On the Exact Polynomial Time Algorithm for a Special Class of Bimatrix Game

Martynas Sabaliauskas and Jonas Mockus

Vilnius University, Institute of Mathematics and Informatics   Akademijos 4, Vilnius LT-08663, Lithuania   E-mail: akatasis@gmail.com, jmockus@gmail.com

Abstract

The Strategy Elimination (SE) algorithm was proposed in [2] and implemented by a sequence of Linear Programming (LP) problems. In this poster an efficient explicit solution is developed and the convergence to the Nash Equilibrium is proven.

Introduction

This poster investigates an example of quadratic bimatrix game model that describes the competition between inspectors and violators. We call them Inspector Games (IG). The IG is similar but not identical to the well known Inspection Game [3]. To solve large scale game problems and to prepare examples of game theory studies, it is essential to use polynomial time algorithms. No polynomial time algorithm is known for obtaining Nash Equilibrium (NE) of Bimatrix Games (BG) in general [4]. Therefore, an important task is to define a subset BG problems where NE can be obtained in polynomial time. In the poster this is done for Inspector Game.

Bimatrix Game. Profit Functions

A Bimatrix Game a finite non-cooperative game between two players. The payoff of the first player is expressed by a matrix $u(i,j)$ where $i = 1,...,m$ denote moves (pure strategies) of the first player and $j = 1,...,n$ represent moves of the second. The payoff of the second player is $v(i,j)$. The expected profit U and V of the first and second players are defined as expected values of payoffs $u(i,j)$ and $v(i,j)$

$$U(x,y) = \sum_{i,j} x_i u(i,j) y_j \quad \text{and} \quad V(x,y) = \sum_{i,j} x_i v(i,j) y_j.$$

Here $x = (x_1,...,x_m)$ and $y = (y_1,...,y_n)$ are probabilities (mixed strategies) of moves i and j.

In the Matrix Game (MG) $u(i,j) = -v(i,j)$. In the Bimatrix Game $u(i,j) \neq -v(i,j)$. In the Quadratic Bimatrix Game (QBG) $m = n$.

Strategy Elimination Algorithm

The general idea of the Strategy Elimination Algorithm (SE) [2] is to eliminate strategies that lead to non-positive solutions of the Irrelevant Frauf (IF) model [2]. First an IF solution for both players x, y is generated. If for some k $x_k \leq 0$ or $y_k \leq 0$ and the solution of the Direct Search Algorithm (DS) [2] is not found, then both the strategies x_k and y_k are eliminated from the set of feasible strategies by setting them to zero $x_k = y_k = 0$.

Iterative application of this method generates a sequence of BG. The iteration stops if a positive IF solution of the reduced QBG is obtained, a DS solution is found, or just a single element remains. Thus no more than n such steps are needed.

Inspector Game. Profit Functions

Denote by $x = (x_1,...,x_m),\ x_i \geq 0,\ \sum_i x_i = 1$ the inspection vector and by $y = (y_1,..,y_m),\ y_j \geq 0,\ \sum_j y_j = 1$ the violation vector. Here x_i denotes the inspection probability of the object i and y_j means the violation probability of the object j. Denote by $u(i,j)$ the inspector's payoff when the object i is inspected and the object j is violated. Denote by $v(i,j)$ the violator's payoff when the object i is inspected and the object j is violated. Functions $U(x,y)$ and $V(x,y)$ denote the inspector's and violator's profit functions using inspection and violation vectors x, y. These vectors define probabilities of inspection and violation. Payoffs of the Inspector Game (IG)

$$u(i,j) = \begin{cases} p_i g_i q_i, & \text{if } i = j, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

and

$$v(i,j) = \begin{cases} -p_i q_j g_j + (1-p_i) q_j g_j, & \text{if } i = j, \\ q_j g_j, & \text{otherwise}. \end{cases} \tag{2}$$

Here $p_i \in (0,1]$ is the probability of detecting the violation if it happens in the object i, $q_i \in (0,1]$ is the probability of the violation in the object i, and $g_i \in (0,\infty)$ is the payoff (potential) of the violation in the object i. The profit functions at given inspection and violation vectors x and y

$$U(x,y) = \sum_{i=1}^{m} x_i p_i q_i g_i y_i,$$

$$V(x,y) = \sum_{i=1}^{m} x_i \Big(-2 p_i q_i g_i y_i + \sum_{j=1}^{m} q_j g_j y_j \Big).$$

Here $u(i,j) \neq v(i,j)$, thus the IG is a bimatrix game [1].

Irrelevant Fraud Model

The Irrelevant Fraud (IF) model defines conditions where no incentive to break the contract solution (x^*, y^*) exists.

$$\alpha = \sum_j u(i,j) y_j, \quad i = 1,...,m, \tag{3}$$

$$\beta = \sum_i x_i v(i,j), \quad j = 1,...,m, \tag{4}$$

$$\sum_i x_i = 1,\ \sum_j y_j = 1,\ x_i, y_j \geq 0, \quad i,j = 1,...,m. \tag{5}$$

Explicit Solution of the IF Model

It follows from (3), (4), (1), (2), and (5) excluding inequalities that

$$y_i = \frac{1}{p_i q_i g_i \sum_{j=1}^{m} \frac{1}{p_j q_j g_j}}, \quad i = 1,...,m,$$

and

$$x_i = \frac{2 + q_i g_i \sum_{j=1}^{m} \frac{1}{p_j q_j g_j} - \sum_{j=1}^{m} \frac{1}{p_j}}{2 p_i q_i g_i \sum_{j=1}^{m} \frac{1}{p_j q_j g_j}}, \quad i = 1,...,m.$$

Then the expected payoffs can be expressed as

$$U(x,y) = \frac{1}{\sum_{j=1}^{m} \frac{1}{p_j q_j g_j}}, \quad V(x,y) = \frac{\sum_{j=1}^{m} \frac{1}{p_j} - 2}{\sum_{j=1}^{m} \frac{1}{p_j q_j g_j}}.$$

The following condition provides positive solutions $x_i > 0,\ i = 1,...,m$.

$$\min\Big(\{q_1 g_1, ..., q_m g_m\}\Big) > V(x,y) = \frac{\sum_{j=1}^{m} \frac{1}{p_j} - 2}{\sum_{j=1}^{m} \frac{1}{p_j q_j g_j}}. \tag{6}$$

Computing Time of the Explicit and Linear Programming Solutions

$m =$	200	400	600	800	10^3	10^4	10^5	10^6	10^7
LP	20	60	360	1500	-	-	-	-	-
ES	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.02	0.2	2

This table illustrates the efficiecy of the explicit solution (ES) comparing to linear programming (LP) implemention used in [2]. The CPU time in seconds (by Intel(R), Core(TM)2, 2.4GHz, RAM 2GB, Linux, Fedora 8) concides with the data of the chart for the dimension m.

Equilibrium Testing in the QBG

The equilibrium can be tested directly by the conditions of Nash equilibrium. First define the 'contract' profits U^*, V^* assuming that both players keep the contract solution (x^*, y^*)

$$U^* = \sum_{i,j} x_i^* u(i,j) y_j^*, \quad V^* = \sum_{i,j} x_i^* v(i,j) y_j^*.$$

Then define the maximal profits U_{max}, V_{max} of players assuming that opposite players keep the contract solution (x^*, y^*)

$$U_{max} = \max_x \sum_{i,j} x_i u(i,j) y_j^*, \quad V_{max} = \max_y \sum_{i,j} x_i^* v(i,j) y_j.$$

The contract profits U^*, V^* are compared with the values U_{max}, V_{max}, and if

$$U^* \geq U_{max}, \quad V^* \geq V_{max}$$

then the contract solution $x_i^* > 0,\ y_j^* > 0$ is recorded as an equilibrium strategy.

Convergence of the ES to the NE

Theorem. A pair of SE solutions (x,y) is a Nash equilibrium of IG.

Proof. After some iterations SE returns positive vectors x^*, y^* according to the conditions (3), (4) and (5) excluding inequalities. These vectors are calculated for reduced matrices $u^*(i,j)$ and $v^*(i,j)$. Denote by $R = \{r_1, r_2, ..., r_k\},\ k \leq m$ a subset of a complete set M of inspection objects $1, 2, ..., m$. If the SE solution $x_i^*,\ y_i^*,\ i \in R$ is positive then according to the conditions of [2] that is NE in subset R. From (6) follows that the general solution $x = (x_1, ..., x_m),\ x_j = 0,\ j \in M \setminus R$ satisfies inequality $\max(\{q_i g_i \mid i \in M \setminus R\}) < V(x^*, y^*)$. Then we can write that

$$U_{max} = \max\Big(\{p_i q_i g_i y_i^* \mid i \in R\}\Big) = U(x^*, y^*),$$

$$V_{max} = \max\Big(\{q_i g_i - 2 p_i q_i g_i x_i^* \mid i \in R\} \cup \{q_j g_j \mid j \in M \setminus R\}\Big) = V(x^*, y^*).$$

That means that SE solution (x,y) is NE for all inspection objects.

The Pseudocode

$InspectorSolver(m, p, q, g)$
$S_1 \leftarrow 0,\ S_2 \leftarrow 0$
for $i \leftarrow 1$ **to** m **do**
    $x_i \leftarrow 0,\ y_i \leftarrow 0,\ h_i \leftarrow i,\ f_i \leftarrow \frac{1}{p_i},\ d_i \leftarrow \frac{f_i}{q_i g_i}$
    $S_1 \leftarrow S_1 + d_i,\ S_2 \leftarrow S_2 + f_i$
$NoSolutions \leftarrow$ **true**, $n \leftarrow m$
while $NoSolutions$ **do**
    $NoSolutions \leftarrow$ **false**, $S_3 \leftarrow \frac{S_2 - 2}{S_1},\ j \leftarrow 1$
    for $i \leftarrow 1$ **to** n **do**
        if $\frac{f_{h_i}}{d_{h_i}} \geqslant S_3$ **then**
            $h_j \leftarrow h_i,\ j \leftarrow j + 1$
        else
            $S_1 \leftarrow S_1 - d_{h_i},\ S_2 \leftarrow S_2 - f_{h_i},$
            $NoSolutions \leftarrow$ **true**
    $n \leftarrow j - 1$
$S_4 \leftarrow \frac{2 - S_2}{2}$
for $i \leftarrow 1$ **to** n **do**
    $y_{h_i} \leftarrow \frac{d_{h_i}}{S_1},\ x_{h_i} \leftarrow y_{h_i} S_4 + \frac{f_{h_i}}{2}$
return x, y.

Web-Based Software

The software is implemented as an Java applet on the web `http://optimum2.mii.lt/` and can be started by pointing 'Start Bimatrix Game' in the section 'Discrete optimization'. By clicking the label 'Generate' one generates a number of inspection objects. The Label 'Calculate' starts NE search.

Concluding Remarks

We need polynomial time algorithms to solve large scale game problems and to explore numerically examples in studies of game theory. No polynomial time algorithm obtaining Nash Equilibrium (NE) is known for Bimatrix Games (BG) in general. Therefore, an important task is to define a subset of BG problems where NE can be reached in polynomial time. In this poster an exact polynomial time algorithm is described for a special class of Bimatrix Game called as the Inspector Game.

References

[1] F. Forgo, J. Szep, and F. Szidarovszky. *Introduction to the Theory of Games*. Kluwer Academic Publishers, 1999.

[2] J. Mockus. A web-based bimatrix game optimization model of polynomial complexity. *Informatica*, 20:79-98, 2009.

[3] G. Owen. *Game Theory*. Academic Press, San Diego, 1995.

[4] R.W. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 20:642-662, 2008.