

戦略数 $m \times n$ のナッシュ均衡の計算

日大生産工(学部) ○平山 圭太

日大生産工 吉田 典正

日大生産工 飯沼 守彦

日大生産工 齋藤 敏雄

1. はじめに

私たちは、日常生活においてさまざまな意思決定を行っている。私たちの意思決定はほかの人々の意思決定に影響を及ぼし、逆に、他の人々の意思決定は私たちの意思決定に影響を及ぼすよう、相互に影響し合っている。このような状況をゲーム的状况といい、数理的に分析する意思決定の理論をゲーム理論²⁾という。

本研究では、文献¹⁾のサポート列举の手法を用い、ゲーム理論における非協力ゲームの解の一種であるナッシュ均衡^{1,2,3,4)}を求めるプログラムを作成し、戦略数がプレイヤーごとに異なる $m \times n$ のナッシュ均衡の計算の測定を行う。

2. ナッシュ均衡の計算方法

ナッシュ均衡の計算には、純粋戦略と混合戦略の2種類の方法がある。表1の利得行列を例に計算方法を説明する。なお、表において、左をAの利得、右をBの利得とする。また、利得行列は、縮退していない¹⁾ものとする。

表1 3×3 の利得行列

A/B	B1	B2	B3
A1	-28 5	6 46	-40 11
A2	-17 33	-49 37	41 1
A3	16 49	-22 -33	-23 -28

2.1 純粋戦略

① B1に対してA1の利得がA2, A3と比較し最大か判断する。

② A1が最大ならば、A1に対してB1の利得がB2, B3と比較し最大か判断する。

③ この処理を全ての A_i, B_i に対して繰り返し、A, Bともに最大の場所があればナッシュ均衡である。表1のナッシュ均衡は $((1, 0, 0), (0, 1, 0))$ と $((0, 0, 1), (1, 0, 0))$ である。

2.2 混合戦略

① A, Bの戦略の集合をそれぞれ

$M = \{0, \dots, m-1\}, N = \{m, \dots, m+n-1\}$ とする。

$k = 2, \dots, \min(m, n)$ に関して、 M, N の k 個の要素を持つ部分集合を I, J とし、次の処理を行う。

② I, J に対して、式(1)を解き、Aの戦略 $x = (x_0, \dots, x_{m-1})$ とBの戦略 $y = (y_m, \dots, y_{m+n-1})$ を求める。表1で $I = \{0, 2\}, J = \{3, 4\}$ の場合の例を式(2), (3)に示す。

$$\sum_{i \in I} x_i b_{ij} = v \text{ for } j \in J, \sum_{i \in I} x_i = 1$$

$$\sum_{j \in J} a_{ij} y_j = u \text{ for } i \in I, \sum_{j \in J} y_j = 1 \quad (1)$$

$$\begin{bmatrix} x_0 & x_2 & v \end{bmatrix} \begin{bmatrix} 5 & 46 & 1 \\ 49 & -33 & 1 \\ -1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$\Rightarrow x_0 = \frac{2}{3}, x_2 = \frac{1}{3}, v = \frac{59}{3}$$

$$\begin{bmatrix} -28 & 6 & -1 \\ 16 & -22 & -1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_3 \\ y_4 \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

$$\Rightarrow y_3 = \frac{7}{18}, y_4 = \frac{11}{18}, u = -\frac{65}{9}$$

③ 戦略の組 x, y が最適応答戦略になっているか調べ、最適応答戦略であれば、ナッシュ均衡である。 x が y に対して最適応答戦略であるとは、式(4)の条件を満たすことである。 y に関しても同様である。

$$x_i > 0 \Rightarrow (Ay)_i = u = \max \{ (Ay)_k \mid k \in M \} \quad (4)$$

式(2), (3)の例では、

$$\begin{bmatrix} \frac{2}{3} & 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 5 & 46 & 11 \\ 33 & 37 & 1 \\ 49 & -33 & -28 \end{bmatrix} = \begin{bmatrix} \frac{59}{3} & \frac{59}{3} & -2 \end{bmatrix}$$

$$\begin{bmatrix} -28 & 6 & -40 \\ -17 & -49 & 41 \\ 16 & -22 & -23 \end{bmatrix} \begin{bmatrix} \frac{7}{18} \\ \frac{11}{18} \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{65}{9} \\ -\frac{329}{9} \\ -\frac{65}{9} \end{bmatrix}$$

表 2. A,B の戦略数に対するナッシュ均衡の計算時間(sec)

A/B	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0.000186	0.000194	0.000204	0.000222	0.000236	0.000252	0.000267	0.000295	0.000327	0.000358	0.000394	0.000434	0.000476	0.0005245
3	-	0.000221	0.000259	0.000305	0.000397	0.000526	0.00067	0.000872	0.001121	0.001428	0.001759	0.002167	0.002635	0.0031936
4	-	-	0.000368	0.00058	0.000878	0.001347	0.00204	0.002941	0.00414	0.005852	0.007901	0.010567	0.013782	0.0177204
5	-	-	-	0.001056	0.00193	0.003435	0.005744	0.009134	0.014359	0.021688	0.031037	0.043866	0.062463	0.0839197
6	-	-	-	-	0.004071	0.007945	0.014564	0.02572	0.042485	0.071799	0.111018	0.175429	0.254169	0.350973
7	-	-	-	-	-	0.016952	0.0356	0.0649	0.120736	0.207119	0.342944	0.558641	0.877373	1.32438
8	-	-	-	-	-	-	0.076021	0.156759	0.302627	0.56217	0.961998	1.64869	2.71674	4.29187
9	-	-	-	-	-	-	-	0.332338	0.709362	1.34946	2.53818	4.4431	7.53363	12.6335
10	-	-	-	-	-	-	-	-	1.46392	2.95446	5.99522	11.2592	20.4004	34.8447
11	-	-	-	-	-	-	-	-	-	6.45525	13.5289	27.2762	50.0331	92.1095
12	-	-	-	-	-	-	-	-	-	-	29.5602	60.3725	119.852	228.571
13	-	-	-	-	-	-	-	-	-	-	-	132.281	269.067	541.385
14	-	-	-	-	-	-	-	-	-	-	-	-	561.679	1182.1
15	-	-	-	-	-	-	-	-	-	-	-	-	-	2518.35

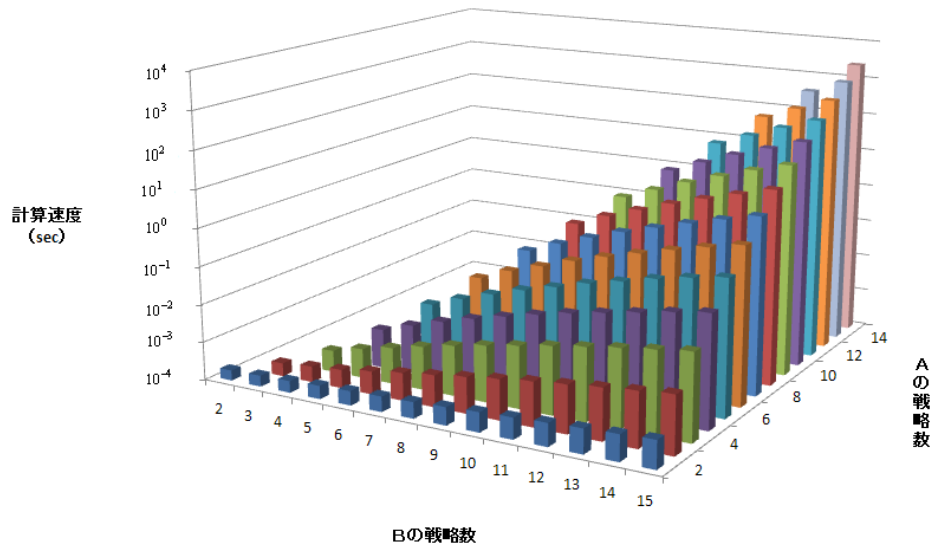


図 1. 計算時間の変化のグラフ

となり、ナッシュ均衡である。

3. 計算結果

ナッシュ均衡を求めるプログラムを作成し、A,B の利得を乱数で与え、A, B の戦略数がそれぞれ 15 個までの計算結果を表 2 および図 1 に示す。今回の実験では、戦略数 $(2 \times 2) \sim (9 \times 15)$ を 100 回、 $(10 \times 10) \sim (11 \times 15)$ を 10 回、 $(12 \times 12) \sim (13 \times 15)$ を 5 回、 $(14 \times 14) \sim (14 \times 15)$ を 2 回、 (15×15) を 1 回、それぞれ繰り返し計算しその平均をとった。なお、使用した CPU は Core i7-2600K 3.4GHz であり、単位は sec である。また、「 2×3 」と「 3×2 」の計算量は等しいため、表 2 の下半分は省略する。

4. まとめ

本研究では、 $m \times n$ の利得行列のナッシュ均衡を求めるプログラムを作成し、計算時間を測定した。今後は他の計算方法などを検証し、より効

率的な計算方法の実装を行っていく予定である。

参考文献

- 1) B. von Stengel, Equilibrium Computation for Two-Player Games in Strategic and Extensive Form, in Algorithmic Game Theory, Cambridge, 2007.
- 2) 岡田章, ゲーム理論, 有斐閣, 1996.
- 3) Naoki Shiba, Numerical Computation for Identifying Nash Equilibria Using Shapley's Labeling Method, International Conference on Innovative Computation, Information and Control, 2009.
- 4) 柴直樹 後藤雅樹, 有限ゲームにおけるナッシュ均衡の数値的解法, 経営情報学会春季全国研究発表大会, 2009.
- 5) 武藤滋夫, ゲーム理論入門, 日本経済新聞出版社, 2001.