Lietuvos matematikos rinkinys
Proc. of the Lithuanian Mathematical Society, Ser. A
Vol. **54**, 2013, 85–90

ISSN 0132-2818
www.mii.lt/LMR/

On the exact polynomial time algorithm for a special class of bimatrix game

Jonas Mockus, Martynas Sabaliauskas

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

Abstract. The Strategy Elimination (SE) algorithm was proposed in [2] and implemented by a sequence of Linear Programming (LP) problems. In this paper an efficient explicit solution is developed and the convergence to the Nash Equilibrium is proven.
Keywords: game theory, polynomial algorithm, Nash equilibrium.

Introduction

This paper 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 in general [4]. Therefore, an important task is to define a subset BG problems where NE can be obtained in polynomial time. In the paper this is done for Inspector Game.

1 Bimatrix game

1.1 Profit functions

The payoff of the first player is expressed by a matrix $u(i,j)$ where $i = 1,\dots,m$ denote moves (pure strategies) of the first player and $j = 1,\dots,n$ 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 $v(i,j)$

$$U(x,y) = \sum_{i,j} x_i u(i,j) y_j,$$

and

$$V(x,y) = \sum_{i,j} x_i v(i,j) y_j.$$

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

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

2 Search for the Nash equilibrium

2.1 Equilibrium testing in the quadratic bimatrix game

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^*)

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

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

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

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

$$U^* \geqslant U_{\max}, \qquad V^* \geqslant V_{\max},$$

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

2.2 Irrelevant Fraud model

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

$$U = \sum_j u(i,j) y_j, \quad i = 1, \ldots, m, \tag{1}$$

$$V = \sum_i x_i v(i,j), \quad j = 1, \ldots, m, \tag{2}$$

$$\sum_i x_i = 1, \qquad \sum_j y_j = 1, \quad x_i \geqslant 0,\ y_j \geqslant 0,\ i,j = 1, \ldots, m. \tag{3}$$

In this paper, an explicit solution is presented. If the solution happens to be positive, meaning that $x_i > 0$, $y_j > 0$, $i, j = 1, \ldots, m$, than according to the conditions of [2] that is NE. Variables x_i and y_j can be zero or negative, too. Then the quadratic elimination (QE) algorithm [2] can be used which provides NE in a specific bimatrix game called as the Inspector Game.

3 Inspector game

3.1 Profit functions

Denote by $x = (x_1, \ldots, x_m)$, $x_i \geqslant 0$, $\sum_i x_i = 1$ the inspection vector and by $y = (y_1, \ldots, y_m)$, $y_j \geqslant 0$, $\sum_j y_j = 1$ 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)

$$u(i,j) = \begin{cases} p_i g_i q_i, & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

and

$$v(i,j) = \begin{cases} -p_i q_j g_j + (1-p_i) q_j g_j, & \text{if } i = j, \\ q_j g_j, & \text{otherwise.} \end{cases} \tag{5}$$

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

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

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

3.2 Explicit solution of the irrelevant Fraud model

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

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

and

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

Then the expected payoffs can be expressed as

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

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

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

3.3 Strategy elimination algorithm

The general idea of the Strategy Elimination Algorithm (SE) is to eliminate strategies that lead to non-positive IF solutions. First an IF solution for both players x, y is generated. If for some k, $x_k \leqslant 0$ or $y_k \leqslant 0$ 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.

Theorem 1. *A pair of SE solutions (x, y) is a Nash equilibrium of Inspector Game.*

Proof. After some iterations SE returns positive vectors x^*, y^* according to the conditions (1), (2) and (3) excluding inequalities. These vectors are calculated for reduced matrices $u^*(i, j)$ and $v^*(i, j)$. Denote by $R = \{r_1, r_2, \ldots, r_k\}$, $k \leqslant m$ a subset of a complete set M of inspection objects $1, 2, \ldots, m$. 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 $x = (x_1, \ldots, x_m)$, $x_j = 0$, $j \in M \setminus R$ satisfies inequality

$$\max\big(\{q_i g_i \mid i \in M \setminus R\}\big) < V\big(x^*, y^*\big). \tag{7}$$

Then, from (7) we can write that

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

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

Theorem 2. *Complexity of the SE algorithm is of polynomial.*

Proof. The SE algorithm reads the lists at most $\sum_{k=0}^{m-1}(m-k)$ times that means the polynomial time complexity.

The algorithm is implemented as an Java applet and is on the web: http:// optimum2.mii.lt/. The pseudo-code follows:

Algorithm 1 *InspectorSolver*(m, p, q, g)

1. $S_1 \leftarrow 0$, $S_2 \leftarrow 0$
2. **for** $i \leftarrow 1$ **to** m **do**
3. $\quad x_i \leftarrow 0$, $y_i \leftarrow 0$, $h_i \leftarrow i$, $f_i \leftarrow \frac{1}{p_i}$, $d_i \leftarrow \frac{f_i}{q_i g_i}$
4. $\quad S_1 \leftarrow S_1 + d_i$, $S_2 \leftarrow S_2 + f_i$
5. *NoSolutions* $\leftarrow$ **true**, $n \leftarrow m$
6. **while** *NoSolutions* **do**
7. $\quad$ *NoSolutions* $\leftarrow$ **false**, $S_3 \leftarrow \frac{S_2 - 2}{S_1}$, $j \leftarrow 1$
8. $\quad$ **for** $i \leftarrow 1$ **to** n **do**
9. $\quad\quad$ **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

Table 1. Computing time of the Explicit (EP) and LP solutions.

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

Table 1 illustrates the efficiecy of explicit solution comparing to Linear Programming (LP) implementation 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.

4 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.

5 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 paper 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.

REZIUMĖ

Efektyvus algoritmas specialiam bimatricinio lošimo uždaviniui spręsti
J. Mockus, M. Sabaliauskas

Nėra žinomų polinominio sudėtingumo algoritmų bimatriciniams lošimams (BM) spręsti. Darbe pateikiamas analitinis specialios BM klasės uždavinių sprendinys ir įrodomas jo konvergavimas į Nash'o pusiausvyrą. Sprendinys realizuotas Java kalba ir laisvai prieinamas tinkle.

Raktiniai žodžiai: lošimų teorija, polinominis algoritmas, Nash'o pusiausvyra.