then tested on several economic applications. In particular, a River Basin Pollution problem is considered where coupled environmental constraints are crucial for the relevant model definition. Numerical runs demonstrate fast convergence of the algorithm for a wide range of parameters.

Keywords: computational economics, Nash equilibrium, coupled constraints, relaxation algorithm

1. Introduction

The focus of this paper is on studying computational-economics methods to find Nash equilibria in infinite games of varying complexity. Game theory problems are renowned for being hard to solve, especially multi-player, non-zero sum, and dynamic games. Such games are even more difficult to solve if the payoff functions are non-differentiable and/or the constraints are coupled.[1] Games of that kind are typical of environmental economics[2] and we envisage this to be an important application area of the methods studied in this paper.

We use the optimum response functions [2,3,12,17,23] to construct an algorithm to search for a Nash equilibrium point. This is an attractive approach because it relies on minimization of a multivariate function; a well studied topic. At each iteration of such an algorithm the multivariate Nikaido–Isoda function [8] is minimized using a standard nonlinear programming routine. The paper [23] has set the foundations for a sequential improvement of the Nikaido–Isoda function through the *relaxation algorithm*. It was proved that the relaxation algorithm converges to a Nash equilibrium for a wide class of games that includes non-differential payoffs and coupled constraints.

An important variation of the relaxation algorithm is considered in this paper. We investigate a steepest-descent step-size control of the relaxation algorithm. The convergence of this modified relaxation algorithm is formally proved and numerically tested.

In this paper we also conduct numerical testing of the relaxation algorithm [23] and of its modified counterpart. In particular, we solve a River Basin Pollution game [6], which includes coupled constraints. The numerical experiments reported here were conducted using customarily developed software in the MATHEMATICA and MATLAB programming environments. Some of the material and examples provided in this paper are based on the working paper [4].

The River Basin Pollution game solved in this paper can be treated as a generic example of how to ensure agents' environmental compliance. Lagrange multipliers are computed as a byproduct of a constrained equilibrium and can be used as Pigouvian nominal taxes [18] to compel agents to limit their polluting activities. This approach to environmental modeling and management can be used in situations in which a local legislator can be identified as an elective representative of different interest groups, see [10].

Other approaches to reduce pollution are also popular. Charging a fixed price for each unit of pollution is a possibility, see, e.g., [7]. One may also target economic variables (e.g., supply, demand, transportation) with penalties for failure to comply with regulation, see [14]. Another approach is to build a model that includes marketable pollution permits. To formulate such a model, and find an equilibrium, variational inequalities are applied in [13].[3]

* Research supported by VUW GSBGM and IGC grants.

[1] We use this term defined by Rosen in [19] to describe a set of relations that constrain players' joint actions.

[2] For a non-differentiable environmental game see [11]; for a coupled constraint game solved via Rosen's algorithm see [6].

[3] For more on how this approach is used for the description and solution of equilibrium problems arising in economics and operations research see, for example, [5,15].

The organization of this paper is as follows. Section 2 has a tutorial character and provides an introduction to some basic concepts. Section 3 presents the relaxation algorithm [23]. In section 4, the convergence proof for the relaxation algorithm with an optimized step size is given. Section 5 provides tests and examples of how the algorithm works. The paper ends with conclusions in section 6. All definitions, theorems, and such are numbered consecutively in each section.

2. Definitions and concepts

There are $i = 1, \ldots, n$ players participating in a game. Each player can take an individual action, which is represented by a vector x_i in the Euclidean space $\mathbb{R}^{m_i}$. All players together can take a collective action, which is a combined vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_n}$. We adopt the convention that one player's action has a subscript (e.g., x_i), and a collective action is in boldface (e.g., $\mathbf{x}$). Let us denote by $X_i \subseteq \mathbb{R}^{m_i}$ an action set of player i and use $\phi_i : X_i \to \mathbb{R}$ for his payoff function. Denote the collective action set by X. By definition, $X \subseteq X_i \times \cdots \times X_n \subseteq \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_n} = \mathbb{R}^m$. Let $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ be elements of the collective action set $X_1 \times \cdots \times X_n$. An element

$$(y_i|\mathbf{x}) \equiv (x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$$

of the collective action set can be interpreted as a collection of actions when the ith agent "tries" y_i while the remaining agents are playing $x_j,\ j = 1, 2, \ldots, i-1, i+1, \ldots, n$.

Definition 2.1. Let $X \subseteq X_i \times \cdots \times X_n \subseteq \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_n} = \mathbb{R}^m$ be the collective action set, and the functions $\phi_i : X_i \to \mathbb{R}$ be the payoff functions of players $i = 1, \ldots, n$. A point $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ is called the *Nash equilibrium point*,[4] if, for each i,

$$\phi_i(\mathbf{x}^*) = \max_{(x_i|\mathbf{x}^*) \in X} \phi_i(x_i|\mathbf{x}^*). \tag{1}$$

Now, we introduce the Nikaido–Isoda function [8].

Definition 2.2. Let ϕ_i be the payoff function of player i. Then the *Nikaido–Isoda function* $\Psi : (X_1 \times \cdots \times X_n) \times (X_1 \times \cdots \times X_n) \to \mathbb{R}$ is defined as

$$\Psi(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \left[\phi_i(y_i|\mathbf{x}) - \phi_i(\mathbf{x})\right]. \tag{2}$$

It follows from the definition of the Nikaido–Isoda function that

$$\Psi(\mathbf{x}, \mathbf{x}) \equiv 0. \tag{3}$$

Each summand of the Nikaido–Isoda function can be thought of as the change in the payoff of a player when his action changes from x_i to y_i while all other players continue to play according to $\mathbf{x}$. The function thus represents the sum of these changes in payoff functions. Note that the *maximum* value that this function can take by changing $\mathbf{y}$, for a given $\mathbf{x}$, is always nonnegative, owing to (3). Also, the function is nonpositive for all feasible $\mathbf{y}$ when $\mathbf{x}^*$ is a Nash equilibrium point, since, at an equilibrium, no player can make a unilateral improvement to their payoff, and so each summand in this case can be zero at most.

From here, we reach the conclusion that when the Nikaido–Isoda function cannot be made (significantly) positive for a given $\mathbf{x}$, we have (approximately) reached the Nash equilibrium point. We use this observation in constructing a termination condition for our algorithm; that is, we choose an ε such that, when $\max_{\mathbf{y} \in X} \Psi(\mathbf{x}^s, \mathbf{y}) < \varepsilon$, we have achieved the equilibrium $\mathbf{x}^s$ to a sufficient degree of precision.

Definition 2.3. An element $\mathbf{x}^* \in X$ is referred to as a *Nash normalized equilibrium point*[5] if

$$\max_{\mathbf{y} \in X} \Psi(\mathbf{x}^*, \mathbf{y}) = 0. \tag{4}$$

The two following lemmas establish a relationship between Nash equilibrium and Nash normalized equilibrium points:

Lemma 2.4 [1]. A Nash normalized equilibrium point is also a Nash equilibrium point.

Lemma 2.5 [1]. A Nash equilibrium point is a Nash normalized equilibrium point if the collective action set X satisfies

$$X = X_1 \times \cdots \times X_n, \qquad X_i \subset \mathbb{R}^{m_i}. \tag{5}$$

An algorithm which uses the Nikaido–Isoda function to compute the Nash normalized equilibrium will be presented in the next section. Because of lemma 2.4, the computed point is obviously a Nash equilibrium. Here we note that at each iteration of the algorithm we wish to move towards a point which is an "improvement" on the one that we are at. To this end, let us put forward the following definition.

Definition 2.6. The *optimum response function* (possibly multi-valued) at the point $\mathbf{x}$ is

$$Z(\mathbf{x}) = \arg\max_{\mathbf{y} \in X} \Psi(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, Z(\mathbf{x}) \in X. \tag{6}$$

In brief terms, this function returns the set of players' actions whereby they all attempt to unilaterally maximize their payoffs.

We now introduce some more technical definitions to be used in the convergence theorems.

[4] Notice that this definition allows for coupled constraint equilibria, see [19]. We compute an equilibrium of this kind in section 5.3.

[5] We follow Aubin's terminology [1]. Notice that Rosen [19] also defines a *normalized equilibrium*, which has a different meaning.

Definition 2.7 [16,21]. Let X be a convex subset of the Euclidean space $\mathbb{R}^m$. A continuous function $f: X \to \mathbb{R}$ is called *weakly convex* on X if for all $\mathbf{x} \in X$, $\mathbf{y} \in X$, $0 \leqslant \alpha \leqslant 1$ the following inequality[6] holds:

$$\alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y}) \geqslant f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) + \alpha(1-\alpha) r(\mathbf{x},\mathbf{y}),$$

where the remainder $r: X \times X \to \mathbb{R}$ satisfies

$$\frac{r(\mathbf{x},\mathbf{y})}{\|\mathbf{x}-\mathbf{y}\|} \to 0, \quad \text{as } \mathbf{x} \to \mathbf{z},\ \mathbf{y} \to \mathbf{z}, \tag{7}$$

for all $\mathbf{z} \in X$.

Definition 2.8. A function $f(\mathbf{x})$ is called *weakly concave* on X if the function $-f(\mathbf{x})$ is weakly convex on X.

Let $\Psi: X \times X \to \mathbb{R}$ be a function defined on a product $X \times X$, where X is a convex closed subset of the Euclidean space $\mathbb{R}^m$. Further, we consider that $\Psi(\mathbf{x},\mathbf{z})$ is weakly convex on X with respect to the first argument, i.e.,

$$\alpha\Psi(\mathbf{x},\mathbf{z}) + (1-\alpha)\Psi(\mathbf{y},\mathbf{z}) \geqslant \Psi(\alpha\mathbf{x} + (1-\alpha)\mathbf{y},\mathbf{z}) + \alpha(1-\alpha) r_{\mathbf{z}}(\mathbf{x},\mathbf{y}) \tag{8}$$

for all $\mathbf{x},\mathbf{y},\mathbf{z} \in X$, $0 \leqslant \alpha \leqslant 1$, and

$$\frac{r_{\mathbf{z}}(\mathbf{x},\mathbf{y})}{\|\mathbf{x}-\mathbf{y}\|} \to 0, \quad \text{as } \|\mathbf{x}-\mathbf{y}\| \to 0, \text{ for all } \mathbf{z} \in X.$$

We suppose that the function $\Psi(\mathbf{z},\mathbf{y})$ is weakly concave with respect to the second argument on X, i.e.,

$$\alpha\Psi(\mathbf{z},\mathbf{x}) + (1-\alpha)\Psi(\mathbf{z},\mathbf{y}) \leqslant \Psi(\mathbf{z},\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) + \alpha(1-\alpha)\mu_{\mathbf{z}}(\mathbf{x},\mathbf{y}) \tag{9}$$

for all $\mathbf{x},\mathbf{y},\mathbf{z} \in X$, $0 \leqslant \alpha \leqslant 1$, and also

$$\frac{\mu_{\mathbf{z}}(\mathbf{x},\mathbf{y})}{\|\mathbf{x}-\mathbf{y}\|} \to 0 \quad \text{as } \|\mathbf{x}-\mathbf{y}\| \to 0, \text{ for all } \mathbf{z} \in X.$$

Definition 2.9. The function $\Psi(\mathbf{x},\mathbf{y})$ is referred to as *weakly convex-concave*, if it satisfies conditions (8) and (9).

The family of weakly convex-concave functions includes the family of smooth functions [16] as well as many non-differentiable functions.

We now present an elementary example to illustrate these definitions.

Example 2.10. Let us consider a convex quadratic function $f: \mathbb{R} \to \mathbb{R}$

$$f(x) = x^2.$$

This function is both weakly convex and weakly concave.

[6] Recall that for a function to be "just" *convex* we require

$$\alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y}) \geqslant f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}).$$

This function is convex, i.e.,

$$\alpha f(x) + (1-\alpha) f(y) \geqslant f(\alpha x + (1-\alpha) y).$$

Consequently, the function $f(x)$ is weakly convex with $r(x,y) = 0$.

To show that the function $f(x)$ is weakly concave, we must find an $\mu(x,y)$ such that, for all $x,y \in \mathbb{R}$ and $\alpha \in [0,1]$,

$$\alpha f(x) + (1-\alpha) f(y) \leqslant f(\alpha x + (1-\alpha) y) + \alpha(1-\alpha)\mu(x,y).$$

That is, if and only if

$$\begin{aligned} & \alpha x^2 + (1-\alpha) y^2 \leqslant (\alpha x + (1-\alpha) y)^2 + \alpha(1-\alpha)\mu(x,y) \\ \Leftrightarrow\ & \alpha x^2 + (1-\alpha) y^2 \leqslant \alpha^2 x^2 + (1-\alpha)^2 y^2 + 2\alpha(1-\alpha) xy + \alpha(1-\alpha)\mu(x,y) \\ \Leftrightarrow\ & \alpha(1-\alpha) x^2 + \alpha(1-\alpha) y^2 - 2\alpha(1-\alpha) xy \leqslant \alpha(1-\alpha)\mu(x,y) \\ \Leftrightarrow\ & (x-y)^2 \leqslant \mu(x,y). \end{aligned}$$

So it is sufficient to select $\mu(x,y) = (x-y)^2$. Also, check that

$$\frac{\mu(x,y)}{\|x-y\|} = \frac{(x-y)^2}{|x-y|} = |x-y| \to 0, \quad \text{as } |x-y| \to 0.$$

So we conclude that $f(x) = x^2$ is weakly concave. (However, as this is a one-variable function, it cannot be weakly convex-concave.)

The functions $r_{\mathbf{z}}(\mathbf{x},\mathbf{y})$, $\mu_{\mathbf{z}}(\mathbf{x},\mathbf{y})$ were introduced with the concept of weak convex-concavity. In the case of $\Psi(\mathbf{x},\mathbf{y})$ being a twice continuously differentiable function with respect to both arguments on $X \times X$, the residual terms satisfy (see [21])

$$r_{\mathbf{y}}(\mathbf{x},\mathbf{y}) = \frac{1}{2}\langle A(\mathbf{x},\mathbf{x})(\mathbf{x}-\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle + o_1(\|\mathbf{x}-\mathbf{y}\|^2) \tag{10}$$

and

$$\mu_{\mathbf{x}}(\mathbf{y},\mathbf{x}) = \frac{1}{2}\langle B(\mathbf{x},\mathbf{x})(\mathbf{x}-\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle + o_2(\|\mathbf{x}-\mathbf{y}\|^2), \tag{11}$$

where $A(\mathbf{x},\mathbf{x}) = \Psi_{\mathbf{xx}}(\mathbf{x},\mathbf{y})|_{\mathbf{y}=\mathbf{x}}$ is the Hessian of the Nikaido–Isoda function with respect to the first argument and $B(\mathbf{x},\mathbf{x}) = \Psi_{\mathbf{yy}}(\mathbf{x},\mathbf{y})|_{\mathbf{y}=\mathbf{x}}$ is the Hessian of the Nikaido–Isoda function with respect to the second argument, both evaluated at $\mathbf{y} = \mathbf{x}$. Moreover, if the function $\Psi(\mathbf{x},\mathbf{y})$ is convex with respect to $\mathbf{x}$, then $o_1(\|\mathbf{x}-\mathbf{y}\|^2) = 0$, and if the function $\Psi(\mathbf{x},\mathbf{y})$ is concave with respect to $\mathbf{y}$ then $o_2(\|\mathbf{x}-\mathbf{y}\|^2) = 0$. These observations simplify evaluation of the remainder terms, which will be needed for assessment of the convergence conditions of the relaxation algorithms considered in the next two sections.

The key convergence condition is that the difference of the residual terms (weakly) dominates a strictly increasing function

$$r_{\mathbf{y}}(\mathbf{x},\mathbf{y}) - \mu_{\mathbf{x}}(\mathbf{y},\mathbf{x}) \geqslant \beta(\|\mathbf{x}-\mathbf{y}\|), \quad \mathbf{x},\mathbf{y} \in X, \tag{12}$$

where β is the strictly increasing function (i.e., $\beta(t_2) > \beta(t_1)$ if $t_2 > t_1$) and $\beta(0) = 0$. This relationship implies[7] existence and uniqueness of the normalized Nash equilibrium point on a convex compact set X.

If the function $\Psi(\mathbf{x}, \mathbf{y})$ is convex-concave, then (see [21])

$$r_{\mathbf{y}}(\mathbf{x}, \mathbf{y}) - \mu_{\mathbf{x}}(\mathbf{y}, \mathbf{x}) = \langle Q(\mathbf{x}, \mathbf{x}), \mathbf{x} - \mathbf{y} \rangle, \qquad (13)$$

where

$$Q(\mathbf{x}, \mathbf{x}) = A(\mathbf{x}, \mathbf{x}) - B(\mathbf{x}, \mathbf{x}) \\ = \Psi_{\mathbf{xx}}(\mathbf{x}, \mathbf{y})|_{\mathbf{y}=\mathbf{x}} - \Psi_{\mathbf{yy}}(\mathbf{x}, \mathbf{y})|_{\mathbf{y}=\mathbf{x}}.$$

The equality (13) implies that the convergence condition (12) follows from the strict positive-definiteness of the matrix $Q(\mathbf{x}, \mathbf{x})$, i.e.,

$$\langle Q(\mathbf{x}, \mathbf{x}), \mathbf{x} - \mathbf{y} \rangle \geqslant \nu ||\mathbf{x} - \mathbf{y}||^2, \qquad (14)$$

where ν is a positive constant. Also, positive-definiteness of the matrix $Q(\mathbf{x}, \mathbf{x})$ implies existence and uniqueness of the normalized Nash equilibrium point.

3. The relaxation algorithm

3.1. Statement of the algorithm

Suppose we wish to find a Nash equilibrium of a game and we have some initial estimate of it, say $\mathbf{x}^0$, and $Z(\mathbf{x})$ single-valued. The relaxation algorithm is given by the following formula:

$$\mathbf{x}^{s+1} = (1 - \alpha_s)\mathbf{x}^s + \alpha_s Z(\mathbf{x}^s), \quad s = 0, 1, 2, \ldots, \qquad (15)$$

where $0 < \alpha_s \leqslant 1$. The iterate at step $s + 1$ is constructed as a weighted average of the improvement point $Z(\mathbf{x}^s)$ and the current point $\mathbf{x}^s$. This averaging ensures convergence of the algorithm under certain conditions, as stated in the following theorems 3.1 and 4.2.

It is interesting to note that we can consider the algorithm as either performing a static optimization or calculating successive actions of players in convergence to an equilibrium in a real time process. If all payoffs are known to us, we can directly find the Nash equilibrium using the relaxation algorithm. However, if we only have access to one player's payoff function and all players' past actions, then at each stage in the real time process we choose the optimum response for that player, assuming that the other players will play as they had in the previous period. In this way, convergence to the Nash normalized equilibrium will occur as $s \to \infty$. By taking sufficiently many iterations of the algorithm, it is our aim to determine the Nash equilibrium $\mathbf{x}^*$ with a specified precision.

[7] This is so because the corresponding gradient map is strictly anti-monotone (see theorem 4 in [21]).

3.2. Conditions for existence of a Nash equilibrium and convergence of the relaxation algorithm

The following theorem states the conditions of convergence for the relaxation algorithm. The conditions may look rather restrictive, but in fact a large class of games satisfy them.

Theorem 3.1 [23]. There exists a unique normalized Nash equilibrium point to which the algorithm (15) converges if:

(1) X is a convex compact subset of $\mathbb{R}^m$,

(2) the Nikaido–Isoda function $\Psi : X \times X \to \mathbb{R}$ is a weakly convex-concave function and $\Psi(\mathbf{x}, \mathbf{x}) = 0$ for $\mathbf{x} \in X$,

(3) the optimum response function $Z(\mathbf{x})$ is single-valued and continuous on X,

(4) the residual term $r_{\mathbf{z}}(\mathbf{x}, \mathbf{y})$ is uniformly continuous on X with respect to $\mathbf{z}$ for all $\mathbf{x}, \mathbf{y} \in X$,

(5) the residual terms satisfy

$$r_{\mathbf{y}}(\mathbf{x}, \mathbf{y}) - \mu_{\mathbf{x}}(\mathbf{y}, \mathbf{x}) \geqslant \beta(\|\mathbf{x} - \mathbf{y}\|), \quad \mathbf{x}, \mathbf{y} \in X, \qquad (16)$$

where $\beta(0) = 0$ and β is a strictly increasing function (i.e., $\beta(t_2) > \beta(t_1)$ if $t_2 > t_1$),

(6) the relaxation parameters α_s satisfy

(a) $\alpha_s > 0$,

(b) $\sum_{s=0}^{\infty} \alpha_s = \infty$,

(c) $\alpha_s \to 0$ as $s \to \infty$.

Notice that the convex set X is able to represent coupled constraints and that the key condition (16) may be satisfied in case of non-differentiable payoff functions.[8]

4. The relaxation algorithm with an optimized step size

4.1. Step size optimization

In order for the algorithm to converge, we may choose any sequence $\{\alpha_s\}$ satisfying the final condition of theorem 3.1. However, it is of computational importance to attempt to optimize the convergence rate.

Suitable step sizes may be obtained by trial and error, and we have found that using a constant step of $\alpha_s \equiv 0,5$ leads to a quick convergence in most of our experiments. In this case, we can think of the α_s as being constant until our convergence conditions are reached, and thereafter decaying with factors $\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$.

Further, we suggest a method for a step size optimization.

[8] Proving validity of condition 5 for non-differentiable functions can be difficult. However, an approach similar to the one used to prove diagonal strict concavity in [11] can be recommended.

Definition 4.1. Suppose that we reach $\mathbf{x}^s$ at iteration s. Then α_s^* is *one-step optimal* if it minimizes the optimum response function at $\mathbf{x}^{s+1}$. That is, if

$$\alpha_s^* = \arg\min_{\alpha\in[0,1)} \Big\{ \max_{\mathbf{y}\in X} \Psi\big(\mathbf{x}^{s+1}(\alpha), \mathbf{y}\big)\Big\}. \qquad (17)$$

(Recall that $\mathbf{x}^{s+1}$ depends on α.)

This method is intuitively appealing. Indeed, we are trying to minimize the maximum of the Nikaido–Isoda function to satisfy the equilibrium condition. We have found that optimizing the step sizes leads to fewer iterations, but that each step takes longer to complete than when using constant step sizes.

4.2. Conditions for convergence of the relaxation algorithm with an optimized step size

The following theorem shows that the relaxation algorithm with the optimized step sizes converges under the conditions of theorem 3.1.

Theorem 4.2. There exists a unique normalized Nash equilibrium point to which the algorithm (15) converges if:

(1) X is a convex compact subset of $\mathbb{R}^m$,

(2) the Nikaido–Isoda function $\Psi: X\times X \to \mathbb{R}$ is a weakly convex-concave function and $\Psi(\mathbf{x},\mathbf{x}) = 0$ for $\mathbf{x}\in X$,

(3) the optimum response function $Z(\mathbf{x})$ is single-valued and continuous on X,

(4) the residual term $r_{\mathbf{z}}(\mathbf{x},\mathbf{y})$ is uniformly continuous on X with respect to $\mathbf{z}$ for all $\mathbf{x},\mathbf{y}\in X$,

(5) the residual terms satisfy

$$r_{\mathbf{y}}(\mathbf{x},\mathbf{y}) - \mu_{\mathbf{x}}(\mathbf{y},\mathbf{x}) \geqslant \beta\big(\|\mathbf{x}-\mathbf{y}\|\big), \quad \mathbf{x},\mathbf{y}\in X, \ (18)$$

where $\beta(0) = 0$ and β is a strictly increasing function (i.e., $\beta(t_2) > \beta(t_1)$ if $t_2 > t_1$),

6. the relaxation parameters α_s satisfy

$$\alpha_s = \arg\min_{\alpha\in[0,1)} \Big\{ \max_{\mathbf{y}\in X} \Psi\big(\mathbf{x}^{s+1}(\alpha), \mathbf{y}\big)\Big\}. \qquad (19)$$

In fact, this theorem differs from theorem 3.1 only in condition 6. However, this difference substantially changes the convergence proof (compare [23]).

4.3. Proof of convergence of the algorithm with an optimized step size

Existence of a normalized equilibrium point $\mathbf{x}^*$ follows from Kakutani's theorem (see, for example, [2]). The convergence proof relies on the nonnegativeness of a Lyapunov function and is provided in an appendix.

In the next section we will consider some games in order to gain an appreciation for how the above algorithms work in conjunction with the Nikaido–Isoda function.

5. Application examples

5.1. A simple two player game with identical payoff functions

Consider a two player game where both players are maximizers, the action space for each player is $\mathbb{R}$, and the payoff functions of players 1 and 2 are identical. Players have the following payoff function on the region $X = \{(x_1,x_2)\colon -10 \leqslant x_1, x_2 \leqslant 10\}$:

$$\phi_i(\mathbf{x}) = -\frac{(x_1+x_2)^2}{4} - \frac{(x_1-x_2)^2}{9}. \qquad (20)$$

The Nikaido–Isoda function in this case is

$$\begin{aligned}\Psi(\mathbf{x},\mathbf{y}) = 2&\left\{\frac{(x_1+x_2)^2}{4} + \frac{(x_1-x_2)^2}{9}\right\}\\ &- \left\{\frac{(y_1+x_2)^2}{4} + \frac{(y_1-x_2)^2}{9}\right\}\\ &- \left\{\frac{(x_1+y_2)^2}{4} + \frac{(x_1-y_2)^2}{9}\right\}. \qquad (21)\end{aligned}$$

All conditions of the convergence of theorems 3.1 and 4.2 are satisfied. Condition 5 follows from the strict positive-definiteness of the matrix $Q(\mathbf{x},\mathbf{x}) = \Psi_{\mathbf{xx}}(\mathbf{x},\mathbf{y})|_{\mathbf{y}=\mathbf{x}} - \Psi_{\mathbf{yy}}(\mathbf{x},\mathbf{y})|_{\mathbf{y}=\mathbf{x}}$. The Nash equilibrium for this game is $\mathbf{x} = (0,0)$.

The optimum response function is calculated to be $Z(\mathbf{x}) = -\frac{5}{13}(x_2,x_1)$. In this case, it is also relatively simple to see how to optimize α_s; since both players have the same payoff function, the optimal α_s is the one which optimizes the payoff function

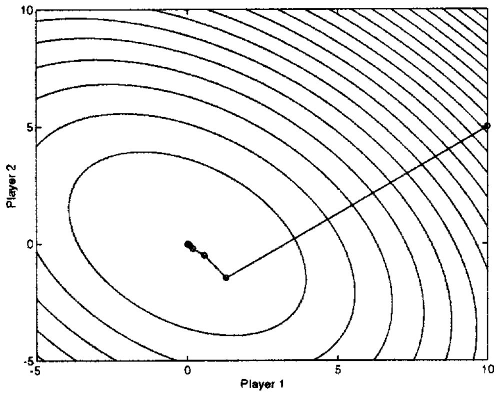
$$\phi_i\big(\mathbf{x}^{s+1}\big) = \phi_i\big(\alpha_s\mathbf{x}^s + (1-\alpha_s)Z\big(\mathbf{x}^s\big)\big), \ 0 < \alpha_s \leqslant 1. \ (22)$$

Calculations of our MATLAB program with starting guess $\mathbf{x} = (10,5)$, and optimized step sizes α_s are given in table 1.

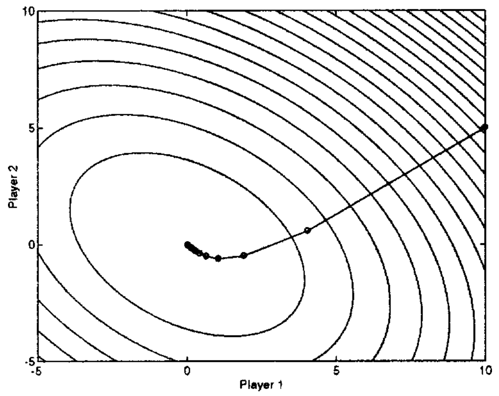
We can now make a comparison between the optimized and nonoptimized α_s, see figure 1. The first graph, with the optimization performed, shows a much quicker convergence. In contrast, the second one, which has $\alpha_s \equiv 0.5$, shows a smoother but much slower convergence. The third shows step sizes of $\alpha_s \equiv 1$ (that is, a non-relaxed algorithm). We would clearly prefer to use optimized step sizes if this could be achieved easily.

Table 1
Convergence of the example in section 5.1.

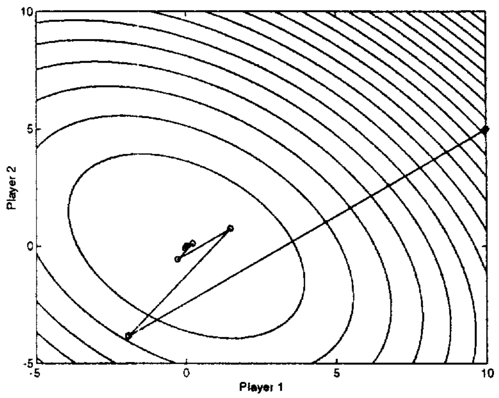
Iteration(s)	$\mathbf{x}^s$	α_s
0	$(10, 5)$	0.7309
1	$(1.2849, -1.4668)$	1.0000
2	$(0.5639, -0.4942)$	1.0000
3	$(0.1901, -0.2169)$	1.0000
4	$(0.0834, -0.0731)$	1.0000
5	$(0.0281, -0.0321)$	1.0000
6	$(0.0123, -0.0108)$	0.5001
7	$(0.0082, -0.0078)$	



(a)



(b)



(c)

Figure 1. Convergence of the example in section 5.1 with starting guess (10, 5) using (a) optimized α_s, (b) $\alpha \equiv 0.5$ and (c) $\alpha \equiv 1$. The isolines are those of the identical payoff function.

5.2. The quantity setting duopoly

In this model, the players' payoff functions are not necessarily the same. Consider a situation where two firms sell an identical product on the same market [9]. Each firm will want to choose its production rates in such a way as to maximize its respective profits. Let x_i be the production of firm i and let d, λ and ρ be constants. The market price is

$$p(\mathbf{x}) = d - \rho(x_1 + x_2) \qquad (23)$$

and the profit made by firm i is

$$\phi_i(\mathbf{x}) = p(\mathbf{x})x_i - \lambda x_i = \big[d - \lambda - \rho(x_1 + x_2)\big]x_i.$$

The Nikaido–Isoda function in this case is

$$\begin{aligned}\Psi(\mathbf{x},\mathbf{y}) = {} & \big[d - \lambda - \rho(y_1 + x_2)\big]y_1 \\ & - \big[d - \lambda - \rho(x_1 + x_2)\big]x_1 \\ & + \big[d - \lambda - \rho(x_1 + y_2)\big]y_2 \\ & - \big[d - \lambda - \rho(x_1 + x_2)\big]x_2, \qquad (24)\end{aligned}$$

leading to an optimum response function of

$$Z(\mathbf{x}) = \frac{d-\lambda}{2\rho}(1,1) - \frac{1}{2}(x_2, x_1). \qquad (25)$$

All convergence conditions of theorems 3.1 and 4.2 are satisfied. For positive ρ, condition 5 of these theorems follows from the strict positive-definiteness of the matrix

$$Q(\mathbf{x},\mathbf{x}) = \Psi_{\mathbf{xx}}(\mathbf{x},\mathbf{y})\big|_{\mathbf{y}=\mathbf{x}} - \Psi_{\mathbf{yy}}(\mathbf{x},\mathbf{y})\big|_{\mathbf{y}=\mathbf{x}} = 2\rho\begin{pmatrix}2 & 1\\ 1 & 2\end{pmatrix}.$$

It is a classical result that the Nash equilibrium in this game is

$$x_i^N = \frac{d-\lambda}{3\rho}$$

with corresponding payoff

$$\phi_i\big(x_i^N\big) = \frac{(d-\lambda)^2}{9\rho}$$

(see, for example [9]). To show the convergence of the algorithm in this case, let us assign values to the parameters. Let $d = 20$, $\lambda = 4$ and $\rho = 1$, then $x^N = (\frac{16}{3}, \frac{16}{3})$.

The convergence of the algorithm for the quantity setting duopoly is displayed in figure 2. Note that in the left hand panel $\alpha_s \equiv 0.5$. The right hand panel shows the optimized algorithm convergence. As in the previous example, the optimized step size requires fewer iterations to converge.

5.3. River Basin Pollution game

We will now use the relaxation algorithm to solve a game with coupled constraints. These constraints mean that the players' action set is now a general convex set $X \subset \mathbb{R}^n$ rather than as previously $X = X_1 \times \cdots \times X_n \subseteq \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_n}$, each separate X_i being convex.

5.3.1. Formulation of the River Basin Pollution game

In this game [6], we consider three players $j = 1, 2, 3$ located along a river. Each agent is engaged in an economic activity (paper pulp producing, say) at a chosen level x_j, but the players must meet environmental conditions set by a local authority.

Pollutants may be expelled into the river, where they disperse. Two monitoring stations $\ell = 1, 2$ are located along the river, at which the local authority has set maximum pollutant concentration levels.

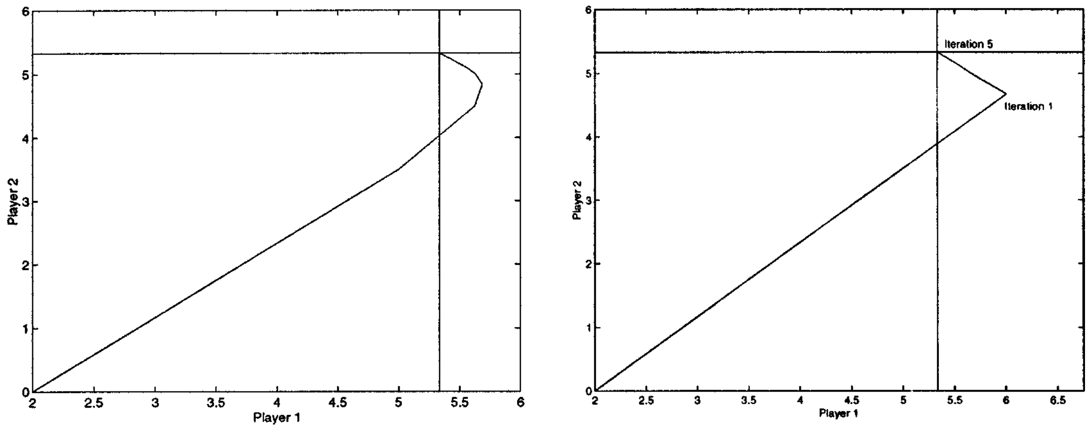


Figure 2. Convergence of the example in section 5.2.

Table 2
Constants for the River Basin Pollution game.

Player j	c_{1j}	c_{2j}	e_j	u_{j1}	u_{j2}
1	0.10	0.01	0.50	6.5	4.583
2	0.12	0.05	0.25	5.0	6.250
3	0.15	0.01	0.75	5.5	3.750

The revenue for player j is

$$\mathcal{R}_j(\mathbf{x}) = \left[d_1 - d_2(x_1 + x_2 + x_3)\right]x_j \tag{26}$$

with expenditure

$$\mathcal{F}_j(\mathbf{x}) = (c_{1j} + c_{2j}x_j)x_j. \tag{27}$$

Thus the net profit for player j is

$$\begin{aligned}\phi_j(\mathbf{x}) &= \mathcal{R}_j(\mathbf{x}) - \mathcal{F}_j(\mathbf{x})\\ &= \left[d_1 - d_2(x_1 + x_2 + x_3) - c_{1j} - c_{2j}x_j\right]x_j.\end{aligned} \tag{28}$$

The constraint on emission that is imposed by the local authority at location ℓ is

$$q_\ell(\mathbf{x}) = \sum_{j=1}^{3} u_{j\ell}e_j x_j \leqslant K_\ell, \quad \ell = 1, 2. \tag{29}$$

The economic constants d_1 and d_2 determine the inverse demand law and are set to 3.0 and 0.01, respectively. The values for constants c_{1j} and c_{2j} are given in table 2, and $K_\ell = 100$, $\ell = 1, 2$.

The $u_{j\ell}$ are the decay and transportation coefficients from player j to location ℓ, and e_j is the emission coefficient of player j, also given in table 2.

5.3.2. *Solution to the River Basin Pollution game*

The above game, in which agents maximize profits (28) subject to actions satisfying jointly convex constraints (29), is a coupled constraint game. We will use the relaxation algorithm to compute an equilibrium to this game in the sense of definition 2.1. This algorithm computes a normalized equilibrium point (in Aubin's sense, see definition 2.3), which is one of the many Nash equilibria that solve this game.[9]

The Nikaido–Isoda function in this case is

$$\begin{aligned}\Psi(\mathbf{x},\mathbf{y}) &= \sum_{j=1}^{3}\left(\phi_j(y_i|\mathbf{x}) - \phi_j(\mathbf{x})\right)\\ &= \left[d_1 - d_2(y_1 + x_2 + x_3) - c_{11} - c_{21}y_1\right]y_1\\ &\quad + \left[d_1 - d_2(x_1 + y_2 + x_3) - c_{12} - c_{22}y_2\right]y_2\\ &\quad + \left[d_1 - d_2(x_1 + x_2 + y_3) - c_{13} - c_{23}y_3\right]y_3.\end{aligned}$$

Notice that the region defined by equation (29) is convex. Condition 5 of theorem 3.1 follows from the strict positive-definiteness of the matrix $Q(\mathbf{x},\mathbf{x})$:

$$\begin{aligned}Q(\mathbf{x},\mathbf{x}) &= \Psi_{\mathbf{xx}}(\mathbf{x},\mathbf{y})|_{\mathbf{y}=\mathbf{x}} - \Psi_{\mathbf{yy}}(\mathbf{x},\mathbf{y})|_{\mathbf{y}=\mathbf{x}}\\ &= \begin{pmatrix} 4c_{21}+4d_2 & 2d_2 & 2d_2\\ 2d_2 & 4c_{22}+4d_2 & 2d_2\\ 2d_2 & 2d_2 & 4c_{23}+4d_2\end{pmatrix}\\ &= 2d_2\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}\\ &\quad + 2\begin{pmatrix} 2c_{21}+d_2 & 0 & 0\\ 0 & 2c_{22}+d_2 & 0\\ 0 & 0 & 2c_{23}+d_2\end{pmatrix}.\end{aligned}$$

[9] As known, the equilibrium with coupled constraints can be a multiple equilibrium and depends on how the burden of satisfying the constraints is to be distributed among the players. Once such a distribution is fixed, we can identify an equilibrium corresponding to that distribution. In fact, for *diagonally strictly concave* games, there is a one-to-one correspondence between the "burden" and the equilibrium. Our game is diagonally strictly concave, however, exploiting this feature here is beyond the scope of this paper. For a formal definition of Rosen's coupled constraint equilibrium and explanations see [19]. In our numerical experiments, we assume that the players share the hardship of satisfying the constraints evenly (or "in solidarity", i.e., the burden of satisfying the constraints satisfaction is distributed equally among all the players, see [4]).

Table 3
Convergence in the River Basin Pollution game for $\alpha \equiv 0.5$.

Iteration(s)	x_1^s	x_2^s	x_3^s	α_s
0	0	0	0	0.5
1	9.68	8.59	1.90	0.5
2	14.85	12.62	2.655	0.5
3	17.65	14.49	2.913	0.5
4	19.18	15.35	2.961	0.5
5	20.03	15.73	2.934	0.5
⋮	⋮	⋮	⋮	⋮
10	21.07	16.03	2.762	0.5
⋮	⋮	⋮	⋮	⋮
20	21.14	16.03	2.728	0.5

Table 4
Convergence in the River Basin Pollution game using the optimized step size.

Iteration(s)	x_1^s	x_2^s	x_3^s	α_s^*
0	0	0	0	0.5
1	19.35	17.19	3.79	1
2	20.71	16.11	3.043	1
3	20.88	16.07	2.924	0.5
4	21.08	16.03	2.776	1
5	21.10	15.03	2.757	0.5
⋮	⋮	⋮	⋮	⋮
10	21.14	16.03	2.73	0.4627
⋮	⋮	⋮	⋮	⋮
20	21.14	16.03	2.729	0.7534

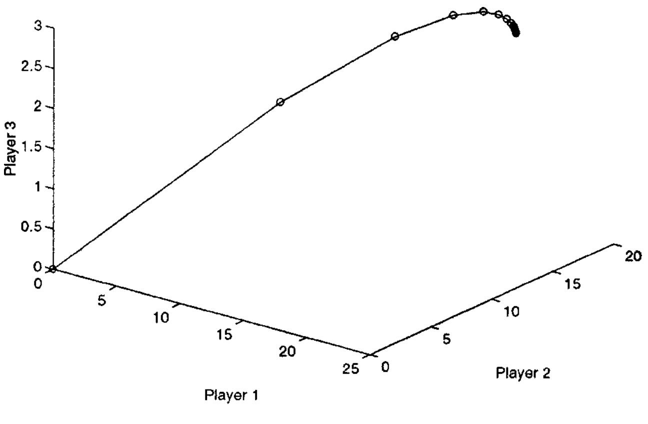


Figure 3. Convergence in section 5.3 with starting guess (0, 0, 0), $\alpha \equiv 0.5$.

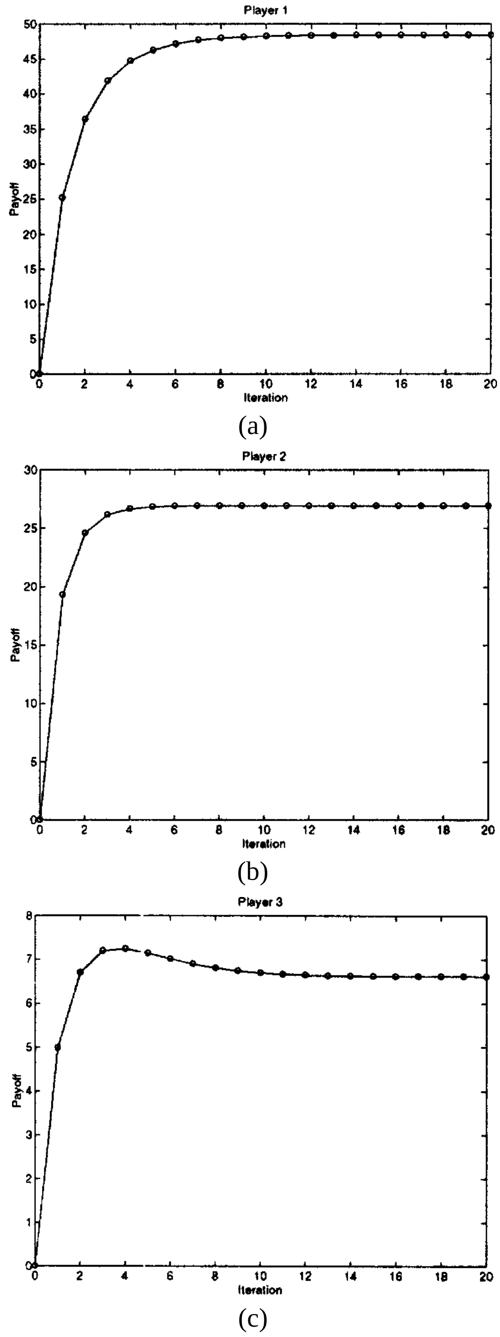


Figure 4. Progression of payoffs of figure 3.

The matrix $Q(\mathbf{x}, \mathbf{x})$ was calculated using the analytical capabilities of the package MATHEMATICA. Other convergence conditions of theorems 3.1 and 4.2 also are satisfied.

We used a starting guess of $\mathbf{x} = (0, 0, 0)$ in our MATLAB program. The convergence for $\alpha \equiv 0.5$ is shown in table 3. The calculations for the algorithm with optimized step sizes are given in table 4.

The convergence with $\alpha \equiv 0.5$ is displayed as a line in the 3D action space in figure 3. This game was also solved in [6] using Rosen's algorithm and was found to have equilibrium $\mathbf{x} = (21.149, 16.028, 2.722)$, giving net profits $\mathbf{z} = (48.42, 26.92, 6.60)$. The first constraint is active, i.e., $q_1(\mathbf{x}) = K_1 = 100$; the second constraint is inactive ($q_2(\mathbf{x}) = 81.17$). Our solution of $\mathbf{x} = (21.14, 16.03, 2.728)$ is within ± 0.01 of the solution achieved in [6] using Rosen's gradient projection algorithm [19].

5.3.3. Applying the Pigouvian taxes

Now that the Nash normalized equilibrium has been found, we can compel the players to obey it by applying Pigouvian taxes. In this way we create a new, unconstrained game.

For each constraint, we place a tax on each player to the amount of

$$T_\ell(\mathbf{x}) = \lambda_\ell \max(0, q_\ell(\mathbf{x}) - K_\ell), \tag{30}$$

where λ_ℓ, $\ell = 1, 2$, is a penalty coefficient for violating the ℓth constraint. Since $T_\ell(\mathbf{x})$ is a nonsmooth penalty function, there will always exist coefficients λ_ℓ sufficiently large to ensure that agents adhere to the environmental constraints (29). In other words, for "big" λ_ℓ, the waste produced by agents' optimal solutions will satisfy the environmental standards.

Applying the taxes as above leads us to the modified payoff functions ϕ_j^*,

$$\phi_j^*(\mathbf{x}) = R_j(\mathbf{x}) - F_j(\mathbf{x}) - \sum_\ell T_\ell(\mathbf{x}).$$

The new equilibrium problem with payoff functions ϕ_j^* and *uncoupled* constraints has the Nash equilibrium point x^{**} defined by the equation

$$\phi_j^*\left(\mathbf{x}^{**}\right) = \max_{x_j \geqslant 0} \phi_j^*\left(x_j | \mathbf{x}^{**}\right), \qquad j = 1, \ldots, n. \tag{31}$$

We make a conjecture based on the general theory of nonsmooth optimization (see, for example, [20]) that, for the environmental constraints' satisfaction, the penalty coefficients λ_ℓ should be greater than (or equal to) the Lagrange multipliers corresponding to the constraints (29). However, the new (unconstrained) Nash equilibrium $\mathbf{x}^{**}$ may not equal the old (constrained) Nash normalized equilibrium $\mathbf{x}^*$. In our numerical experiments we set λ_ℓ to equal the "final" Lagrange multipliers[10] for constraint ℓ that was observed during the calculation of the constrained equilibrium by algorithm (15). For this setup, the unconstrained equilibrium $\mathbf{x}^{**}$ is equal to the constrained equilibrium $\mathbf{x}^*$, see figure 5.

For our game, only the first constraint $\ell = 1$ was active. Thus, in the River Basin Pollution game (with the parameter values given in table 2), the payoff function for player j becomes

$$\begin{aligned}\phi_j^*(\mathbf{x}) &= R_j(\mathbf{x}) - F_j(\mathbf{x}) - T_1(\mathbf{x}) \\ &= \left[d_1 - d_2(x_1 + x_2 + x_3) - c_{1j} - c_{2j}x_j\right]x_j \\ &\quad - \lambda_1 \max\left(0, \sum_{j=1}^{3} u_{j1}e_j x_j - K_1\right).\end{aligned} \tag{32}$$

The computations of section 5.3.2 gave us the maximum Lagrange multiplier value for the active constraint $\lambda_1 = 0.5774$. Cross-sectional graphs (see figure 5) of the modified payoff functions illustrate that each payoff function achieves its maximum at the point $\mathbf{x}^*$.

[10] We were using a nonlinear programming subroutine (MATLAB `constr`) to maximize the Nikaido–Isoda function and the Lagrange multipliers were available at each iteration of the algorithm (15).

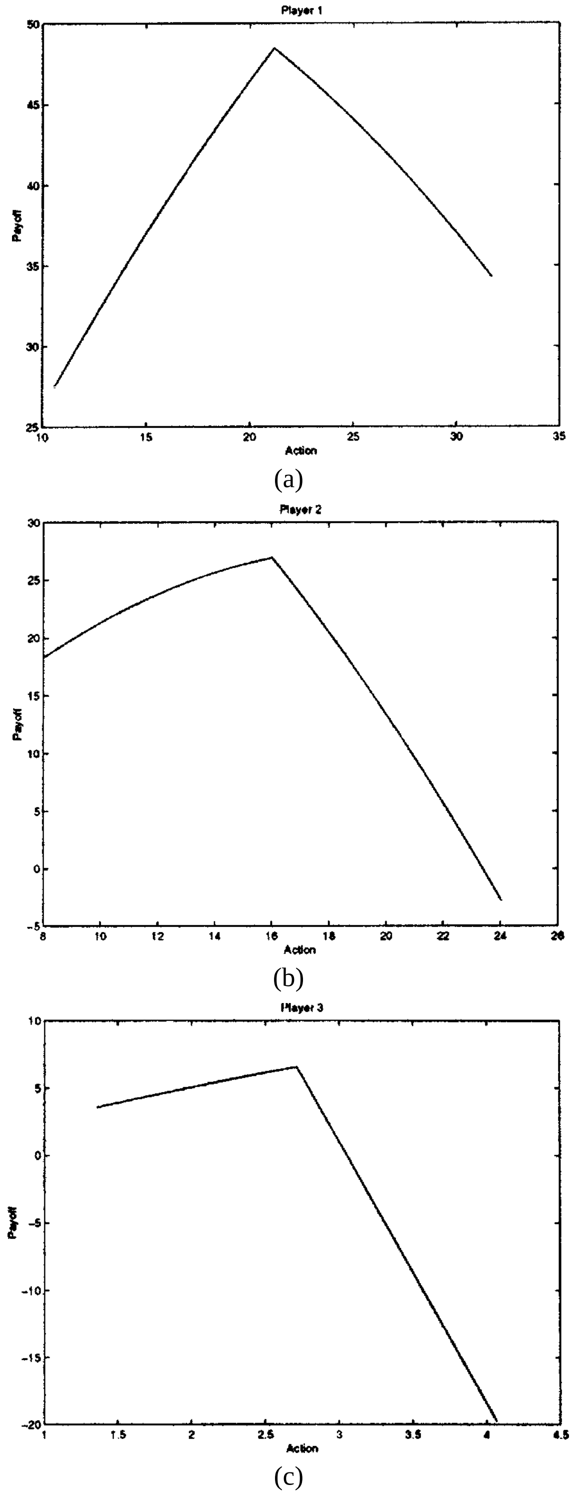


Figure 5. Payoff functions for players 1, 2, 3 with Pigouvian taxes applied.

6. Conclusions

The Nikaido–Isoda function was introduced to find a Nash equilibrium using optimization approaches. This, together with the relaxation methodology, allows us to find the Nash equilibrium of quite general infinite games (nonzero sum, multi-player, and coupled constraints) in a much simpler fashion than had been previously possible. Software was developed to implement these ideas and has been successfully used to solve some examples.

The use of the algorithm with an optimized step size resulted in a much smaller number of iterations needed for convergence. However, the computation time of one iteration depends on the effectiveness of the minimization method. In a few cases, the method[11] was inefficient, in that the overall computation times were comparable with the relaxation algorithm with the constant step size.

The numerical experiments showed that the relaxation algorithm is an efficient computational tool to solve Nash equilibrium problems under quite general conditions.

Appendix

Proof of convergence of the algorithm with an optimized step size

We need to establish existence and uniqueness of a normalized equilibrium point.

Existence of a normalized equilibrium point $\mathbf{x}^*$ follows from Kakutani's theorem (see, for example, [2]).

Let us denote by $G(\mathbf{x}) = \partial_{\mathbf{y}}\Psi(\mathbf{x},\mathbf{y})|_{\mathbf{y}=\mathbf{x}}$ the differential of the function $\Psi(\mathbf{x},\mathbf{y})$ [21] with respect to the second argument at point $(\mathbf{x},\mathbf{x})$. The following inequality was proved in [21]:

$$\langle g(\mathbf{x}) - g(\mathbf{y}), \mathbf{y} - \mathbf{x}\rangle \geqslant r_{\mathbf{y}}(\mathbf{x},\mathbf{y}) - \mu_{\mathbf{x}}(\mathbf{y},\mathbf{x}),$$

where

$$\mathbf{x},\mathbf{y} \in X,\ \ g(\mathbf{x}) \in G(\mathbf{x}),\ \ g(\mathbf{y}) \in G(\mathbf{y}).$$

Condition 5 of theorem 4.2 implies strict anti-monotonicity[12] of the gradient multivalued map $G(\mathbf{x})$ and, consequently, uniqueness of the normalized equilibrium point $\mathbf{x}^*$. The normalized equilibrium point is also a Nash equilibrium point (see lemma 2.4).

To prove convergence of the algorithm we use the following Lyapunov function:

$$V(\mathbf{x}) = \Psi(\mathbf{x}, Z(\mathbf{x})).$$

The idea of the proof relies on the contradiction between the nonnegativeness of the Lyapunov function and the fact that if there were no convergence to an equilibrium, $V(\mathbf{x})$ would have to be negative. The function $V(\mathbf{x})$ equals zero at the normalized equilibrium point $\mathbf{x}^*$,

$$V(\mathbf{x}^*) = 0,$$

and

$$V(\mathbf{x}^*) > 0, \quad \mathbf{x} \in X,\ \mathbf{x} \neq \mathbf{x}^*.$$

[11] As said, the MATLAB `constr` function was used to optimize α.

[12] The multivalued map $G(\mathbf{y})$ is called strictly anti-monotone if

$$\langle g(\mathbf{x}) - g(\mathbf{y}),\ \mathbf{y} - \mathbf{x}\rangle > 0 \quad \text{for all } \mathbf{x},\mathbf{y} \in X,$$
$$g(\mathbf{x}) \in G(\mathbf{x}), \quad g(\mathbf{y}) \in G(\mathbf{y}),$$

see, for instance, [21].

With condition 6 of the theorem, we can prove that the Lyapunov function $V(\mathbf{x}^s)$ cannot increase on the trajectory generated by the algorithm:

$$\begin{aligned} V(\mathbf{x}^{s+1}) &= \Psi(\mathbf{x}^{s+1}, Z(\mathbf{x}^{s+1})) \\ &= \Psi((1-\alpha_s)\mathbf{x}^s + \alpha_s Z(\mathbf{x}^s), Z(\mathbf{x}^{s+1})) \\ &= \min_{0\leqslant\alpha\leqslant 1} \big(\Psi((1-\alpha)\mathbf{x}^s + \alpha Z(\mathbf{x}^s), Z((1-\alpha)\mathbf{x}^s \\ &\quad + \alpha Z(\mathbf{x}^s)))\big) \\ &\leqslant \Psi(\mathbf{x}^s, Z(\mathbf{x}^s)) = V(\mathbf{x}^s). \end{aligned} \tag{33}$$

Let us denote by $\mathbf{x}^{s+1}(\alpha)$ the trajectory point which depends upon a free parameter α at iteration s, i.e.,

$$\mathbf{x}^{s+1}(\alpha) = (1-\alpha)\mathbf{x}^s + \alpha Z(\mathbf{x}^s).$$

Evidently, $\mathbf{x}^{s+1} = \mathbf{x}^{s+1}(\alpha_s)$, where step size α_s is obtained by optimization rule (19).

Weak convexity of the function $\Psi(\mathbf{x},\mathbf{y})$ with respect to $\mathbf{x}$ implies

$$\begin{aligned} V(\mathbf{x}^{s+1}) &= \Psi(\mathbf{x}^{s+1}, Z(\mathbf{x}^{s+1})) \\ &= \Psi((1-\alpha_s)\mathbf{x}^s + \alpha_s Z(\mathbf{x}^s), Z(\mathbf{x}^{s+1}(\alpha_s))) \\ &\leqslant \Psi((1-\alpha)\mathbf{x}^s + \alpha Z(\mathbf{x}^s), Z(\mathbf{x}^{s+1}(\alpha))) \\ &\leqslant (1-\alpha)\Psi(\mathbf{x}^s, Z(\mathbf{x}^{s+1}(\alpha))) \\ &\quad + \alpha\Psi(Z(\mathbf{x}^s), Z(\mathbf{x}^{s+1}(\alpha))) \\ &\quad - (1-\alpha)\alpha r_{Z(\mathbf{x}^{s+1}(\alpha))}(\mathbf{x}^s, Z(\mathbf{x}^s)), \end{aligned}$$

where $0 \leqslant \alpha \leqslant 1$. Since

$$\Psi(\mathbf{x}^s, Z(\mathbf{x}^{s+1}(\alpha))) \leqslant \Psi(\mathbf{x}^s, Z(\mathbf{x}^s))$$

and

$$\Psi(\mathbf{x}^s, \mathbf{x}^s) = 0,$$

then

$$\begin{aligned} V(\mathbf{x}^{s+1}) &\leqslant (1-\alpha)\Psi(\mathbf{x}^s, Z(\mathbf{x}^s)) + \alpha\Psi(\mathbf{x}^s, \mathbf{x}^s) \\ &\quad + \alpha\Psi(Z(\mathbf{x}^s), Z(\mathbf{x}^{s+1}(\alpha))) \\ &\quad - (1-\alpha)\alpha r_{Z(\mathbf{x}^{s+1}(\alpha))}(\mathbf{x}^s, Z(\mathbf{x}^s)). \end{aligned} \tag{34}$$

Weak concavity of the function $\Psi(\mathbf{x},\mathbf{y})$ with respect to the variable $\mathbf{y}$ implies

$$\begin{aligned} &(1-\alpha)\Psi(\mathbf{x}^s, Z(\mathbf{x}^s)) + \alpha\Psi(\mathbf{x}^s, \mathbf{x}^s) \\ &\quad \leqslant \Psi(\mathbf{x}^s, (1-\alpha)Z(\mathbf{x}^s) + \alpha\mathbf{x}^s) \\ &\qquad + (1-\alpha)\alpha\mu_{\mathbf{x}^s}(Z(\mathbf{x}^s), \mathbf{x}^s) \\ &\quad \leqslant V(\mathbf{x}^s) + (1-\alpha)\alpha\mu_{\mathbf{x}^s}(Z(\mathbf{x}^s), \mathbf{x}^s). \end{aligned} \tag{35}$$

Combining (34) and (35) we have

$$\begin{aligned} V(\mathbf{x}^{s+1}) &\leqslant V(\mathbf{x}^s) + (1-\alpha)\alpha\big[\mu_{\mathbf{x}^s}(Z(\mathbf{x}^s), \mathbf{x}^s) \\ &\quad - r_{Z(\mathbf{x}^{s+1}(\alpha))}(\mathbf{x}^s, Z(\mathbf{x}^s))\big] \\ &\quad + \alpha\Psi(Z(\mathbf{x}^s), Z(\mathbf{x}^{s+1}(\alpha))) \\ &= V(\mathbf{x}^s) + (1-\alpha)\alpha\big[\mu_{\mathbf{x}^s}(Z(\mathbf{x}^s), \mathbf{x}^s) \\ &\quad - r_{Z(\mathbf{x}^s)}(\mathbf{x}^s, Z(\mathbf{x}^s))\big] \end{aligned}$$

$$+ (1-\alpha)\alpha\big[r_{Z(\mathbf{x}^s)}\big(\mathbf{x}^s, Z(\mathbf{x}^s)\big) - r_{Z(\mathbf{x}^{s+1}(\alpha))}\big(\mathbf{x}^s, Z(\mathbf{x}^s)\big)\big] + \alpha\Psi\big(Z(\mathbf{x}^s), Z(\mathbf{x}^{s+1}(\alpha))\big). \quad (36)$$

Further, in view of condition 5 of the theorem,

$$V(\mathbf{x}^{s+1}) \leqslant V(\mathbf{x}^s) - (1-\alpha)\alpha\beta\big(\|\mathbf{x}^s - Z(\mathbf{x}^s)\|\big) + (1-\alpha)\alpha\big[r_{Z(\mathbf{x}^s)}\big(\mathbf{x}^s, Z(\mathbf{x}^s)\big) - r_{Z(\mathbf{x}^{s+1}(\alpha))}\big(\mathbf{x}^s, Z(\mathbf{x}^s)\big)\big] + \alpha\Psi\big(Z(\mathbf{x}^s), Z(\mathbf{x}^{s+1}(\alpha))\big). \quad (37)$$

Suppose that the statement of the theorem is not valid. In this case, there exists a subsequence $\mathbf{x}^{s_k}$ of the sequence $\mathbf{x}^s$, and there exists $\varepsilon > 0$ such that

$$\|\mathbf{x}^{s_k} - Z(\mathbf{x}^{s_k})\| \geqslant \varepsilon > 0, \quad k = 1, 2, \ldots. \quad (38)$$

The continuity of the function $Z(\mathbf{x})$ (condition 3) implies

$$\|Z(\mathbf{x}^{s_k+1}(\alpha)) - Z(\mathbf{x}^{s_k})\| \to 0, \quad \text{as } \alpha \to 0, \quad (39)$$

and, consequently, condition 2 implies

$$\Psi\big(Z(\mathbf{x}^{s_k}), Z(\mathbf{x}^{s_k+1}(\alpha))\big) \to 0, \quad \text{as } \alpha \to 0. \quad (40)$$

Since the residual term $r_{\mathbf{z}}(\mathbf{x}, \mathbf{y})$ is uniformly continuous with respect to $\mathbf{z}$, then with limit equation (39)

$$\big[r_{Z(\mathbf{x}^{s_k})}\big(\mathbf{x}^{s_k}, Z(\mathbf{x}^{s_k})\big) - r_{Z(\mathbf{x}^{s_k+1}(\alpha))}\big(\mathbf{x}^{s_k}, Z(\mathbf{x}^{s_k})\big)\big] \to 0, \quad \text{as } \alpha \to 0. \quad (41)$$

If α is sufficiently small, then (40) and (41) imply

$$\big[r_{Z(\mathbf{x}^{s_k})}\big(\mathbf{x}^{s_k}, Z(\mathbf{x}^{s_k})\big) - r_{Z(\mathbf{x}^{s_k+1}(\alpha))}\big(\mathbf{x}^{s_k}, Z(\mathbf{x}^{s_k})\big)\big] + (1-\alpha)^{-1}\Psi\big(Z(\mathbf{x}^{s_k}), Z(\mathbf{x}^{s_k+1}(\alpha))\big) \leqslant 2^{-1}\beta(\varepsilon). \quad (42)$$

Further, combining (37), (38), and (42), for sufficiently small α, we have

$$\begin{aligned} V(\mathbf{x}^{s_k+1}) &\leqslant V(\mathbf{x}^{s_k}) - (1-\alpha)\alpha\beta\big(\|\mathbf{x}^s - Z(\mathbf{x}^s)\|\big) \\ &\quad + (1-\alpha)\alpha\big[r_{Z(\mathbf{x}^{s_k})}\big(\mathbf{x}^{s_k}, Z(\mathbf{x}^{s_k})\big) \\ &\quad - r_{Z(\mathbf{x}^{s_k+1})}\big(\mathbf{x}^{s_k}, Z(\mathbf{x}^{s_k})\big) \\ &\quad + (1-\alpha)^{-1}\Psi\big(Z(\mathbf{x}^{s_k}), Z(\mathbf{x}^{s_k+1})\big)\big] \\ &\leqslant V(\mathbf{x}^{s_k}) - (1-\alpha)\alpha\beta(\varepsilon) + (1-\alpha)\alpha\big[2^{-1}\beta(\varepsilon)\big] \\ &\leqslant V(\mathbf{x}^{s_k}) - 2^{-1}(1-\alpha)\alpha\beta(\varepsilon). \end{aligned}$$

It follows from the last inequality that

$$V(\mathbf{x}^{s_k}) \to -\infty, \quad \text{as } k \to \infty.$$

This contradicts nonnegativeness of the function $V(\mathbf{x}^{s_k})$, which proves the theorem.

References

[1] J.P. Aubin, *Mathematical Methods of Game and Economic Theory* (Elsevier, Amsterdam, 1980).

[2] V.Z. Belenkii, V.A. Volkonskii, S.A. Ivankov, A.B. Pomaranski and A.D. Shapiro, *Iterative Methods in Game Theory and Programming* (Nauka, Moscow, 1974) (in Russian).

[3] T. Başar, Relaxation techniques and asynchronous algorithms for on-line computation of non-cooperative equilibria, Journal of Economic Dynamics and Control 11 (1987) 531–549.

[4] S. Berridge and J.B. Krawczyk, Relaxation algorithms in finding Nash equilibria, Economic Working papers Archive (1997). URL: http://econwpa.wustl.edu/eprints/comp/papers/9707/9707002.abs

[5] M.C. Ferris and J.S. Pang, eds., *Complementarity and Variational Problems: State of the Art* (SIAM, Philadelphia, PA, 1997).

[6] A. Haurie and J.B. Krawczyk, Optimal charges on river effluent from lumped and distributed sources, Environmental Modeling and Assessment 2 (1997) 93–106.

[7] W.D. Montgomery, Markets in licenses and efficient pollution control programs, Journal of Economic Theory 5 (1972) 395–418.

[8] H. Nikaido and K. Isoda, Note on noncooperative convex games, Pacific Journal of Mathematics 5, Supp. 1 (1955) 807–815.

[9] J.B. Krawczyk, *ECON 407 – Economic Dynamics B, Introduction to Dynamic Games with Application – Course Notes*, Works and Services, Printing Unit, Victoria University of Wellington (May 1995) p. 1774.

[10] J.B. Krawczyk, Modelling and economics of effluent management in municipalities, in: *Modelling Change in Integrated Economic and Environmental Systems*, eds. S. Mahendrarajah, A.J. Jakeman and M.J. McAleer (Wiley, New York, 1998).

[11] J.B. Krawczyk, O. Pourtallier and M. Tidball, A steady-state satisfactory solution to an environmental game with piece-wise defined payoffs, Natural Resource Modeling 11(4) (winter 1998) 301–329.

[12] S. Li and T. Başar, Distributed Algorithms for the Computation of Noncooperative Equilibria, Automatica 23 (1987) 523–533.

[13] A. Nagurney and K. Dhanda, Variational inequalities for marketable pollution permits with technological investment opportunities: the case of oligopolistic markets, Mathematics and Computer Modeling 26(2) (1997) 1–25.

[14] A. Nagurney, S. Thore and J. Pan, Spatial market policy modeling with goal targets, Operations Research 44 (1996) 393–406.

[15] A. Nagurney, *Network Economics: A Variational Inequality Approach* (Kluwer Academic, Boston, MA, 1993).

[16] E.A. Nurminski, Subgradient method for minimizing weakly convex functions and ε-subgradient methods of convex optimization, in: *Progress in Nondifferentiable Optimization: Collaborative Proceedings CP-82-S8* (International Institute for Applied Systems Analysis, Laxenburg, Austria, 1982) pp. 97–123.

[17] G.P. Papavassilopoulos, Iterative techniques for the Nash solution in quadratic games with unknown parameters, SIAM J. Control and Optimization 34(4) (1986) 821–834.

[18] A. Randall, *Resource Economics* (Wiley, New York, 1987).

[19] J.B. Rosen, Existence and uniqueness of equilibrium points for concave n-person games, Econometrica 33(3) (1965) 520–534.

[20] N.Z. Shor, *Minimization Methods for Non-Differentiable Functions* (Springer, Berlin, 1985).

[21] S. Uryasev, On the anti-monotonicity of differential mappings connected with general equilibrium problem. Optimization, A Journal of Mathematical Programming and Operations Research (Berlin) 19(5) (1988) 693–709.

[22] S. Uryasev, *Adaptive Algorithms for Stochastic Optimisation and Game Theory* (Nauka, Moscow, 1990) (in Russian).

[23] S. Uryasev and R.Y. Rubinstein, On relaxation algorithms in computation of noncooperative equilibria, IEEE Transactions on Automatic Control 39(6) 1263–1267.