大局的な状況に応じて打ち方の変更を行う麻雀AI

海津　純平[1]　吉仲　亮[1]　篠原　歩[1]

概要：麻雀は大局的な状況に応じて戦略を変更することが重要である．例えば，自分の現在順位が低いときは高得点を狙う戦略が，自分がトップのときはゲームを早く終了させるための戦略が有効な戦略とされる．また，自分が親のときは連荘をするために，早上がりを狙う戦略も有効である．本論文では，先行研究を基に，大局的な状況に応じて打ち方を変更する麻雀 AI を提案する．大局的な状況として，ゲームの進み具合，現在順位，現在の役割を考慮する．また，計算機実験によって，先行研究と対戦を行い，1 ～ 4 位率，平均順位，平均点数に対する評価を行った．その結果，提案手法が有用であることを示すことができた．

A Mahjong AI to change the strategy depending on the global situation

Jumpei Kaizu[1]　Ryo Yoshinaka[1]　Ayumi Shinohara[1]

Abstract: Selecting a suitable strategy depending on the global situation is important to win the Mahjong game. In general, a player at the lowest place should select a strategy that can obtain higher scores, while a player at the first place should select another strategy to win early the hand. In addition, when you are the dealer, to continue the dealer, winning the hand as quickly as possible is also an effective strategy. This paper proposes a Mahjong AI to change the strategy depending on the global situation on the basis of our previous study. We consider game progress, the current ranking and the current role as the global situation. In addition, we performed some computer experiments that compare the proposed method with previous methods, and we evaluated the ratio of each rank, the average rank, the average score. The experimental results indicate that the proposed method is effective.

1. 背景

近年，多人数不完全情報ゲームのひとつである麻雀に対して，様々な AI の研究が行われている．麻雀は，自分の行動選択や相手の手牌，自分と相手の点数状況など考慮しなければならない要素が多い．そのため，人間のトッププレイヤ以上の実力をもった人工知能は我々の知る限りまだ作成されていない [1–4].

本研究では，大局的な状況に応じて打ち方の変更を行う麻雀 AI を考える．多人数の麻雀においては，自分の手牌だけでなく，自分の現在順位，点差，何局目かなど様々な状況を考慮した打ち方の戦略を選ぶ必要がある．例えば，自分が最下位のときは上がり点を高くする打ち方にすべきであり，自分がトップのときはゲームを早く終了させるために早上がりをする打ち方にすべきである．また，自分の役割に応じて打ち方を変える場合もある．これまでの我々の研究では，小松らの手法 [5]，原田らの手法 [6] を基にし，一つのパラメータを変更することで打ち方を変えることができる手法を提案した [7]．この研究では，一人麻雀を対象にしていたが，一般的な麻雀は多人数で行うものであるため，本論文では，多人数の麻雀に対して実験を行っている．大局的な状況としてゲームの進み具合と現在順位，自分の役割を考慮する．実験では，市川が開発した麻雀 AI 対戦サーバ Mjai [8] を用いて，先行研究の手法と，大局的な状況に応じてパラメータを変更する場合を対戦させる．このことから，打ち方を変えない AI と打ち方を変える AI を対戦させ，打ち方の変更が有用であるかを確かめている．

2. 麻雀のルール

ゲームは，**局**と呼ばれる単位に分割されている．局を特定の回数繰り返すことでゲームが終了する．一般的に 1

1 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

ゲーム 8 局の**半荘**を行う．それぞれの局で，プレイヤの 1 人が**親**という役割を担当し，その他のプレイヤが**子**という役割になる．局の始まりには，**山**と呼ばれる伏せられた牌の集合から，各プレイヤは 13 枚の牌を取る．山は伏せられているためプレイヤはどの牌を持ってくるかはわからない．プレイヤが所持する牌を**手牌**と呼び，他プレイヤには公開されない．

局が始まるとプレイヤは親から順番に，山から 1 枚の牌を取り 14 枚になった手牌の中から 1 枚の牌を場に捨てる行為を繰り返してゲームを進行する．手牌が 13 枚のとき，山から取った牌または他プレイヤが捨てた牌を加えた 14 枚の牌の組合せが特定の条件を満たしていればプレイヤは**上がり** *1，その組合せに応じた点数を得ることができる．また，上がったプレイヤが親の場合．点数は 1.5 倍となる．局は，山の残り牌数が特定数になるか，プレイヤの一人が上がったときに終了する．子であるプレイヤが上がった場合，親の役割が別のプレイヤへ移り，次の局が始まる．親であるプレイヤが上がった場合，**連荘**となり，親の役割は移らず，もう一度同じ局を行う．ゲームが終わったときの得点数で勝敗を競う．

上がりに必要な 14 枚の組合せに**役**と呼ばれる特定のパターンがあることが上がり条件となる．複数の特定のパターンが組合せにある場合，複数の役があることになる．さらに，特定のパターンではない例外である役の一つに**リーチ**という役がある．リーチは行動名でもあり，あと 1 枚の牌を加えたら上がりに必要な組合せができることを宣言する．リーチをしたあとは，山から 1 枚の牌を取る番のとき，取ってきた牌をそのまま捨てることを繰り返す．このとき，山から取った牌または他プレイヤが捨てた牌を加えた 14 枚の牌が，上がりに必要な組合せとなった場合，役のリーチが成立し上がることができる．上がり時の役の数や種類，上がり時の状況により点数は計算される．

3. 麻雀 AI

3.1 先行研究

本研究が基づく先行研究では，山から 1 枚牌を取った時，どの牌を捨てるか決める手法を提案している [7]．まず，手牌 14 枚とランダムに選んだ残り巡数分の牌の中から，得られる報酬が最大となる組合せを求める．そして，手牌の中で，求めた組合せに含まれていない牌に報酬を与える．報酬の式を以下に示す．

$$R = \alpha \frac{P}{M} + (1-\alpha)\frac{U}{13} \tag{1}$$

ここで，P は求めた組合せによって得られる上がり点数，U は手牌の中で交換の必要がない牌の枚数であり，元の手牌の牌の多重集合を T，求めた組合せの牌の多重集合を

*1 本来，和了（ほーら）と言うがこの論文では単に上がるという表現を用いる．

Algorithm 1: 先行研究

Input: 手牌 $T = \{T_1, T_2, \cdots, T_{14}\}$，シミュレーション回数 N，残り巡目 k
Output: 捨てる牌 T_s

価値を格納する要素数 14 の配列 V の各要素を 0 にする;
for $i = 1$ **to** N **do**
  k 枚の牌をランダムに選ぶ;
  $R = \alpha\frac{P}{M} + (1-\alpha)\frac{U}{13}$ が最大となる組合せ T' を求める;
  for $j = 1$ **to** 14 **do**
    if 牌 T_j **が組合せ** T' **に含まれない then**
      $V_j \leftarrow V_j + R$;
$s \leftarrow \arg\max_{1 \le j \le 14} V_j$;
output T_s;

T' としたとき，$U = |T \cap T'|$ である．α はバランスパラメータである．式 (1) の第 1 項は上がり点の最大値 M(親は 48000，子は 36000) で，第 2 項は手牌の交換が不必要な枚数の最大値 13 で，それぞれ正規化している．このようなシミュレーションを N 回繰り返し，最も報酬の高い牌を，捨てる牌として選択する．

P が高いほど，高い点数を得ることができる．また，U の値が大きいということは，少ない交換回数で上がることができるということである．これらの 2 つの値を，パラメータ α を用いて調整することで，得点または早上がりを優先した打ち方を選択することができる．式 (1) を用いた先行研究の手法を Algorithm 1 に示す．

3.2 リーチのアルゴリズム

先行研究で対象としていた一人麻雀とは違い，本研究は一般的な 4 人麻雀を対象としているためリーチをすることができる．麻雀において，リーチは重要であるため，先行研究の手法にリーチのアルゴリズムを組み込んだ．捨てる牌を決め，その牌を捨てることでリーチができる場合，リーチをするようにしている．リーチのアルゴリズムを組み込んだ手法を Algorithm 2 に示す．

3.3 提案手法

大局的な状況に応じて打ち方を変更する AI を実現するために，先行研究の手法の α を局ごとに変更する．局の開始時に，現在順位と局数，また自分の役割から α を決定する．現在順位が 1 位で最後の局であるときは，早くゲームを終了させるために α を小さくし早上がりを目指す．現在順位が 4 位で序盤の局であるときは，高得点を目指すために α を大きくする．また，親の役割のときは連荘があるため，親で現在順位 4 位のときは，上がるために α を小さくし早上がりを目指す．現在順位，局数，自分の役割と α の関係を表 1 に示す．

Algorithm 2: リーチのアルゴリズム

```
Input: 手牌 T = {T_1, T_2, ..., T_14}, シミュレーション回数
       N, 残り巡目 k
Output: 捨てる牌 T_s, リーチをするか否か
価値を格納する要素数 14 の配列 V の各要素を 0 にする;
for i = 1 to N do
    k 枚の牌をランダムに選ぶ;
    R = α P/M + (1 − α) U/13 が最大となる組合せ T' を求める;
    for j = 1 to 14 do
        if 牌 T_j が組合せ T' に含まれない then
            V_j ← V_j + R;
s ← arg max_{1≤j≤14} V_j;
if T_s を捨てることでリーチができる then
    output T_s, リーチをする;
else
    output T_s, リーチをしない;
```

表 1 現在順位と局数，自分の役割に対する α の値

		1 位	2 位	3 位	4 位
子	1 局目	0.0	0.0	0.0	0.0
	2 局目	0.2	0.5	0.6	0.9
	3 局目	0.2	0.5	0.6	0.9
	4 局目	0.2	0.5	0.6	0.9
	5 局目	0.1	0.4	0.7	1.0
	6 局目	0.1	0.4	0.7	1.0
	7 局目	0.0	0.3	0.8	1.0
	8 局目	0.0	0.3	0.8	1.0
親		0.6	0.4	0.2	0.0

4. 実験

4.1 予備実験

計算機実験により，打ち方を変更する AI が有用か確かめる．予備実験として，実際に打ち方が変更できているかを確かめる．麻雀 AI 対戦サーバ Mjai を用いて，先行研究の手法を，取ってきた牌をそのまま捨てるだけの AI 3 つと対戦させる．1 回のゲームは半荘とし，100 回行う．シミュレーション回数は 1000 回とする．α の値を 0 から 1 まで 0.1 刻みで動かし，平均上がり得点と平均上がり巡目を確かめた．結果を表 2 に示す．α が小さいときは早上がりができており，大きいときは上がり得点が高くなっている．α を変えることで，打ち方の変更ができている．

また，麻雀 AI 対戦サーバ Mjai を用いて，α の値が違う先行研究の手法 4 を対戦させた．先行研究の手法の $\alpha = 0.0$，$\alpha = 0.3$，$\alpha = 0.7$，$\alpha = 1.0$ を対戦させる．1 回のゲームは半荘とし，1000 回行う．シミュレーション回数は 1000 回とする．結果を表 3 に示す．先行研究の手法同士で対戦させたときでも，α を変えることで，打ち方の変更ができていることが確認できた．

表 2 α と打ち方の関係

		平均上がり得点	平均上がり巡目
先行研究	$\alpha = 0.0$	4611	13.24
	$\alpha = 0.1$	4874	12.74
	$\alpha = 0.2$	4573	13.07
	$\alpha = 0.3$	5052	13.00
	$\alpha = 0.4$	4858	13.12
	$\alpha = 0.5$	5114	13.31
	$\alpha = 0.6$	5461	13.18
	$\alpha = 0.7$	5595	13.53
	$\alpha = 0.8$	5652	13.58
	$\alpha = 0.9$	5844	13.80
	$\alpha = 1.0$	6427	13.95

表 3 α の値が異なる AI 同士を対戦させたときの打ち方の結果

		平均上がり得点	平均上がり巡目
先行研究	$\alpha = 0.0$	3734	12.23
	$\alpha = 0.3$	4236	12.29
	$\alpha = 0.7$	5022	12.39
	$\alpha = 1.0$	6110	13.26

4.2 提案手法の有用性

打ち方の変更ができているため，大局的な状況に応じて α を変更する手法の有用性を確かめる．先行研究の手法の $\alpha = 0.0$，$\alpha = 0.9$，$\alpha = 1.0$ と提案手法を対戦させる．$\alpha = 0.9$ と対戦させたのは，先行研究の実験において，このパラメータが多人数麻雀で強いと考察されたためである [7]．リーチ無しの場合とリーチ有りの場合に分けて対戦させる．リーチ有りの場合は先行研究の手法, 提案手法ともにリーチのアルゴリズムを組み込んだ手法（Algorithm 2）とする．1 位〜4 位率，平均順位，平均得点について比較する．平均点数はゲーム終了時に持っていた点数の平均である．1 回のゲームは半荘とし，1000 回行う．シミュレーション回数は 1000 回とする．

リーチ無しの場合の結果を表 4，リーチ有りの場合の結果を表 5 に示す．リーチ無しの場合では，1 位率や 4 位率，平均順位，平均点数において先行研究の手法の $\alpha = 0.9$ よりも悪い結果となってしまった．しかし，リーチ有りの場合では，これらの点において先行研究の手法よりも良い結果となった．リーチをすることで，手牌に特定のパターンがなくても上がることができる．すなわち，他の役がなくてもリーチを宣言することで，上がったときに役としてリーチがあるため上がることができる．リーチを手法に組み込むことで上がりやすくなるため，α が小さいときはより早上がりになる．また，リーチをして上がると役としてリーチがある．役の数が多いほど点数が上がるため，α が大きいときはより高得点となる．このことから，リーチをしていないときよりも打ち方を変更することが有効に働いたと考えられる．これらの結果から，大局的な状況に応じて打ち方の変更を行う麻雀 AI が有用であると言える．

Mjai の作者である市川は mjai-manue という麻雀 AI を

表 4　先行研究との比較実験の結果 (リーチ無し)

	提案手法	先行研究		
	α 変更	$\alpha = 0.0$	$\alpha = 0.9$	$\alpha = 1.0$
1 位率 (%)	27.3	16.8	**32.5**	23.4
2 位率 (%)	24.9	25.8	24.9	24.4
3 位率 (%)	24.2	29.1	21.5	25.2
4 位率 (%)	23.6	28.3	**21.1**	27.0
平均順位	2.44	2.69	**2.31**	2.56
平均点数	25483	22705	**27344**	24468

表 5　先行研究との比較実験の結果 (リーチ有り)

	提案手法	先行研究		
	α 変更	$\alpha = 0.0$	$\alpha = 0.9$	$\alpha = 1.0$
1 位率 (%)	**28.2**	25.3	24.7	21.8
2 位率 (%)	25.2	25.8	26.3	22.7
3 位率 (%)	22.5	24.8	24.1	28.6
4 位率 (%)	**24.1**	**24.1**	24.9	26.9
平均順位	**2.43**	2.48	2.49	2.61
平均点数	**25998**	25813	25097	23093

表 6　mjai-manue との比較実験の結果

	提案手法	市川作	先行研究	
	α 変更	mjai-manue	$\alpha = 0.0$	$\alpha = 1.0$
1 位率 (%)	27.5	**32.8**	21.9	17.8
2 位率 (%)	22.8	35.1	25.6	16.5
3 位率 (%)	23.8	24.0	24.6	27.6
4 位率 (%)	25.9	**8.1**	27.9	38.1
平均順位	2.48	**2.07**	2.59	2.86
平均点数	25444	**32022**	23049	19485

作成している [9]．この mjai-manue と提案手法，先行研究の手法の $\alpha = 0.0$，$\alpha = 1.0$ を対戦させてみた．リーチは有りとし，実験内容は先述した内容と同じである．結果を表 6 に示す．1 位率や 4 位率，平均順位，平均点数において mjai-manue が一番良い結果となった．麻雀は自分が捨てた牌で他プレイヤが上がった場合，自分は上がったプレイヤに上がり点を支払わなければならない．mjai-manue は自分の手牌だけを考慮せず，他プレイヤに上がらせないように行動選択をしているため，4 位率がかなり低くなっていると考えられる．このように，他プレイヤに上がらせないにように行動選択することは麻雀 AI において重要であることが分かる．

5. まとめと今後の課題

本論文では，状況に応じて打ち方を変更する麻雀 AI の有用性を確かめた．計算機実験により，打ち方を固定する場合と対戦させ，変更する場合が良い結果を得ることができた．

現在，打ち方のパラメータの設定は，人手で決めた値となっている．良いパラメータ設定でないためか，リーチ無しの場合では先行研究よりも悪い結果となってしまった．

今後の課題として，大局的な状況に応じた良いパラメータの自動設定を試みる．大局的な状況に応じて，パラメータを自動的に調整する関数を作成することを考えている．また，何回もゲームを行い，ゲームの結果から関数の調整をするような学習を行い，良い戦略を取れる関数を作成したい．また，今回の手法はリーチの選択が単純なアルゴリズムになっている．リーチをするか否かを，シミュレーションから決定する手法にしたいと考えている．mjai-manue との対戦でわかるように，自分の捨てる牌で他プレイヤが上がらないように行動選択することは麻雀 AI において重要である．他プレイヤが上がれそうな牌を考慮する手法を作成する必要がある．

参考文献

[1] 北川竜平, 三輪誠, 近山隆. 麻雀の牌譜からの打ち手評価関数の学習. ゲームプログラミングワークショップ 2007 論文集, pp. 76–83, 2007.

[2] 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系研究科, 2010.

[3] 根本佳典, 古宮嘉那子, 小谷善行. CRF を用いた麻雀の不完全情報推定. ゲームプログラミングワークショップ 2012 論文集, pp. 155–158, 2012.

[4] Naoki Mizukami and Yoshimasa Tsuruoka. Building a computer Mahjong player based on Monte Carlo simulation and opponent models. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 275–283. IEEE, 2015.

[5] 小松智希, 成澤和志, 篠原歩. 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. 情報処理学会研究報告. GI, Vol. 2012-GI-28, No. 8, pp. 1–8, 2012.

[6] 原田将旗, 古宮嘉那子, 小谷善行. 麻雀における手牌と残り牌からの上がり探索による着手決定アルゴリズム CHE. 情報処理学会研究報告. GI, Vol. 2014-GI-31, No. 13, pp. 1–4, 2014.

[7] 海津純平, 成澤和志, 篠原歩. 一人麻雀における打ち方を考慮した評価指標に関する研究. ゲームプログラミングワークショップ 2015 論文集, pp. 172–178, 2015.

[8] 市川宙. Mjai 麻雀対戦サーバ. https://github.com/gimite/mjai.

[9] 市川宙. Mahjong AI mjai-manue. https://github.com/gimite/mjai-manue.