On-line computation of incentive Stackelberg solution

Mitri Kitti
mitri.kitti@hut.fi
Contents

1 Introduction 2

2 Incentive Stackelberg solution 3

3 On-line computational method for finding the incentive solution 5
   3.1 Updating the incentive coefficients ............................ 7
   3.2 Algorithm for solving the incentive problem .................. 8

4 Example I: Quadratic objective function 9
   4.1 Fixed-point iteration ............................................. 9
   4.2 Broyden-method .................................................. 11

5 Example II: A duopoly model of coordination 13
   5.1 Symmetric firms ................................................. 15
   5.2 Asymmetric firms ............................................... 16
   5.3 Duopoly model with Nash-equilibria solved iteratively ......... 16

6 Discussion and conclusions 18

7 Appendix 23
   7.1 Necessary conditions for the follower’s problem ............... 23
   7.2 Broyden-method .................................................. 24
   7.3 Convergence theorems .......................................... 25
## 1 Introduction

A two-player Stackelberg game is a game where the other player (the leader) announces his decision (action) to the other player (the follower), who takes this decision into account when designing his reaction. The players may then act simultaneously, or the follower may take his action first. In the case where the leader can observe the follower’s action he can give his strategy as a function of follower’s action, i.e., in a form of an incentive. The follower chooses his best response to the leader’s incentive policy. Such Stackelberg games are called reversed or incentive Stackelberg games. If the one shot game is repeated, i.e., the same game is played several times, perhaps infinitely many times, then the game is called repeated incentive Stackelberg game. The most favorable incentive policy for the leader would be such that the follower’s best response leads to the optimal outcome of the one shot game for the leader. Hence, the leader seeks the policy that leads to his global optimum in each round. In this paper we study incentive stackelberg games where the leader does not know the follower’s objective function, i.e., the information is incomplete, but the game can be repeated.

Situations that can be described by Stackelberg games arise in many real world problems, where the parties have asymmetric positions. For example, in a large organization, the headquarter may execute resource allocation, profit sharing and penalty or reward policies that induce the subdivisions to work in accordance with the interest of the entire organization. Especially, many situations in economics where one of the parties is government can be formulated as Stackelberg games with the government trying to design such subsidy or penalty programs that induce the other players, for example competing firms to act cooperatively.

In [10] Zheng and Başar have proved the existence of an affine incentive strategy for a two player game and presented a possible construction of such strategy. Further results on the existence and constructions are given, e.g., in [2]. Direct computation of the incentive strategy requires perfect knowledge of the follower’s cost function or at least the follower’s reaction function, which will generally not be available to the leader. When the game is repeated the leader can learn his incentive policy from the follower’s responses.

We call an on-line algorithm a method that is real time implementable. In a repeated game an algorithm is real time implementable if the players can update their decisions using only such information that is available in the game. If the algorithm for computing game solutions is such that its use does not require the players to know each others’ objective functions then it can be called a distributed algorithm. In case of incomplete information on-line methods are also distributed methods. Here the term incomplete information refers to lack of knowledge about the structure of a game. In decision theory, incomplete information has also been used to refer
to situations characterized by a decision maker without precisely defined preferences.

In many real world problems the leader can not know the follower’s preferences and therefore it would be challenging to construct a procedure for solving the problem with incomplete information. Nevertheless, the on-line computation of the incentive solution without any initial knowledge on the follower’s cost function has been studied only little in literature. In [8] Vallée and Başar have studied the computation of the incentive solution using genetic algorithms, however, they do not implement the genetic algorithm as an on-line procedure. Ting [7] has studied an adaptive method for finding the incentive strategy. The adaptive method can be used in on-line computation but it needs some initial structural knowledge about the follower’s cost function.

In this paper, we shall present an on-line computational method for updating the incentive policy in a manner that leads to the leader’s global optimum. The method we shall present requires only follower’s best response in each iteration step and it does not require any additional information. This is possible because the leader’s problem can be modified to a form of a system of nonlinear equations that is defined by the follower’s best responses. The on-line computational method can be formed by solving the system of equations iteratively using an algorithm that requires only follower’s action in each iteration step.

The paper is arranged as follows. We define the incentive problem and introduce the notion of an incentive Stackelberg solution in Section 2. In Section 3 we present an algorithm for finding the incentive solution. In Sections 4 and 5 we test the algorithm presenting some numerical examples.

2 Incentive Stackelberg solution

We suppose that a two person incentive Stackelberg game is defined by the cost functions $J_L(u, v)$ and $J_F(u, v)$, for the leader and the follower respectively. The decision variables of the leader and the follower are $u$ and $v$, which belong to the action spaces $U$ and $V$. In this work we suppose that $U = \mathbb{R}^n$ and $V = \mathbb{R}^m$ and we use Euclidean norm ($\| \cdot \|$) in these spaces. Both players are trying to minimize their cost functions. Let us suppose that there exists a global optimum

$$(u^*, v^*) = \operatorname{arg} \min_{u \in U, v \in V} J_L(u, v)$$

(1)
for the leader’s cost function. In a repeated game the total cost for a player is the discounted sum of his costs of single rounds. For example for the leader the total cost is

$$
\sum_{i=1}^{T} \alpha^t J_L(u_i, v_i),
$$

(2)

where $\alpha \in (0, 1)$ is a discount factor and $T$ is the number of rounds in the repeated game.

Let $\gamma: V \to U$ be a policy for the leader belonging to a strategy space $C \subset \{\gamma| \gamma: V \to U\}$. The set $C$ is assumed to consist of affine policies of the form

$$
\gamma(v) = u_0 + Qv,
$$

(3)

where $Q$ is a linear mapping ($n \times m$ matrix) from $V$ into $U$ and $u_0$ is a fixed vector in $U$. The leader chooses his action according of the form (3), after the follower has chosen his action first, i.e., $u = \gamma(v)$. The follower’s action is the best response for the chosen incentive. The incentive design problem solved by the leader is then defined as follows. Find $\gamma \in C$, i.e., find $u_0$ and $Q$ such that

$$
v^* = \arg \min_{v \in V} J_F(\gamma(v), v),
$$

(4)

and

$$
u^* = \gamma(v^*).
$$

(5)

We call the $\gamma$ that solves the problem (4) and (5) an incentive at $(u^*, v^*)$. Suppose such $u_0$ and $Q$ exist. Equation (5) implies that we must have $u_0 = u^* - Qv^*$. Hence, we can without loss of generality seek incentives in the form

$$
\gamma(v) = u^* + Q(v - v^*).
$$

(6)

When the leader gives an incentive in the form (6) the follower has to solve an optimization problem

$$
\min_{v \in V} J_F(u, v)
$$

(7)

with $u = u^* + Q(v - v^*)$.

(8)

When the leader’s policy (8) is plugged into the follower’s objective function in (7) we obtain an unconstrained optimization problem with respect to the follower’s decision variables $v$, i.e., the objective function is $\tilde{J}_F(v)$. Supposing that the follower’s objective function is strictly
pseudoconvex we get the necessary and sufficient conditions for the follower’s optimum by setting
the gradient of the new objective function $\nabla \tilde{J}_F(v)$ zero. Applying the chain rule we get

$$Q^t \nabla_u J_F(u, v) + \nabla_v J_F(u, v) = 0. \quad (9)$$

An incentive policy at the point $(u^*, v^*)$ has to satisfy (9). Suppose $\nabla_u J_F(u^*, v^*) \neq 0$, then
one possible $Q$ satisfying (9) at $(u^*, v^*)$ is given by

$$Q = -\frac{\nabla_u J_F(u^*, v^*) \nabla_u J_F(u^*, v^*)^t}{\|\nabla_u J_F(u^*, v^*)\|^2}. \quad (10)$$

It is easy to see that if $Q$ is a solution to problem (4), (5) and (6), then $Q$ defines an affine
manifold on the hyperplane

$$\nabla_u J_F(u^*, v^*)^t(u - u^*) + \nabla_v J_F(u^*, v^*)^t(v - v^*) = 0. \quad (11)$$

Though there always exists an affine incentive at $(u^*, v^*)$ when $\nabla_u J_F(u^*, v^*) \neq 0$, the compu-
tation of this incentive requires either perfect knowledge of the followers cost function or some
learning process of the follower’s preferences.

In this paper we formulate the leader’s problem in a form that can be solved using standard
iterative methods that require only follower’s optimal response in each iteration. This is done
by parameterizing the matrix $Q$ and by searching iteratively such parameters that the incentive
leads to the optimal outcome for the leader.

3 On-line computational method for finding the incentive solu-
tion

The algorithm for finding the incentive presented in this section does not require explicit inform-
ation about the follower’s objective function. The main idea is that the leader gives his policy
in the form (6) with

$$Q = \frac{p_1 p_2^t}{\|p_1\|^2}. \quad (12)$$

where $p_1 \neq 0$ and $p_2$ are suitable parameter vectors. The follower’s optimization problem is
now

$$\min_{v \in V} J_F(u, v) \quad (13)$$

with $u = u^* - \frac{p_1 p_2^t}{\|p_1\|^2}(v - v^*). \quad (14)$
When $J_F$ is strictly pseudoconvex and the constraints are linear the problem (13) has a solution for every $\mathbf{p}_1 \in \mathbb{R}^n$, $\mathbf{p}_1 \neq 0$ and $\mathbf{p}_2 \in \mathbb{R}^m$.

Note that, as a result of this problem $\mathbf{u}$ and $\mathbf{v}$ will be functions of the vectors $\mathbf{p}_i$, $i = 1, 2$, and the resulting point $(\mathbf{u}(\mathbf{p}_1, \mathbf{p}_2), \mathbf{v}(\mathbf{p}_1, \mathbf{p}_2))$ will be on a submanifold (14) of the hyperplane $\mathbf{p}_1^t(\mathbf{u} - \mathbf{u}^*) + \mathbf{p}_2^t(\mathbf{v} - \mathbf{v}^*) = 0$. We will call the vectors $\mathbf{p}_i$, $i = 1, 2$, incentive coefficient vectors. Supposing that $\nabla_{\mathbf{u}} J_F(\mathbf{u}^*, \mathbf{v}^*) \neq 0$, then vectors

$$
\mathbf{p}_1 = \alpha \nabla_{\mathbf{u}} J_F(\mathbf{u}^*, \mathbf{v}^*) \quad \quad (15)
$$

$$
\mathbf{p}_2 = \alpha \nabla_{\mathbf{v}} J_F(\mathbf{u}^*, \mathbf{v}^*)
$$

with $\alpha \neq 0$, define an incentive at $(\mathbf{u}^*, \mathbf{v}^*)$. Note that, as $\alpha$ can be arbitrary large real number, the set of possible incentive coefficient vectors that define an incentive at $(\mathbf{u}^*, \mathbf{v}^*)$ is unbounded. Hence, when the incentive solution is searched in the form (6) with $Q$ defined by (12) in terms of $\mathbf{p}_1$ and $\mathbf{p}_2$ the solution set for the leader’s problem is unbounded.

Follower’s response to the policy defined by $\mathbf{p}_1$ and $\mathbf{p}_2$ will be used in the next iteration to generate a new policy. Let us denote

$$
\mathbf{p} = \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix}.
$$

The leader can update his incentive strategy according to the difference of the point $(\mathbf{u}(\mathbf{p}), \mathbf{v}(\mathbf{p}))$ and his global optimum. The updating of the coefficients is similar to the updating process used in generating Pareto optimal points in constraint proposal method described by Ehtamo et al [3]. The method presented for generating Pareto efficient points is based on updating a system of linear constraints going through some reference point. Now the reference point is the global optimum of the leader’s objective function and we are updating a submanifold on a hyperplane. Moreover, the manifold is given by the leader instead of a mediator. In the following subsections we shall describe updating the manifold or the coefficient vectors in more detail.

Note that if $\mathbf{p}_1 \neq 0$ and $\mathbf{p}_2 \neq 0$, then it can be shown that at the point $(\mathbf{u}(\mathbf{p}), \mathbf{v}(\mathbf{p}))$ it holds that

$$
\nabla_{\mathbf{u}} J_F(\mathbf{u}(\mathbf{p}), \mathbf{v}(\mathbf{p})) = \alpha \mathbf{p}_1
$$

$$
\nabla_{\mathbf{v}} J_F(\mathbf{u}(\mathbf{p}), \mathbf{v}(\mathbf{p})) = \alpha \mathbf{p}_2
$$

for some $\alpha \neq 0$ or otherwise

$$
\nabla_{\mathbf{u}} J_F(\mathbf{u}(\mathbf{p}), \mathbf{v}(\mathbf{p})) \perp \mathbf{p}_1 \quad \text{and} \quad \nabla_{\mathbf{u}} J_F(\mathbf{u}(\mathbf{p}), \mathbf{v}(\mathbf{p})) = 0.
$$

For details see section 7.1.
3.1 Updating the incentive coefficients

The leader’s problem is to find incentive coefficient vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ such that the solution of problem (13) and (14) is the leader’s global optimum $(\mathbf{u}^*, \mathbf{v}^*)$. Let $\mathbf{d}(\mathbf{p})$ denote the difference vector

\[
\mathbf{d}(\mathbf{p}) = \begin{pmatrix} \mathbf{d}_1(\mathbf{p}) \\ \mathbf{d}_2(\mathbf{p}) \end{pmatrix} = \begin{pmatrix} \mathbf{u}(\mathbf{p}) - \mathbf{u}^* \\ \mathbf{v}(\mathbf{p}) - \mathbf{v}^* \end{pmatrix} = \begin{pmatrix} -\frac{\mathbf{p}_1 \mathbf{p}_2^T}{\|\mathbf{p}_1\|^2}(\mathbf{v}(\mathbf{p}) - \mathbf{v}^*) \\ \mathbf{v}(\mathbf{p}) - \mathbf{v}^* \end{pmatrix}.
\]  

(19)

Using this notation the leader’s problem is to find a solution to the system of nonlinear equations

\[
\mathbf{d}(\mathbf{p}) = 0.
\]  

(20)

We have modified the original incentive problem (4), (5) and (6) to a form of finding a root of a system of nonlinear equations. The explicit form of $\mathbf{v}(\mathbf{p})$ is not known for the leader. Nevertheless, $\mathbf{v}(\mathbf{p})$ for given $\mathbf{p}$ is obtained from the follower’s reaction. Hence, there are two possible class of methods that can be used in solving (20), fixed-point iteration and quasi-Newton methods. In this paper we study fixed-point iteration and Broyden’s first method.

Let $\mathbf{p}^i$ denote the incentive coefficients in the iteration. Updating of the incentive constraint using fixed-point iteration is done according to

\[
\mathbf{p}^{i+1} = \mathbf{p}^i - \mu \mathbf{d}(\mathbf{p}^i),
\]  

(21)

where $\mu$ is an iteration parameter.

Another method for updating $\mathbf{p}$ is the Broyden’s first method that is a quasi-Newton method. Using Newton’s method requires knowing the inverse of the Jacobian matrix of the function $\mathbf{d}(\mathbf{p})$ in each iteration step. This information is not available for the leader. In the Broyden-method the Jacobian matrix is replaced by an approximation that satisfies Quasi-Newton and Broyden’s conditions (for more details see section 7.2). The inverse of such matrix in each iteration step is obtained from the following equation

\[
H^{i+1} = H^i + \frac{\mathbf{s}_i - H^i \mathbf{y}^i}{\mathbf{s}^i H^i \mathbf{y}^i} \mathbf{s}^i H^i,
\]  

(22)

where $\mathbf{s}_i = \mathbf{p}^{i+1} - \mathbf{p}^i$ and $\mathbf{y}^i = \mathbf{d}(\mathbf{p}^{i+1}) - \mathbf{d}(\mathbf{p}^i)$. Incentive coefficients are now updated according to

\[
\mathbf{p}^{i+1} = \mathbf{p}^i - H^i \mathbf{d}(\mathbf{p}^i).
\]  

(23)

We summarize the method for updating the incentive coefficients and the algorithm for finding the incentive solution using these methods in next subsection.
3.2 Algorithm for solving the incentive problem

The algorithm is based on the idea of finding the incentive strategy in a parameterized form. Instead of finding all \( n \times m \) components of matrix \( Q \) we suppose the the form (12), which includes only \( n + m \) unknown parameters. Moreover, when \( \nabla_v J_F(u^*, v^*) \neq 0 \), then vectors \( p_1 \) and \( p_2 \) that solve the equation (20) define the hyperplane (11) on which all possible affine incentive solutions are. An algorithm for finding the solution to (20) is the following

**Initialization step**

Let \( \epsilon > 0 \) be the termination scalar and \( T \) number of rounds in the repeated game. Choose an initial value \( p^1 \) for the incentive coefficients. If using fixed point iteration choose \( \mu \) and if using Broyden’s method choose a matrix \( H^1 \). Let \( i = 1 \) and go to the main step.

**Main step**

1. Playing of round \( i \):
   
   (a) The leader gives an incentive in the form of (14) with the coefficients \( p^i \).
   
   (b) The follower finds the optimum with the given incentive constraints, i.e., solves the problem (13). This results to a solution \( v^i = v(p^i) \).
   
   (c) Leader chooses his action according to (14) resulting to \( u^i = u(p^i) \).

2. If \( i = T \), then stop the game is over. If \( 1 < i < T \) and \( \|d(p^i)\| < \epsilon \), the incentive strategy has been found, repeat step 1 until the game is over. Otherwise go to step 3.

3. The leader updates the incentive coefficients using either fixed point iteration (21) or the Broyden-method (22) and (23). This results to new coefficients \( p^{i+1} \). Replace \( i \) by \( i + 1 \) and repeat step 1.

The termination criterion in step 2 is only one possible choice. Another appropriate criterion is for example \( |J_L(u^i, v^i) - J_L(u^*, v^*)| < \epsilon \) or \( |J_L(u^i, v^i) - J_L(u^*, v^*)|/|J_L(u^*, v^*)| < \epsilon \) when \( J_L(u^*, v^*) \neq 0 \).

As noted in Section 2 there exists a solution to (20), and vectors that are of the form (15) solve the system. Often, when the system has a solution, the iteration converges if the initial guess is close enough to the solution [5]. If the iteration is of the form \( p^{i+1} = G(p^i) \) with \( G : \mathbb{R}^n \to \mathbb{R}^n \), then according to Ostrowski’s theorem if \( \rho(G(p^*)) < 1 \) then \( p^* \) is a point of attraction for the iteration. The symbol \( \rho \) denotes the spectral radius (see Section 7.3 definitions 4, 5 and Theorem 1). For the fixed-point iteration of (20) this condition is satisfied when \( d \) is Lipschitz continuous and uniformly monotone [4] (Section 7.3 Definition 6 and Theorem 2).
Now the function \( d(p) \) depends on the follower’s best response \( v(p) \) and the required properties are not generally satisfied. Another way to prove the convergence of iteration is to prove that the algorithmic mapping \( G \) is closed and prove some additional properties for the iteration that guarantee the convergence [9] (Section 7.3 Theorem 3).

Broyden’s method is not stationary with respect to the iteration operator \( G \) because the updating of \( H \) results a new operator in each iteration step. Hence, the Broyden’s method is of the form \( p^{i+1} = G(p^i, i) \) (Section 7.3 Definition 2). For the Broyden’s first and second methods in a nonlinear case it has been proved the local 2n-step quadratic convergence rate [4]. In this paper we shall not study the convergence of the iteration in more detail.

In the following two sections we present numerical examples using the algorithm described in this section. In the first example we study a standard incentive problem with a quadratic followers’ objective function. The last example concerns a duopoly model.

4 Example I: Quadratic objective function

In this section we suppose that the followers objective function is a quadratic function of the decision variables:

\[
J_F(u, v) = \frac{1}{2} u^t K u + \frac{1}{2} v^t L v. \tag{24}
\]

For a given \( p \) the follower’s optimization problem results to

\[
v(p) = \left( L + \frac{p_2 p_1^t}{\|p_1\|^2} K \frac{p_1 p_2^t}{\|p_2\|^2} \right)^{-1} \frac{p_2 p_1^t}{\|p_1\|^2} K (u^* + \frac{p_1 p_2^t}{\|p_2\|^2} v^*), \tag{25}
\]

using (9) and (14) and supposing that the inverse exists.

4.1 Fixed-point iteration

We suppose that the decision variables \( u, v \in \mathbb{R} \) and we set the parameters as follows: \( u^* = v^* = 1 \), \( K = 1 \), \( L = 3 \). In Figure 1 we present graphical illustration of the first four steps of fixed-point iteration in \( v, u \)-plane with initial values \( p_1 = p_2 = 1 \). The dotted curves in the figure are the isoprofit contours of \( J_F \). The iteration converges rapidly close to the incentive at \((u^*, v^*)\) but it has some nonlinear properties. In the first iteration step the distance of \((u(p), v(p))\) from \((u^*, v^*)\) does not decrease and the value of \( J_F \) does not grow towards \( J_F(u^*, v^*) \), see also Figure 2.
Figure 1: Graphical illustration of fixed-point iteration in $v,u$-plane. $\zeta d = 1d$.

Figure 2: Norm of difference vector and the value of $J_F$ in fixed-point iteration, $\zeta = 0.5$ and initial $p_1 = p_2$. $i=1$. 

Figure 3: Graphical illustration of fixed-point iteration in $v,u$-plane.
In Figure 3 we see that during the iteration the norm of vector $p$ grows and the distance of the point $(p_1, p_2)$ from the point of attraction $p^*$ decreases. Note that, the point of attraction $p^*$ of the iteration depends on the initial guess and parameter $\mu$ and it can not be known a priori. All we know from $p^*$ is that $p^* = \alpha \nabla J_F(u^*, v^*)$ for some $\alpha \neq 0$. In the figure we have made an approximation that $p^* \approx p^{40}$. The relation between the point of attraction and the value of $\mu$ seems to be linear (Figure 4), when the initial guess is fixed. Choosing a large fixed-point parameter causes the convergence to a solution that has a large norm.

![Figure 3: Norm of vector p in fixed-point iteration, $\mu = 3$ and initial $p_1 = p_2 = 1$.](image)

Note that the convergence of the fixed point iteration can not be explained by Ostrowski's theorem. One can notice this by checking numerically the value of the spectral radius of the Jacobian matrix of operator $I - \mu d$. The convergence of the fixed-point iteration seems to result from the fact that the solution set of (20) is unbounded due to the parametrization of matrix $Q$. Nevertheless, the convergence is not global. For example if we set $\mu = 2$ then $p_1^2 = 0$ and the iteration does not converge. Moreover, the convergence rate of the iteration depends on the choice of parameter $\mu$ and the initial guess. For example if the initial point is such that the scale of the vectors $p_1$ and $p_2$ differs very much, then choosing a large value for $\mu$ may accelerate the convergence dramatically (Figure 5).

### 4.2 Broyden-method

In this example we suppose that $u, v \in \mathbb{R}^2$ and $u^* = v^* = (1 \ 1)^t$. For the matrices $K$ and $L$ we choose identity matrices. In figure 6 we have applied Broyden-method to the incentive problem with initial point $p = (1 \ 1 \ 5 \ 5)^t$ and the initial matrix $H$ a $4 \times 4$ identity matrix.
Figure 4: Norm of the point of attraction as a function of $\eta$.

Figure 5: Norm of the point of difference vector with $\eta = \eta_0(\times) \eta_0 = \eta(+) = \eta^0(\circ) = \eta^0(\circ)$. Initial $p_1 = p_2 = 0.001$. 

$\eta = \varepsilon d = \varepsilon d \eta$ Initial $\eta_0 = \eta_0(\times) \eta_0 = \eta(+) = \eta^0(\circ) = \eta^0(\circ)$.
The iteration converges rapidly close to the incentive solution. The Brody method has similar convergence properties as fixed-point iteration: the norm of the incentive coefficient vectors grows and the distance from the point of attraction decreases (Figure 7).

### Example II: A duopoly model of coordination

Let $q_1$ and $q_2$ be the quantities produced by the two firms. Let us denote $q = (q_1, q_2)$ and $H$ the identity matrix.

The figure: Norm of the difference vector for the Brody method with initial point $p = (1, 2)$.

The iteration converges rapidly close to the incentive solution. Brody method has similar convergence properties as fixed-point iteration: the norm of the incentive coefficient vectors grows and the distance from the point of attraction decreases (Figure 7).
where \( a_i, i = 0, \ldots, 3 \) are positive constants.

Let functions \( J_1 \) and \( J_2 \) be duopolists’ profit functions:

\[
J_i(q_i, u) = p(q_i, u)q_i - \frac{1}{2}c_i q_i^2, \quad i = 1, 2.
\]  

(27)

Government sets some target production levels \( q^* \) and some target \( q^*, u^* \). Let us denote \( b = (b_1, b_2)^t \). The government’s incentive is now of the form

\[
u_b(q) = u_0 + b^t(q - q^*).
\]

(28)

For a given incentive the firms play Nash against each other. In the Nash-equilibrium \( (q_1^N(b), q_2^N(b)) \) the following pair of equations is solved:

\[
\begin{align*}
\frac{\partial J_1}{\partial q_1} &= 0 \\
\frac{\partial J_2}{\partial q_2} &= 0.
\end{align*}
\]

(29)

As there are three parties in the game, we have to modify the algorithm presented in section 3. We replace the step that involves the follower’s best response for a given incentive by solving the Nash-equilibrium for the two firms. The leader plays as if there was only one follower.

First we present some iterations when the Nash-equilibrium is solved from (29). In subsection 5.3 we present iterations where the exact Nash-solution is not solved from (29) but it is solved iteratively by using relaxed asynchronous Gauss-Seidel algorithm presented in [1].
5.1 Symmetric firms

We set the values of the parameters as follows: \(a_0=5, a_1 = a_2 = a_3 = 1/2, c_0 = 10\) and \(c_1 = c_2 = 1\). The objective functions of the firms are same and therefore the firms are symmetric. In the government’s target \(q_1^* = q_2^* = 1.6949\) and \(u^* = 0.1695\). The incentive at the leader’s global optimum is defined by \(b = -(1 \ 1)^t\). To use the algorithm described in this paper we shall parameterize \(b\):

\[
b(\theta_1, \theta_2) = -\theta_1 \theta_2 / p_1^2 = -\theta_2 / \theta_1,
\]

where \(\theta_1 \in \mathbb{R}\) and \(\theta_2 \in \mathbb{R}^2\). The government’s problem is to find \(\theta_1\) and \(\theta_2\) that define the incentive at the team optimal solution.

When using Broyden-method the iteration does not converge if the initial guess for \(\theta_1\) and \(\theta_2\) is not close enough the solution. The reason is that the term \(s\) in (22), (23) diverges but the term \(y\) stays close to zero vector at the same time. Therefore the terms of the matrix \(H\) grow infinitely large. The behavior results from the poor initial guess and the nature of the equations (29). However, when the initial guess is good enough as in Figure 9, the Broyden’s iteration converges normally. Fixed-point iteration (Figure 8) does not suffer the same phenomenon, though it may converge slowly if the initial guess is badly chosen. As can be seen in Fig. 8 and Fig. 9 in six iteration steps (or periods) the government has induced the two firms very close to his optimum. Compared to the work required when using genetic algorithms [8] the amount of work is much less using on-line method with fixed-point iteration.

Figure 8: Fixed-point iteration with \(\mu = 100\) and initial values \(\theta_1 = 1, \theta_2 = (2 \ 2)^t\), quantities \(q_1\) (+) and \(q_2\) (o) in left, value of \(u\) in right. The dashed line is the value in the leader’s optimum.
Figure 9: Broyden’s iteration with initial values \( \theta_1 = 1, \theta_2 = (2, 2)^t \) and initial \( H \) an identity matrix, quantities \( q_1 \) (+) and \( q_2 \) (o) in left, value of \( u \) in right. The dashed line is the corresponding value in the exact Nash-equilibria of the current iteration.

5.2 Asymmetric firms

We set the values of the parameters as follows: \( a_0 = 5, a_1 = a_2 = a_3 = 1/2, c_0 = 10, c_1 = 2 \) and \( c_2 = 1 \). Now, the profit functions of the firms are asymmetric. In the government’s global optimum \( q_1^* = 1.0152, q_2^* = 2.0305 \) and \( u^* = 0.1523 \) and the incentive is defined by \( b = -(2 \cdot 0.5)^t \).

In the fixed-point iteration of Figure 10 we note that the iteration converges rapidly close to a solution but it reaches the exact solution very slowly. Broyden’s iteration converges faster than fixed-point iteration with above values of the parameters (Figure 11). If we choose for the initial values of \( \theta_1 \) and \( \theta_2 \) same values as in Fig. 8 and Fig. 9, then the Broyden-method does not converge.

5.3 Duopoly model with Nash-equilibria solved iteratively

In this subsection we study the duopoly model of coordination in a situation where the government does not wait for the duopolists to find the equilibrium. The firms are both playing Nash against each other, but as they do not know the objective function of each other they are trying to find the Nash-equilibrium iteratively. The government gives new incentive before the equilibrium is reached. Government updates his incentive using the on-line method described in Section 3. The duopolists update their production quantities by solving the Nash-equilibria using an on-line procedure. The Nash-equilibria is solved iteratively in each stage of the game.
Figure 10: Fixed-point iteration with $\mu = 100$ and initial values $\theta_1 = 1$, $\theta_2 = (1 \ 0)^t$, quantities $q_1$ (+) and $q_2$ (o) in left, value of $u$ in right. Dashed line is the value in the leader’s optimum.

Figure 11: Broyden’s iteration with initial $\theta_1 = 1$, $\theta_2 = (1 \ 0)^t$ and initial $H$ an identity matrix, quantities $q_1$ (+) and $q_2$ (o) in left, value of $u$ in right. Dashed line is the value in the leader’s optimum.
by using relaxed asynchronous Gauss-Seidel procedure, but the government limits the number of iteration steps. Following equations describe the Gauss-Seidel procedure for solving the Nash-equilibrium with a given incentive

\[
\begin{align*}
q_1^{k+1} &= \alpha q_1^k + (1 - \alpha) \left[ \arg \min_{q_1 \in \mathbb{R}} J_1(q_1, q_2^k, u_b(q_1, q_2^k)) \right], \\
q_2^{k+1} &= \beta q_2^k + (1 - \beta) \left[ \arg \min_{q_2 \in \mathbb{R}} J_2(q_1^{k+1}, q_2, u_b(q_1^{k+1}, q_2)) \right],
\end{align*}
\]

(31)

where parameters $\alpha$ and $\beta$ are relaxation parameters [1].

In the algorithm presented in Section 3.2 we replace the step 1 (b) by an iteration described by (31) with the number of steps $n$ determined by the government. We simulated the game with symmetric firms (Subsection 5.1) and parameters $n = 3$, $\alpha = \beta = 0.1$. As can be seen in Figure 12 fixed-point iteration converges rapidly to an incentive at the leader’s global optimum though $n$ is very small. The convergence of Broyden-method (Figure 13) is not as good as for fixed-point iteration. After seven iteration steps the iteration starts to diverge but in eleventh step the iteration converges again. Note that, in Fig. 12 and Fig. 13 duopolists play three rounds in each iteration step.

If the number of steps $n$ is reduced the convergence of the iteration decelerates. In Figure 14 $n = 1$ and the fixed-point iteration still converges rapidly but slower than when $n = 3$. The good convergence of the iteration partly results from the choice of initial $q$. We have chosen such initial values for the quantities that satisfy the market equilibrium equation 29 with the initial incentive.

6 Discussion and conclusions

Because of the asymmetric structure of the game the two-player repeated reversed Stackelberg game, the leader is able to update the incentive according to the follower’s responses without any prior knowledge on the follower’s objective function.

In this report we have presented a method for finding an incentive Stackelberg solution. The incentive strategy is parameterized with two incentive coefficient vector’s that define an affine manifold. In each round of the game the leader gives the follower a new incentive constraint. The follower optimizes his objective function and gives his response to the leader who updates the constraint accordingly. The updating procedure presented is based on the idea of solving numerically a system of equations.

We presented two methods for updating the incentive coefficients: fixed-point iteration and Broyden’s first method. These methods are typical procedures used in solving a system of
Figure 12: Fixed-point iteration with initial $\theta_1 = 1, \theta_2 = (2 \ 2)^t$ and $n = 3, \mu = 100$. The dashed line is the corresponding value in the exact Nash-equilibrium of the current iteration or incentive.

Figure 13: Broyden-method with initial $\theta_1 = 1, \theta_2 = (2 \ 2)^t$ and initial $H$ identity matrix, $n = 3$. The dashed line is the corresponding value in the exact Nash-equilibrium of the current iteration.
nonlinear equations. The methods can be applied in on-line computation because they require only such information that is available during the game.

We presented two numerical examples to show the use of the on-line method. The first example was a simple quadratic case, in the second example we studied the use of the algorithm in a situation of two competing firms and a government coordinating them.

The quadratic case showed some properties of the iterative procedures. Convergence of the fixed-point iteration depends on the initial guess for the incentive coefficients and the value of the fixed-point parameter. When choosing a larger fixed-point parameter, the norm of the point of attraction for the iteration grows. Fixed-point iteration may suffer from poor convergence rate and oscillatory behavior of the iteration. Applying a random relaxation may reduce the oscillations.

The convergence of the fixed-point iteration results from the parameterization of the incentive, which causes the solution set to be unbounded. Generally, the convergence of the fixed-point iteration requires strong properties from the system for which the solution is searched. Broyden’s iteration converges rapidly in quadratic case but in the duopoly model the convergence of the iteration requires that the initial guess is very close to a solution.

The method described in this paper has several benefits to methods previously applied to the incentive problem. The most important benefit is that the method has some potential in on-line computation. To compare the number of operations done when using the on-line procedure to
the use of genetic algorithm we can say that the difference is considerable. Genetic algorithm may require long simulations and each round requires a whole population of follower's answers to the leader's population of incentives. On-line computation requires same information as genetic algorithms but compared to the genetic algorithms the amount of work is less.
References


7 Appendix

7.1 Necessary conditions for the follower’s problem

**Definition 1.** Let $S$ be a nonempty open set in $\mathbb{R}^n$ and let $J : S \to \mathbb{R}$ be differentiable on $S$. The function $J$ is said to be

- **pseudoconvex** if for each $x_1, x_2 \in S$ with $\nabla J(x_1)^t(x_2 - x_1) \geq 0$ we have $J(x_2) \geq J(x_1)$

- **strictly pseudoconvex** if for each $x_1, x_2 \in S$ with $\nabla J(x_1)^t(x_2 - x_1) > 0$ we have $J(x_2) > J(x_1)$

In section 2 we obtained a necessary and sufficient condition (9) for the follower’s problem when $J_F$ is pseudoconvex. When $Q$ is of the form (12) we get

$$\frac{p_2 p_1^t}{\|p_1\|^2} \nabla_u J_F(u, v) = \nabla_v J_F(u, v)$$

(32)

for the necessary and sufficient condition. Together with the incentive constraint (14) the above equation defines $u(p)$ and $v(p)$. The following lemma shows that the information leader gets is a point $u(p)$ and $v(p)$ in which the gradients of the follower’s objective function are parallel to vectors $p_1$ and $p_2$ or $p_2 \perp \nabla_u J_F(u(p), v(p))$.

**Lemma 1.** If $p_1 \neq 0$, $p_2 \neq 0$, then either

$$\nabla_u J_F(u(p), v(p)) = \alpha p_1$$

(33)

or otherwise

$$\nabla_u J_F(u(p), v(p)) \in \{x | p_1^t x = 0\}.$$  

(34)

In both cases

$$\nabla_v J_F(u(p), v(p)) = \alpha p_2 \quad \text{with} \quad \alpha = \frac{p_1^t \nabla_u J_F(u(p), v(p))}{\|p_1\|^2}.$$  

(35)

**Proof.** Let us denote $\alpha = p_1^t \nabla_u J_F(u(p), v(p)) / \|p_1\|^2$. From necessary conditions (32) we obtain $\nabla_v J_F(u(p), v(p)) = \alpha p_2$ when $p_1 \neq 0$. Moreover, we notice that if condition (34) holds, then $\alpha = 0$ and $\nabla_v J_F(u(p), v(p)) = 0$ and otherwise we get from (32) $p_1^t \nabla_u J_F(u(p), v(p)) = \alpha \|p_1\|^2$, which implies $p_1^t \nabla_u J_F(u(p), v(p)) - \alpha p_1 = 0$, i.e., $\nabla_u J_F(u(p), v(p)) = \alpha p_1$ and the proof is complete. \(\square\)

If $p_2 = 0$ then the leader gets no information about $\nabla_u J_F(u_1, v_1)$.  

23
7.2 Broyden-method

Quasi-Newton methods are based on the notion of sequential approximation of the transpose of the Jacobian matrix $J(d(x))^t$ or its inverse $(J(d(x))^t)^{-1}$, when original system to solved is $d(x) = 0$ and the Newton iteration is defined by

$$x_{k+1} = x_k - (J(d(x_k))^t)^{-1}d(x_k).$$  \hspace{1cm} (36)

If $J(d(x))^t$ is approximated, the iteration process has the form

$$x_{k+1} = x_k - B_k^{-1}d(x_k).$$  \hspace{1cm} (37)

When the matrix $(J(d(x))^t)^{-1}$ is approximated, the computation is done by

$$x_{k+1} \approx x_k - H_k d(x_k).$$  \hspace{1cm} (38)

If the mapping $d$ is sufficiently smooth, for small variations $s_k = x_{k+1} - x_k$ we have

$$d(x_{k+1}) = d(x_k) + J(d(x))^t s_k.$$  \hspace{1cm} (39)

Requiring that the matrix $B_{k+1}$ satisfies

$$d(x_{k+1}) = d(x_k) + B_{k+1} s_k.$$  \hspace{1cm} (40)

Denoting $y_k = d(x_{k+1}) - d(x_k)$ and rewriting the above relation we get

$$B_{k+1} s_k = d_k.$$  \hspace{1cm} (41)

Similarly we get for $H_{k+1}$

$$H_{k+1} y_k = x_k.$$  \hspace{1cm} (42)

These relations are called quasi-Newton conditions. They do not determine uniquely the matrices $B_{k+1}$ and $H_{k+1}$ and some further hypothesis is needed for the computation of the matrices.

The matrices are uniquely defined when we require that the approximation does not change in some $n - 1$-dimensional subspace $\pi_k \subset \mathbb{R}^n$ not containing $s_k$ for $B_{k+1}$ and for for $H_{k+1}$ not containing $y_k$. For $B_{k+1}$ this can be written as

$$B_{k+1} x = B_k x \ \forall \ x \in \pi_k.$$  \hspace{1cm} (43)

and for $H_{k+1}$

$$H_{k+1} x = H_k x \ \forall \ x \in \pi_k.$$  \hspace{1cm} (44)
The unique solution for $B_{k+1}$ from (41) and (43) is

$$B_{k+1} = B_k + \frac{y_k - B_k s_k}{c_k s_k} c_k,$$

(45)

and for $H_{k+1}$ from (42) and (44) is

$$H_{k+1} = H_k + \frac{s_k - H_k y_k}{z_k^i y_k} z_k,$$

(46)

where $c_k$ is a vector orthogonal to $\pi_k$ (note that $\pi_k$ is different for the two formulas). If the matrices $B_k$ and $H_k$ are nonsingular and $H_k = B_k^{-1}$, then for $c_k = B_k z_k H_k^{-1} = B_k^{-1}$. Broyden’s first method is defined by 46 with $c_k = s_k$ and second by 46 with $c_k = y_k$.

7.3 Convergence theorems

**Definition 2.** Iterative process is said to be defined by (single valued) stationary mapping $G$ if $x_{k+1} = G(x_k)$, if the iterative process is defined by $x_{k+1} = G(x_k, k)$ we say that iteration is defined by a nonstationary mapping.

**Definition 3.** Fixed-points of a one-to-one mapping $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are the points belonging to the set $\{x \in \mathbb{R}^n | G(x) = x\}$.

**Definition 4.** For any complex $n \times n$ matrix $G$ the spectral radius $\rho(G)$ of $G$ is defined as the maximum of $|\lambda_1|, \ldots, |\lambda_n|$, where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $G$.

**Definition 5.** Let $G : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$. The $x^*$ is a point of attraction of the iteration $x^{k+1} = G(x^k)$ if there is an open neighborhood $S$ of $x^*$ such that $S \subset D$ and, for any $x^0 \in S$, the iterates $\{x^k\}$ defined by $x^{k+1} = G(x^k)$ all lie in $D$ and converge to $x^*$.

**Theorem 1 (Ostrowski’s Theorem).** Suppose that $G : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ has a fixed-point at $x^* \in \text{int}(D)$ and is (Fréchet-) differentiable at $x^*$. If the spectral radius of $J(G(x*))$ satisfies $\rho(J(G(x*))) = \sigma < 1$, then $x^*$ is a point of attraction of the iteration $x^{k+1} = G(x^k)$ ($J$ denotes the Jacobian).

For proof see [5].

**Definition 6.** We say that the mapping $d$ is:
• **Lipschitz (with constant \( l \)) if** \( \|d(x) - d(y)\| \leq l\|x - y\| \quad \forall x, y \in \mathbb{R}^n \);

• **monotone if** \( 0 \leq (d(x_1) - d(x_2))^t(x_1 - x_2) \quad \forall x_1, x_2 \in \mathbb{R}^n \);

• **uniformly monotone if** there exists a constant \( m > 0 \) such that

\[
m\|p^1 - p^2\| \leq (d(p^1) - d(p^2))^t(p^1 - p^2) \quad \forall p^1, p^2 \in U \times V.
\]

**Theorem 2.** Let the mapping \( d : \mathbb{R}^n \to \mathbb{R}^n \) be Lipschitz with constant \( l \) and uniformly monotone with monotonicity constant \( m \). Then system \( d(x) = 0 \) has a unique solution \( x = x^* \), and for any \( \mu \in (0, \frac{2m}{l^2}) \) the iteration \( x_{k+1} = x_k - \mu d(x_k) \) converges to \( x^* \) globally and for the convergence rate holds an estimate \( \|x_k - x^*\| \leq \frac{\mu q^k}{1 - q}\|d(x_0)\| \), where \( q = \sqrt{1 - 2m\mu + l^2}\mu^2} \).

For proof see [4].

**Definition 7.** A point-to-set map \( G : \mathbb{R}^n \to \mathbb{R}^n \) is closed if \( x_k \to x \ k \in \mathbb{K}, \ y_k = G(x_k) \ k \in \mathbb{K} \) and \( y_k \to y \ k \in \mathbb{K} \) imply \( y = G(x_k) \) (\( \mathbb{K} \) denotes an infinite set of positive integers).

**Theorem 3.** Let the point-to-set map \( G : \mathbb{R}^n \to \mathbb{R}^n \) determine an algorithm that given a point \( x_1 \in \mathbb{R}^n \) generates the sequence \( \{x_k\}_{k=1}^{\infty} \). Also let a solution set \( \Omega \subset \mathbb{R}^n \) be given. Suppose

(a) All points \( x_k \) are in a compact set \( X \subset \mathbb{R}^n \).

(b) There is a continuous function \( Z : \mathbb{R}^n \to \mathbb{R} \) such that: if \( x \) is not a solution, then for any \( y \in G(x) \) \( Z(y) > Z(x) \) and if \( x \) is a solution then either the algorithm terminates or for any \( y \in G(x) \) \( Z(y) \geq Z(x) \).

(c) The map \( G \) is closed at \( x \) if \( x \) is not a solution.

Then either the algorithm stops at a solution, or the limit of any convergent subsequence is a solution.

For proof see [9].