多人数ゲームの順位を決定するゲーム木探索

後藤智章　乾伸雄　小谷善行
東京農工大学

概要
2 人零和有限確定完全情報ゲームの結果は、ゲーム木探索によって先手必勝か後手必勝か引き分けかに分類される。本論文では多人数の零和有限確定完全情報ゲームで同様のことを求める手法を示す。多人数ゲームにおいては、2 人ゲームの場合とは異なり、それぞれのプレイヤにとっての可能性がある順位の列挙を決定する。また、実際に多人数ゲームにこの手法と高速化手法を導入し、解とノード数を調べた。

Game Tree Search for Ranking in Multiplayer Games

Tomoaki GOTO　Nobuo INUI　Yoshiyuki KOTANI
Tokyo University of Agriculture and Technology
tgotoh@fairy.ei.tuat.ac.jp　nobu@cc.tuat.ac.jp　kotani@cc.tuat.ac.jp

Abstract
The result of two-person deterministic zero-sum limited games with perfect information can be classified to "first move wins", or "second move wins" or "draw" by game tree search. This paper shows a searching technique on multiplayer deterministic zero-sum limited with perfect information games as well as for two players. Unlike the case of two-person game, we only determine the possibilities of ranking for each player in multiplayer game. We introduced this concept to game tree search and the fast search method to multiplayer games. We experimentally investigate the solution and the number of nodes.

1. はじめに

非協力 n 人ゲームの解は、ナッシュ均衡点[1][2]が知られている。また、ナッシュ均衡点より満足な解を求める研究は、均衡点の精緻化[1]と呼ばれる。本論文では、零和有限確定完全情報の非協力 n 人ゲームにおける妥当な戦略を定義して、その戦略における解を計算機で求めることを目的とする。以下では、今回扱うゲームを多人数ゲームと呼ぶこととする。

多人数ゲームにおいて、一人のプレイヤがある目的を必ず成功できるかどうかを解とするときは、自分を OR ノード、敵全てを AND ノードとし、目的成功を真、目的失敗を偽として探索することで解を求めることができる。これは不確定性のある問題を決定論的に解決する[3] こととなる。また、2 人ゲームのゲーム木探索では、両者が自分にとっての評価関数の値が最も高い手を選択する仮定の下で探索を行うミニマックス戦略が用いられる。これは多人数ゲームでも評価値をベクトル化することで実現することができ、M3 サーチ[4]などが知られている。

本論文では、多人数ゲームにおける 1 人の目的成功だけではなく、全員の可能性のある順位を求めることを目的とする。そこで、末端まで探索する AND/OR 木探索をベクトル化したような探索法を用いる。

今回扱う多人数ゲームの順位は、n 人の場合に 1 位〜n 位まで決定するゲームであり、あらかじめ何位まで決定するかわかっているものとする。つまり、末端ノードには順位（1 位だけを決める場合は勝ち・負け）が、それぞれのプレイヤに関して割り当てられている問題を扱うことになる。

多人数ゲームの解である各プレイヤの順位は、他のプレイヤの自由度によって影響されることがあり、一意に決定しないことがほとんどである。しかし、それぞれのプレイヤが協力しないで最善

をつくした場合における、可能性のある順位を限定することは可能である。そこで、本稿では確率的な分析をせず、順位がありうるかありえないかを求めるアルゴリズムを提案した。

論文の構成は次のようになる。第2節で多人数ゲームにおける各プレイヤの戦略の定義とアルゴリズムの数式を示し、第3節で高速化手法を示す。第4節で実際のゲームを簡略化したゲームにおける実験結果を示す。第5節で考察を示し、最後に第6節でまとめを行う。

2. 多人数ゲームの順位決定アルゴリズム

2.1 アルゴリズムの方針

多人数ゲームの解を求めたいときは、対象となるゲームにおけるプレイヤの戦略とは何かを考える必要がある。

多人数ゲームには、自分にとっての評価値が同じであり、かつ、他のプレイヤへの影響が異なる手が複数ある場合に、他のプレイヤの順位が不確定になるという課題がある。特に、本稿のように順位を求める場合は、評価値の種類が評価関数による値と比べて2値〜n値程度と少ないため、図1のように評価が不明になることが頻繁に起こる。

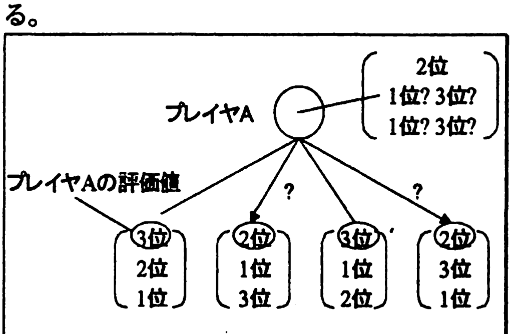

図1 手の選択の例

多人数ゲームの順位を決定する場合、上記で示した課題を考慮する必要がある。そこで、全てのプレイヤが必ず上の順位を目指す仮定の下で、可能性のある順位を全て列挙する手法を用いる。ここで、図2に示される、Bが1位になる手が選択される可能性が高いと考えられるような場合がある。今回の手法では、非協力ゲームにおける手は確率では表せないものとして、この場合でも全て同じように列挙する。また、詰将棋のような受け側（受け側は負けが確定している。）が少しでもゲームを長引かせようとする戦略も考えないこととする。これは、全てのプレイヤがゲームの神様のような存在であり、全ての手の末端を把握していることを想定している。

このような手法で複数並んだ順位に関しても、少しでも上の順位を狙う戦略における手の優越関係を考えて、遷移する可能性のある順位を全て列挙していくことにする。

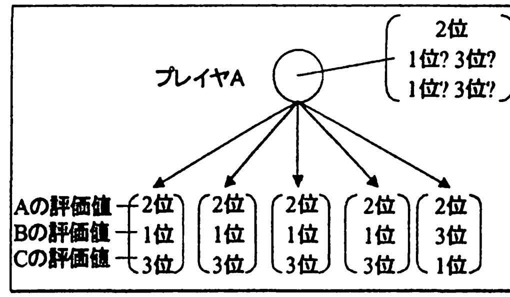

図2 プレイヤBが1位になりそうな例

2.2 アルゴリズムの公式化

アルゴリズムは、式(1)〜(7)で示される。評価値は(1)のような列ベクトルで表され、各要素は各プレイヤの「可能性のある順位の集合」である。プレイヤIの「可能性のある順位の集合」は(2)のような形となる。評価関数は、式(3)で示される。葉ノード以外では、選択可能性がある子ノードの集合$S(P)$に関して式(4)の演算をすることで求められる。一方、葉ノードでは、静的評価関数で算出されるゲーム終了時の各プレイヤの順位を評価値とする。式(4)は、ベクトルの各要素の和集合を求める式である。式(5)、(6)で示される$S(P)$は、選択する可能性があるノードの集合であり、明らかに優れている他のノードが存在するノードは選択しないことを示している。式(7)は、「可能性のある順位の集合」であるI_Pと$I_{P'}$が異なり、I_Pの最低順位が$I_{P'}$の最高順位を上まっているか等しければ、I_Pが$I_{P'}$に対して優れていることを示す。

$$F(P)=\begin{bmatrix}\vdots\\ I_p\\ \vdots\end{bmatrix}\quad \cdots (1)$$

$$I_p=\{i_{1p},i_{2p},\cdots i_{n-1p},i_{np}\}\quad \cdots (2)$$

$$F(P)=\begin{cases}\bigcup_{P'\in S(P)} F(P') & \text{if } P \text{ is not leaf node}\\ Evaluation(P) & \text{if } P \text{ is leaf node}\end{cases}$$

$$\cdots (3)$$

$$\bigcup_i X_i=\begin{bmatrix}\bigcup_i x_{i1}\\ \bigcup_i x_{i2}\\ \vdots\\ \bigcup_i x_{in}\end{bmatrix}\ \text{where}\ X_i=\begin{bmatrix}x_{i1}\\ x_{i2}\\ \vdots\\ x_{in}\end{bmatrix}\quad \cdots (4)$$

$$S(P)=\{P' \mid P'\in C(P), {}^{\forall}P''\in C(P),$$

$$\text{turn of } P=I, I_{P''}\not\succ I_{P'}\}\quad \cdots (5)$$

$$C(P)=\text{set of children of } P\quad \cdots (6)$$

$$I_{P'}\succ I_{P''}\Leftrightarrow I_{P'}\neq I_{P''}, i_{\min P'}\geq i_{\max P''}\quad \cdots (7)$$

2.3 アルゴリズムの実行例

4 人で 1 位〜4 位を決めるゲームにおける例を、図 3 に示す。この例では、プレイヤ A にとっての可能性のある順位に関して、(ア)と(ウ)が(エ)より明らかに劣っているので、プレイヤ A に選択されないとする。また、(イ)と(エ)はどちらが優れているかわからないので、どちらも選択の可能性があるとする。

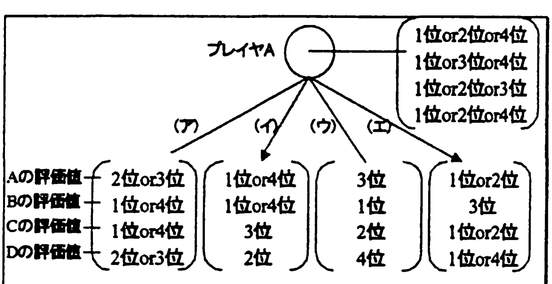

図 3 アルゴリズムの実行例

3 高速化

3.1 枝刈り

ノードの評価値が、探索中の手番が 1 位だけで、他のプレイヤが 1 位以外の全順位の可能性があるようになった場合に枝刈りができる。4 人で 1 位〜4 位を決めるゲームにおける枝刈りの例を図 4 に示す。

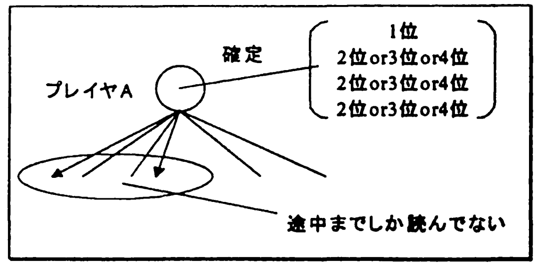

図 4 枝刈りの例

3.2 反復深化法 (iterative deepening)

深さ優先探索において、最初から末端まで読まずに、何手読みをするのかを表す深さのしきい値を決め、しきい値を徐々に増やす反復深化法という手法がある。今回は、深さのしきい値を 2 ずつ増やして実験を行った。

3.3 トランスポジションテーブル

ゲーム木探索において、探索中に既に探索した局面と同一な局面が現れることがあり、再利用することで効率化できる[6][7]。この場合の、以前読んだときの局面の情報を蓄える表はトランスポジションテーブルと呼ばれる。

反復深化法を用いた AND/OR 木探索では、テーブルに書かれている値が「末端まで読んだ評価値」である場合と、「まだ末端まで読んでないが深さのしきい値まで読んでわからなかった」という情報である場合がある。前者は、その局面の評価値として再利用できる。後者は、その局面から何手読んだかを覚えておき、今から読む予定の手数を上まっていれば、このしきい値ではわからないと判断できる。今回提案した探索法でもこの方法を導入した。

4. 実験

4.1 実験したゲームのルール

実験は、カードゲームの大富豪を完全情報に変更し、単純化したゲームを用いた。ルールを次に列挙する。

- 2 人以上で遊ぶゲームである
- 最初に 1〜n の数字が書かれたカードが配られ、テーブルには何も置かない
- 順番にテーブルにあるカードより大きい数字のカードを 1 枚出す
- カードを出せなければパスをする

- テーブルに何も無ければ何でも出せる
- テーブルにカードがあればパスもできる
- 最後にカードを出した人以外全員がパスをしたら、テーブルのカードを無くして、最後にカードを出した人の手番とする
- 早くカードが無くなった人ほど順位が上になる

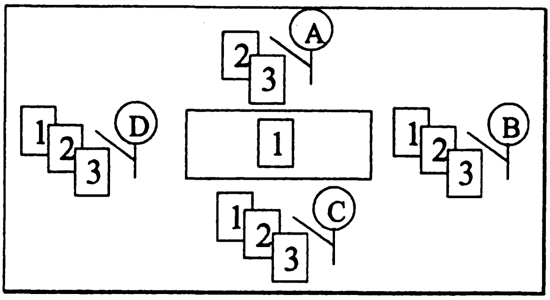

図5 単純化した大富豪 (4人、カードが1〜3)

4.2 実験内容とトランスポジションの仕様

4.1 のゲームを、複数種類のルールと高速化手法で実験を行い、それぞれのノード数、最大深さ、解を測定した。ルールはプレイヤの人数、カードの種類、順位種類の3つを変更した。探索法として、高速化手法の組み合わせで6種類の実験を行った。

候補手の順番は数字の少ない順でパスは最後とした。なお、パスしかない局面もノード数としてカウントし、カードを全て出したプレイヤに関してはノード数としてカウントしないこととした。また、探索ノード数が 2^{23} を超えたときは解けないものとした。

トランスポジションテーブルのサイズは 2^{20} とし、16回までオープンハッシュを行い、空き領域がない場合は上書きすることにした。ハッシュコードの情報は、次の5つである。

- 手番
- 各プレイヤが持っているカード
- テーブルに置かれているカード
- カードを最後に出したプレイヤ
- 順位が確定しているのが誰か

また、ハッシュ表の情報は次の4つである。

- ハッシュコード(64bit)
- 末端の値であるかのフラグ
- この時点で順位が確定してない者の評価値
- この局面から読んだ局面数

図6は、左を先に末端まで探索した場合のトランスポジションテーブルを利用することができる場合の例である。テーブルには、まだ順位が確定してないプレイヤが今後どうなるのかを記録することになる。

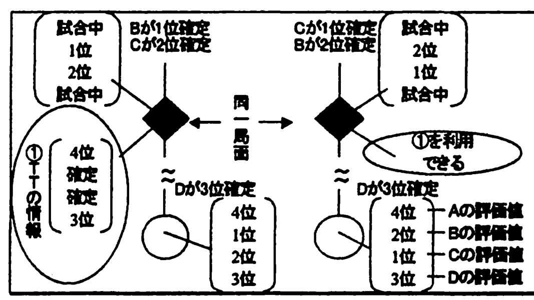

図6 TTを利用できる例

4.3 実験結果

付録に実験結果のノード数、深さ、解を示す。

得られた解は、カードが3枚以下の場合 (例:図7) は、手番の順が順位の順という結果となった。手番で順位が確定しており、各プレイヤに一つの可能性のある順位が割り当てられる形となる。

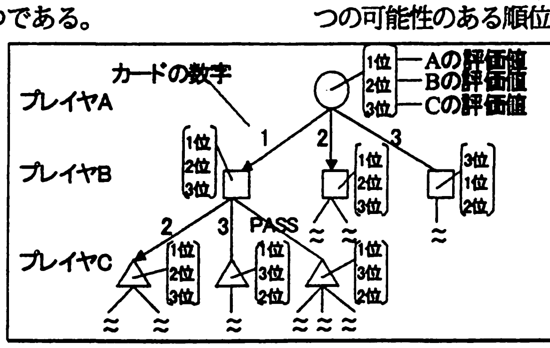

図7 3人で1位〜3位までを決めるゲームで、カードの種類が1〜3の例

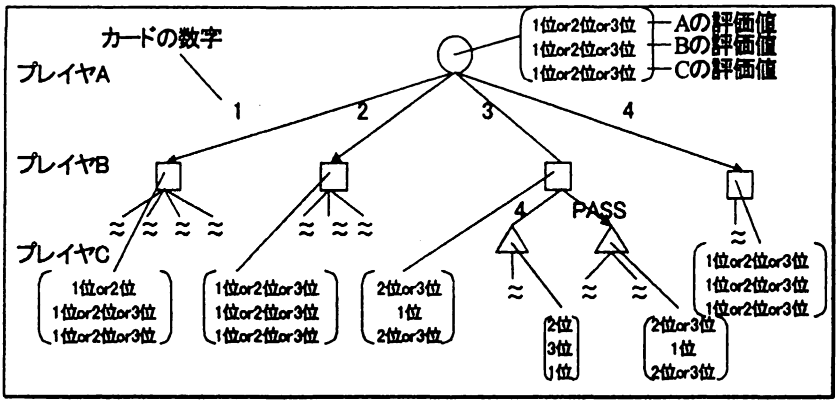

図8　3 人で 1 位～3 位までを決めるゲームで、カードの種類が 1～4 の例

また、4 枚以上になると手番で順位が確定（例：図8）していないという結果となった。

ノード数に関しては、枝刈り+トランスポジションの場合のノード数が最も少なく、反復深化法だけの場合のノード数が最も多い傾向があった。

4.4　実験結果の一例

実験結果の一例である図 7 について説明する。この例は 1 位～3 位までを決める 3 人ゲームで、カードの種類が 1～3 の場合である。一番上のルート局面の評価値は、プレイヤ A 、B 、C が、それぞれ 1 位、2 位、3 位となるのがわかる。ノード数は、全探索で 6482、最も効率が良かった枝刈り+トランスポジションで 2193 となった。

5　実験結果の考察

実験結果から、次のことが考えられる。

(1)枝刈りの有無について

全探索と枝刈りありのノード数の比率は次のような傾向が見られた。

- 順位種類数を増加させると、全探索と枝刈りありのノード数の比率が減少する
- カードの種類数を増加させると、全探索と枝刈りありのノード数の比率が増加する
- 人数を増加させると、全探索と枝刈りありのノード数の比率が増加する

また、順位種類数を 3 以上にすると、枝刈りをしても、全探索とほとんど変わらないノード数になる傾向が見られた。つまり、1 位だけを決めてゲームが終了するルールではないときはほとんど枝刈りが無いようであった。

(2)最大深さについて

枝刈り、トランスポジションを用いることで探索する最大深さが減少することがあった。この場合は全探索と比べて、かなり少ないノード数となる。一方、反復深化法を用いても最大深さは変わらなかった。

(3)トランスポジションと反復深化法について

トランスポジションは有効に働き、ルールによっては数 10 倍の高速化が見られた。また、トランスポジションを使っても反復深化法が有効ではなかった理由は、枝刈りがほとんど無いことと、本ゲームは平均候補主数が少ない（カードが 5 枚でも約 2.0 程度である。）ために、しきい値ごとのノード数の増加が少ないことが理由であると考えられる。

6.　おわりに

本稿では、多人数ゲームの解として各プレイヤの可能性のある順位の決定を行う探索法を提案した。更に、実際に多人数ゲームに導入して解を求める実験を複数の効率化手法に関して行った。その結果、手法で得られる解と、効率化によるノード数の変化を知ることができた。

今回の探索法で得られた解は、ゲームが少し複雑になると全てのプレイヤが全ての順位になりうるという結果となった。このことから、多人数ゲームのルールが複雑になると、それぞれが協力無しに最善な戦略を行った場合において手番による順位確定が無くなっていき、必勝法が確定しないということが予想できる。

また、効率化手法として、枝刈り・反復深化法・トランスポジションテーブルに関して実験を行った結果、枝刈りと反復深化法はあまり有効ではなかったが、トランスポジションテーブルは有効であることがわかった。枝刈りが有効ではなかったのは、ノードの評価値を確定できる条件の厳しさから予想通りの結果である。また、反復深化法が

有効ではなかったのは、枝刈りが少ないことが原因であると考えられる。

今後の課題として、多人数ゲームにおける戦略の考察、他のゲームにおける実験、高速化手法の応用が考えられる。トランスポジションテーブルに関しては、類似局面を利用することによって更に効率化できることが考えられる。また、多人数ゲームにおける少しずつ順位が確定していくゲームでは、反復深化法で末端になる前の情報を利用することで、高速化をする手法が考えられるだろう。

参考文献

[1] 岡田章: ゲーム理論, 有斐閣, 1996.
[2] 出口弘: ゲームプログラミングと社会科学, ゲームプログラミング, pp.231-243, 共立出版, 1998.
[3] 作田誠, 飯田弘之: 問題解決における不確定性パラダイムの提案と適用, ゲーム情報学 No.003, pp.49-56, 情報処理学会, 2000.
[4] 橋本剛, 平沢雅彦, 梶原洋一郎, 佐々木宣介, 飯田弘之: 四人将棋プログラムの基本的アルゴリズム, ゲーム情報学 No.001, pp.96-106, 情報処理学会, 1999.
[5] 大川貴之, 桜井貴文, 小谷善行, 辻尚史: 多人数ゲームにおける枝刈りと四人将棋への応用, ゲーム情報学 No.007, pp.73-80, 情報処理学会, 2002.
[6] David Levy, Monty Newborn,小谷善行 監訳: コンピュータチェス, サイエンス社, 1994.
[7] 山下宏: YSS, コンピュータ将棋の進歩 2, pp.112-142, 共立出版, 1998.

多人数カードゲームの解を求めた場合におけるノード数と解の比較

表1　3人で順位種類が1位～2位の場合

ルール	人数	3人	3人	3人	3人	3人
	順位種類	1位～2位	1位～2位	1位～2位	1位～2位	1位～2位
	カード種類	1だけ	1～2	1～3	1～4	1～5
ノード数 (最大深さ)	全探索	2(2)	17(7)	1318(18)	334471(27)	over 2^23
	枝刈り	2(2)	10(6)	147(14)	12651(25)	2212593(34)
	枝刈り+ID	2(2)	23(6)	1022(14)	131852(25)	over 2^23
	枝刈り+TT	2(2)	10(6)	141(14)	7422(24)	252465(30)
	枝刈り+ID+TT	2(2)	23(6)	757(14)	58651(24)	4530303(30)
	TT	2(2)	17(7)	1048(18)	38847(26)	748409(30)
解（可能性のある順位）		手番順	手番順	手番順	全員全順位	全員全順位

表2　3人で順位種類が1位～3位の場合

ルール	人数	3人	3人	3人	3人	3人
	順位種類	1位～3位	1位～3位	1位～3位	1位～3位	1位～3位
	カード種類	1だけ	1～2	1～3	1～4	1～5
ノード数 (最大深さ)	全探索	5(5)	65(11)	6482(21)	1778308(30)	over 2^23
	枝刈り	5(5)	65(11)	6403(21)	1448338(30)	over 2^23
	枝刈り+ID	11(5)	194(11)	27545(21)	7902482(30)	over 2^23
	枝刈り+TT	5(5)	52(11)	2193(21)	50719(26)	815392(33)
	枝刈り+ID+TT	11(5)	165(11)	8179(21)	431875(26)	over 2^23
	TT	5(5)	52(11)	2199(21)	51666(26)	849917(33)
解（可能性のある順位）		手番順	手番順	手番順	全員全順位	全員全順位

表3　順位種類が1位～2位でカードが1～3の場合

ルール	人数	2人	3人	4人	5人	6人
	順位種類	1位～2位	1位～2位	1位～2位	1位～2位	1位～2位
	カード種類	1～3	1～3	1～3	1～3	1～3
ノード数 (最大深さ)	全探索	130(9)	1318(18)	17248(31)	281332(48)	5503606(69)
	枝刈り	26(8)	147(14)	759(22)	3771(32)	18512(44)
	枝刈り+ID	108(8)	1022(14)	7772(22)	57314(32)	412243(44)
	枝刈り+TT	26(8)	141(14)	677(22)	3037(32)	13084(44)
	枝刈り+ID+TT	93(8)	757(14)	4549(22)	31146(32)	197678(44)
	TT	115(9)	1048(18)	9197(31)	74163(48)	553917(64)
解（可能性のある順位）		手番順	手番順	手番順	手番順	手番順

表4　5人で順位種類が1位～2位の場合

ルール	人数	5人	5人	5人	5人
	順位種類	1位～2位	1位～2位	1位～2位	1位～2位
	カード種類	1だけ	1～2	1～3	1～4
ノード数 (最大深さ)	全探索	2(2)	37(11)	281332(48)	over 2^23
	枝刈り	2(2)	14(8)	3771(32)	over 2^23
	枝刈り+ID	2(2)	48(8)	57314(32)	over 2^23
	枝刈り+TT	2(2)	14(8)	3037(32)	4197767(63)
	枝刈り+ID+TT	2(2)	48(8)	31146(32)	over 2^23
	TT	2(2)	37(11)	74163(48)	over 2^23
解（可能性のある順位）		手番順	手番順	手番順	全員全順位

表5　4人で順位種類が1位～4位の場合

ルール	人数	4人	4人	4人	4人
	順位種類	1位～4位	1位～4位	1位～4位	1位～4位
	カード種類	1だけ	1～2	1～3	1～4
ノード数 (最大深さ)	全探索	9(9)	444(20)	427475(38)	over 2^23
	枝刈り	9(9)	444(20)	427475(38)	over 2^23
	枝刈り+ID	29(9)	1627(20)	2743396(38)	over 2^23
	枝刈り+TT	9(9)	261(20)	27715(34)	1491486(46)
	枝刈り+ID+TT	29(9)	914(20)	167986(34)	over 2^23
	TT	9(9)	261(20)	27715(34)	1550145(43)
解（可能性のある順位）		手番順	手番順	手番順	全員全順位

表6　5人でカードが1～3の場合

ルール	人数	5人	5人	5人	5人
	順位種類	1位～2位	1位～3位	1位～4位	1位～5位
	カード種類	1～3	1～3	1～3	1～3
ノード数 (最大深さ)	全探索	281332(48)	2823760(53)	over 2^23	over 2^23
	枝刈り	3771(32)	2823760(53)	over 2^23	over 2^23
	枝刈り+ID	57314(32)	over 2^23	over 2^23	over 2^23
	枝刈り+TT	3037(32)	209416(49)	276400(49)	288426(49)
	枝刈り+ID+TT	31146(32)	1741328(49)	2639200(49)	2892609(49)
	TT	74163(48)	209416(49)	276400(49)	288426(49)
解（可能性のある順位）		手番順	手番順	手番順	手番順