カルキュレーションのためのセミランダムプレイ

杉山 隆志　　飯田 弘之
s5058@cs.inf.shizuoka.ac.jp　iida@cs.inf.shizuoka.ac.jp
静岡大学情報学部

概 要

本稿は不完全情報一人ゲームにおけるプレイヤのモデル化を論じる．モデル化の手法としてセミランダムプレイを用いる．セミランダムプレイは，先読みが導入されたランダムプレイであり，元々は二人完全情報ゲームで考案されたものである．そのセミランダムプレイの概念を不完全情報一人ゲームに適用する．一人遊びカードゲームの代表的なゲームであるカルキュレーションを題材に，ランダムプレイおよびセミランダムプレイを計算機上に実装し，それぞれの自動プレイ実験を行い，いくつかのデータが得られたので報告する．

Semi-Random Self-Play for Calculation

Takashi Sugiyama　and　Hiroyuki Iida

Department of Computer Science, Shizuoka University

ABSTRACT

This paper explores a way to build models of players of different strength for single-agent games with incomplete-information using the semi-random self-play method. For this purpose, we have chosen the domain of CALCULATION, a single-agent (card) game with incomplete-information. The semi-random self-play method was first proposed in the domain of two-person games with complete-information. We then consider how to design a random/semi-random self-play method for a single-agent game with incomplete-information. This paper reports the results of experiments performed by the semi-random self-play method we propose.

1 はじめに

実力の異なる様々なレベルのプレイヤをモデル化することは，ゲームプログラミングあるいはゲーム情報学的観点から重要な意義がある [1]．その一つとして最近注目されているのは，あるゲームの情報学的性質を調査するために，平均合法手，平均終局手数，引き分け率，といった情報を得たいとき，様々なレベルのプレイヤによるデータを得ることである [2]．

本稿では，不完全情報一人ゲームでのプレイヤのモデル化を検討する．そのために，不完全情報一人ゲームとして，典型的な一人遊びカードゲームであるカルキュレーションを題材とする．プレイヤのモデル化の手法としてセミランダムプレイを用いる．セミランダムプレイは，先読みが導入されたランダムプレイであり，元々は二人完全情報ゲームで考案されたものである [3]．ここでは，そのセミランダムプレイの概念を不完全情報一人ゲームに適用する．

まず最初に，カルキュレーションについて簡単に説明する．次に，ランダムプレイの概念を述べた後，その実装を行い，ランダムプレイによる自動プレイの実験結果を考察する．さらに，カルキュレーションのためのセミランダムプレイを提案し，それを実装して同様な実験を行い，その結果を考察する．

2 カルキュレーション

カルキュレーションは，トランプの一人遊びゲームとして古くから親しまれているだけでなく，そのプログラミングも1970年代頃から今日に至るまで盛んである．本節では，カルキュレーションのルール，現在までのカルキュレーションプログラミングに関する研究の経過を振り返る．

2.1 ルール

カルキュレーションはジョーカーを除いた52枚のトランプを用いてプレイされる．ゲームの目的は，以下に示すような系列を作ることである(10は"0"と表記)．そして，十分にシャッフルされたカード(の系列)に対する成功率を競う([4]など引用)．

台 1: A,2,3,4,5,6,7,8,9,0,J,Q,K
台 2: 2,4,6,8,0,Q,A,3,5,7,9,J,K
台 3: 3,6,9,Q,2,5,8,J,A,4,7,0,K
台 4: 4,8,Q,3,7,J,2,6,0,A,5,9,K

プレイ手順は以下の通りである．

【1. 開始】 52枚のカードをよく切って，裏返しにしたものを山とする．

【2. カードを引く】 山からカードを1枚引き，状況にしたがって次の動作を選ぶ．

- **【台に出す】** 4つの台のうちのどれかに出すことができるときは出してもよい．
- **【スタックに置く】** 台に出さなかったときは，4つのスタックのどれかに表向きに置く．

【3. カードを移す】 スタックの先頭のカードのうち，台に出すことのできるものがあれば，それを移してもよい．これは何回でも行えるし，全く行わなくてもよい．スタックとスタック，台と台の間のカード移動はできない．以上を終えたら上記2に戻る．

【4. 終了の判定】 52枚のカードを引き終り，台の4つの列が完成すれば終了(成功)．スタックにカードが残り，それ以上台に移せなくなると終了(失敗)である．

2.2 これまでの研究経過

近年，コンピュータによるカルキュレーション成功率の進展が目覚しい．従来の成果は，花澤[7]の要約によれば，最高記録が3スタックで約44.6%，4スタックで93%強である．ところが最近，田中[5][6]によって達成された結果は顕著な飛躍である．3スタックで約72%，4スタックで約99%という成功率である．十分にシャッフルされたカードの系列の中には，理論的に必ず失敗してしまうものがある割合で存在するだろうから，約99%の成功率というのは，ほぼパーフェクトなプレイと評価できる．

従来はカルキュレーションプログラミングもチェスライクゲームのプログラミング同様，評価関数を用いるなどの経験則を利用する手法が一般的であった．一方，飛躍的成果を示した田中[6]の手法は，エキスパートが用いるような経験則を一切用いず，しらみつぶし的な先読み探索によるものである．

プレイヤのモデル化という観点からすれば，田中[6]が作成したカルキュレーションのコンピュータプレイヤは，人間名人を超える超エキスパートプレイヤに相当する．つまり，カルキュレーションで名人レベルのゲームのデータが入手できる環境ができたことになる．

同様に，将棋や碁などのプロ棋士の試合データから平均合法手，平均終局手数，引き分け率といった情報をわれわれは入手してきた．しかし，こういったデータは，与えられたゲームの性質を分析するには十分でない．ランダムプレイヤ(超々初心者)から名人レベルまで，様々なレベルのプレイヤによる統計的データを調査すべきである．

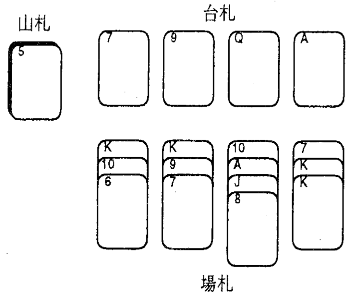

図 1: カルキュレーションのプレイ例(4スタックの場合)

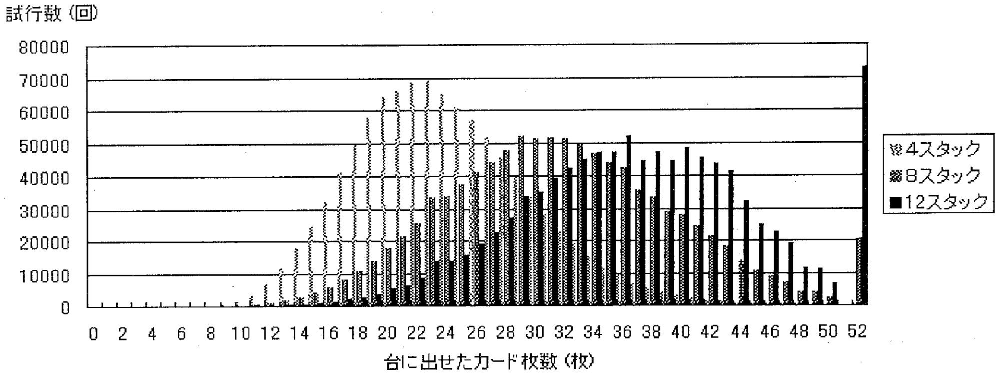

図 2: ランダムプレイの結果 (台に置けた枚数と試行数 (/10^6 回))

3　ランダムプレイ

本節では，セミランダムプレイを設計する前段階として，カルキュレーションでのランダムプレイを検討する．

チェスライクゲームのような完全情報二人ゲームでは，プレイヤの選択は基本的には自分の手番において自分の駒を動かす (将棋では打つことも) 指し手だけであるから，ランダムプレイの定義は簡単である [3]．その場合，可能な選択肢からそれぞれを等確率で選択するのが自然である．一方，カルキュレーションの場合，プレイヤにとって指し手を選択すべき意志決定の場面が，(1) 山札からカードを引いたとき，そして，(2) カードをスタックに置いた後 (2.1 節参照) など複数あるので，妥当なランダムプレイの定義を得ることはそれほど簡単ではない．

以下，まず最初にカルキュレーションのランダムプレイを定義し，次に，その定義に基づいて，計算機上にランダム自動プレイを実装する．そして，多数のランダムなカード出現列に対し，ランダム自動プレイの実験を行う．

3.1　ランダムプレイの定義

カルキュレーションにおいて，プレイヤが指し手を選択する場面を以下に列挙する．

【(1) 山札からカードを引いたとき】
可能なら，カードを台へ出す
カードをスタックへ置く

【(2) カードを置いた後】
可能なら，スタックから台へカードを置く
新たに山札からカードを一枚引く

ここでわれわれは，(1) の場面で，いま山札から一枚引いたカードを直接台へ出す選択と，そのカードを一旦スタックへ置き，その後そのカードを同じ台へ出す選択 (次の場面 (2) で行われれる) は等価であることに注意しなければならない．

本研究では，プログラム実装に際して，スタックへカードをランダムに置くことに着目して，ランダムプレイを以下のように定義する．

定義 1 カルキュレーションのランダムプレイは，**R1**，**R2** によって定義されるプレイである．

R1：山札からカードを一枚引き，そのカードをランダムにスタックへ置く．そして，**R2** へ．

R2：スタックから台へカードを出せる場合，スタックからその台へカードを移す．このとき，出せるカードはスタック *1* から *4* (左側から) の順で優先し，移す台も同様に台 *1* から *4*(左側から) の順で優先する．山札から *52* 枚引いたら終了し，成功か失敗かを判定する．それ以外は，**R1** へ．

表 1: ランダムプレイの結果 (台の枚数と成功数)

	台に出せたカードの平均枚数	成功数 (台に 52 枚カードを出した回数)
4 スタック	23.85	652
8 スタック	31.77	20337
12 スタック	36.97	73028

3.2 実験結果と考察

上述した定義に基づくランダム自動プレイを 1,000,000 回試行した．その結果を表 1および図 2に示す．ここではゲームの成功か失敗かの情報と，どれだけカードを台に出せたかを集計した．本実験では，スタック数として 4，8，12 の 3 通りの場合を行った．乱数生成は，Microsoft 社の VisualC++ 6.0 に提供されている`rand()` 関数を利用した．複数の状態で同じ条件になるよう再現性のある乱数を用いた．

図 2から，われわれは以下の知見を得た．

- 台に配置できる平均カード枚数は正規分布を描く (図 2 参照).
- 成功率はスタック数の増加に対し指数的に増加する．
- 台に配置できる平均カード枚数は，スタック数の増加に対し増分が減ってゆく．
- カードの全出現列の中で 6.0×10^{-4} % 程度の系列がランダムプレイでも成功できる．

4 セミランダムプレイ

前節で定義したランダムプレイを基に，カルキュレーションのためのセミランダムプレイについて検討する．先に定義したランダムプレイは，スタックへカードを置くときにランダムに選択を行うものである．セミランダムプレイを定義するためには，そのような場面で何らかのルールに基づいて選択する必要がある．そして，様々なプレイヤモデルを得るために，強さ (成功率) の指標を持たせなければならない．

以下では，強さの指標は「ある台に対し，どれだけカードを連続してスタックに並べられるか」に焦点を当てて，セミランダムプレイの定義を試みる．そして，ランダムプレイと同様に計算機上にセミランダムプレイを実装し，自動プレイの実験を行う．

4.1 セミランダムプレイの定義

ランダムプレイの定義 1における **R1** を以下のように拡張することで，セミランダムプレイを定義する．

定義 2 カルキュレーションのセミランダムプレイは，**R1′**，**R2** によって定義されるプレイである．

R1′: 手札から一枚カードを引き，以下の規則に基づきスタックへカードを置く．

(a) 0 を含む任意の自然数 i (先読みの深さに相当) の長さの範囲内で，順番を飛ばさずにスタックに並べられるか調べ，一番うまく並べられるスタックへカードを置く．

(b) うまく並べられない場合は，スタックにランダムにカードを置く．

R2: スタックから台へカードを出せる場合，スタックからその台へカードを移す．このとき，出せるカードはスタック 1 から 4 (左側から) の順で優先し，移す台も同様に台 1 から 4 の順で優先する．山札から 52 枚引いたら終了し，成功か失敗かを判定する．それ以外は，**R1′** へ．

ここで，$i = 0$ の場合のセミランダムプレイはランダムプレイに等しい．以下に，手札を引いたとき $i = 1,2,3$ のそれぞれの場合でどういうふるまいをするか，簡単な例を示し説明する．

手札: 5
スタック 1:　6　10
スタック 2:　7　9　K
スタック 3:　8　J　A　10

$i = 1$ の場合，5 の次に台に出せるカードは，台 1 から台 4 でそれぞれ順に 6,7,8,9 である．この場合，どのスタックの先頭にも，次に台に出せる可能性のあるカードが置かれている．今回の実装では，より単純に左側優先で番号の小さいスタックに置かれるようにした．よって，スタック 1 の先頭にカード 5 が置かれる．

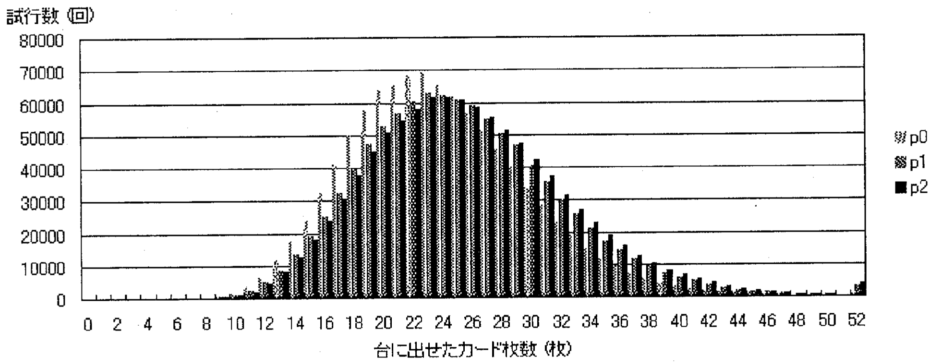

図 3: セミランダムプレイの結果 (スタック数 4 での台に置けた枚数と試行数 (/10^6 回))

$i = 2$ の場合，5 の次に台に出せるカード，および，その次に台に出せるカードが順に置かれているスタックを探す．この場合は 6-7，7-9，8-J，9-K と先頭から並んでいるスタックを探す．その結果，スタック 2 とスタック 3 が該当するので，左側優先によりスタック 2 にカード 5 を置くことになる．

$i = 3$ の場合，6-7-8，7-9-J，8-J-A と並んでいるスタックを探し，スタック 3 にカード 5 を置くことになる．

以上の定義により，**R1′ (a)** の深さ i を変化することにより，様々なプレイヤモデルが得られるはずである．

4.2 実験結果と考察

本稿では，ランダムプレイで行った実験と同様に，定義 2に基づいてセミランダム自動プレイを 1,000,000 回試行した．ここでは，i を変化させたプレイヤモデルを P_i と表現する．スタック数を 4 に固定し，$P_{0\sim5}$ のそれぞれについてランダムプレイ実験と同様のデータを収集した．結果を，表 2と図 3($P_{0\sim2}$ のみ) に示す．

結果より，以下の知見を得ることができる．

- i の増加に対し，成功率，台に置く枚数の増加は頭落ちになる．これは，カードがスタックに順に並ぶ可能性は，長さが長くなるほど指数的に低くなるためと考えられる．
- 定義 2において i を増加させてゆくと，成

表 2: セミランダムプレイの結果 (台の枚数と成功数)

	台に出せたカードの平均枚数	成功数 (台に 52 枚カードを出した回数)
P_0	23.85	652
P_1	25.27	3151
P_2	25.58	3888
P_3	25.60	3898
P_4	25.60	3899
P_5	25.60	3898

功率は，$3900/10^6 = 0.0039\%$ 程度に収束すると考えられる．

5 考察

本研究ではカルキュレーションにおいて，ランダムプレイおよびセミランダムプレイを定義し，それに基づき計算機上の自動プレイによりいくつかのデータを取得した．

ランダムプレイでは，カルキュレーションの性質に則った正当なデータが得られたと判断する．また，ランダムプレイにおける成功率が得られたことは，非常に有意義であったと考えている．

セミランダムプレイについては，本稿の定義 2 によると，成功率 0〜0.0039% の範囲内でのみプ

レイヤモデルを得られることとなる．これは非常に微小な範囲であるため，今後は完全プレイヤまで表現可能な定義にさらに拡張する必要性があると考えている．

6 今後の課題

最後に，今後の課題についていくつか示しておく．

- カルキュレーションにおける完全プレイヤを表現できるようなセミランダムプレイの提案．
- 一人遊びカードゲームへのセミランダムプレイの定義．
- 一人遊びカードゲームの進化論的変遷の調査．
- 不完全一人情報ゲームにおけるセミランダムプレイの定義への拡張，さらに，不完全情報ゲームへのセミランダムプレイの定義の拡張．

参考文献

[1] 飯田弘之 (1997). 様々な実力を持つゲームプレイヤのモデル化,「情報学研究」(渓村・飯田 編), 静岡大学情報学部, pp.1–24.

[2] 佐々木宣介, 橋本剛, 梶原羊一郎, 飯田弘之 (1999). チェスライクゲームにおける普遍的指標, 情報処理学会ゲーム情報学研究会 99-GI-1, pp.91–98.

[3] Y. Kajihara, M. Sakuta and H. Iida (1999). Semi-Random Play in Game Playing, *Proceedings of the Joint International Conference on Advanced Science and Technology (JICAST'99)*, pp.205–208.

[4] 竹内郁雄 (1986). GPCC ウルトラナノピコ問題, bit, **18(3)**, pp.341.

[5] 田中哲朗 (1999). 経験則を用いないカルキュレーションのプレイ, *Proceedings of Game Programming Workshop '99, IPSJ Symposium Series*, **99(14)**, pp.76–83.

[6] 飯田弘之, 田中哲朗, 佐々木宣介 (2000). Games and Puzzles Computer Contest (G-PCC) '99 課題: 3 スタックカルキュレーション & 四人将棋シングル戦, 第 41 回プログラミングシンポジウム報告集, pp81–85.

[7] 花沢正純 (1997). カリキュレーション (計算),「bit 別冊 ゲームプログラミング」(松原・竹内 編), 共立出版, pp.109–117.