# R-Rivals のナッシュ均衡戦略

田中哲朗[1,a)]

**概要**：R-Rivals はカナイセイジ氏がデザインした 2 人用カードゲームで 2014 年に発売された．R-Rivals はサイコロのようなランダムな要素を含まない 2 人零和ゲームだが，同時手番ゲームなのでじゃんけんと同様にナッシュ均衡が存在するのは双方のプレイヤが混合戦略を用いたときになる．
本研究では，同値な状態は同じであるとみなして，R-rivals のすべての状態を数えた．そして，得られた約 54 万の状態すべてのゲーム値を計算した．その結果，初期状態で出せる 8 枚のカードのうち，道化，姫，将軍を除く 5 枚のカードをある確率で出すのが最適な戦略であること，ポイントが 0-3 から勝てる状態があることなどの性質をみつけることができた．

# Nash Equilibrium Strategy for R-Rivals

Tetsuro Tanaka[1,a)]

***Abstract:*** R-Rivals is a two-player card game designed by Seiji Kanai and released in 2014. R-Rivals is a two-player zero-sum game that does not contain random elements like dice, but since it is a simultaneous turn game, it has a Nash equilibrium of mixed strategies like rock-paper-scissors.
In this study, we counted all the states of R-rivals, considering equivalence to be the same. Then, we calculated the game value of all the obtained states (about 540,000). As a result, we found various properties, such as the optimal strategy is to play five cards out of the eight cards that can be played in the initial state, except for the clown, princess, and general, with a certain probability, and that there are many states where the score can be won from 0-3.

## 1. はじめに

R-Rivals はカナイセイジ氏がデザインした 2 人用カードゲーム「R」をリメイクして 2014 年に発売されたカードゲームである．以下にゲームの概要を記す [*1]．

- 2 人のプレイヤで遊ぶ．
- 各プレイヤ用に，それぞれ 8 種類 (道化，姫，密偵，暗殺者，大臣，魔術師，将軍，王子) のカード 1 枚ずつの合計 16 枚のカードを使う．カードには種類ごとに「強さ」(0 から 7) が割り当てられている．
- 各プレイヤはゲーム開始時に 8 種類のカード 1 枚ずつの手札を持ってゲームを開始する．
- 通常，各プレイヤがカード 1 枚ずつ選んで伏せて出し (以下では「同時出し」と呼ぶ)，同時に表にして勝敗を確認する．これを「勝負」と呼ぶ．通常は「強さ」の大きいカードが「勝負」に勝つが，カードの「能力」により「強さ」の大小以外が勝敗に影響することがある．
- 1 回「勝負」に勝つと，通常 1 ポイントを獲得する．獲得ポイントの合計が 4 ポイントを超えたらそのゲームはそのプレイヤの勝ちとなる．
- 「勝負」に引き分けると，次の「勝負」に持ち越しとなる．引き分けが続いた後に，ある「勝負」で勝敗が決定すると，そこで勝ったプレイヤが引き分けが続いたすべての「勝負」に勝ったものとしてポイントを獲得する．
- どちらのプレイヤも 4 ポイント以上を獲得せずに 8 回の「勝負」を終えた場合はそのゲームは引き分けとなる．
- 8 種類のカードはそれぞれ以下の能力を持つ．カッコ

[1] 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
[a)] ktanaka@g.ecc.u-tokyo.ac.jp
[*1] ゲームの説明書にはさまざまなオプションルールが紹介されているが，本研究では基本ルールのみを対象とする．

の中は「強さ」を表す

**道化 (0)** 相手が「魔術師」のときは負けるが，それ以外は引き分けとなる．

**姫 (1)** 相手が「王子」の時は，「勝負」だけでなくゲームに勝つ．

**密偵 (2)** 次の「勝負」で相手はカードを先に出して表にして (以下では「先出し」と呼ぶ)，それを見て自分がカードを出せる (以下では「後出し」と呼ぶ)．双方が密偵を出した場合は能力がキャンセルされて次の「勝負」も同時に出す．

**暗殺者 (3)** 相手が「王子」を出した場合は負けるが，それ以外ではその「勝負」は「強さ」の小さい方が勝つ．

**大臣 (4)** このカードで勝つと 2 ポイントを獲得するものとする．この「勝負」も含めて引き分けが続いて，最後に勝ったときはこの「勝負」について 2 ポイントが獲得する．

**魔術師 (5)** 「同じ「勝負」で出された」相手のカードの能力を無効化する．前の「勝負」出された将軍の効果は無効化できない．

**将軍 (6)** 次の「勝負」で出す自分のカードの「強さ」に 2 を加える．同じ「勝負」で相手が魔術師を出した場合は，この効果は無効化されて次の「勝負」のカードの「強さ」は変化がない．一方，次の「勝負」で相手が魔術師を出しても次の「勝負」での「強さ」への「+2」は有効である．

**王子 (7)** 相手が「姫」の時はゲームに負ける．

「暗殺者」のカードの効果を「強さ」を逆転させると考えると，双方が「暗殺者」を出した時に「強さ」の大きいほうが勝つと考えることも可能である．通常は双方が「暗殺者」を出した時は「強さ」が等しいが，事前に片方が「将軍」を出した時は，「強さ」が異なるので問題になる．本研究ではこのカードの効果を「強さ」の小さい方が勝つとして，双方が「暗殺者」を出した時も「強さ」の小さい方が勝つものとする．これを変更したときの影響については，第 4.6 節で解析する．

各プレイヤの視点からは各「勝負」の前の段階で，完全な情報が得られている．また，ゲームの勝ちで得られる利得を 1, 負けを -1, 引き分けを 0 とすると，零和ゲームとなっている．

同時着手ゲームなので，ナッシュ均衡を実現するためには「じゃんけん」と同じように同じ状態 (state) で確率的にアクション (着手) を決定する混合戦略 (mixed strategy) が必要になる．

このゲームについては，文献 [1] でいくつかの経験的な戦略に関する解析がおこなわれているが，本研究では，R-Rivals の全状態の数がそれほど大きくないことを利用して，ナッシュ均衡を与える戦略を求める．この解析結果を用いて，R-Rivals というゲームに関するさまざまな性質を得ることができる．

## 2. 状態の表現

各プレイヤが混合戦略を用いるという前提で，各プレイヤの戦略 (strategy) は，プレイヤの視点での状態 $s \in S$ で可能なアクションの集合を $A(s)$ としたとき，各アクション $a_i \in A(s)$ を選択する確率 $\pi(s) = (p_1, p_2, \ldots, p_{|A(s)|})$ で表せる．

各プレイヤが手札を出すタイミングでの状態数を求める．一般的にはそれまでの「勝負」で各プレイヤが出したカードの履歴 (history) で状態を表現することは可能であるが，状態数が大きくなる [*2]．だが，R-Rivals ではカードの能力の多くがその「勝負」か，次の「勝負」までしか効力を持たないので，以下の情報だけで状態を表現できる．

$mask_{1,2}$ 各プレイヤの残りカード．8 種類のカードの有無なので，各プレイヤごとに 0 から $2^8 - 1$ までの整数で表せる．

$score_{1,2}$ 「勝負」前の各プレイヤの獲得ポイント．それぞれ 0-3 まで．

$wscore_{1,2}$ 次の「勝負」に各プレイヤが勝った時に加算されるポイント．獲得ポイントと合わせて 4 を超える場合は，「4-獲得ポイント」としても同じ状態とみなせる．引き分けが重なった時に「勝負」に勝つと，引き分けだった「勝負」に勝ったものとしてポイントを獲得するため．途中で「大臣」が出されることもあるので，各プレイヤが勝った時のポイントはプレイヤごとに定義している．

$spy$ 前の「勝負」でプレイヤ 1 のみが「密偵」を出した時は 1, プレイヤ 2 のみが「密偵」を出した時は-1 とする．なお，どちらも「密偵」を出さなかったとき，相手が「魔術師」を出して能力が無効化されたとき，両プレイヤが「密偵」を出したときは 0 とする．

$dgeneral$ 前の「勝負」が引き分けで，プレイヤ 1 のみが「大臣」を出した時は＋ 2, プレイヤ 2 のみが「大臣」を出した時は -2 とする．

$prevcard$ 通常は -1 だが，手番のプレイヤが前の「勝負」で出した「密偵」の能力が有効な「勝負」では，まず相手が手札を出すのでそのカードをここに入れる．

これにより，状態数を 537,103 に減らすことができた．表 1 に自分の残りカード数ごとの状態数を示す．

なお，状態数を更に減らすためのアイデアとして，`(([6, 7], [4, 7]), (3, 3), (1, 1), 0, 0, -1)` と `(([6, 7], [4, 7]), (2, 3), (2, 1), 0, 0, -1)` を

[*2] 8 回目の「勝負」の前の時点での履歴の上限は，${}_8P_7 \times {}_8P_7 = 25,401,600$ で，それまでに勝敗が決定している場合もあるのでここから減る．7 回目までは，もっと少ないので計算機で扱える範囲ではある．

表 1　残りカード枚数ごとの状態数

| 残りカード枚数 | 同時出し | 先出し | 後出し |
|---|---|---|---|
| 8 | 1 | 0 | 0 |
| 7 | 50 | 6 | 42 |
| 6 | 1902 | 273 | 1638 |
| 5 | 27872 | 4574 | 22870 |
| 4 | 110786 | 22425 | 89700 |
| 3 | 102091 | 25777 | 77331 |
| 2 | 25374 | 6779 | 13558 |
| 1 | 2600 | 727 | 727 |

区別しないというアイデアもある．現在のポイントと次の「勝負」を勝ったときのポイントが異なっているが，どちらも，次の「勝負」を勝った方が即，勝つ状態であり負けた時のポイントが何点になるかは影響がないので，同一視しても良い．ただし，基本的にはゲーム木の末端近くなので，状態数の削減には大きく寄与しないと考えて，現時点では採用していない．

## 3. ナッシュ均衡戦略の計算

全部で 537,103 の状態に対して，ナッシュ均衡での手番のプレイヤにとっての利得の期待値と，それを与えるナッシュ均衡を与えるポリシーを一つ求めることにする．

自分の残りカード数が 1 の状態は，そのカードを出してゲームの勝敗が決定するので，それを利得とする．自分の残りカード数 n におけるナッシュ均衡での手番プレイヤの利得の期待値がすべて求まっていると仮定すると，以下のようにして，自分の残りカード数 n+1 におけるナッシュ均衡での手番のプレイヤにとっての利得の期待値と，それを与えるナッシュ均衡を与えるポリシーを求めることができる．

**同時出し**　両プレイヤの可能なアクションの組合せで，次の状態の利得行列を作成して，線形計画法で利得の期待値とナッシュ均衡ポリシーの一つを得られる．

**先出し**　深さ 2 の min-max 木を作成する．木の末端は自分の残りカード数が 1 つ少ない状態になるので利得の期待値が計算できる．その木で得られる min-max 値が元の状態の利得の期待値となり，木の根での bestmove を確率 1 で選択する手がナッシュ均衡ポリシーとなる．

**後出し**　深さ 1 の max 木を作成する．木の末端は自分の残りカード数が 1 つ少ない状態になるので利得の期待値が計算できる．その木で得られる max 値が元の状態の利得の期待値となり，木の根での bestmove を確率 1 で選択する手がナッシュ均衡ポリシーとなる．

同時アクションの二人零和ゲームの混合戦略のナッシュ均衡を線形計画法で求める手法は文献 [5], [6] にあるように古くから知られているが，以下では具体的に説明する．

各プレイヤが $n$ 枚のカードを持っているとき，プレイヤ 1 が可能なアクションを $action_1$, プレイヤ 2 が可能なアクションを $action_2$ として以下のように表す．

$$action_1 = \{A_0, A_1, A_2, \ldots, A_{n-1}\}$$

$$action_2 = \{B_0, B_1, B_2, \ldots, B_{n-1}\}$$

ここで，それぞれが $A_i, B_j$ を選択した時の，プレイヤ 1 の立場からの利得の期待値を $R_{i,j}$ とする．二人零和ゲームを課程しているので，プレイヤ 2 の立場からの利得の期待値は $-R_{i,j}$ となる．ここで，プレイヤ 1 が確率 $\pi = \{p_0, p_1, \ldots, p_{n-1}\}$ でそれぞれの手を選択するものとする．確率の定義により，

$$\sum_{i=0}^{n-1} p_i = 1$$

$$0 \leq p_i (0 \leq i \leq n-1)$$

となる．敵が $B_j$ を選択した時の自分の利得は $R(j, \pi) = \sum_{i=0}^{n-1} p_i R_{i,j}$ となる．相手が最善の手を選択した時の自分の利得を $v$ とすると，$0 \leq j \leq n-1$ の $j$ について，

$$v \leq \sum_{i=0}^{n-1} p_i R_{i,j}$$

が成り立つ．この制約のもとで，$v$ を最大化する線形計画法を解く．

$$x = (v, p_0, p_1, \ldots, p_{n-1})^T$$

$$c = (-1, 0, 0, \ldots, 0)^T$$

として，$c_T x$ を最小化することになる．

$$\sum_{i=0}^{n-1} p_i = 1$$

の条件は，

$$A_{eq} = (0, 1, 1, \ldots, 1)$$

$$b_{eq} = 1$$

として，

$$A_{eq} x = b_{eq}$$

と書ける．また，

$$v \leq \sum_{i=0}^{n-1} p_i R_{i,j}$$

の条件は，

$$A_{ub} = \begin{pmatrix} 1 & -R_{0,0} & -R_{1,0} & \cdots & -R_{i,0} & \cdots & -R_{n-1,0} \\ 1 & -R_{0,1} & -R_{1,1} & \cdots & -R_{i,1} & \cdots & -R_{n-1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & -R_{0,n-1} & -R_{1,n-1} & \cdots & -R_{i,n-1} & \cdots & -R_{n-1,n-1} \end{pmatrix}$$

$$b_{ub} = (0, 0, \ldots, 0)^T$$

として，

$$A_{ub}x \leq b_{ub}$$

と書け，$0 \leq p_i$ という変数の制約を加えれば 標準的な線形計画法ソルバーで解くことができる．

本研究では，線形計画法ソルバーとしては，scipy.optimize.linprog を用い，単体法で解く python プログラムを作成した．10 数分ですべての状態のナッシュ均衡での利得の期待値とナッシュ均衡を与えるポリシーの一つを計算できた．[*3]

## 4. ナッシュ均衡戦略の性質

得られたナッシュ均衡戦略に関していくつかの性質を調べた．

### 4.1 初期状態

初期状態 (手札が 8 枚の時) のナッシュ均衡戦略を求めるための，利得行列を小数点以下 5 桁まで表示したものは表 2 のようになる．

**表 2** 初期状態の利得行列

| P1＼P2 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | -0.01177 | -0.49893 | 0.04516 | -0.10607 | 0.00193 | 0.07208 | 0.35812 |
| 1 | 0.01177 | 0 | -0.49910 | 0.32692 | -0.61282 | -0.01548 | 0.01164 | 1 |
| 2 | 0.49893 | 0.49910 | 0 | 0.31530 | -0.42120 | -0.01222 | 0.01580 | 0.28045 |
| 3 | -0.04516 | -0.32692 | -0.31530 | 0 | 0.31017 | -0.01332 | 0.36280 | 0.05922 |
| 4 | 0.10607 | 0.61282 | 0.42120 | -0.31017 | 0 | -0.02830 | -0.05954 | -0.05382 |
| 5 | -0.00193 | 0.01548 | 0.01222 | 0.01332 | 0.02830 | 0 | -0.26833 | -0.28208 |
| 6 | -0.07208 | -0.01164 | -0.01580 | -0.36280 | 0.05954 | 0.26833 | 0 | -0.23217 |
| 7 | -0.35812 | -1 | -0.28045 | -0.05922 | 0.05382 | 0.28208 | 0.23217 | 0 |

これを解いて得られるナッシュ均衡戦略としては，「密偵」，「暗殺者」，「大臣」，「魔術師」，「王子」を約 $(0.22176, 0.27191, 0.22685, 0.23428, 0.04520)$ の確率で出す手が利得の期待値 0 を達成する (利得行列の rank の関係で解の一意性がいえる)．初期状態で「道化」，「姫」，「将軍」を出して，相手がナッシュ均衡戦略をとった時の利得の期待値はそれぞれ，約 -0.10578, -0.11922, -0.03627 となるので，それらの手を確率 0 以上で選択するのは不利になることが確かめられた．

### 4.2 有効なアクションの数

得られたナッシュ均衡戦略において，残りカード枚数ごとの有効な (0 より大きい確率の) アクション数を同時着手について求めたものを表 3 に示す．先出し，後出しでは得られた戦略では有効なアクション数は常に 1 となるので，省略する．

初期状態 (カード枚数 8 枚) の時の有効アクション数が，5 なのに対して，カード枚数 7 枚で有効アクション数が 7 の状態が ([0, 2, 3, 4, 5, 6, 7], [0, 2, 3, 4, 5, 6, 7]), (0, 0), (2, 2), 0, 0, -1) と ([0, 1, 3, 4, 5, 6, 7], [0, 1, 3, 4, 5, 6, 7]), (0, 0), (2, 2), 0, 0, -1) の 2 個存在する．最初に両方が「姫 (1)」を出して引き分けた状態と，両方が「密偵 (2)」を出して引き分けた状態になる．

[*3] 解析に用いたプログラム，および解析結果は `https://github.com/tanakat01/r-rivals` で公開している．

**表 3** 残りカード枚数ごとの有効なアクション数 (同時着手)

| カード枚数＼アクション数 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 2600 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 19333 | 6041 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 58522 | 33351 | 10218 | 0 | 0 | 0 | 0 | 0 |
| 4 | 38200 | 33424 | 32451 | 6711 | 0 | 0 | 0 | 0 |
| 5 | 4005 | 4997 | 10469 | 7522 | 879 | 0 | 0 | 0 |
| 6 | 73 | 169 | 539 | 769 | 334 | 18 | 0 | 0 |
| 7 | 2 | 4 | 7 | 12 | 23 | 0 | 2 | 0 |
| 8 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

### 4.3 各カードが何ターン目の勝負で出されるか

ナッシュ均衡戦略同士の対戦で，どのカードが何ターン目の勝負で出される確率が高いかを表 4 に示す．「姫 (1)」が 0.40603 の確率で使われないうちに勝負がつくカードだということは，ゲームの性質から予想されるとおりであるが，次に使われない確率が高いカードが「密偵 (2)」だというのは，そこそこ意外と言えるかもしれない．

ナッシュ均衡戦略同士の対戦における引き分け対戦の確率は 0.07585 となった．

### 4.4 ランダムプレイヤを相手にしたときの勝率

プレイヤ 2 をランダムプレイヤとした時の，得られたナッシュ均衡ポリシーの利得の期待値を求めたところ，約 0.46078 となった．利得は負け-1, 引き分け 0, 勝ち 1 なので勝率に換算すると 7.3 割程度になる．一般にナッシュ均衡ポリシーは一意でないので，ナッシュ均衡戦略であり，ランダムプレイヤに対する勝率がこれよりも大きくなる戦略は存在する可能性はあるが，どの戦略に対しても期待値が 0 以上であり，ランダムプレイヤに 7 割以上の勝率となる戦略が存在することが確かめられたことになる．

なお，完全なランダムプレイヤでは，「密偵」によって出す順序が決まっているときに，「相手が先に「姫」を出したのに，自分が後から「王子」を出す」，「相手が「姫」をもっているのに，自分が先に「王子」を出す」という負けが確定することがあきらかなアクションも等確率で生成してしまう．この 2 つのパターンを除いて等確率で手を選択する戦略に対するナッシュ均衡戦略の利得の期待値は約 0.43970 となった．

### 4.5 大逆転

現在の獲得ポイントが高いほうが勝率が高いと予想される．これを確かめるために，P1 のポイントと，P2 のポイントごとの「P1 の勝ち，勝ちでも負けでもない，P2 の勝ち」の状態数を数えたものを図 5, 図 6 に示す．

P1 のポイントが 0, P2 のポイントが 3 でも勝てるケースがかなり多いが，これには，たとえば前のターンで「密偵」を出して，自分に「姫」が残っているのに相手が「王

表 4　各カードが何ターン目の勝負で出されるか

| cards ＼ turn | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | not used |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.00000 | 0.14488 | 0.11867 | 0.12525 | 0.14147 | 0.10949 | 0.04932 | 0.01382 | 0.29709 |
| 1 | 0.00000 | 0.09852 | 0.07259 | 0.08968 | 0.09935 | 0.09615 | 0.08730 | 0.05038 | 0.40603 |
| 2 | 0.22176 | 0.16342 | 0.11464 | 0.06882 | 0.03578 | 0.01979 | 0.01056 | 0.00348 | 0.36175 |
| 3 | 0.27191 | 0.15931 | 0.14973 | 0.12193 | 0.05189 | 0.03662 | 0.02154 | 0.01174 | 0.17533 |
| 4 | 0.22685 | 0.17773 | 0.14569 | 0.11136 | 0.07806 | 0.05767 | 0.04432 | 0.01909 | 0.13923 |
| 5 | 0.23428 | 0.05530 | 0.08505 | 0.09827 | 0.10931 | 0.07653 | 0.02990 | 0.00869 | 0.30268 |
| 6 | 0.00000 | 0.07570 | 0.14545 | 0.16684 | 0.15398 | 0.09199 | 0.05191 | 0.01520 | 0.29893 |
| 7 | 0.04520 | 0.12515 | 0.14961 | 0.11707 | 0.09065 | 0.07277 | 0.05207 | 0.02613 | 0.32136 |
| SUM | 1.00000 | 1.00000 | 0.98142 | 0.89921 | 0.76050 | 0.56102 | 0.34692 | 0.14853 | |

表 5　同時出しの状態で，P1 のポイントと，P2 のポイントごとの「P1 の勝ち，勝ちでも負けでもない，P2 の勝ち」の状態数

| P1 ＼ P2 | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | (580, 2191, 580) | (2371, 5830, 1346) | (2939, 9582, 2199) | (2500, 11147, 4843) |
| 1 | (1346, 5830, 2371) | (3234, 10284, 3234) | (3277, 14135, 4333) | (2643, 13545, 8720) |
| 2 | (2199, 9582, 2939) | (4333, 14135, 3277) | (3994, 14165, 3994) | (3149, 10342, 6597) |
| 3 | (4843, 11147, 2500) | (8720, 13545, 2643) | (6597, 10342, 3149) | (2499, 4426, 2499) |

子」を出した状態が含まれていたり，引き分けがたまっていて，P1 のポイントが 0 でも次の「勝負」に勝てば 4 点になる状態も含まれていて，「大逆転」といえるほどではない．

引き分けによって貯まった「勝負」がなく，P1 のポイントが 0, P2 のポイントが 3 の必勝状態は 405 個見つかったが，いずれも，前のカードで前の「勝負」で自分が「密偵」を出した後出しの状態だった．そのうち，自分が「姫」を持っているのに相手が先出しで「王子」を出した状態を除くと 284 個だった．

あの 1 つとして，(([3, 4, 5, 6, 7], [1, 2, 3, 4, 6]), (0, 3), (1, 1), 1, 0, 3) をみてみる．これは相手が，[姫 (1), 密偵 (2), 暗殺者 (3), 大臣 (4), 将軍 (6)] の中から「暗殺者 (3)」を先出しした状態である．ここで「王子 (7)」を出すと「暗殺者」の能力に関わらず，勝ち (([3, 4, 5, 6], [1, 2, 4, 6]), (1, 3), (1, 1), 0, 0, -1) に遷移する．次は「将軍 (6)」を出してその後，手数はかかるが勝ちが確定する．

### 4.6　双方が暗殺者を出したときの処理

「暗殺者」の効果に関してはここまでは説明書に基づいて「強さ」が小さい方が勝つとして，双方が「暗殺者」を出した時も「強さ」が小さい方が勝つと解釈した．

一方で説明書の説明には合致しないが，「暗殺者」の効果を「強さ」の大小が逆転すると解釈して，双方が「暗殺者」を出した時には，再逆転して「強さ」が大きいほうが勝つとするルールも成立する．

双方が「暗殺者」を出したときは，「強さ」は同じなので，この 2 つのルールで違いがないと思われるかもしれないが，前の「勝負」でどちらかが片方が「将軍」を出した場合は，「将軍」を出したプレイヤの「強さ」は本来の「暗殺者」の「強さ」3 に+2 されて 5 となっているので，どちらが勝つかが変わることになる．

このルールの変更を行なった時に，状態の勝率やナッシュ均衡のポリシーにどの程度変化があるかを調べた．

まず，状態数が変化する．たとえば，このルールのもとでは，(([0, 1, 2, 4, 5, 7], [0, 1, 4, 5, 6, 7]), (2, 0), (1, 1), 0, 0, -1) という状態が存在する．これは，自分が 6, 3, 相手が 2, 3 をこの順に出した時の状態だが，オリジナルのルールでは，2 回目の「勝負」で相手が勝ち，(([0, 1, 2, 4, 5, 7], [0, 1, 4, 5, 6, 7]), (1, 1), (1, 1), 0, 0, -1) となってします．カードの種類ごとの状態数は表 7 のように変わる．

このルールのもとでの，ナッシュ均衡戦略も求めた．初期状態 (手札が 8 枚の時) のナッシュ均衡戦略としては，「密偵」,「暗殺者」,「大臣」,「魔術師」,「王子」を約 $(0.22157, 0.27272, 0.22684, 0.23458, 0.04429)$ の確率で出す手が利得の期待値 0 を達成する．オリジナルルールと比べると，「密偵」,「大臣」,「王子」を出す確率を少し減らし，「暗殺者」,「魔術師」を出す確率を少し増やしている．

勝率の変化が大きい状態の例として，(([3, 6], [3, 4]), (3, 3), (1, 1), 1, -2, 3) をみてみる．これは，前の「勝負」で自分の「密偵 (2)」を出し，この「勝負」で，相手が「暗殺者 (3)」を出した状態である (相手は前回「将軍 (6)」を出して「強さ」を +2 している)．

暗殺者逆転ルールでは，自分が「将軍 (6)」を出すと相手の勝ち．「暗殺者 (3)」を出しても「強さ」3 と 5 で相手の勝ちとなる．それに対して，しかし，オリジナルルールでは「暗殺者 3」を出して「強さ」が 3 と 5 でこの「勝負」は自分が勝ち，ゲーム全体も勝つ．すなわち，勝敗が入れ

**表 6** 手番出しの状態で，P1 のポイントと，P2 のポイントごとの「P1 の勝ち，勝ちでも負けでもない，P2 の勝ち」の状態数

| P1 ＼ P2 | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | (1273, 237, 305) | (4368, 697, 1083) | (5920, 2314, 1792) | (5591, 5077, 3443) |
| 1 | (4188, 556, 1342) | (8186, 1917, 2298) | (8942, 6455, 3139) | (9539, 10515, 7323) |
| 2 | (7245, 1303, 1793) | (12674, 4217, 2139) | (15130, 8765, 3371) | (13122, 7437, 7500) |
| 3 | (10761, 2053, 1502) | (20450, 4307, 1947) | (20515, 4598, 2892) | (10705, 1767, 3734) |

**表 7** 残りカード枚数ごとの状態数 (暗殺者逆転ルール)

| 残りカード枚数 | 同時出し | 先出し | 後出し |
|---|---|---|---|
| 8 | 1 | 0 | 0 |
| 7 | 50 | 6 | 42 |
| 6 | 1904 | 273 | 1638 |
| 5 | 27882 | 4576 | 22880 |
| 4 | 110804 | 22438 | 89752 |
| 3 | 102081 | 25776 | 77328 |
| 2 | 25372 | 6779 | 13558 |
| 1 | 2600 | 727 | 727 |

替わる状態であると入れる．

ルールの違いによる全体の傾向の違いをみるために，ナッシュ均衡戦略のすべての状態でのエントロピーの平均を求めたところ，オリジナルルールでは，0.33500, 暗殺者逆転ルールでは 0.33512 となった．複数の手を確率的に出す状態が増え，より戦略的になったのかもしれないが，ほとんど影響がないとも言える．

### 4.7 厳密解ソルバの利用

ゲーム終了時の利得が 勝ち (1), 引き分け (0), 負け (-1) の整数値であり，可能な move が有限であることから，すべての状態のゲーム値は有理数で表現可能である．そのため，線形計画法の近似解ではなく厳密解を求めるソルバを使えば，ゲーム値を厳密に求めることができることになる．Python 言語には，分子と分母を多倍長整数であらわす fractions.Fracion があるので，扱いも容易である．

線形計画法の厳密解ソルバとして有名なものは，文献 [2] で紹介されている QSOpt_ex [3], SoPlex [4] などがあるが，本研究では　Dmitry Shintyakov 氏が作成した Python 言語のみで書かれた Simplex 法プログラム [*4] を使用した [*5]．

厳密解が比較的簡単な有理数で表されることを期待したが，残念ながらそうはならなかった．たとえば，前節で述べたように，ナッシュ均衡戦略では，初期状態で「密偵 (2)」を 0.22176 の確率で出すが，厳密解では，「1405076054.... / 63361401685....」のように分母，分子共に 10 進で 3241 桁になった．厳密解のデータ自体は github で公開している．

[*4] `https://github.com/dmishin/pylinprog`

[*5] SoPlex は Python 言語から容易に使えなかった．QSOpt_ex は python-qsoptex (`https://pypi.org/project/python-qsoptex/`) を用いて Python から呼び出せるが，1 万 7 千回位ソルバを呼び出すと Bus error を起こす障害が複数の環境で発生して，解決できなかった

## 5. おわりに

本研究では R-Rivals で，ゲーム値 (利得の期待値) を求める上で同じとみなせる状態をまとめて，手番からみた約 50 万の状態を求めた．その上で，それぞれの状態の手番プレイヤからみたゲーム値を求めた．

その結果，初期状態で可能な 8 枚のカードのうち，「道化」,「姫」,「将軍」を除く 5 枚のカードをある確率で出すのが最適な戦略であること，ポイントが 0-3 から勝てる状態があることなど様々な性質をみつけることができた．

**謝辞**　本研究は JSPS 科研費 18K11600 の助成を受けて行われた．

### 参考文献

[1] 中松 稜, and 高橋 和子 ”カードゲーム「アールライバルズ」における戦略”, 情報処理学会ゲーム情報学研究会研究報告, vol. 2021-GI-46, No. 8 (2021): 1-4.

[2] 品野勇治, and 藤井浩一. ”使ってみよう線形計画ソルバ (特集 はじめよう線形計画法).” オペレーションズ・リサーチ= Communications of the Operations Research Society of Japan: 経営の科学 64.4 (2019): 238-245.

[3] Cook, William, et al. ”An exact rational mixed-integer programming solver.” International Conference on Integer Programming and Combinatorial Optimization. Springer, Berlin, Heidelberg, 2011.

[4] Gleixner, Ambros M., Daniel E. Steffy, and Kati Wolter. ”Iterative refinement for linear programming.” INFORMS Journal on Computing 28.3 (2016): 449-464.

[5] Balinski, M. L., and Albert W. Tucker. ”Duality theory of linear programs: A constructive approach with applications.” Siam Review 11.3 (1969): 347-377.

[6] Lucas, William F. ”An overview of the mathematical theory of games.” Management Science 18.5-part-2 (1972): 3-19.