複数クライアントによる麻雀の協調支援手法の提案

浅尾 豪[1,a)]　大西 建輔[2]

概要：特定の相手を支援, つまりその相手に必要な牌を捨てることで有利にする麻雀支援クライアントの開発をおこなってきた. 本研究では, 4 人麻雀において特定の相手を 3 人で協調して支援し, 支援対象の捨て牌から相手の待ち牌を推定することを目標とした. その結果, 非支援時, 単独での支援時と比べ, 待ち牌の推定, 支援対象の出和了率の向上が確認できた. また, 局終了時に待ち牌が残っている場合は, 平均して 8 番目にあることがわかった.

A method of entertaining mah-jong client by cooperative support

Go Asao[1,a)]　Kensuke Onishi[2]

Abstract: We have investigate entertaining mah-jong client i.e., the client supports a target by discarding useful tile for the target. In this study, we consider that mah-jong with one target and three support clients. We propose that a cooperative support method by support clients. By computational experiment, the rate of the target's winning by proposed method increases than that by non-support and individual support methods. When the game is finished without winning, the winning tile of the target has about 8th rank in an order of discard.

1. はじめに

昨今, 将棋, 囲碁などの分野ではコンピュータによるクライアントの開発が盛んにおこなわれ, 強いクライアントの開発が報告されている. 麻雀の分野においても強いクライアント開発の研究 [1][2] がなされている.

一方で, 私たちはユーザーを楽しませるクライアントを作成しようと考えた. その一環として, 安西の作成した支援麻雀をするクライアントの開発 [3] がある. 支援とは特定の相手（以下支援対象）を多く勝たせ, 楽しいと感じさせることを目的とした戦法である. 支援麻雀クライアントでは支援対象に対して危険だと判断した牌を捨てることで, 他の対戦相手よりも支援対象を多くあがらせようとした. このクライアントでは, 支援をおこなわないクライアントと比べて支援対象の 1 位率, 3 位率が上昇した. また, 安西の研究を参考に, 支援麻雀クライアントの開発, 手法の提案をおこなった [4]. この手法は支援対象以外に標的を定め, 標的から和了ることで支援クライアントの見破られ易さを隠ぺいしようとした. しかし, この手法では支援クライアントの隠ぺい率を向上させることはできなかった. これまでの支援麻雀クライアントは支援対象の待ち牌の推定をおこなわないため, あまり高い精度で支援をおこなうことができていない.

本研究では, 支援対象の捨て牌から待ち牌を推定することを目標とした. しかし, 4 人麻雀で待ち牌を推定することは困難である. そのため, 支援クライアントが協調して支援することで, 捨て牌から待ち牌を推定することが可能なのかの検証をおこなった.

2. 支援手法

支援をおこなうためには支援対象に振り込む必要がある. そのため, 支援時は支援対象に対してより危険と判断する牌を優先的に捨てるようにする. 牌 $p_i(0 \leq i \leq 33)$ に設定した値を危険度と呼び, $rank_i$ とする. 危険度は支援対象の捨て牌から現物牌, スジ牌, 裏スジ牌を求め, 対応する

[1] 東海大学大学院理学研究科
Graduate School of Science, The University of Tokai
[2] 東海大学
The University of Tokai
[a)] 4bsfm001@mail.tokai-u.jp

Algorithm 1 支援対象の捨て牌 $p[i]$ への処理

Var: $p[]$ 各牌; $risk[]$ 各牌の危険度;
1: $risk[i] \leftarrow$ 現物
2: **for** $j = 0$ to 33 **do**
3: **if** $risk[j] >$ スジの値 **then**
4: **if** $p[j] = p[i]$ のスジ牌 **then**
5: $risk[j] \leftarrow$ スジの値
6: **end if**
7: **if** $p[j] = p[i]$ の裏スジ牌 **then**
8: $risk[j] \leftarrow risk[j] * 2$
9: **end if**
10: **end if**
11: **end for**

Algorithm 2 染めへの対応

Var: $p[]$ 各牌; $risk[]$ 各牌の危険度;
Type: $TMan = 0..8$（萬子）;
$TPin = 9..17$（筒子）;
$TSou = 18..26$（索子）;
Var: $teban$ 局開始時から支援対象が捨てた回数;
Var: $countm, countp, counts$
支援対象が捨てた萬子, 筒子 , 索子の枚数;
Var: $flagm, flagp, flags$ 既に危険度を下げたかどうか;
1: **if** $teban > 7$ **then**
2: **if** $(countm > 4) \wedge (\neg flagm)$ **then**
3: **for all** $i \in TMan$ **do**
4: $risk[i] \leftarrow risk[i]/2$
5: **end for**
6: $flagm = TRUE$
7: **end if**
8: **if** $(countp > 4) \wedge (\neg flagp)$ **then**
9: **for all** $i \in TPin$ **do**
10: $risk[i] \leftarrow risk[i]/2$
11: **end for**
12: $flagp = TRUE$
13: **end if**
14: **if** $(counts > 4) \wedge (\neg flags)$ **then**
15: **for all** $i \in TSou$ **do**
16: $risk[i] \leftarrow risk[i]/2$
17: **end for**
18: $flags = TRUE$
19: **end if**
20: **end if**

値を設定した. まず, 指標となる牌の危険度を科学する麻雀 [5] の降りる際の牌の危険度を参考に, 初期値を**表 1** の基本危険度に設定した. 支援対象の捨て牌 p_i に対してそれぞれの牌の危険度を変更する. この手法をまとめたものが **Algorithm 1** である.

Algorithm 1 を説明する. まず, 支援対象の捨て牌 p_i の危険度 $risk_i$ を現物の値にする（1 行目）. 次に, p_i のスジ牌 p_j の危険度 $risk_j$ がスジの値（1, 9 牌なら 4 であり, 2, 8 牌なら 16 である）よりも大きいならば, $risk_j$ をスジの値とする（5 行目）. 同様に, p_i の裏スジ牌 p_j に対して, もし $risk_j$ がスジの値より大きいならば, $risk_j$ を 2 倍にする（8 行目）.

一方, Algorithm 1 だけでは染め手に対して対策がなされていない. 染め手の場合, 同一種類の牌を多く捨てている傾向がある. 染め手に対策をすることで通常よりも支援対象の待ち牌の推定精度を向上させられると考えられる. そこで, 局開始時から捨て牌の種類を数牌ごとに記憶しておき, その枚数をそれぞれ $countm, countp, counts$ とおく. そして, 各種類の捨て牌数が 5 枚以上で, 手番数が 8 順目以降の場合, その種類の数牌の危険度を一度だけ半分にする. これをまとめたのが **Algorithm 2** である.

表 1　牌の危険度
Table 1　Risk of mahjong tile

スジ	基本危険度	初期枚数	初期順位
現物	0	0	0
単騎字牌	2	0	0
スジの 1 牌, 9 牌	4	0	0
単騎以外の字牌	8	7	28
スジの 2 牌, 8 牌	16	0	0
スジの 3 牌, 7 牌	32	0	0
スジでない 1 牌, 9 牌	64	6	22
片側スジの 4, 5, 6 牌	128	0	0
スジでない 2 牌, 8 牌	256	6	16
スジでない 3 牌, 7 牌	512	6	10
スジでない 4, 5, 6 牌	1024	9	1

また, 局開始時から支援すると, 支援対象が聴牌する前に 4, 5, 6 牌のような数牌を捨ててしまい, 結果としてこれらの牌の危険度が下がり, 支援が上手くいかない場合が多々あった. そこで, 支援開始手番を設定した. 支援開始手番までは危険度の低い牌を捨て, 支援開始手番以降は危険度の高い牌を捨てる. 今回の実験では, 事前に取得した支援対象の聴牌手番の平均値である 12 順目と, 総聴牌数の内, 2 割以上が聴牌していた 6 順目を支援開始手番とした. 支援中は支援対象が聴牌していると仮定して牌を捨てている. そのため, 自分が捨てた牌で振り込まなかった場合, その牌は支援対象の待ち牌ではないと判断し, 自分の捨てた牌 p_i の危険度 $risk_i$ を 1/4 とした.

また, 今回支援対象以外は支援クライアントである. そこで, 支援開始手番以後の支援対象以外の捨て牌 p_i の危険度 $risk_i$ も 1/4 とし, 協調支援を実現した.

3. 計算機実験

3.1 実験概要

作成した支援クライアント 3 体と自作の支援対象クライアント 1 体の計 4 体で 4 人麻雀をおこない, 支援対象に対する支援をおこなった. 対戦回数は半荘約 1300 回, 計 13000 局以上実行した. 実行環境はまうじゃん for Java[6] を使用した. 収集データとして支援対象の各局ごとの待ち牌を出力した. また, 支援クライアントの各手番ごとの支援対象に対する各牌の危険度の一覧を出力した.

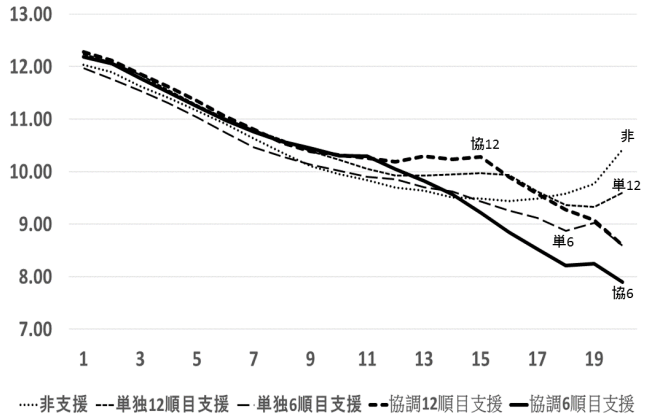

図 1　待ち牌の順位
Fig. 1　Rank of wait mahjong tile

表 2　クライアントごとの支援対象の出和了率
Table 2　Ron rate of supported target for each client

	出和了率
非支援	10.90%
単独 12 順目	14.30%
単独 6 順目	16.54%
協調 12 順目	15.17%
協調 6 順目	17.66%

3.2　実験結果

支援対象の危険度の一覧データから, 支援対象が聴牌した局での, 各手番 j ($0 \leq j \leq 20$) 時における支援対象の待ち牌の危険度の順位を求め, その平均を $rank_j$ とする. **図 1** は非支援, 単独支援, 協調支援（支援開始手番がそれぞれ 6 順目, 12 順目）をおこなったそれぞれの支援クライアントの $rank_j$ の推移である. 図 1 より, 単独, 協調のどちらとも 6 順目支援は, 手番が進んだ時点の $rank_j$ が 12 順目支援時よりも小さくなっている. また, 17 順目以降では, 協調支援の $rank_j$ が単独支援よりも小さくなっている.

表 2 は, 各クライアントと対戦した支援対象の出和了率（総局数中の出和了回数）である. 表 2 より, 協調支援は単独支援と比べて 1%から 2%ほど支援対象の和了率が向上している. 従って, 協調支援は, 単独支援よりも支援対象に多く振り込むことができていた. また, 最も出和了率が高いのは, 協調 6 順目支援の場合であり, 図 1 の $rank$ が最も低くなる場合に対応している.

しかし, 協調 6 順目支援でも最終的な $rank_j$ は 7.90 である. 待ち牌が 34 枚中 8 番目に危険な牌では読みの精度が高いとは言えない. 従って今回の手法だけでは十分な待ち牌の推定ができているとは言えない.

4.　考察

4.1　順位の期待値

順位の平均 $rank_j$ がどのような値をとるかを計算で確認する. まず, 1 順目の順位が正しい結果となっているかを期待値を求めて比較する. 1 順目の牌の危険度 $risk_i$ は表 1 の基本危険度に設定している. そのため, 1 順目の順位は必ず表 1 の初期順位になる. まず, 支援対象の待ち牌が 34 種類の中のただ 1 つのみの場合を考える. 待ち牌がそれぞれの種類である確率は全体の枚数分のその種類の牌の枚数である. 従って, 待ち牌が 1 種類の場合の期待順位 E_1 は

$$E_1 = (1*9 + 10*6 + 16*6 + 22*6 + 28*7)/34 = 13.44$$

となる.

次に, 待ち牌が 2 種類の場合も同様に期待値を計算する. 待ち牌が 2 種類以上の場合, 待ち牌の順位は順位が低い方の値を採用している. 仮に, 待ち牌が 1, 4 萬だった場合, 初手の順位が低い 4 萬の順位が待ち牌の順位となる. 待ち牌が 2 種類の場合の期待順位 E_2 は

$$E_2 = (1*49 + 10*120 + 16*192 + 22*264 + 28*531)/1156 = 7.87$$

となる.

最後に待ち牌が 1 枚である時と 2 枚である時の比は今回の実験では**表 4** に示すように 24:17 であった. 従って 1 順目の期待順位は

$$E = (24*E_1 + 17*E_2)/41 = 11.29$$

である. この結果は図 1 の 1 順目の順位とほぼ一致している. このことから 1 順目の順位は期待値通りであったことが分かった.

2 順目以降は牌の危険度 $risk_i$ が順次変更されていくため, 確率を求めることが困難となる. 2 順目以降の期待順位の算出は今後の課題である.

表 3　初手の待ち牌の確率
Table 3　Initial probability of waiting tile

牌の種類	1 種類の確率	2 種類の確率
字牌	7/34	49/1156
1, 9 牌	6/34	120/1156
2, 8 牌	6/34	192/1156
3, 7 牌	6/34	264/1156
4, 5, 6 牌	9/34	531/1156

表 4　待ち牌の種類数
Table 4　The number of waiting tile

待ち牌が 1 種類	待ち牌が 2 種類	合計
2401	1706	4107

4.2　危険度の値

次に設定した危険度の値が信用できるかを検証する. **表 5** は支援対象が聴牌状態であった 4107 局における, 各種類

表 5　各種類の待ち牌の総数
Table 5　Total number of waiting tile for each class

	総数 (割合)		1 種類あたり	
	両面以外	両面	両面以外	両面
1, 9 牌	280(4.76%)	321(5.46%)	46.67	53.50
2, 8 牌	798(13.57%)	269(4.58%)	133.0	44.83
3, 7 牌	1239(21.08%)	293(4.98%)	206.5	48.83
4, 5, 6 牌	789(13.42%)	883(15.02%)	87.67	98.11
字牌	1007(17.13%)	0(0.00%)	143.8	0.00

ごとの支援対象の待ち牌の総数と, それぞれの総数を牌の種類数で割った値をまとめたものである. 1, 9 牌が待ち牌であったのは全体の 10.22%(=4.76%+5.46%) であった. それに比べて, 4, 5, 6 牌が待ち牌となっていたのは全体の 28.44%(=13.42%+15.02%) であった. 従って, 1, 9 牌の危険度を低く設定したことは正しかったと考えられる. また, 4, 5, 6 牌が待ち牌となっていた割合は全体で最も高かった. 従って, 4, 5, 6 牌の危険度を最も高く設定したことは正しかったと考えられる.

一方, 1 種類あたりの比率では 4, 5, 6 牌が待ち牌となる比率よりも, 3, 7 牌が待ち牌となる比率の方が高くなっている. この原因が支援対象側にあるのか, 支援側にあるのかは今後究明していく予定である. そのため, 様々なクライアントと対戦させ, 支援対象側にあるのかを調べていく予定である. また, 字牌で待つ回数も少なくない. 今後はもう少し字牌の危険度を高めて同様の実験を繰り返す予定である.

5. おわりに

読みの精度が低い原因として支援対象が聴牌する前に捨てた牌が待ち牌となり, 危険度を下げてしまっている可能性が考えられる. 聴牌前に待ち牌となる牌を捨ててしまっている状況は, 対局 23069 局中, 4467 回あり, 全体の 2 割程度だった. 今後, この状況が待ち牌の推定に影響しているかを調べていく予定である.

また, 待ち牌の内訳は両面待ちのみで考えると 4, 5, 6 牌が多いが, 全体では 3, 7 牌が高くなっていた. この結果から危険度の値を修正する必要があると考えられる. しかし, 今回の支援対象が両面待ちをしなかっただけとも考えられる. 従って, 今後は様々なクライアントと対戦させ, より多くのデータを取っていきたい.

最後に, 提案手法の期待順位は初手以外求めることはできなかった. 今後は, マルコフ連鎖などを用いて提案手法の期待順位を確認していく予定である.

参考文献

[1] 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, 2010 年度東京大学工学系研究科電気系工学専攻修士論文 (2010).
[2] 水上直紀, 中張遼太郎, 浦晃, 三輪誠, 鶴岡慶雅, 近山隆：降りるべき局面の認識による単独麻雀プレイヤの 4 人麻雀への適用, ゲームプログラミングワークショップ 2013 論文集, pp1-7(2013).
[3] 安西諒祐：接待をする麻雀クライアントの開発とその評価, 組み合わせゲーム・パズル ミニプロジェクト, 第 8 回研究集会, http://www.alg.cei.uec.ac.jp/itohiro/Games/Game130301.html(2013).
[4] 浅尾豪：狙い撃ち手法による支援麻雀クライアントの開発, FIT2014(第 13 回情報科学技術フォーラム), 第 2 分冊, pp.289-290(2014).
[5] とつげき東北：科学する麻雀, 講談社現代新書 (2004).
[6] 石畑恭平：コンピュータ麻雀のアルゴリズム, 工学社 (2007).