Vol.2015-ICS-179 No.21
2015/3/20

不完全情報ゲームにおける ゲームプレイングエージェントの試作と抽象化手法の改良

谷口 慶一郎[1]　伊藤 孝行[1]

概要：不完全情報ゲームにおいては，ナッシュ均衡に基づく戦略を取ることは有効な手段であるが，ゲーム木のサイズが膨大であることから，厳密なナッシュ均衡を求めることは困難である．そこで，ゲーム木の抽象化を用いることで，ゲーム木を縮小し，抽象化されたゲーム木に対して近似ナッシュ均衡を求める研究が行われている．近似ナッシュ均衡の精度は，どの程度元のゲーム木の情報を落とすことなく抽象化するかに大きく依存する．そこで本研究では，不完全情報ゲームとしてポーカーの種目の一つである 2 player Limit Texas Hold'em を題材として，従来の抽象化手法の改良手法を提案する．k-means 法による抽象化を行う際に，ゲーム木の偶然手番を表す特徴量として，Hand Strength Distribution (HSD) と Opponent Cluster Hand Strength(OCHS) が用いられている．しかし，OCHS は，HSD と比較して偶然手番を表す情報量が極端に少ないという問題点がある．そこで，OCHS のクラスタ数を変更することで改良を行い，OCHS のクラスタ数を変更したエージェント間で対戦する実験を行い，OCHS のクラスタ数を変更することで，良い抽象化が行うことができるかを検証した．

1. はじめに

不完全情報ゲームでは，ナッシュ均衡に基づく戦略（ナッシュ均衡戦略）を取ることは有効な手段である．近年の先行研究（[1] [2] [3] [4] [5] [6]）では，ポーカーを展開形ゲームとして捉えゲーム木で表し，ゲーム木からナッシュ均衡を求める試みがなされている．しかし，不完全情報ゲームの厳密なナッシュ均衡を求めることは，空間的，時間的に困難である．そこで，不完全情報ゲームのナッシュ均衡を探索するため，部分的に同じゲーム木をまとめる，偶然手番の数を減らすなどの手段でゲーム木のサイズを縮小する，ゲーム木の抽象化が行われる．そして，抽象化されたゲーム木から近似的なナッシュ均衡を求めて実際のゲームの戦略として用いる．近似ナッシュ均衡の精度は，どの程度元のゲーム木の情報を落とすことなく抽象化するかに大きく依存する．

従来の抽象化手法のひとつとして，k-means 法による偶然手番の抽象化が挙げられる [8] [10] [11]．k-means 法による偶然手番の抽象化を行う際には，ゲーム木の偶然手番を表す特徴量を表す必要がある．先行研究では偶然手番を表す特徴量として，Hand Strength Distribution (HSD)[10] と Opponent Cluster Hand Strength (OCHS)[10] が用いられている．しかし，OCHS は HSD と比較して，表すことができる情報量が極端に少ないという問題点がある．そこで本研究の目的は，OCHS のクラスタ数を変更することによる，OCHS の改良とする．OCHS のクラスタ数を 6, 8, 10, および 12 に設定したエージェントをそれぞれ作成し対戦させる．対戦結果から，OCHS のクラスタ数を増やすことで，良い抽象化が実現可能であることを示す．

2. 背景と関連研究

2.1 2 player Limit Texas Hold'em

本研究で題材とする 2 player Limit Texas Hold'em はポーカーの一種で 2 人ゲームである．2 枚の手札と 5 枚の場のカードから，5 枚を選びプレイヤーの役とする．チップを賭ける Round が 4 回ある．Round 1 では，両プレイヤーに 2 枚ずつ手札が配られる．Round 2, 3 および 4 では，それぞれ 3 枚，1 枚および 1 枚の場のカードが開かれる．また各 Round では，プレイヤーは 3 種類の行動を取ることができる．手札を捨てゲームを諦める Fold，相手と同額を賭ける Call，および賭け金を上乗せする Raise である．各 Round で行うことができる Raise 回数は 4 回で，Raise 額は Round 3 および 4 では，Round 1 および 2 の倍額である．両プレイヤーが Fold することなく，Round 4 を終えた時点で，より強い役を作ったプレイヤーが全賭け金を得ることができる．

[1] 名古屋工業大学 工学部 情報工学科
Nagoya Institute of Technology

ⓒ 2015 Information Processing Society of Japan

2.2 展開形ゲーム

ゲームは多くの場合，いくつかの手番から成っており，それぞれの状況ごとにどのような行動をとるかを考える必要がある．手番の系列としてゲームを表現する方法として展開形ゲームがある．ポーカーの場合，カードが配られることも含めて，プレイヤーの行動の系列によって利得が決まるため，展開形ゲームで表すことができる．また，展開形ゲームはゲーム木で表すことができる．ゲーム木は，ゲームの状態を表すノードと，あるプレイヤーが，ある行動を取ったことを表すエッジから成る．ノードは，あるプレイヤーが行動を選択する人的手番と，どのプレイヤーの意思にも無関係な選択が行われる偶然手番がある．ポーカーの場合，トランプを配るノードが偶然手番である．また，ゲーム木の葉は，各プレイヤーがいくらの利得を得るかを表す．

2.3 ポーカーエージェントの関連研究

【ルールや数式に基づいた戦略をとるエージェント】

ルールの記述や数式を基に行動を決定する知識ベースのプレイヤーは数多く研究されている [14]．しかし，ルールに基づいたエージェントは作成者の熟練度に依存することや，膨大なルールの記述によるルール間の衝突が問題になっている．

【対戦相手のモデル化を用いたエージェント】

現実の世界においても，ゲームの対戦相手の戦略を予測することは有効な手法である．そこで，対戦相手のモデル化を行う研究がなされている [15]．しかし，複数の戦略を切り替えて用いるエージェントの場合，対戦相手のモデル化の精度だけでなく，モデルの学習速度や戦略が切り替わったときに素早く適応できることが必要である．モデル化の例として，行動の頻度（Action Frequencies）を利用する手法 [15] が挙げられる．また，ニューラルネットワークを用いた対戦相手のモデル化を行う手法も提案されている．

【抽象化を用いたエージェント】

ゲーム木の抽象化

Texas Hold'em のゲーム木は膨大であり，計算量的に近似ナッシュ均衡を計算することが困難である．そこで，ゲーム木のサイズを縮小する抽象化の研究がなされている [2] [3] [8] [7]．もとのゲーム木の情報を損失することなく抽象化するほど，より良い近似ナッシュ均衡を計算することができる．

部分木の抽象化

部分木の抽象化は，ゲーム木の中で同じ形を持った部分木を探し，集約する手法である [8]．全く同じパターンのゲーム木を集約するため，もとのゲーム木の情報を損失することなく，ゲーム木のサイズを縮小することができる．しかし，部分木の抽象化だけでは，ナッシュ均衡を探索できるサイズまで縮小することができない．

表 1 各 Round における人的手番と偶然手番の数

Round	人的手番	偶然手番
1	8	1,326
2	70	25,989,600
3	630	1,221,511,200
4	5,670	2,428,287,420

手札の抽象化

Texas Hold'em ではカードのマーク間に強さ関係がないことに着目して，偶然手番の削減を図っている [7] [9] [10]．しかし，ナッシュ均衡を計算する上では，マークの抽象化だけでは不十分である．そこで，手札の集約を行う．先行研究では，カードの強さ，カードが持っているポテンシャルを考え，Bucket という形でカードをまとめることで手札の集約を行っている [7]

プレイヤーの手番の抽象化

偶然手番の抽象化ではなく，プレイヤーの手番を抽象化することによってゲーム木の縮小を図る手法である [9]．本研究では取り扱っていない No Limit Texas Hold'em での抽象化に有効な手法である．

3. ゲーム木の抽象化とナッシュ均衡探索

3.1 2 player Limit Texas Hold'em のゲーム木と偶然手番の抽象化の必要性

2 player Limit Texas Hold'em のゲーム木における，各 Round 毎の偶然手番の数および人的手番の数を表 1 に示す．

表 1 から分かるように，2 player Limit Texas Hold'em においては，人的手番の数と比較して，偶然手番の数が非常に多いことが分かる．そこで，偶然手番を抽象化することで，ゲーム木全体の縮小を図る．

3.2 近似ナッシュ均衡の探索までの流れ

2 player Limit Texas Hold'em のゲーム木の抽象化を行う際に用いたアルゴリズム，およびゲーム木から近似ナッシュ均衡を求めるアルゴリズムについて述べる．はじめに，偶然手番の抽象化の第一段階として Hand Isomorphism (HI)[12] を用いる．HI は情報無損失な抽象化アルゴリズムで，カードのマーク間に強さ関係がないことに着目したアルゴリズムである．次に，偶然手番の抽象化の第二段階として，k-means 法を用いた偶然手番のクラスタリングを行う [3], [7], [10]．各 Round 毎にクラスタリングを行うことで偶然手番の削減を図る．偶然手番を表す指標として，Round 2 および 3 では，Hand Strength Distribution (HSD) を用い，Round 4 では，Opponent Cluster Hand Strength (OCHS) を用いる．詳細は後述する．以上の抽象化手法を用いて抽象化されたゲーム木から，CounterFactual Regret Minimization (CFR)[13] を用いて近似ナッシュ均衡を計算する．

表 2 HI を適用したときの各 Round の偶然手番の数

Round	Hand Isomorphism なし	Hand Isomorphism あり
1	1,326	169
2	25,989,600	1,286,792
3	1,221,511,200	55,190,538
4	56,189,515,200	2,428,287,420

3.3 Hand Isomorphism (HI)

Hand Isomorphism (HI)[12] は，情報無損失な抽象化である．Texas Hold'em にはマークについての強さがないことに着目して，性質が同じカードの組み合わせの集合にインデクシングを行う．

例えば，2♡ と 2♠ は，新たにカードが開かれる前は同じものとして扱うことができる．しかし，後に 3♠ が開かれた場合は，2♠ は 2♡ より強くなり，2♡ と 2♠ は同等ではなくなる．これは，Texas Hold'em にはフラッシュという 5 枚のカードがすべて同じマークの役があるからである．図 2 に，HI を用いた後の Round 毎の偶然手番の数を示す．

表 2 より，情報無損失で大幅な偶然手番の削減できていることが分かる．

3.4 k-means 法による抽象化

偶然手番の抽象化手法の 1 つである k-means 法を用いたクラスタリングによる抽象化について述べる．k-means 法を用いた抽象化は各 Round 毎に行う．ただし，Round 1 では他の Round に比べて，偶然手番の数が少ないため k-means 法による抽象化は行わない．Round 1 を除いた各 Round のクラスタ数は，それぞれ 9000 とする．クラスタ数 9000 という数は，[10] で用いられたクラスタ数である．

k-means 法を用いたクラスタリングを行うためには，データ点である各偶然手番を表す指標が必要になる．偶然手番を表す指標として，Round 2 および 3 では Hand Strength Distribution (HSD)[10] を用い，Round 4 では Opponent Cluster Hand Strength (OCHS)[10] を用いる．

【Hand Strength (HS)】

Hand Strength (HS) は，Round 4 における，自分の手札の引き分けを考慮した勝率である [10]．Round 4 においては，場のカードは 5 枚すべてが開かれているため，自分の手札 2 枚と合わせて役が確定している状態である．従って，相手の手札の組み合わせをすべて考え，そのうちいくつの手札に対して勝利したか，引き分けたかを計算することができる．勝利した組み合わせ数を w，引き分けた組み合わせ数を d とする．式 (1) に HS の式を示す．

$$\text{Hand Strength(HS)} = \frac{\mathrm{w} + 0.5 \times \mathrm{d}}{{}_{45}\mathrm{C}_2} \qquad (1)$$

式 1 の分母の ${}_{45}\mathrm{C}_2$ は，相手の手札の総数を表している．

【Hand Strength Distribution (HSD)】

Hand Strength Distribution (HSD) は，開かれていないカードがある Round の偶然手番が，どの程度の頻度で，どの程度の Hand Strength (HS) になるかを表す確率分布である [10]．HSD は，0 から 1 までの HS を 0.02 刻みにした 50 個のヒストグラムで表す．また，各ヒストグラムは，開かれていないカードが開かれたときに，ある HS になる確率を表す．HSD の例として，Round 2 において手札が J♣ 9♡，場のカードが 4♢ 5♢ 9♢ のときの HSD を図 1 に示す．

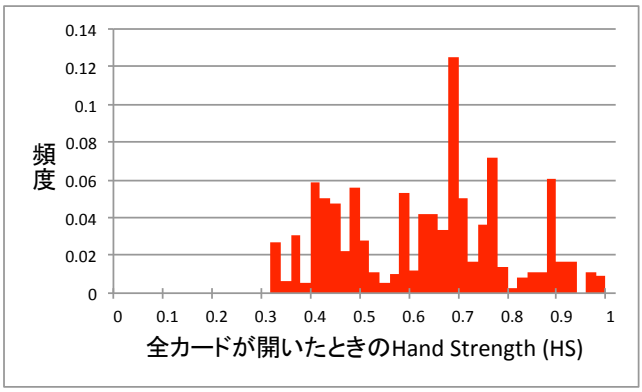

図 1 Round 2 手札 J♣ 9♡ 場のカード 4♢ 5♢ 9♢ のときの HSD

図 1 の横軸は HS を表し，縦軸はある HS になる頻度を表している．例えば，全カードが開いた時に 0.68 から 0.7 の HS になる確率は約 0.13 であることを表している．各ヒストグラムは，場のカードがすべて開かれたときに，対応する範囲の HS になる確率を表している．従って，HSD ではすべてのヒストグラムの和は 1 になる．本研究では，HSD を Round 2 および 3 で用いる．

【Opponent Cluster Hand Strength (OCHS)】

Hand Strength Distribution (HSD) は，Round 4 では用いることができない．Round 4 では開いていないカードがないため，HSD ではなく，Hand Strength (HS) しか計算できないためである．そこで，Round 4 では，HSD の代わりに Opponent Cluster Hand Strength (OCHS) を特徴量として用いる [10]．本研究では，OCHS を Round 4 の偶然手番を表す指標として用いるが，OCHS は全 Round に適用可能である．

手札の組み合わせは ${}_{52}\mathrm{C}_2 = 1{,}326$ 通り存在する．実際には Hand Isomorphism によって，169 通りを考えればよい．手札 169 通りをいくつかのクラスタに分類したものを Opponent Cluster (OC) という．本研究ではまず，クラスタ数が 8 の OC を考える．OC は Round 1 の偶然手番，すなわち手札を HSD で表し，Earth mover's distance (EMD) を距離指標としてクラスタリングすることで求めることができる．手札 169 通りを 8 つにクラスタリングした結果を，図 2 に示す．

図 2 の Suited はマークが同じ手札の組み合わせを表し，Unsuited はマークが異なる手札の組み合わせを表す．例えば，手札の数字が 5 と 6 の場合，マークが同じときはクラスタ 3 に分類され，マークが異なるときはクラスタ 1 に

ⓒ 2015 Information Processing Society of Japan

Unsuited \ Suited	2	3	4	5	6	7	8	9	T	J	Q	K	A
2	4	1	1	1	1	1	2	2	2	4	4	4	6
3	1	6	1	1	1	1	2	2	3	4	4	6	6
4	1	1	6	1	1	2	2	2	3	4	4	6	6
5	1	1	1	6	3	3	3	3	3	4	4	6	6
6	1	1	1	1	7	3	3	3	3	4	5	6	6
7	1	1	1	2	3	7	3	3	5	5	5	6	7
8	1	1	2	2	3	3	8	3	5	5	5	6	7
9	2	2	2	2	3	3	3	8	5	5	5	7	7
T	2	2	2	2	3	3	3	5	8	5	7	7	7
J	2	2	4	4	4	4	5	5	5	8	7	7	7
Q	4	4	4	4	4	4	5	5	5	5	8	7	7
K	4	4	4	6	6	6	6	6	7	7	7	8	7
A	6	6	6	6	6	6	6	7	7	7	7	7	8

図 2 手札 169 種類を 8 個のクラスタに分類した結果（[10] Table 1）

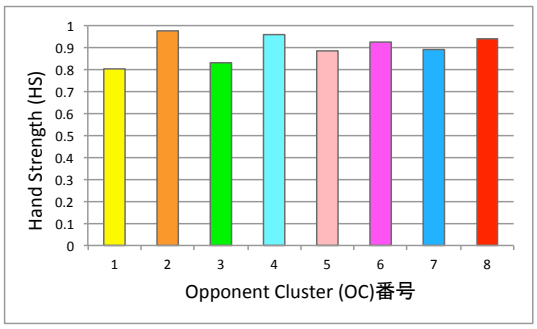

図 3 Round 4 手札 J♣ 9♡ 場のカード 4♢ 5♢ 9♢ 6♣ J♠ のときの OCHS

分類される．また，クラスタ番号が大きいクラスタに属している手札ほど，強い手札であることを表している．

各 Opponent Cluster に対する Hand Strength (HS) または E[HS] をヒストグラムで表したものが，Opponent Cluster Hand Strength (OCHS) である．Round 4 では，すべての場のカードが開かれているため，各 OC に対する引き分けを考慮した勝率は HS である．従って，Round 4 の OCHS は，HS のヒストグラムで表されることになる．一方，Round 1，2 および 3 では，開かれていない場のカードが存在する．そこで，各 OC に対する引き分けを考慮した勝率は，開かれていないカードの全パターンを考慮した上で計算される．従って，Round 1，2 および 3 の OCHS は，HS の期待値のヒストグラムで表すことができる．Round 4 において手札が J♣ 9♡，場のカードが 4♢ 5♢ 9♢ 6♣ J♠ のときの OCHS を，図 3 に示す．

図 3 の横軸は OC 番号を表し，縦軸は OC 番号に対する HS を表す．図 3 において，各ヒストグラムの色は，図 2 の色に対応している．また，HSD では表現できなかった Round 4 の偶然手番を，OCHS では表現できていることがわかる．

3.5 CounterFactual Regret minimization (CFR)

CounterFactual Regret minimization (CFR)[13] は，Regret Matching を用いた regret 最小化アルゴリズムの，展開形ゲームへの応用である．CFR は，従来の線形計画法によるナッシュ均衡探索に比べて，より大きなゲーム木を扱うことができる．

CFR では，ゲーム木中の各人的手番について regret を計算する．ここで regret とは，ある行動を取らなかったときの後悔度合いを表す．ある人的手番 I において，I で取ることができるすべて行動について regret を計算し，regret が大きい行動が選ばれる確率が高くなるように戦略を更新する．

4. Opponent Cluster Hand Strength のクラスタ数の変更による抽象化の改良

4.1 Opponent Cluster Hand Strength の問題点

本研究では，偶然手番の特徴を表す指標として，Round 2 および 3 では Hand Strength Distribution (HSD) を用い，Round 4 では Opponent Cluster Hand Strength (OCHS) を用いる．しかし，HSD は各偶然手番を 50 本のヒストグラムで表すのに対し，OCHS は各偶然手番を 8 本のヒストグラムで表している．従って，OCHS は HSD により偶然手番を表す情報量が極端に少ない．つまり，Round 4 の偶然手番は，Round 2 および 3 の偶然手番と比較して，特徴を表すことができていない．OCHS と HSD の特徴量の表現力の違いは，k-means 法によるクラスタリング結果の違いになって現れる．すなわち，Round 4 の偶然手番のクラスタリング結果は，Round 2 および 3 のクラスタリング結果ほど良い結果を得ることができない点が，OCHS を偶然手番の特徴量として用いたときの問題点である．

4.2 Opponent Cluster Hand Strength のクラスタ数の変更

Hand Strength Distribution (HSD) の情報量より少ない理由は，ヒストグラムの本数の違いにある．HSD のヒストグラムの数は 50 本であるのに対し，OCHS のヒストグラムの数は 8 本である．そこで本研究では，OCHS のヒストグラムの本数を 10 本および 12 本に増やす．すなわち，手札の総数 1,326 通りを 10 個および 12 個の Opponent Cluster (OC) に分類し，それぞれのクラスタに対して引き分けを考慮した勝率，つまり Hand Strength (HS) の期待値をヒストグラムで表す．また，OCHS のヒストグラム数が抽象化の結果に影響するかを検証するため，OCHS のヒストグラムの本数を 6 本に設定した抽象化も行う．図 4，図 5，および図 6 に，それぞれ OC の数が 6 個，10 個，および 12 個のときの手札のクラスタリングの結果を示す．

図 4，図 5，および図 6 の Suited はマークが同じ手札の組み合わせを表し，Unsuited はマークが異なる手札の組み合わせを表す．

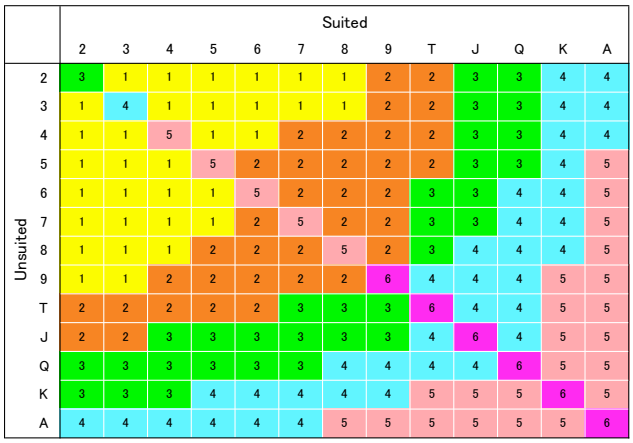

図 4 手札 169 種類を 6 個のクラスタに分類した結果

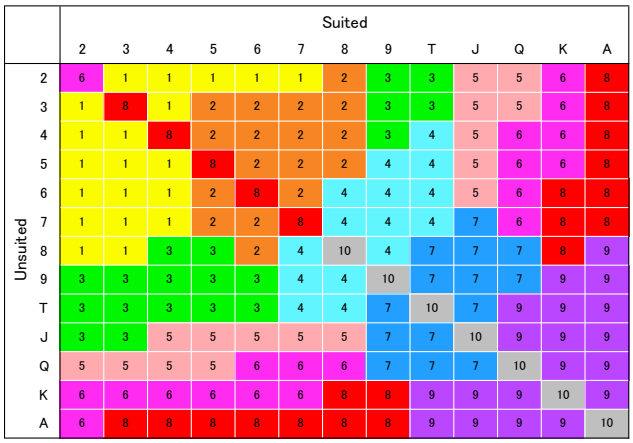

図 5 手札 169 種類を 10 個のクラスタに分類した結果

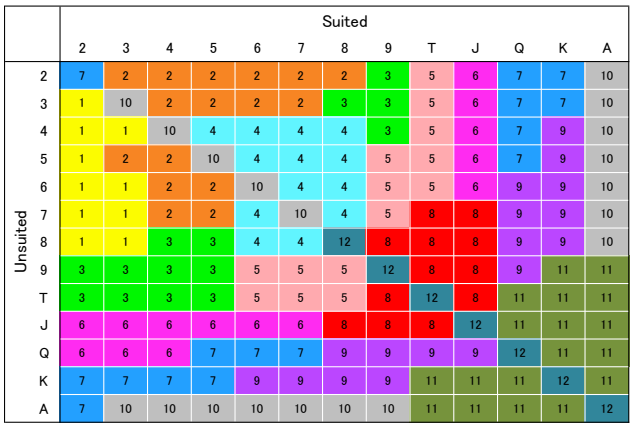

図 6 手札 169 種類を 12 個のクラスタに分類した結果

4.3 k-means 法の実行環境と計算時間

本研究では，各 Round の偶然手番をクラスタリングする際に k-means 法を用いた．Round 2 および 3 では，各データ点は Hand Strength Distribution (HSD) で表されており，ヒストグラムの数は 50 本であるため，各データ点は要素 50 の配列で表される．Round 4 では，各データ点は Opponent Cluster Hand Strength (OCHS) で表されている．ヒストグラムの数は，6 本，8 本，10 本，および 12 本のときがある．それぞれ OCHS のクラスタ数を 6，8，10，および 12 に設定したときの本数である．

表 3 各 Round の抽象化の計算時間

Round	計算時間（日）
2	約 0.3 日
3	約 3 日
4 (クラスタ数 6)	約 4.2 日
4 (クラスタ数 8)	約 5 日
4 (クラスタ数 10)	約 5.5 日
4 (クラスタ数 10)	約 6.1 日

表 4 各 Round における Raise に関する制限

Round	Raise 額	回数
1	10	3
2	10	4
3	20	4
4	20	4

次に各 Round の偶然手番の抽象化を行った実行可能について述べる．OS が Ubuntu 14.04，CPU が Intel(R) Core(TM) i7-5820K CPU@3.30GHz を 12 個，メモリが 64GB の計算機を用いて計算を行った．また，コードは C 言語で記述されており，OpenMP を用いて並列処理を行っている．各 Round ごとの抽象化の計算時間を表 3 に示す．

5. 評価実験と結果

5.1 実験目的

Opponent Cluster (OC) のクラスタ数を 6，8，10，および 12 に設定したときのポーカープレイングエージェントを，それぞれクラスタ 6 エージェント，クラスタ 8 エージェント，クラスタ 10 エージェント，およびクラスタ 12 エージェントとして作成する．本実験では，クラスタ 6 エージェント，クラスタ 8 エージェント，クラスタ 10 エージェント，およびクラスタ 12 エージェント間の対戦結果から，Opponent Cluster Hand Strength のクラスタ数を増やしたときの有効性を示すことを目標とする．

5.2 実験設定

【ルール】

本実験で用いるポーカーのルールは，2 player Limit Texas Hold'em とする．また，両プレイヤーがゲームの初めに強制的に賭けるチップであるスモールブラインドとビッグブランドは，それぞれ 5 と 10 とする．スモールブラインドを出すプレイヤーとビッグブランドを出すプレイヤーは，1 ゲーム毎に入れ替わる．1 ゲームとはカードが配られてから，勝敗が決まりチップの移動が行われるまでのことをいう．また，各 Round における Raise に関する制限を表 4 に示す．表 4 において，Raise 額は，Raise を行った際に上乗せする金額を表す．また，表 4 の回数は，ある Round において，Raise を行うことができる回数の上限を表す．Raise を行うことができる回数の上限は，1 プレイ

ヤーあたりではなく，両プレイヤ合わせての回数の上限である．

【ポーカープレイングエージェントの作成】

本実験では，偶然手番の抽象化に k-means 法を用いる．抽象化対象は，Round 2, 3, および 4 で，Round 1 は偶然手番の数が少ないため抽象化を行わない．偶然手番の特徴を表す指標として，Round 2, 3 では Hand Strength Distribution (HSD) を用い，Round 4 では Opponent Cluster Hand Strength (OCHS) を用いる．ここで，Opponent Cluster (OC) の数を 6 に設定した抽象化と，OC の数を 8 に設定した抽象化，OC の数を 10 に設定した抽象化，および OC の数を 12 に設定した抽象化の 4 つの抽象化を行う．4 つの抽象化されたゲーム木に対して，それぞれ CounterFactual Regret minimization (CFR) を適用して，行動戦略を計算する．CFR を用いた学習の回数は全ゲーム木とも 30 万回で，学習に用いたデータは全ゲーム木で共通である．

【ポーカープレイングエージェント対戦環境 ： ACPC コード】

本実験では，4 つのエージェント間の対戦を行うための環境として，Annual Computer Poker Competition で使用されているサーバーコード（ACPC コード）を用いる．Annual Computer Poker Competition (ACPC) は，Association for the Advancement of Artificial Intelligence (AAAI) が毎年開催しているポーカープレイングエージェントの競技会である [16].

【対戦回数】

Opponent Cluster Hand Strength (OCHS) のクラスタ数を変更した 4 つのエージェントのリーグ戦を行う．本実験では，対戦回数を次のように設定する．手札が両プレイヤーに配られてから，勝敗が決まりチップの移動が行われるまでの流れを 1 ゲームとして，3,000 ゲーム行う．さらに 3,000 ゲームを 1 セットとして 3,000 セット行う．また，3,000 セットを 1 試合として，9 試合行う．従って，合計 $3,000 \times 3,000 \times 9 = 81,000,000$ ゲームを行う．1 セット 3000 ゲーム中の各ゲームで配られるカード，および場のカードはランダムに生成する．ただし，すべての i セット目の第 k ゲームと，j セット目の第 k ゲームで配られるカード，および場のカードは同一である．つまり，本実験では，$3,000 \times 9 = 27,000$ 種類の配られるカード，および場のカードの組み合わせで対戦する．27,000 種類の配られるカード，および場のカードの組み合わせは，それぞれランダムで生成されるため，重複した組み合わせがある可能性がある．

Annual Computer Poker Competition では，2 人のポーカープレイングエージェントの対戦は 3000 回行われるため [16]，本実験でも 3,000 ゲームを 1 セットとして対戦を行う．

また，本実験では，3,000 ゲーム 1 セットを 3000 セット行う．クラスタ 6 エージェント，クラスタ 8 エージェント，クラスタ 10 エージェント，およびクラスタ 12 エージェントの戦略は行動戦略である．つまり，あるゲーム状態で選択する行動は，確率的に決定される．従って，1 セットの試行だけでは，選択される確率の低い行動が選択され，結果として負けてしまうことが起こりうる．そこで，すべての i セット目の第 k ゲームと，j セット目の第 k ゲームで配られるカード，および場のカードを同一にし，1 セットの内容を繰り返すことで，偶然による結果のばらつきを抑制する．

表 5 各エージェント間の対戦結果（mbb/g）

	C 6	C 8	C10	C12	平均
C 6		-9.66	-14.31	-29.80	-17.92
C 8	9.66		-17.87	-31.38	-13.2
C10	14.31	17.87		-10.20	7.33
C12	29.80	31.38	10.20		23.79

【実験結果】

作成した 4 つのエージェント間の対戦結果を，表 5 に示す．表 5 では，クラスタ 6 エージェント，クラスタ 8 エージェント，クラスタ 10 エージェント，およびクラスタ 12 エージェントをそれぞれ C6, C8, C10, および C12 と表している．また，表 5 中の対戦結果の単位は mbb/g である．mbb/g とは，milli-big-blinds/game の省略形で，全試合を終えた後の利得を 1/1000 ビッグブランドで表し，全試合数である 8100 万で割った値である．本実験では，ビッグブランドは 10 と設定している．例えば，C8 と C6 の結果を見ると，C8 が 9.66 mbb/g で勝利している．実際には，$9.66 \times 10 \times 1,000 \times 81,000,000 \approx 7,827,800$ だけ利益を得ている．

【考察】

表 5 より，OCHS のクラスタ数を大きく設定したエージェントは，OCHS のクラスタ数を小さく設定したエージェントに対して勝利していることが分かる．また，クラスタ数の差が大きいほど，対戦結果の差も大きくなる傾向が見られる．OCHS のクラスタ数を大きく設定することで，Round 4 における偶然手番の特徴量をより表すことができる．つまり，クラスタ数を大きく設定したクラスタリングは，クラスタ数を小さく設定したクラスタリングと比較して，良い抽象化が可能であると推察できる．表 5 で示したように，OCHS のクラスタ数を大きく設定したエージェントは，OCHS のクラスタ数を小さく設定したエージェントに対して勝利できることから，OCHS のクラスタ数を大きく設定することで良い抽象化を行うことが可能であることが示された．

クラスタ数の差が大きいほど，対戦の結果の差も大きくなることから，OCHS のクラスタ数を増やすほどより強いエージェントを作成できると推察される．しかし，クラスタ数を大きくすることで，k-means 法を用いたクラスタリ

ングの計算時間が増加するという問題点がある．k-means 法のオーダーは超多項式であるため，三角不等式を用いた計算の省略手法などを用いることや，より多くの CPU を用いて並列処理を行う必要がある．

6. おわりに

本研究では，Opponent Cluster Hand Strength (OCHS) の改良を行った．OCHS は，ゲーム木における偶然手番の特徴量を表す指標であり，k-means 法を用いた偶然手番の抽象化を行う際に用いる．OCHS が提案手法された論文では，Opponent Cluster (OC) の数は 8 であったが，本研究では OC の数を 6，10，および 12 に設定した．OC の数を増やすことで，OC の数が小さいときと比較して，偶然手番の特徴量をより表現できるため，結果として質の良い抽象化が期待できる．そこで，OC の数が 6，8，10，および 12 のときのポーカープレイングエージェントをそれぞれ作成し，実験を行った．実験の結果，OC の数を大きく設定したエージェントが，OC の数を小さく設定したエージェントより優れているという結果を得ることができた．

実験の結果から，OC の数を増やすことで，OC の数が小さいときと比較して，偶然手番の特徴量をより表現できるため，結果として質の良い抽象化が行えることが確認された．偶然手番の抽象化を行うためには，偶然手番の特徴を表現する指標が必要である．偶然手番の特徴を正確に表現することができる指標を用いれば，より良い抽象化を行うことができる．本研究から，偶然手番の特徴を正確に表現する手段として，偶然手番の特徴を表現する指標のヒストグラムの数を増やすことが有効であるという知見を得られた．

今後の課題として，多人数ゲームへの拡張が挙げられる．本研究で扱った 2 player Limit Texas Hold'em は，2 人ゲームである．ポーカーのみならず，ゲームの多くは多人数で行われることも多い．そこで，多人数ゲームへの拡張を行う必要がある．ゲームのプレイヤーが増えるに連れて，ゲーム木のサイズも大きくなる．従って，ゲーム木の抽象化が必要になる．多人数ゲームのゲーム木の抽象化手法としては，ゲームを少人数で行ったときの戦略を，多人数に拡張する手法などがある．

参考文献

[1] Hoda, S., Gilpin, A., Pena, J., and Sandholm, T.: *Smoothing techniques for computing Nash equilibria of sequential games*, Mathematics of Operations Research 35(2), 494-512. Subsumes our WINE-07 paper (2010).

[2] Gilpin, A. and Sandholm, T.: *Speeding Up Gradient-Based Algorithms for Sequential Games (Extended Abstract)*, Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2010).

[3] Sandholm, T.: *The State of Solving Large Incomplete-Information Games, and Application to Poker*, AI Magazine, special issue on Algorithmic Game Theory, Winter, 13-32 (2010).

[4] Sam Ganzfried and Tuomas Sandholm.: *Game theory-based opponent modeling in large imperfect-information games*, In The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '11, pp. 533-540, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[5] Bowling, M., Risk, A., Bard, N., Burch, N., Davidson, J., Hawkin, R., Holte, R., Johanson, M., Kan, M., et al: *A Demonstration of the Polaris Poker System*, Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2009).

[6] Billing, D., Burch, N., Davidson, A., Holte, R., Schaeffer, J., Schauenberg, T., and Szafron, D.: *Approximating Game-Theoretic Optimal Strategies for Full-scale Poker*, Proceedings of the 18th International Joint Conference on Artificial Intelligence in August (2003)

[7] Gilpin, A., Sandholm, T., and Sørensen, T.: *Potential-Aware Automated Abstraction of Sequential Games, and Holistic Equilibrium Analysis of Texas Hold'em Poker*, Proceedings of the National Conference on Artificial Intelligence (AAAI) (2007).

[8] Gilpin, A. and Sandholm, T.: *Lossless abstraction of imperfect information games*, Journal of the ACM, 54 (5), October. Early versions: Finding Equilibria in Large Sequential Games of Imperfect Information, EC-06, and CMU-CS-05-158, 2005 (2007).

[9] J. Hawkin, R. Holte, and D. Szafron.: *Automated action abstraction of imperfect information extensive-form games*, In National Conference on Artificial Intelligence (AAAI), 2011.

[10] Michael Johanson, Neil Burch, Richard Valenzano, and Michael Bowling.: *Evaluating State-Space Abstractions in Extensive-Form Game*, Proceedings of the Twelfth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pp. 271-278, 2013.

[11] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron.: *Approximating Game-Theoretic Optimal Strategies for Full-scale Poker*, Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), pp.661-668, 2003.

[12] Kevin Waugh.: *A Fast and Optimal Hand Isomorphism Algorithm*, In AAAI Workshop on Computer Poker and Imperfect Information, 2013.

[13] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione.: *Regret Minimization in Games with Incomplete Information*, In Advances in Neural Information Processing Systems 20 (NIPS), pp. 905-912, 2008.

[14] D. Billings, A. Davidson, J. Schaeffer, and D. Szafron.: *The challenge of poker*, Artificial Intelligence Vol. 134, No. 1, pp. 201-240, 2002.

[15] D. Billings, D. Papp, J. Schaeffer, and D. Szafron.: *Opponent modeling in poker*, Proceeding of The National Conference On Artificial Intelligence, pp. 493-499. JOHN WILEY & SONS LTD, 1998.

[16] *Annual Computer Poker Competition*, http://www.computerpokercompetition.org/

ⓒ 2015 Information Processing Society of Japan