役を構成するゲームに対する効率的な行動決定アルゴリズムの提案

小松 智希[1,a)]　成澤 和志[1]　篠原　歩[1]

概要：探索空間が非常に広く，評価関数が作りにくいゲームにおいて行動決定を行う手法にモンテカルロ法があり，囲碁や大貧民などのゲームに対して有効な手法であることがわかってきた．しかし，麻雀のように探索空間全体に対して得点が得られる組み合わせが少ないゲームでは，モンテカルロ法は報酬を得ることができるプレイアウトの回数が少ないため，十分な効果を発揮することができない．本論文では，麻雀におけるモンテカルロ法の非効率性を実験的に検証する．また，プレイアウトにおいて効率的に報酬を得ることができる手法を提案し，モンテカルロ法と比較することで実験的に有効性を示す．

キーワード：モンテカルロ法，不完全情報ゲーム

Effective algorithm for decision making on hand-composing game

Komatsu Tomoki[1,a)]　Narisawa Kazuyuki[1]　Shinohara Ayumi[1]

Abstract: Monte Carlo methods have been successfully applied for playing games, and have outperformed previous algorithm in such games as Go and Daihinmin. However, as we will experimentally show, it is not very effective for some games like Mahjong, where random simulation can rarely get rewards. Without positive rewards, players have little reason to choose better actions. In this paper, we propose a new algorithm to overcome this difficulty. It virtually simulates many play-outs in each trial simultaneously, so that many of play-outs can get positive rewards, even for this kind of games. We show some preliminary experiments that convinced us that the approach is promising.

Keywords: Monte Carlo method，incomplete information game

1. はじめに

モンテカルロ法 [7] は代表的なゲーム AI 設計手法の 1 つであり，様々なゲームに応用されている．[3], [5], [6] この手法は，未知の情報やプレイヤの行動をランダムに仮定したシミュレーションを繰り返すことで行動の良さを推定する．このシミュレーションはプレイアウトと呼ばれる．プレイアウトはゲームが終了するまで行い，実際のゲームの得点を推定に利用する．そのため，モンテカルロ法はゲームの途中の盤面を評価する関数を必要とせず，あらゆるゲームに対して容易に適用することができる．

モンテカルロ法の問題点としてプレイアウト中のプレイヤの行動をランダムに決定するため効率が悪いことがあげられる．この問題点に対して，モンテカルロ法を探索木に適用したモンテカルロ木探索 [1] が提案されている．モンテカルロ木探索は，プレイアウトの結果をもとに探索木を展開していくため，プレイアウト中の行動はランダムに選ぶよりも良いものを選択でき，プレイアウトで点数を得る可能性が高くなる．この手法を発展させた UCT [4] は探索空間が広く，評価関数の設計が難しいとされる囲碁 [2] だけでなく，多人数不完全情報ゲームの 1 つである大貧民に

1　東北大学大学院情報科学研究科
Tohoku University, GSIS, Sendai, Miyagi 980–8579, Japan
a)　tomoki_komatsu@shino.ecei.tohoku.ac.jp

おいて非常に効果的であることが示されている [6], [8], [9]．しかし，同じ多人数不完全情報ゲームである麻雀に対しては，UCT はあまり効果的ではないことが示されている [10]．

本論文では，多人数不完全情報ゲームの 1 つである麻雀に対する手法を提案する．麻雀は，多人数不完全情報ゲームとして分類されるが，本論文では麻雀の難しさとして以下の 4 つをあげる．

(不完全性)　他プレイヤの牌が見えない

(ランダム性)　次に手に入る牌がわからない

(多人数性)　他プレイヤの影響が大きい

(希少性)　役を作る条件が厳しい

不完全性やランダム性，多人数性については他の多人数不完全情報ゲームにおいてもよくみられる性質であり，これらの性質をもつゲームに大貧民などがある．希少性とは，考えられる全ての組み合わせに対して，上り条件を満たす役の数が非常に少ないことを意味する．麻雀は希少性が高いゲームであり，麻雀の難しさを表す最も重要な要素の 1 つである．

本論文では，麻雀の希少性に焦点をあてるため，多人数ではなく一人で行うことで，不完全性と多人数性を排除した一人麻雀を定義する．ランダム性は麻雀において不可欠な要素であるため，そのままとする．本論文では希少性が高い一人麻雀に対して効率的な行動選択を行う手法を提案する．

本論文は，第 2 章で麻雀に対してモンテカルロ法を適用した時の問題点を分析する．第 3 章で提案手法の具体的なアルゴリズムを述べる．第 4 章で提案手法を麻雀に適用し，有効性を検証する．第 5 章でまとめと今後の課題について述べる．

2. 準備

本論文は麻雀における効率的な行動決定を行うことを目標とする．麻雀は本来 4 人で行うゲームだが，対戦相手を考慮した行動決定は非常に難しい．そのため，本論文では麻雀のルールを変更し対戦相手の影響がない状況での行動決定を行うことにする．

2.1 一般的な麻雀のルール

麻雀は 34 種類 (図 1)，136 枚の牌を用いて行う 4 人用のゲームである．牌は，萬子 (マンズ)，索子 (ソーズ)，筒子 (ピンズ) と呼ばれる 1 から 9 までの数字が割り当てられた 3 種類の数牌と，東，西，南，北，発，中が印字されたものと，何も印字されていない 7 種類の字牌と呼ばれる牌で構成され，同一の牌はそれぞれ 4 枚存在する．

ゲームは，プレイヤ全員に 13 枚の牌を配るところから始まる．各プレイヤが所持している牌を手牌といい，他プレイヤには公開されない．プレイヤに配られた牌以外は伏せられた状態でセットされており，これらの牌を山と呼ぶ．

図 1　ゲームで使う全種類の牌

図 2　順子の例

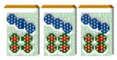

図 3　刻子の例

図 4　雀頭の例

図 5　和了りの条件を満たした牌の組み合わせの例

プレイヤは山から牌を 1 枚ツモリ*1，この牌と手牌の 14 枚の牌で役と呼ばれる和了がり*2形を作る．役ができない場合は，ツモ牌と手牌の中から 1 枚の不要な牌を場に捨てることを繰り返すことで，ゲームは進行する．プレイヤが和了るためには，(1) 自分で山から取ってきた牌で和了り形を作る (ツモ上り)，(2) 和了り形に必要な最後の牌を他のプレイヤが場に捨てた場合 (ロン上り) の 2 通りがある．ゲームは局と呼ばれる単位に分割されている．局は山の残り枚数が一定数になるか，プレイヤの一人 (または複数が同時に) が和上ったときに終了する．局を規定回数繰り返すことでゲームは終了する．ゲーム終了時に一番高い点数を持っているプレイヤが勝者となる．

3 連続の数字の同種数牌の組み合わせを順子 (シュンツ)，3 枚の同一の牌の組み合わせを刻子 (コウツ)，2 枚の同一の牌の組み合わせを雀頭 (ジャントウ) という．順子，刻子，雀頭の例を図 2， 3， 4 に示す．順子と刻子のことを面子 (メンツ) と呼ぶ．プレイヤは，雀頭を 1 つ，面子を 4 つで構成される役を作ると和了ることができる*3．和了る条件を満たした例を図 5 に示す*4．

プレイヤが和了るとプレイヤ間で点数の移動が起こる．麻雀には，役と呼ばれる特定の和了りパターンが複数存在し，役に応じて点数が変動する．本論文で扱う役を表 1 に示す．また，役は 1 つの和了り形に対して複数成立することができる．役が成立すると役の難易度に合わせた飜が与えられ，複数の役がある場合には飜は足し合わせたものとなる．例えば，図 5 では門前清自摸和，断幺九，三色同順の 3 つの役が成立しており，飜数は 4 飜となる．また，

*1　麻雀では山から牌を 1 枚取ってくることをツモるといい，ツモで取ってきた牌をツモ牌という
*2　麻雀では役を作って上がることを和了る (あがる) という
*3　七対子，国士無双を除く
*4　索子の 7 でツモ和了している

表 1 本論文で採用する役一覧

役数	役名		
一飜役	門前清自摸和	断幺九	平和
	一盃口	役牌	場風
二飜役	三色同順	一気通貫	混全帯幺九
	七対子	三暗刻	三色同刻
	小三元	混老頭	
三飜役	混一色	純全帯幺九	二盃口
六飜役	清一色		
役満	国士無双	四暗刻	大三元
	字一色	大四喜	緑一色
	清老頭	九蓮宝燈	

表 2 飜数と符に対する点数

	1 飜	2 飜	3 飜	4 飜
20 符		2,000	3,900	7,700
30 符	1,500	2,900	5,800	11,600
40 符	2,000	3,900	7,700	12,000
50 符	2,400	4,800	9,600	12,000
60 符	2,900	5,800	11,600	12,000

表 3 5 飜以上に対する点数

飜数	点数
5 飜	12,000
6,7 飜	18,000
8,9,10 飜	24,000
11,12 飜	36,000
13 飜〜	48,000

ゲーム開始時に 1 種類の牌がドラとして指定され，この牌が手牌に含まれている枚数と同じ飜が与えられる．

和了った場合に移動する点数は，役に応じた飜数と符と呼ばれる基準から計算される．符とは手牌の構成要素から計算される値であるが，計算方法が複雑なため詳細は省略する．本論文で用いる点数表を表 2 と表 3 に示す．表にもある通り，5 飜以上を獲得した場合は符によらず点数が決定される．図 5 の飜数は 4 飜，符は 30 符であるため，この組み合わせの和了り点数は 11600 点である．

2.2 本論文で扱う麻雀のルール

本論文では対戦相手の影響がなく，不完全な情報が存在するゲームでどのように行動決定をするかのみに着目するため，麻雀のルールを一人で行うものに変更し，一人麻雀と呼ぶ．本論文で扱う一人麻雀のルールを以下のように定める [*5]．また，表現を一般的にするため以降，局のことをゲームと呼ぶ．

1. ランダムに牌をプレイヤに 13 枚与える．
2. ドラをランダムに決定する．
3. プレイヤに牌を 1 枚ランダムに与える．
4. 和了る条件を満たすなら，点数を計算し，プレイヤに得点を与えゲームを終了する．
5. 和了れない場合は，牌を 1 枚捨てる．
6. 捨てた牌の総数が 18 枚なら，ゲームを終了する．そうでなければ，3 に戻る．

Algorithm 1: モンテカルロ法

```
1  報酬を格納する配列 V を初期化する
2  可能な行動を列挙する
3  for i = 1 to 可能な行動の個数 do
4  |  確定しない情報を，ランダムに推定する
5  |  for j = 1 to N do
6  |  |  i 番目の行動を行う
7  |  |  while ゲームが終了していない do
8  |  |  |  全ての行動をランダムに決定しゲームを進行する
9  |  |  end while
10 |  |  ゲームの結果をもとに V_i に報酬を加える
11 |  end for
12 end for
13 V_i が最大となる i の行動を返す
```

3. モンテカルロ法の問題点

この章では，麻雀にモンテカルロ法を適用する際に問題となる点を述べる．

3.1 モンテカルロ法の麻雀への適用

モンテカルロ法は不確定な情報やプレイヤの行動をランダムに決定し，ゲームを最後まで行うことで行動の良さを推定する．このシミュレーションのことをプレイアウトという．プレイアウト中に得点を得られたとき，得点に応じた報酬をプレイアウトで選択した行動に与える．このプレイアウトを何度も繰り返し，報酬の平均値が最も大きいものが最良の行動であると判断する．

一般的なモンテカルロ法のアルゴリズムを Algorithm 1 に示す．麻雀における行動とは，手牌とツモ牌[*6]の 14 枚の中から 1 枚を捨てることであるため，可能な行動の個数および報酬を格納する配列 V の大きさは 14 である．5 行目のループは，プレイアウトの回数を表している．10 行目における報酬は，和了り役に応じた点数であり，プレイアウトにおいて和了れない場合，報酬は 0 となる．

3.2 実験

モンテカルロ法の麻雀に対する有効性を検証するため，モンテカルロ法で行動決定を行うプレイヤに 1 万回一人麻雀を行わせ，和了った回数と平均点数を測定する．プレイアウトの回数 N は，10，20，30，$\cdots$，100，200，300，$\cdots$，1000 回に設定しそれぞれ実験を行う．また，比較対象としてランダムに行動決定を行うプレイヤで同様の実験を行う．

*5 場は常に東場，プレイヤは常に東家とする

*6 一人麻雀においては 3. のランダムに 1 枚配られた牌

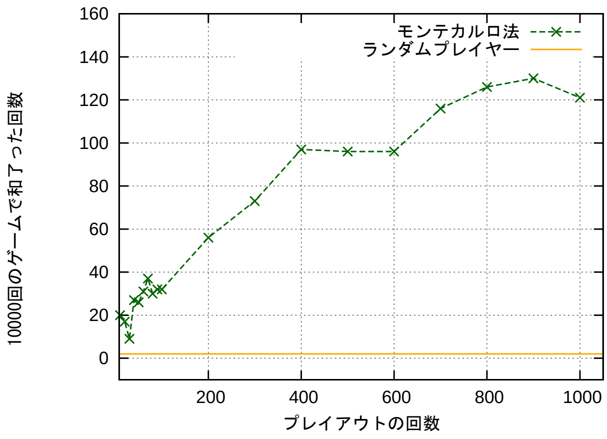

図 6 プレイアウト回数に対する和了った回数

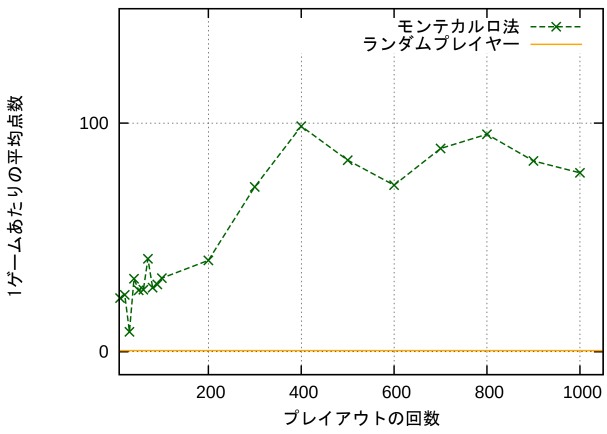

図 7 プレイアウト回数に対する平均得点

和了った回数と平均点数の結果を図 6，7 にそれぞれ示す．この結果からモンテカルロ法はランダムプレイヤよりもよい成績を収めたことが分かる．しかし，モンテカルロ法ではプレイアウトを 1000 回に設定したプレイヤでも，10000 回のゲームで 121 回しか和了れていないため，よい行動選択を行えているとは言えない．

3.3 考察

実験からモンテカルロ法を麻雀に適用してもよい行動決定を行えないことが分かった．この原因はプレイアウト中の行動選択をランダムに行っているため，プレイアウトで和了った回数が極端に少ないことが原因である．前節の実験において，プレイアウト回数 1000 回のモンテカルロ法に 10000 回のゲームを行わせたときのプレイアウトの総数と報酬が与えられた回数を表 4 に示す．結果からほとんどのプレイアウトで和了ることができていないことが分かる．モンテカルロ法はプレイアウトで和了らない場合は，その結果を行動の良さの推定に用いることができない．そのため，モンテカルロ法は麻雀において効率の悪い手法であることがわかる．

表 4 モンテカルロ法の 10000 回のゲームでのプレイアウトの総数と，プレイアウト中で和了った回数と割合

プレイアウトの総数	179,340,000
プレイアウト中で和了った回数	32,652
プレイアウトで和了れる割合 (%)	0.0182067581

4. 提案手法

前章の実験よりモンテカルロ法は，麻雀のように役を作ることが難しいゲームにおいて効率が悪いことが分かった．実験は麻雀に対してしか行わなかったが，上る条件が厳しいゲームに対してこの問題点は同様に生じる．この問題点を克服する手法を提案する．

プレイアウト中に良い行動，ここでは役を作ることができる行動を選択することができればこの問題は解消される．しかし，そもそもモンテカルロ法は何が良い行動なのかを決定するために行う手法であり，プレイアウト中で良い行動選択ができるのであれば，もはやモンテカルロ法を行う理由はない．

プレイアウト中に良い行動選択を行う方法として，ヒューリスティックな関数を用いる方法がある．ヒューリスティックな関数を用いる方法は，現在の盤面を大ざっぱに評価する関数を用意し，その関数から有利と判断できる行動を優先的にプレイアウト中の行動とする手法である．麻雀における簡単なヒューリスティックな関数として考えられるのは，完成している面子数を与えるものなどが考えられる．この関数を導入すると，ランダムに行動選択を行うよりもよい選択を行うことができるが，麻雀上級者は高い点数を得るために，わざと面子を崩す選択をすることもあり，この評価関数が正しいとは言い難い．

4.1 プレイアウトの改良

ここでは，評価関数を作成することなく，モンテカルロ法と同様に実際のゲームの得点のみを価値の推定に用いる手法を提案する．山から取ってくる牌が全て見えている状況下，つまり完全情報ゲームであるならば最良の行動が決定的に求まることを利用する．

まず，残りのツモ牌が K 枚である場合[*7]，ランダムに K 枚の牌を選択する．次に，手牌 14 枚とランダムに決定した K 枚の牌の中で最も点数が高くなる和了り形を作る．この時，和了り形にはなく，初めの手牌には存在する全ての牌に得られた点数を報酬として与える．これを N 回繰り返し，最も報酬が高い牌を行動として選択する．また，この繰り返しのことをモンテカルロ法と同様に，プレイアウトと呼ぶ．

例として，図 8，9，10 に示す場合を考える．残りのツモの回数が 8 回で，手牌が図 8 の時にどの牌を捨てるかを

[*7] 一人麻雀においては初めに残りツモ回数を K と仮定する

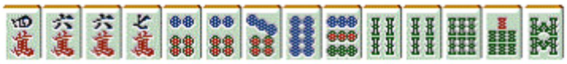

図 8　残り順目が 8 順における手牌の例

図 9　将来得られる牌を仮定した例

図 10　図 8 と図 9 の牌を用いて和了り点が最大になる牌の組み合わせ

Algorithm 2: 提案手法

```
1  配列 V を初期化する
2  for i = 1 to N do
3      ランダムに K 枚の牌を選択する
4      この牌と手牌で，最大の点数が得られる組み合わせを全て求める
5      for j = 1 to 14 do
6          if j 番目の牌が求めた組み合わせに含まれない then
7              V_j ← V_j + 和了り点
8          end if
9      end for
10 end for
11 V_j が最大となる j を返す
```

考える．残りのツモ牌が図 9 に示す 8 枚であるとする．この場合，図 10 に示す牌の組み合わせが最も点数が高い．よって，図 10 の和了り形には存在せず，図 8 の手牌に存在する牌である萬子の 6（左から 2 番目の牌）と筒子の 9（左から 9 番目の牌）に報酬を与える．ここで，萬子の 6 は和了りに含まれているように見えるが，複数枚の同じ牌が手牌にある場合はそれら全てが使われない限り報酬が与えられる．図 10 の和了り形を作るためには，萬子の 6 が必ず 1 枚は捨てられる必要あるためである．この組み合わせの和了り点は 18000 点であるため，萬子の 6 と筒子の 9 に 18000 点の報酬が与えられる．

提案手法のアルゴリズムを Algorithm 2 に示す．N はプレイアウトの回数を表し，K は残りのツモの数を表す．報酬を格納する配列 V の大きさは，前章のモンテカルロ法と同様に 14 である．配列の初期化が終わった後に，手牌とランダムに生成した K 枚の牌を用いて作れる和了りで最大の点数が得られるものを全て求める．j は手牌の中の 1 つの牌を表し，牌 j が求めた和了りの組み合わせに含まれない場合 7 行目で V_j に報酬を加える．もし手牌に牌 j が複数枚含まれているときは，それらが全て使われていない限り和了りに含まれるとは言わない．これを繰り返し行い，V_j が大きくなるような j を最も良い選択であると判断する．

表 5　提案手法とモンテカルロ法において，10000 回のプレイアウトで 10000 回のゲームを行った時の 1 ゲームあたりの和了り形の平均総数と行動決定を行う平均時間 (msec)

	提案手法		モンテカルロ法
K	平均総数	計算時間 (msec)	計算時間 (msec)
17	866921.9	1413.8	285.0
16	539700.5	996.7	267.5
15	329700.2	709.6	251.4
14	192826.8	503.8	235.0
13	108577.2	363.8	218.0
12	59373.0	270.8	201.9
11	30985.1	205.8	184.8
10	15683.6	162.6	168.6
9	7342.1	132.3	152.9
8	3350.0	113.6	135.1
7	1390.1	96.0	117.9
6	530.0	84.7	101.8
5	173.8	75.4	84.9
4	51.1	68.9	67.5
3	15.9	64.0	50.7
2	3.1	59.3	33.7
1	0.3	55.8	17.4

5. 実験

ここでは，提案手法の計算時間，得点効率，プレイアウトの回数の 3 つに焦点を絞り，実験的に評価する．

5.1 計算時間

提案手法で問題となるのは，最大の点数を得られる和了りを求める計算時間である．今回の実装では全ての和了り形を列挙し，点数が最も高いものを選択するという素朴な方法で求めている．14 枚の手牌に対して和了っているかどうかの判定は，バックトラック法を用いて判定できるため，計算時間はほとんどかからない．しかし，多数の牌に対して全ての和了りの組み合わせを求めるには膨大な時間がかかることが予想される．そこで，和了りの組み合わせがどれくらい存在するのかを調べるために以下の実験を行う．

まず，14 枚の牌をランダムに決定する．次に残りのツモ数が K 回であるときの行動選択を 10000 回のプレイアウトを行い決定する ($K = 1, 2, \cdots, 17$)．この手順を各 K に対して，10000 回のゲームを繰り返し，以下の 2 つを計測する．

- 1 ゲームあたりの和了り形の総数の平均
- 1 ゲームあたりの計算時間の平均

また，モンテカルロ法に対しても同様の実験を行い行動選択にかかる計算時間の平均を測定し，比較する．

実験の結果を表 5, 図 11, 図 12 に示す．この結果から

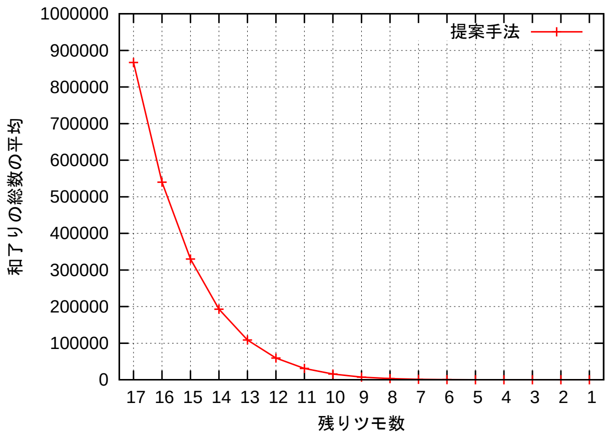

図 11 残りツモ数に対する 1 ゲームの和了り形の総数の平均

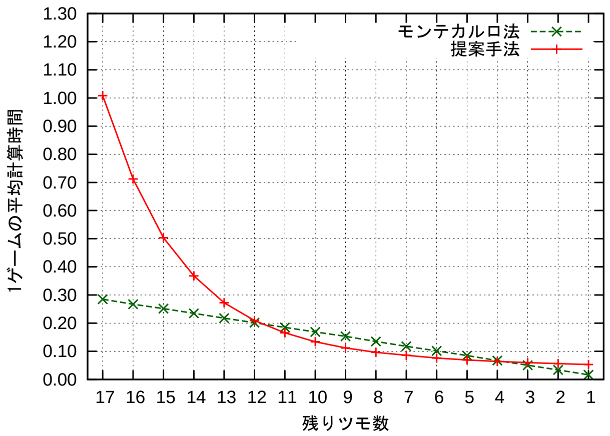

図 12 残りツモ数に対する 10000 回のプレイアウトに要した計算時間の平均

列挙に時間はそこまでかからないことが分かった．図 11，図 12 から和了りの総数と計算時間が残りツモ数に対して指数的に減少していることが分かる．これは麻雀の和了る条件がかなり厳しいためであると考えられる．当然残りツモ数が最大のときに一番計算時間がかかるが，10000 回のプレイアウトを行うのに 1 秒程度しかかからないことから提案手法が現実的な計算時間で実行できることがわかる．

5.2 得点効率

提案手法の利点は 2 つ考えられる．1 つ目は，1 回のプレイアウトで同時に複数の行動 (牌) に対して報酬を与えることができるという点である．2 つ目は，ほとんどのプレイアウトで 0 でない報酬を与えることができるということである．ここでは，実際にどれくらいのプレイアウトにおいて 0 でない報酬を与えることができるかを実験的に示す．

前節と同様に 14 枚の牌をランダムに決定し，残りのツモ数が K 回であるときの行動選択を 10000 回のプレイアウトから決定する実験を行う ($K = 1, 2, \cdots, 17$)．この手順を各 K に対して 10000 回のゲームを繰り返し，以下の 3 つを測定する．

- 1 ゲームあたりの得られた点数の平均

表 6 提案手法とモンテカルロ法において，10000 回のプレイアウトで 10000 回のゲームを行った時の 1 ゲームあたりの平均点数と 0 でない点数を得たプレイアウトの回数の割合 (%)

	提案手法		モンテカルロ法	
K	平均点数	割合 (%)	平均点数	割合 (%)
17	9079.0	97.560	3709.4	0.0046
16	7986.4	93.949	3160.8	0.0042
15	7140.8	87.474	3405.3	0.0039
14	6465.3	77.553	3491.0	0.0039
13	5929.0	64.838	3449.3	0.0035
12	5511.4	50.958	3380.5	0.0034
11	5182.1	37.069	3421.9	0.0032
10	4872.1	25.229	3377.9	0.0028
9	4644.1	15.795	3102.2	0.0025
8	4434.4	9.239	3239.3	0.0022
7	4248.3	4.820	3373.9	0.0022
6	4081.2	2.275	3474.5	0.0016
5	3866.4	0.913	3361.0	0.0013
4	3830.7	0.318	3035.5	0.0011
3	3521.8	0.111	3219.6	0.0014
2	3328.3	0.024	3109.3	0.0004
1	3150.9	0.003	5404.2	0.0004

- 1 ゲームあたりの 0 でない点数を得られたプレイアウト回数の割合 (%)

また，モンテカルロ法に対しても同様の実験を行う．

実験結果を表 6，図 13，図 14 に示す．この結果から，モンテカルロ法に比べて提案手法ではほとんどの場合で報酬が得られ，かつ効率的に価値の推定が行えていることが分かる．

5.3 プレイアウト回数

5.1 節，5.2 節の実験結果から，提案手法は現実的な計算時間で実行可能であり，効率的に価値の推定を行うことが分かった．ここでは，プレイアウト回数に対する性能を評価する．残りツモ数をランダムに仮定し，プレイアウト回数を 10,20,30, $\cdots$,100，200,300,400, $\cdots$,1000 回と変化させ，10000 回のゲームを行った時に以下の 2 つを測定する．

- 1 ゲームあたりの得られた点数の平均
- 1 ゲームあたりの和了ったプレイアウト回数の平均

また，3.2 節でのモンテカルロ法に対する実験と比較する．

実験結果を図 15，図 16 に示す．モンテカルロ法と比較するために，3.2 節で行った実験結果も再掲した．この結果から分かるように，提案手法はモンテカルロ法と比較してよい行動選択ができていることが分かる．また，プレイアウト回数を 400 回以上に設定しても平均点数の向上が見られなくなった．これは，提案手法が一人麻雀で得られる平均点数の限界まで収束したと考えられる．

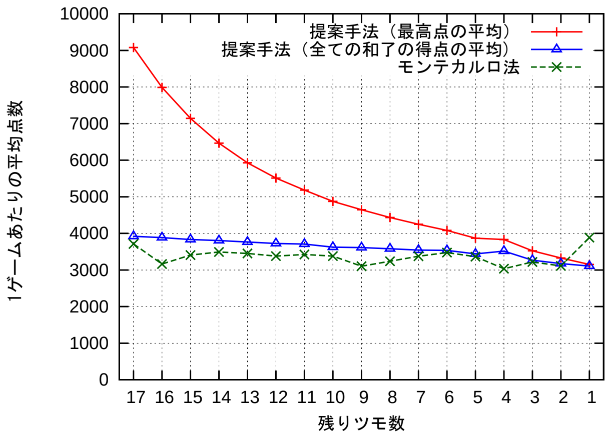

図 13　残りツモ数に対する 1 ゲームあたりの点数の平均

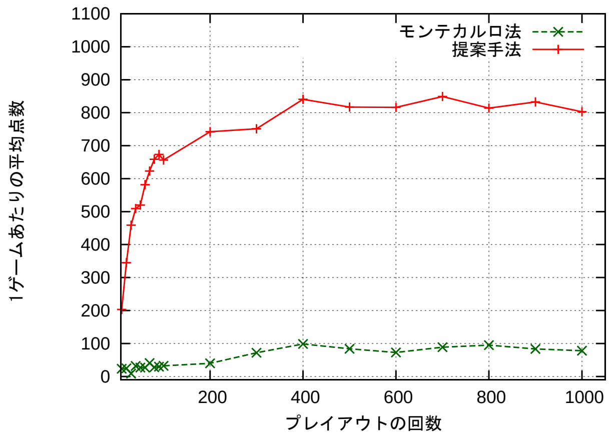

図 15　1 ゲームあたりの平均点数

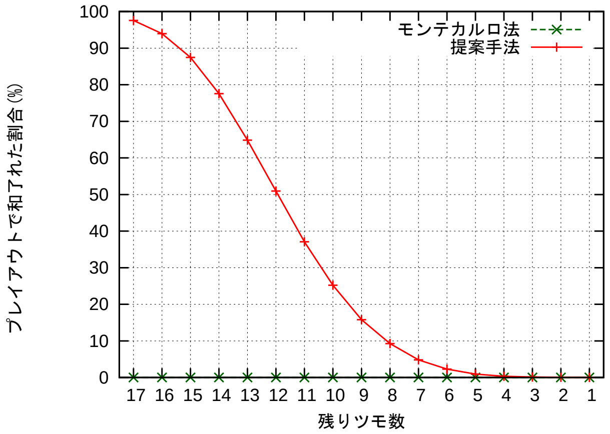

図 14　残りツモ数に対する 1 ゲームあたりの和了ったプレイアウト回数の平均

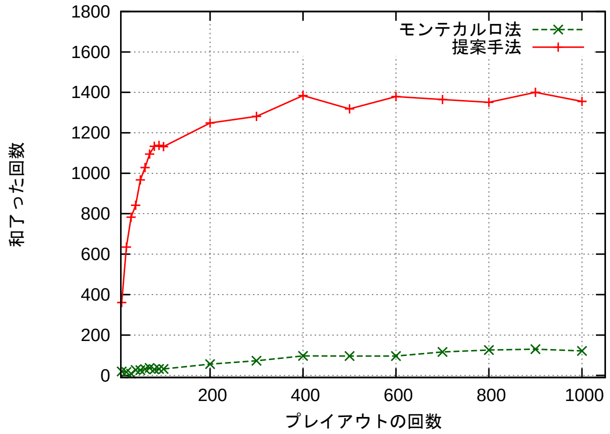

図 16　10 万回のゲームで和了った回数

6.　まとめ

本論文では，モンテカルロ法を麻雀に適用する場合に生じるプレイアウトの効率に対する問題を解決した手法を提案した．

まず，モンテカルロ法ではプレイアウト中の行動をランダムに決定するため，麻雀のように行動選択の数に比べて役を作ることが難しいゲームでは，大半のプレイアウトで報酬を得ることができないことを実験的に示した．次に，本来隠れた情報を既知のものと仮定することで，効率的に報酬を得ることができるプレイアウトを行う手法を提案した．提案方法の利点は，大半の場合に和了りの組み合わせが存在するためほとんどの場合に報酬を与えることができること，同時に複数の牌に対して報酬を与えることができることがあげられる．また，計算時間，プレイアウトの得点効率およびプレイアウトの回数に対して，モンテカルロ法との比較実験を行い，提案手法の有効性を実験的に示した．

麻雀において役を作ることが難しい理由は，和了りの組み合わせの総数が少ないことであるが，ツモる牌が全てわかっていると仮定すると和了りの組み合わせを列挙することは容易である．提案手法では，この性質を利用することで，モンテカルロ法では非効率的な麻雀に対して効率的にプレイアウトを行うことを可能にした．そのため，提案手法は麻雀のように役を作ることが難しいゲームにも同様に適用することができると考えられる．

7.　今後の課題

本論文では，問題を簡単にするため，他のプレイヤによる影響を考慮しないルールを採用したが，本来のルールに則った場合にも対応するアルゴリズムの開発が必要となる．麻雀において，他のプレイヤからの影響を受ける要因として，(1) ロン，(2) 局が途中で終わる，(3) 鳴き，(4) リーチの 4 つが挙げられる．

ロン和了りは他プレイヤが捨てた牌を用いて和了がるため，和了がる機会が多くなるが，逆に自分の牌を用いて和了がられることもある．自分が捨てた牌をでロン和了りをされたとき，点数を他プレイヤに奪われるため，麻雀の上級者は捨て牌から他プレイヤの手の進行度を予測し，他プレイヤの和了りに必要な牌を切らないようにする．本論文の提案手法では，自分が和了ることしか考慮していなかったが，他プレイヤのロン和了りを阻止しつつ，自分が和了るための行動選択が必要となる．

他プレイヤがツモ和了りやロン和了りをした場合，局が途中で終了する．本論文で扱った一人麻雀では，必ず捨て牌が 18 枚になるまで牌をツモることができたが，実際の麻雀では 18 枚をツモることはほとんどない．そのため，点数の高い和了りを目指すだけでなく，他のプレイヤよりできるだけ早く和了がることを優先する必要もある．

鳴きとは，対戦相手が捨てた牌を手に入れることができるルールである．鳴きを行うと必要な牌が手に入る代わりに和了った時の点数が下がる．鳴きを行うかどうかの判断は上級者でも意見が分かれ，非常に難しい問題である．

リーチとは役の 1 つで，和了りに必要な牌が 1 枚になったときにリーチを宣言することで成立する役である．リーチを宣言することで得られる得点は上昇するが，他プレイヤに自分が和了りに近いことを示すために，他プレイヤからのロン和了りは成立しにくくなる．また，リーチを宣言することで，自分が捨てる牌を選択できなくなるため，他プレイヤに和了られる可能性も高くなる．どのような場面でリーチを宣言すればいいのかを判断するのは難しい問題である．

これらの問題を解決し，通常の麻雀に対応したアルゴリズムを開発することが今後の課題である．

参考文献

[1] Cameron Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, pp. 1–43, 2012.

[2] Sylvain Gelly, Yizao Wang, Rémi Munos, and Olivier Teytaud. Modification of UCT with patterns in Monte-carlo go. Technical report, INRIA, 2006.

[3] P. Hingston and M. Masek. Experiments with monte carlo othello. In *IEEE Congress on Evolutionary Computation*, pp. 4059–4064, 2007.

[4] Levente Kocsis and Csaba Szepesvari. Bandit based monte-carlo planning. In *Machine Learning: European Conference on Machine Learning 2006*, Vol. 4212, pp. 282–293. 2006.

[5] G. Van den Broeck, K. Driessens, and J. Ramon. Monte-carlo tree search in poker using expected reward distributions. *Advances in Machine Learning*, pp. 367–381, 2009.

[6] 須藤郁弥. モンテカルロ法を用いた多人数不完全情報ゲームの AI 設計に関する研究. Master's thesis, 東北大学大学院情報科学研究科, 2011.

[7] 玉木久夫. 乱択アルゴリズム. 共立出版, 2008.

[8] 小沼啓, 西野哲朗. コンピュータ大貧民に対するモンテカルロ法の適用. Technical Report 3, 電気通信大学情報通信工学科, 電気通信大学総合情報学科, 2011.

[9] 西野順二, 西野哲朗. 多人数不完全情報ゲームのモンテカルロ木探索における推定の効果. 情報処理学会研究報告. BIO, バイオ情報学, No. 31, pp. 1–4, 2011.

[10] 三木理斗. 多人数不完全情報ゲームにおける最適行動決定に関する研究. Master's thesis, 東京大学大学院工学研究科, 2010.