Lietuvos matematikos rinkinys
Proc. of the Lithuanian Mathematical Society, Ser. A
DOI: 10.15388/LMR.A.2014.15

ISSN 0132-2818
Vol. **55**, 2014
pages 79–84

On the Nash equilibrium in the inspector problem

Martynas Sabaliauskas, Jonas Mockus

Vilnius University, Institute of Mathematics and Informatics
Akademijos 4, LT-08663 Vilnius
E-mail: martynas.sabaliauskas@mii.vu.lt, jmockus@gmail.com

Abstract. Inspector problem represents an economic duel of inspector and law violator and is formulated as a bimatrix game. In general, bimatrix game is NP-complete problem. The inspector problem is a special case where the equilibrium can be found in polynomial time. In this paper, a generalized version of the Inspector Problem is described with the aim to represent broader family of applied problems, including the optimization of security systems. The explicit solution is provided and the Modified Strategy Elimination algorithm is introduced.

Keywords: game theory, Nash equilibrium, polynomial complexity.

1 Introduction, bimatrix games

The bimatrix game is defined by two $m \times n$ dimension payoff matrices. $U = (u_{ij})$ and $V = (v_{ij})$ where $i = 1, \ldots, m$, $j = 1, \ldots, n$ are pure strategies of the first and second player correspondingly [1]. The mixed strategies are denoted by x_i, y_j where x_i, y_j are probabilities. Formally

Definition 1. A vector $x = (x_1, x_2, \ldots, x_m) : \sum_{i=1}^{m} x_i = 1$ is pure strategy of the first player, if $x_i \in \{0, 1\}$, it is mixed if $x_i \geqslant 0$, and it is strictly mixed if $x_i > 0$, $i = 1, \ldots, m$. Strategies of the second player are defined similarly.

Definition 2. The sums $U(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i u_{ij} y_j$ and $V(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i v_{ij} y_j$ are expected payoffs.

Definition 3. A pair (x, y) is the Nash Equilibrium (NE) if it satisfies following conditions:

$$U(x, y) \geqslant U_{\max} = \max\left(\left\{ \sum_{j=1}^{n} u_{ij} y_j \;\middle|\; i = 1, \ldots, m \right\}\right), \tag{1}$$

$$V(x, y) \geqslant V_{\max} = \max\left(\left\{ \sum_{i=1}^{m} x_i v_{ij} \;\middle|\; j = 1, \ldots, n \right\}\right). \tag{2}$$

In [1] the bimatrix games were reduced to bilinear programming and the algorithm defining NE was proposed. However, at the time there are no algorithms for NE of the general bimatrix problem. In this paper an MSE algorithm of polynomial complexity was proposed, and implemented for a special class of bimatrix game, so called Generalized Inspector Problem which properties are defined by Theorem 2. In contrast to the general Lemke algorithm [1], the MSE algorithm requires solutions of a finite sequence of linear programming problems instead of the bilinear programming problem which is NP-complete. In both the matrices U and V of the Inspector Problem, all the column elements and all the raw elements are equal (with exception of the diagonal elements). This enables the explicit solution of complexity $O(m)$.

Theorem 1. *A pair of strictly mixed strategies (x, y) is the Nash equilibrium of the bimatrix game (u_{ij}), (v_{ij}) if and only if there exist real numbers α ir β, such that*

$$\begin{cases} \sum_{j=1}^{n} u_{ij} y_j = \alpha, & i = 1, \dots, m, \\ \sum_{i=1}^{m} x_i v_{ij} = \beta, & j = 1, \dots, n. \end{cases} \tag{3}$$

Proof. Is in [2]. □

2 Inspector problem

A simple version of the Inspector problem was considered in [2, 3]. Here we regard more general setup which allows formulation and solution broader family of problems, including the optimization of the security services. The general inspector problem is defined by matrix $(a_{ij}) - m \times n$, $m \geqslant n$, where elements a_{ij}, $i = 1, \dots, m$, $j = 1, \dots, n$ describe inspection actions and parameters. The set of inspection objects is denoted by $M = \{1, 2, \dots, m\}$, inspection actions are defined by vector

$$b_j = (a_{1j}, a_{2j}, \dots, a_{mj}), \quad j = 1, \dots, n.$$

and inspection parameters are represented by vector

$$c_i = (a_{i1}, a_{i2}, \dots, a_{in}), \quad i = 1, \dots, m.$$

Let's consider quadratic game $m \times m$. Define the payoffs of the first player by matrices (u_{ij}) with elements

$$u_{ij} = \begin{cases} f_u(c_i), & \text{if } i = j, \\ g_u(c_i), & \text{if } i \neq j, \end{cases} \tag{4}$$

and initial conditions:

$$f_u(c_i) \neq g_u(c_i), \quad i = 1, \dots, m.$$

The payoffs of the second player are defined by matrix (v_{ij}) with elements

$$v_{ij} = \begin{cases} f_v(c_j), & \text{if } i = j, \\ g_v(c_j), & \text{if } i \neq j, \end{cases} \tag{5}$$

and initial conditions

$$f_v(c_j) \neq g_v(c_j), \quad j = 1, \dots, m.$$

Here $f_u(c_i)$, $g_u(c_i)$, $f_v(c_j)$ and $g_v(c_j)$ are real functions of vectors c_i and c_j. The objective is to define mixed strategies $x = (x_1, x_2, \dots, x_m)$ and $y = (y_1, y_2, \dots, y_m)$, satisfying equilibrium conditions (1) and (2).

2.1 Optimal strategies

The first player is setting such mixed strategies that the payoff of the second player would be independent on its strategies:

$$\begin{cases} \sum_{j=1}^{m} u_{ij} y_j = \alpha, & i = 1, \dots, m, \\ \sum_{j=1}^{m} y_j = 1. \end{cases} \tag{6}$$

The second player is setting such mixed strategies that the payoff of the first player would be independent on its strategies:

$$\begin{cases} \sum_{i=1}^{m} x_i v_{ij} = \beta, \quad j = 1, \dots, m, \\ \sum_{i=1}^{m} x_i = 1. \end{cases} \tag{7}$$

Then it follows from Theorem 1 that,

$$\alpha = U(x, y), \qquad \beta = V(x, y). \tag{8}$$

2.2 Explicit solution

Let's solve system of Eqs. (4)–(7) with variables x_i, y_i, α, β, $i = 1, \dots, m$. It follows from expressions (4), (6), (8) that

$$\begin{aligned} \alpha &= U(x, y) = \sum_{j=1}^{m} u_{ij} y_j = f_u(c_i) y_i + \sum_{\substack{j=1 \\ j \neq i}}^{m} g_u(c_i) y_j \\ &= f_u(c_i) y_i - g_u(c_i) y_i + g_u(c_i) \sum_{j=1}^{m} y_j = y_i \big(f_u(c_i) - g_u(c_i) \big) + g_u(c_i). \end{aligned}$$

Therefore,

$$y_i = \frac{U(x, y) - g_u(c_i)}{f_u(c_i) - g_u(c_i)}, \quad i = 1, \dots, m. \tag{9}$$

By solving (5), (7), (8) we define that

$$x_i = \frac{V(x, y) - g_v(c_i)}{f_v(c_i) - g_v(c_i)}, \quad i = 1, \dots, m. \tag{10}$$

Then it follows from equalities $U(x, y)$ ir $V(x, y)$. From (6), (9) follows that

$$\sum_{i=1}^{m} y_i = \sum_{i=1}^{m} \frac{U(x, y) - g_u(c_i)}{f_u(c_i) - g_u(c_i)} = U(x, y) \sum_{i=1}^{m} \frac{1}{f_u(c_i) - g_u(c_i)} - \sum_{i=1}^{m} \frac{g_u(c_i)}{f_u(c_i) - g_u(c_i)} = 1.$$

Therefore,

$$U(x, y) = \frac{1 + \sum_{i=1}^{m} \frac{g_u(c_i)}{f_u(c_i) - g_u(c_i)}}{\sum_{i=1}^{m} \frac{1}{f_u(c_i) - g_u(c_i)}}. \tag{11}$$

By solving system of equalities (7), (9) one defines that

$$V(x, y) = \frac{1 + \sum_{i=1}^{m} \frac{g_v(c_i)}{f_v(c_i) - g_v(c_i)}}{\sum_{i=1}^{m} \frac{1}{f_v(c_i) - g_v(c_i)}}. \tag{12}$$

2.3 Search for the Nash equilibrium

In this paper a Modification of the Strategy Elimination Algorithm (MSE) is proposed when elimination of negative solutions $x_i < 0$ and $y_i < 0$ is performed by different players independently. The proof will be given that this algorithm provides the Nash equilibrium. The pseudo-code follows:

Algorithm 1 $MSE(m, f_v(c_1), \ldots, f_v(c_m), g_v(c_1), \ldots, g_v(c_m))$

1. $S_1 \leftarrow 0$, $S_2 \leftarrow 0$
2. **for** $i \leftarrow 1$ **to** m **do**
3. $\quad x_i \leftarrow 0$, $h_i \leftarrow i$, $f_i \leftarrow \frac{1}{f_v(c_i) - g_v(c_i)}$, $S_1 \leftarrow S_1 + f_i$, $S_2 \leftarrow S_2 + f_i g_v(c_i)$
4. $NoSolutions \leftarrow$ **true**, $n \leftarrow m$
5. **while** $NoSolutions$ **do**
6. $\quad NoSolutions \leftarrow$ **false**, $V \leftarrow \frac{1+S_2}{S_1}$, $j \leftarrow 1$
7. $\quad$ **for** $i \leftarrow 1$ **to** n **do**
8. $\quad\quad$ **if** $V \geqslant g_v(c_{h_i})$ **and** $f_{h_i} > 0$ **or** $V < g_v(c_{h_i})$ **and** $f_{h_i} < 0$ **then**
9. $\quad\quad\quad h_j \leftarrow h_i$, $j \leftarrow j + 1$
10. $\quad\quad$ **else**
11. $\quad\quad\quad S_1 \leftarrow S_1 - f_{h_i}$, $S_2 \leftarrow S_2 - f_{h_i} g_v(c_{h_i})$, $NoSolutions \leftarrow$ **true**
12. $\quad n \leftarrow j - 1$
13. **for** $i \leftarrow 1$ **to** n **do**
14. $\quad x_{h_i} \leftarrow (V - g_v(c_{h_i})) f_{h_i}$
15. **return** x

Steps 1–3 perform initialization, calculation of partial payoffs S_1, S_2 and output of parameter h_i defining the inspection objects remaining after elimination. Steps 4–12 are for elimination of negative solutions; during each iteration the non-negativity of solution x_{h_i} is tested (line 8). Iteration is repeated after the elimination of negative solutions x_{h_i} and updating of expected payoff $V^*(x^*, y^*)$. Algorithm stops after elimination of all negative solutions. Steps 13–15 perform output of solutions $x = (x_1, \ldots, x_m)$. The mixed strategy of the second player is defined similarly by the algorithm $MSE(m, f_u(c_1), \ldots, f_u(c_m), g_u(c_1), \ldots, g_v(u_m))$.

Theorem 2. *The MSE algorithm provides Nash Equilibrium* (*NE*) (x^*, y^*) *of General Inspection Problem if*

$$f_v(c_i) < g_v(c_i), \quad i = 1, \ldots, m, \tag{13}$$

$$f_u(c_j) < g_u(c_j), \quad j = 1, \ldots, m. \tag{14}$$

Proof. In the worst case, the MSE algorithm read lists $\sum_{i=0}^{m-1}(m-i)$ times, so the algorithm is finite and of polynomial complexity $O(m^2)$. MSE do not generate strictly negative solutions x^* ir y^* since $\sum_i x_i^* = 1$ ir $\sum_j y_i^* = 1$ and negative solutions are eliminated at each iteration during MSE steps 5–12 until the positive solution x^* is returned after the finite number of iterations If only positive solutions $x^* = (x_{p_1}, x_{p_2}, \ldots, x_{p_k})$ are produced then from Theorem 1 it follows that x^* is the NE regarding the objects $p_1, p_2, \ldots, p_k$. Now we show that x^* also is NE regarding all the objects. Suppose that $M = \{1, 2, \ldots, m\}$ and $P = \{p_1, p_2, \ldots, p_k\} \subseteq M$. Then from (13) it follows

$$\max\left(\left\{g_v(c_i) \,\middle|\, i \in M \setminus P\right\}\right) < V\left(x^*, y^*\right).$$

If $f_v(c_i) < g_v(c_i)$, $i \in P$, then

$$V_{\max} = \max\left(\left\{V\left(x^*, y^*\right)\right\} \cup \left\{\sum_{i=1}^{m} x_i v_{ij} \,\middle|\, j \in M \setminus P\right\}\right) = V\left(x^*, y^*\right),$$

since

$$\sum_{i=1}^{m} x_i v_{ij} = \left(f_v(c_j) - g_v(c_j)\right) x_j + g_v(c_j) \leqslant g_v(c_j) < V\left(x^*, y^*\right), \quad j \in M \setminus P,$$

therefore solution $x^* = (x_{p_1}, x_{p_2}, \dots, x_{p_k})$ satisfy equlibrium condition (2). If solution $y^* = (y_{r_1}, y_{r_2}, \dots, y_{r_l})$ is positive, then from Theorem 1 follows that it is NE regarding $r_1, r_2, \dots, r_l$ objects. Using arguments similar to those applied considering solution x^* we show that y^* is NE regarding all the inspection objects. Suppose that $R = \{r_1, r_2, \dots, r_l\}$. Then it follows from inequality (14) that

$$\max\bigl(\bigl\{g_u(j)\bigm| j \in M \setminus R\bigr\}\bigr) < U\bigl(x^*, y^*\bigr).$$

Consequently, the solution $y^* = (y_{r_1}, y_{r_2}, \dots, y_{r_l})$ satisfies equilibrium condition (1):

$$U_{\max} = \max\bigl(\bigl\{\bigl(f_u(j) - g_u(j)\bigr)y_j + g_u(j)\bigm| j \in R\bigr\} \cup \bigl\{g_u(j) \mid j \in M \setminus R\bigr\}\bigr) = U\bigl(x^*, y^*\bigr).$$

Therefore the solution of MSE algorithm (x^*, y^*) is NE regarding all the inspection objects. $\square$

2.4 Example, optimization of security system

Denote by (a_{ij}) a matrix describing system parameters

$$a_{i1} = p_i, \qquad a_{i2} = q_i, \qquad a_{i3} = g_i, \qquad a_{i4} = b, \quad p_i, q_i, g_i, b > 0,\ i = 1, \dots, m.$$

Here $M = \{1, 2, \dots, m\}$ is a set of objects protected by the system and the protective actionas are represented as

$$b_1 = (p_1, p_2, \dots, p_m), \quad b_2 = (q_1, q_2, \dots, q_m), \quad b_3 = (g_1, g_2, \dots, g_m), \quad b_4 = (b, b, \dots, b).$$

Parameters of protected object are denoted by

$$c_i = (p_i, q_i, g_i, b), \quad i = 1, \dots, m.$$

The payoff of the security system is represented by the matrix (u_{ij}):

$$u_{ij} = \begin{cases} p_i g_i - (1 - p_i) q_i g_i, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

The offenders payoff is described by the matrix (v_{ij}) with elements

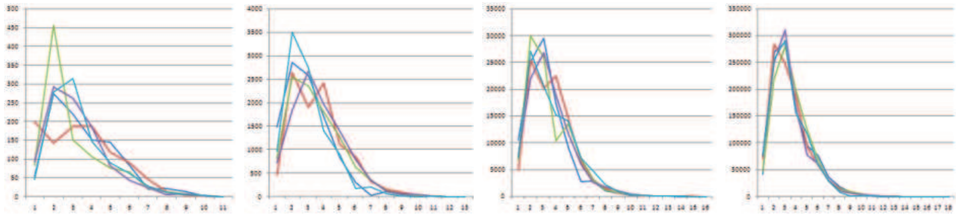
$$v_{ij} = \begin{cases} -p_i b + (1 - p_i) q_i g_i, & \text{if } i = j, \\ q_j g_j, & \text{if } i \neq j. \end{cases}$$

It follows from (9), (10), (11), (12) the explicit, not necessarily positive, solutions:

$$x_i = \frac{1 - \sum_{j=1}^m \frac{g_v(c_i) - g_v(c_j)}{f_v(c_j) - g_v(c_j)}}{(f_v(c_i) - g_v(c_i)) \sum_{j=1}^m \frac{1}{f_v(c_j) - g_v(c_j)}} = \frac{1 + \sum_{j=1}^m \frac{q_i g_i - q_j g_j}{p_j (b + q_j g_j)}}{p_i (b + q_i g_i) \sum_{j=1}^m \frac{1}{p_j (b + q_j g_j)}}, \quad i = 1, \dots, m,$$

$$y_j = \frac{1 - \sum_{i=1}^m \frac{g_u(c_j) - g_u(c_i)}{f_u(c_i) - g_u(c_i)}}{(f_u(c_j) - g_u(c_j)) \sum_{i=1}^m \frac{1}{f_u(c_i) - g_u(c_i)}} = \frac{(g_j(p_j - q_j + p_j q_j))^{-1}}{\sum_{i=1}^m \frac{1}{g_i(p_i - q_i + p_i q_i)}}, \quad j = 1, \dots, m.$$

According to Theorem 2, the NE solution (x, y) can be defined by corresponding iterations of MSE algorithm sequentially eliminating a set of protected objects m_i, where $g_i(p_i - p_i q_i + q_i) > 0$, $i = 1, \dots, m$, and using the explicit solution for the remaining objects.



$m = 10^3$ $m = 10^4$ $m = 10^5$ $m = 10^6$

Fig. 1. Convergence to Nash equilibrium.

3 Experimental investigation

The MSE algorithm was implemented by Maple16. The testing was performed by random generation of 100 samples of the U, V matrices of both the game models. All samples succeeded. The results are in Fig. 1 where the X axis shows the iteration number and the Y axis denotes the number of eliminated solutions during a single iteration. Different lines show different experiments.

4 Conclusion

The Generalized Inspector Problem (GIP) introduced in this paper extends the family of problems. The improved MSE algorithm provides solution of GIP with the convergence rate $O(m)$. The experiments show that the software implementation reflects the mathematical results and indicates the improved convergence rate $O(m \log m)$.

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] J. Mockus and M. Sabaliauskas. On the exact polynomial time algorithm for a special class of bimatrix game. *Liet. matem. rink. Proc. LMS, Ser. A*, **53**:85–90, 2013.

REZIUMĖ

Nešo pusiausvyra inspektoriaus uždavinyje

M. Sabaliauskas, J. Mockus

Šiame straipsnyje inspektoriaus uždavinys formuluojamas kaip ekonominė dvikova tarp įstatymų sergėtojo ir pažeidėjo. Šis uždavinys apibrėžiamas bimatrica, kuri nusako visas galimas lošimo baigtis. Inspektoriaus uždavinys laikomas išpręstu, jei apskaičiuojamos lošėjų strategijos, tenkinančios Nešo pusiausvyros sąlygas. Šiame darbe nagrinėjamas apibendrintas inspektoriaus uždavinys su tikslu jį pritaikyti platesnei uždavinių klasei, įskaitant saugumo sistemų optimizavimą. Pasiūlytas analitinis sprendinys bei modifikuotas strategijų eliminavimo algoritmas.

Raktiniai žodžiai: lošimų teorija, polinominis algoritmas, Nešo pusiausvyra.