AMO - Advanced Modeling and Optimization, Volume 19, Number 2, 2017

A Note on Solving 5-Person Game

R. Enkhbat[1], S. Batbileg[1],
A. Anikin[2], N. Tungalag[1], and A. Gornov[2]

[1] National University of Mongolia
renkhbat46@yahoo.com, batbileg@seas.num.edu.mn,
n.tungalag@num.edu.mn
[2] Matrosov Institute for System Dynamics and Control Theory, SB RAS, Russia
anton@anikin.xyz, gornov@icc.ru

Abstract. The nonzero sum 5-person game has been considered. It is well known that the game can be reduced to a global optimization problem [5, 7, 13]. By extending Mills' result [5], we derive global optimality conditions for a Nash equilibrium. In order to solve the problem numerically, we apply the Curvilinear Multistart Algorithm [2, 3] developed for finding global solutions in nonconvex optimization problems. The proposed algorithm was tested on five person games.

Keywords: Nash equilibrium, nonzero sum game, mixed strategies, curvilinear multistart algorithm.

1 Introduction

It is well known that game theory plays an important role in applied mathematics, mathematical modeling, economics and decision theory. Game theory is a powerful tool for modeling firm competitions at oligopoly market du to J.Nash [17]. There are many works devoted to game theory [6, 8–12, 4]. Most of them deals with zero sum two person games or nonzero sum two person games. Also, two person non zero sum game was studied in [10, 14, 15] by reducing it to D.C programming[1]. The three person game was examined in [2] by global optimization techniques.

This paper considers nonzero sum 5-person game. We consider 5-person game as a special case of the nonzero sum n-person game. Based on the results [16], we develop a computational algorithm for finding a Nash equilibrium. So far, less attention has been paid to computational aspects of game theory, specially N-person game. Aim of this paper to fulfill this gap. The paper is organized as follows. In Section 2, we formulate non zero sum 5-person game and show that it can be formulated as a global optimization problem with polynomial constraints. We formulate the problem of finding a Nash equilibrium for non zero sum 5-person games as a nonlinear programming problem. A Global search algorithm has been proposed in Section 3. Section 4 is devoted to computational experiments.

AMO - Advanced Modeling and Optimization. ISSN: 1841-4311

2 Nonzero Sum 5-person Game

Consider the 5-person game in mixed strategies with matrices $(A_q, \ \ q = 1, 2, 3, 4, 5)$ for players $1, 2, 3, 4, 5$.

$$A_q = \left(a^q_{i_1 i_2 i_3 i_4 i_5}\right), q = 1, 2, 3, 4, 5,$$

$$i_1 = 1, 2, \dots, k_1; \dots ; i_5 = 1, 2, \dots, k_5.$$

Denote by D_p the set

$$D_p = \{u \in \mathbb{R}^p \ |\sum_{i=1}^{p} u_i = 1, \ u_i \geq 0, \ i = 1, \dots, p\}, \ p = k_1, k_2, k_3, k_4, k_5.$$

A mixed strategy for player 1 is a vector $x^1 = (x^1_1, x^1_2, \dots, x^1_{k_1}) \in D_{k_1}$, where x^1_i represents the probability that player 1 uses a strategy i. Similarly, the mixed strategies for q-th player is $x^q = (x^q_1, x^q_2, \dots, x^q_{k_q}) \in D_{k_q}, \ q = 1, 2, 3, 4, 5$. Expected payoff for 1-th person is given by:

$$f_1(x^1, x^2, x^3, x^4, x^5) = \sum_{i_1=1}^{k_1} \sum_{i_2=1}^{k_2} \dots \sum_{i_5=1}^{k_5} a^1_{i_1 i_2 i_3 i_4 i_5} x^1_{i_1} x^2_{i_2} x^3_{i_3} x^4_{i_4} x^5_{i_5}.$$

and for q-th person

$$f_q(x^1, x^2, x^3, x^4, x^5) = \sum_{i_1=1}^{k_1} \sum_{i_2=1}^{k_2} \dots \sum_{i_5=1}^{k_5} a^q_{i_1 i_2 i_3 i_4 i_5} x^1_{i_1} x^2_{i_2} x^3_{i_3} x^4_{i_4} x^5_{i_5},$$

$$q = 1, 2, 3, 4, 5.$$

Definition 2.1 *A vector of mixed strategies* $\tilde{x}^q \in D_{k_q}, \ q = 1, 2, 3, 4, 5$ *is a Nash equilibrium if*

$$\begin{cases} f_1(\tilde{x}^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, \tilde{x}^5) \geq f_1(x^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, \tilde{x}^5), \ \forall x^1 \in D_{k_1} \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ f_q(\tilde{x}^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, \tilde{x}^5) \geq f_q(\tilde{x}^1, \dots, \tilde{x}^{q-1}, x^q, \tilde{x}^{q+1}, \dots, \tilde{x}^5), \ \forall x^q \in D_{k_q} \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ f_5(\tilde{x}^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, \tilde{x}^5) \geq f_5(\tilde{x}^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, x^5), \ \forall x^5 \in D_{k_5}. \end{cases}$$

It is clear that

$$\begin{array}{l} f_1(\tilde{x}^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, \tilde{x}^5) = \max_{x^1 \in D_{k_1}} f_1(x^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, \tilde{x}^5), \\ \dots \quad \dots \quad \dots \quad \dots \\ f_q(\tilde{x}^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, \tilde{x}^5) = \max_{x^q \in D_{k_q}} f_q(\tilde{x}^1, \dots, \tilde{x}^{q-1}, x^q, \tilde{x}^{q+1}, \dots, \tilde{x}^5), \\ \dots \quad \dots \quad \dots \quad \dots \\ f_5(\tilde{x}^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, \tilde{x}^5) = \max_{x^5 \in D_{k_5}} f_n(\tilde{x}^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, x^5). \end{array}$$

Denote by

$$\sum_{i_1=1}^{k_1}\sum_{i_2=1}^{k_2}\dots\sum_{i_{q-1}=1}^{k_{q-1}}\sum_{i_{q+1}=1}^{k_{q+1}}\dots\sum_{i_5=1}^{k_5} a^q_{i_1 i_2 i_3 i_4 i_5} x^1_{i_1}\dots x^{q-1}_{i_{q-1}} x^{q+1}_{i_{q+1}}\dots x^5_{i_5} \triangleq$$
$$\triangleq \varphi_{i_q}(x^1,\dots,x^{q-1},x^{q+1},\dots,x^5) = \varphi_{i_q}(x|x^q),$$
$$i_q = 1,2,\dots,k_q, \quad q = 1,2,3,4,5.$$

For further purpose, it is useful to formulate the following statement.

Theorem 2.2 *[16] A vector strategy* $(\tilde{x}^1,\tilde{x}^2,\tilde{x}^3,\tilde{x}^4,\tilde{x}^5)$ *is a Nash equilibrium if and only if*

$$f_q(\tilde{x}) \geq \varphi_{i_q}(\tilde{x}|\tilde{x}^q) \tag{1}$$

for

$$\tilde{x} = (\tilde{x}^1,\tilde{x}^2,\tilde{x}^3,\tilde{x}^4,\tilde{x}^5)$$
$$i_q = 1,2,\dots,k_q,$$
$$q = 1,2,3,4,5.$$

Theorem 2.3 *[16] A mixed strategy* $\tilde{x}$ *is a Nash equilibrium for the nonzero sum* 5*-person game if and only if there exists vector* $\tilde{p} \in \mathbb{R}^5$ *such that vector* $(\tilde{x},\tilde{p})$ *is a solution to the following bilinear programming problem:*

$$\max_{(x,p)} F(x,p) = \sum_{q=1}^{n} f_q(x^1,x^2,x^3,x^4,x^5) - \sum_{q=1}^{5} p_q \tag{2}$$

subject to :

$$\varphi_{i_q}(x|x^q) \leq p_q, \quad i_q = 1,2,\dots,k_q, \tag{3}$$

3 The Curvilinear Multistart Algorithm

In order to solve the problem, we use curvilinear multistart algorithm. The algorithm was originally developed for solving box-constrained optimization problems, therefore, we convert our problem from the constrained to unconstrained form using penalty function techniques. For each equality constraint $g(x) = 0$, we construct a simple penalty function $\hat{g}(x) = g^2(x)$. For each inequality constraint $q(x) \leq 0$, we also construct the corresponding penalty function as follows:

$$\hat{q}(x) = \begin{cases} 0, & \text{if } q(x) \leq 0, \\ q^2(x), & \text{if } q(x) > 0. \end{cases}$$

Thus, we have the following box-constrained optimization problem:

$$\hat{f}(x) = f(x) + \frac{\gamma}{2} \sum_i \hat{g}_i(x) + \frac{\gamma}{2} \sum_j \hat{q}_j(x) \to \min_X,$$

$$X = \left\{ x \in \mathbb{R}^n | \underline{x_i} \le x_i \le \overline{x_i},\ i = 1, ..., n \right\}.$$

where γ is a penalty parameter, $\underline{x}$ and $\overline{x}$ - are lower and upper bounds. For original x-variables the constraint is the box $[0, 1]$; for p-variables box constraints are $[0, \overline{p_q}]$. Values of $\overline{p_q}$ are chosen from some intervals. An initial value of a penalty parameter γ is chosen not too large (something about 1000) and after finding some local minimums we increase it for searching another local minimum.

Algorithm 1 The Curvilinear Multistart Algorithm

Input: $x^1 \in X$ – initial (start) point; $K > 0$ – iterations count; $\delta > 0$; $N > 0$; $\varepsilon_\alpha > 0$ — algorithm parameters.
Output: Global minimum point x^* and $f^* = f(x^*)$

for $k \leftarrow 1, K$ **do** $f^k \leftarrow f(x^k)$
  generate stochastic point $\tilde{x}^1 \in X$
  generate stochastic point $\tilde{x}^2 \in X$
  generate stochastic α-grid:

$$-1 = \alpha_1 \le .... \le \alpha_i \le -\delta \le 0 \le \delta \le \alpha_{i+1} \le ... \le \alpha_N = 1$$

  Let $\hat{x}(\alpha) = \mathrm{Proj}_X \left(\alpha^2 \left((\tilde{x}^1 + \tilde{x}^2)/2 - x^k \right) + \alpha/2 \left(\tilde{x}^2 - \tilde{x}^1 \right) + x^k \right)$ where $\mathrm{Proj}_X(z)$ - projection of point z onto set X.
  //note that $\hat{x}(-1) = \hat{x}^1$, $\hat{x}(1) = \hat{x}^2$, $\hat{x}(0) = x^k$.
  $f_*^k \leftarrow f^k$
  $\alpha_*^k \leftarrow 0$
  for $i \leftarrow 1, (N-2)$ **do**
//Convex triplet
    if $f(\hat{x}(\alpha_i)) > f(\hat{x}(\alpha_{i+1}))$ **and** $f(\hat{x}(\alpha_{i+1})) < f(\hat{x}(\alpha_{i+2}))$ **then**
//Refining the value of minima using
//Golden-Section search method with accuracy ε_α
      $\alpha_*^k \leftarrow$ GoldenSectionSearch$(f, \alpha_i, \alpha_{i+1}, \alpha_{i+2}, \varepsilon_\alpha)$
      if $f(\hat{x}(\alpha_*^k)) < f_*^k$ **then**
        $f_*^k \leftarrow f(\hat{x}(\alpha_*^k))$
        $\alpha_*^k \leftarrow \alpha_*^k$
      end if
    end if
  end for
//Start local optimization algorithm
  $x^{k+1} \leftarrow$ LOptim$(\hat{x}(\alpha_*^k))$
end for
$x^* \leftarrow x^k$
$f^* \leftarrow f(x^k)$

The proposed algorithm starts from some initial point $x^1 \in X$. At each $k-$th iteration the algorithm performs randomly "drop" of two auxiliary points $\tilde{x}^1$ and $\tilde{x}^2$ and generating a curve (parabola) which passes through all three points x^k, $\tilde{x}^1$ and $\tilde{x}^2$. Then we generate some random grid along this curve and try to found all convex triples inside the grid. For each founded triple we perform refining the triple minima value with using golden section method. The best triple became a start point for local optimization algorithm, the final point of which will be a start point for the next iteration of global method. Details are presented in Algorithms 1 and 2.

Algorithm 2 The Local Optimization Algorithm

Input: $x^1 \in X$ – initial (start) point; $\varepsilon_x > 0$ — accuracy parameter.
Output: Local minimum point x^* and $f^* = f(x^*)$

```
repeat
    d^k = x^k − Proj_X(x^k − ∇f(x^k))
    //Perform local relaxation step, for example, with using standard convex interval capture technique.
    x^{k+1} = argmin_{α≥0} f(x^k + αd^k)
until ||x^{k+1} − x||_2 ≤ ε_x
```

4 Computational Experiments

The proposed method was implemented in C language and tested on compatibility with using the GNU Compiler Collection (GCC, versions: 4.8.5, 4.9.3, 5.4.0), clang (versions: 3.5.2, 3.6.2, 3.7.1, 3.8) and Intel C Compiler (ICC, version 15.0.6) on both GNU/Linux, Microsoft Windows and Mac OS X operating systems.

The proposed algorithm was applied for numerically solving number of problems with 5 players. The problems were created by the well-known GAMUT [18] generator. In all cases, Nash equilibrium points were found successfully.

Problem 1. GAMUT Random Game $2 \times 2 \times 2 \times 2 \times 2$:

f^*	$x^1_*, x^2_*, x^3_*, x^4_*, x^5_*$	$p^*_1, p^*_2, p^*_3, p^*_4, p^*_5$
$-3.09 \cdot 10^{-5}$	$x^1_* = (0.94, 0.06)$	$p^*_1 = 31.97$
	$x^2_* = (0.58, 0.42)$	$p^*_2 = 54.01$
	$x^3_* = (1, 0)$	$p^*_3 = 60.87$
	$x^4_* = (0.9, 0.1)$	$p^*_4 = 67.78$
	$x^5_* = (0.99, 0.01)$	$p^*_5 = 41.19$
$6.12 \cdot 10^{-8}$	$x^1_* = (0.93, 0.07)$	$p^*_1 = 32.06$
	$x^2_* = (0.59, 0.41)$	$p^*_2 = 53.77$
	$x^3_* = (1, 0)$	$p^*_3 = 61.05$
	$x^4_* = (0.89, 0.11)$	$p^*_4 = 67.27$
	$x^5_* = (1, 0)$	$p^*_5 = 42.12$

Problem 2. GAMUT Random Game $6 \times 2 \times 4 \times 5 \times 3$:

f^*	$x^1_*, x^2_*, x^3_*, x^4_*, x^5_*$	$p^*_1, p^*_2, p^*_3, p^*_4, p^*_5$
$1.86 \cdot 10^{-3}$	$x^1_* = (0, 0, 0.66, 0.29, 0, 0.05)$	$p^*_1 = 51.87$
	$x^2_* = (0.49, 0.51)$	$p^*_2 = 44.4$
	$x^3_* = (0.18, 0.32, 0.5)$	$p^*_3 = 53.17$
	$x^4_* = (0, 0.19, 0.06, 0.07, 0.68)$	$p^*_4 = 48.81$
	$x^5_* = (0.42, 0.1, 0.48)$	$p^*_5 = 44.8$
$-7.83 \cdot 10^{-5}$	$x^1_* = (0, 0.29, 0.48, 0, 0.23)$	$p^*_1 = 53.65$
	$x^2_* = (0.35, 0.65)$	$p^*_2 = 37.89$
	$x^3_* = (0, 0.58, 0.22, 0.20)$	$p^*_3 = 51.54$
	$x^4_* = (0, 0.78, 0, 0.19, 0.03)$	$p^*_4 = 52.08$
	$x^5_* = (0.17, 0.17, 0.66)$	$p^*_5 = 53.54$

References

1. Strekalovsky A.S. (1998) Global Optimality Conditions for Nonconvex Optimization. Journal of Global Optimization 12: 415–434.
2. Enkhbat R, Tungalag N., Gornov A., Anikin A. The Curvilinear Search Algorithm for Solving Three-Person Game. Proc. DOOR 2016. CEUR-WS. 2016. Vol. 1623. P. 574–583. http://ceur-ws.org/Vol-1623/paperme4.pdf.
3. Gornov A., Zarodnyuk T. (2014) Computing technology for estimation of convexity degree of the multiextremal function. Machine learning and data analysis 10 (1): 1345–1353. http://jmlda.org
4. Mangasarian O. L. and Stone H. (1964) Two-Person Nonzero Games and Quadratic Programming. J. Mathemat. Anal. Appl. 9: 348–355.
5. Mills H. (1960) Equilibrium Points in Finite Games. J. Soc. Indust. Appl. Mathemat. 8 (2): 397–402.
6. Von Neumann, J., Morgenstern, O. (1944) Theory of games and economic behavior. Princeton University Press
7. Melvin Dresher (1981) The Mathematics of Games of Strategy. Dover Publications.
8. Vorobyev N.N. (1984) Noncooperative games. Nauka.
9. Germeyer YU.B. (1976) Introduction to Operation Reseach. Nauka.
10. Strekalovsky A.S and Orlov,A.V. (2007) Bimatrix Game and Bilinear Programming. Nauka.
11. Owen,G. (1971) Game Theory. Saunders, Philadelphia.
12. Gibbons R. (1992) Game Theory for Applied Economists. Princeton University, Press.
13. Howson J. T. (1972) Equilibria of Polymatrix Games. Management Sci 18: 312–318.
14. Strekalovsky, A. S. and Enkhbat, R. (2014) Polymatrix games and optimization problems. Automation and Remote Control 75(4), 632–645.
15. Orlov Andrei V. and Strekalovsky Alexander S. and Batbileg, S. (2014) On computational search for Nash equilibrium in hexamatrix games. Optimization Letters 10 (2): 369–381.
16. R. Enkhbat and S. Batbileg and N. Tungalag and A. Anikin, A. Gornov (2017) A Computational Method for Solving N-Person Game. The Bulletin of Irkutsk State University, Series ”Mathematics”, Vol. 20.
17. Nash, J.F. Equilibrium points in n-person games. Proc. of the Nat. Acad. of Sci. USA 36, 48–49 (1950).
18. GAMUT – suite of game generators. `http://gamut.stanford.edu`