E-022

不完全情報ゲームにおけるゲーム木探索アルゴリズム
Game-tree Search Algorithm for Incomplete Information Game

小田和 友仁 † Tomohito Otawa　　**上原 貴夫** † Takao Uehara

1　はじめに

本研究は不完全情報ゲームであるブリッジを対象としている．ブリッジは4人でテーブルを囲んでプレイするカードゲームである．対面のプレイヤ同士がパートナとなり，二組に分かれて勝敗を競う．そのため敵との競合はもちろん，味方との協調が非常に重要である [1].

現在のコンピュータブリッジの最善手決定法としては，モンテカルロシミュレーションにもとづくアルゴリズムが主流である [2]．このアルゴリズムは主観的な仮説のみを探索の対象とし，他者の推論を考慮しない．すなわち不完全情報ゲームの最善手決定法としては考慮が不十分である．

そこで我々は先にブリッジのプレイヤをエージェントとしてモデル化する方法を提案した [1][3][4]．このエージェントは他プレイヤのハンドを推論するとともにそのプレイヤから見た自分のハンドも推論し，それらの仮説にもとづいて行動を決定する．このエージェントに実装したabc アルゴリズムは従来の最善手決定アルゴリズムに比べ，探索すべき仮説の組み合わせが非常に多く，処理時間がかかる．そこで $\alpha\beta$ 法やトランスポジションテーブルなどの高速化アルゴリズムに関して拡張した方法を検討する．

2　従来の最善手決定アルゴリズム

見えていないカードの可能な分布（世界）を多数生成し，各世界でダブルダミーブリッジ（4人のハンドを見た上でのプレイ）の MinMax 値を求め，選ばれた候補の中から全体として良さそうな行動を選ぶ方法が何人かの研究者により提案された．Ginsberg の GIB もこの方法をもちいていている [2]．Ginsberg は行動の候補の集合を M としたとき，つぎのようにして一つの行動（つぎに出すカード）を決定している．

[Step1] それまでのビッドおよびプレイと矛盾しないようにカードをくばり，ディールの集合 D を作る

[Step2] 各ディール $d \in D$ ごとに，各行動 $m \in M$ を選んだらどのような結果になるかダブルダミーで評価してスコア $s(m,d)$ を計算する

[Step3] $\sum_d s(m,d)$ が最大となるような行動 m を選ぶ

3　エージェントモデル

著者は各プレイヤを知識と仮説推論機構をもつエージェントとしてモデル化した．ビッドに関する知識はビッドの決定とビッドしたプレイヤのハンドを推論する両方の目的で使われる [3]．プレイに関する知識は主にハンドの推論に使われる [4]．エージェントはそれまでのビッドおよびプレイを観察し，それと矛盾しないような仮説として，ディールの集合 D を作ることができる．また，4章のアルゴリズムで述べる他のプレイヤから見たディールの集合も容易に生成できる．

† 東京工科大学, Tokyo University of Technology

4　abc アルゴリズム

abc アルゴリズムとは他者の推論を考慮した最善手決定アルゴリズムである．プレイヤ a がとりうる行動の候補の集合を M としたとき，つぎのようにして一つの行動（つぎに出すカード）を決定する．

[Step1] それまでのビッドおよびプレイと矛盾しないようにカードをくばり，プレイヤ a から見たディールの集合 $D(a)$ を作る

[Step2] 各ディール $d \in D(a)$ ごとに，つぎの手順で各行動 $m \in M$ を選ぶとどのような結果になるか評価してスコア $s(m,d)$ を計算する

- **[Step2-1]** 各ディール $d \in D(a)$ ごとに，それまでのビッドおよび m を含むプレイと矛盾しないようにカードをくばり，他の2人のプレイヤ b，c の立場から見たディールの集合 $D(b,m,d)$，$D(c,m,d)$ を作る
- **[Step2-2]** 3つのディール $da \in D(a)$，$db \in D(b,m,d)$，$dc \in D(c,m,d)$ の組ごとに，行動 $m \in M$ を選んだ結果として得られるスコア $s(m,[da,db,dc])$ をダブルダミーで計算する
- **[Step2-3]** 各ディール $d \in D(a)$ について $[d,da,db]$ のすべて（n 個）に対する平均値 $\sum s(m,[d,da,db])/n$ を求め，これを $s(m,d)$ とする

[Step3] $\sum_d s(m,d)$ が最大となるような行動 m を選ぶ

Step2-2 のスコア計算には3組のディールを扱うため，MinMax 法を拡張したアルゴリズムをもちいる．MinMax 法の拡張について 4.1 で詳しく説明する．

4.1　MinMax 法の拡張

a　：初期局面 $p0$ におけるプレイヤ
b，c　：ダミーと a 以外のプレイヤ
wa　：a から見たワールド（ディール）
wb　：b から見たワールド（ディール）
wc　：c から見たワールド（ディール）
p　：局面
$s(p)$　：p のつぎの局面
$t(p)$　：p のタイプ（max,min,末端）
　ただし $t(p0) = max$ とする
$u(p)$　：p におけるプレイヤ（a，b，c のいずれか）
$v(p) = [v(p,wa), v(p,wb), v(p,wc)]$
　：p における wa,wb,wc の静的評価
　ただし，定義されない場合は $*$ で表す
$max(*,x) = x$
$min(*,x) = x$
$abc(p) = [a(p), b(p), c(p)]$
　：p におけるプレイヤ a，b，c から見た評価値

以上の定義において，つぎのようなルールをもちいて $abc(p) = [a(p), b(p), c(p)]$ を計算する．

if $t(p) =$ 末端 **then** $abc(p) = v(p)$

if $t(p) = max$ **then**

 if $u(p) = a$ **then**

  $a(p) = max_{p' \in s(p)} a(p')$
  $b(p) = max_{p' \in s(p)} b(p')$
  $c(p) = max_{p' \in s(p)} c(p')$

 if $u(p) = b$ **then**

  $b(p) = max_{p' \in s(p)} b(p')$
  このとき最大値を与える p' の集合を X とする
  $a(p) = max_{p' \in X} a(p')$
  $c(p) = max_{p' \in s(p)} c(p')$

 if $u(p) = c$ **then**

  $c(p) = max_{p' \in s(p)} c(p')$
  このとき最大値を与える p' の集合を X とする
  $a(p) = max_{p' \in X} a(p')$
  $b(p) = max_{p' \in s(p)} b(p')$

if $t(p) = min$ **then**

 if $u(p) = a$ **then**

  $a(p) = min_{p' \in s(p)} a(p')$
  $b(p) = min_{p' \in s(p)} b(p')$
  $c(p) = min_{p' \in s(p)} c(p')$

 if $u(p) = b$ **then**

  $b(p) = min_{p' \in s(p)} b(p')$
  このとき最小値を与える p' の集合を X とする
  $a(p) = min_{p' \in X} a(p')$
  $c(p) = min_{p' \in s(p)} c(p')$

 if $u(p) = c$ **then**

  $c(p) = min_{p' \in s(p)} c(p')$
  このとき最大値を与える p' の集合を X とする
  $a(p) = min_{p' \in X} a(p')$
  $b(p) = min_{p' \in s(p)} b(p')$

図 1 に abc アルゴリズムにおけるゲーム木の例を示す．West がプレイヤ a にあたり，South，East がプレイヤ b，c にあたる．ダミーである North の局面ではパートナの South が代行して手を決定するため，中断における評価値の最小値を選択する．

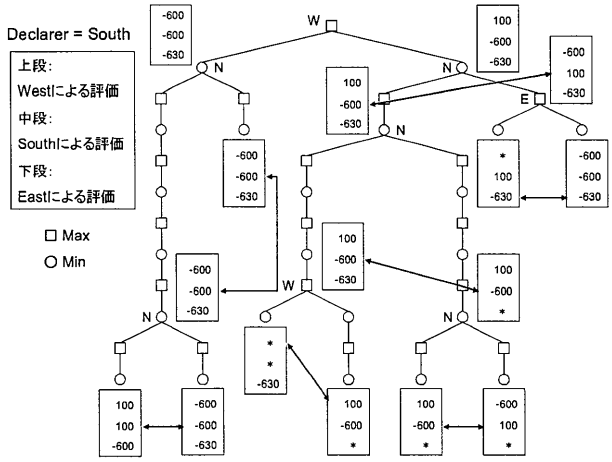

図 1　abc アルゴリズムのゲーム木

4.2　$\alpha\beta$ 法の拡張

他者の推論を考慮したアルゴリズムは 3 者の評価値にもとづく MinMax 探索であるため，$\alpha\beta$ 法の適用が可能である．しかし，ルートプレイヤ以外の選択ではルートプレイヤはその選択に従属するため，ルートプレイヤの評価値でカットオフするのは難しい．また，一者の判断でカットオフしてしまうと，その他のプレイヤの判断で破綻してしまう．そこで，ルートプレイヤ以外の 2 者がカットできると判断した枝のみカットの対象とする方法を検討している．

4.3　トランスポジションテーブルの拡張

トランスポジションテーブルにゲーム木の各局面における評価値を記憶し，同じ局面でのゲーム木探索を省くことにより abc アルゴリズムを高速化する手法の適用を検討する．

1 つのディールを対象とする従来のアルゴリズムではゲーム木の局面を表現するために，4 人の残りハンドとつぎのプレイヤを表すハッシュ値をもちいる．ハッシュテーブルに記録するために，局面のハッシュ値をつぎのようにして作る．52 枚のカードの内の 1 枚を各プレイヤが持っていることを表わす 208 個の乱数と，つぎのプレイヤを表す 4 個の乱数を用意する．そして残りのハンドに対応するすべての乱数と，つぎのプレイヤに対応する乱数の排他的論理和をとりハッシュ値とする．この方法はインクリメンタルにハッシュ値を作成できるので遷移していくゲーム木の局面を表すのに都合がよい．

3 種類のディールを対象とする abc アルゴリズムでは同じ方針でおこなうと煩雑になるのでつぎの方法をとる．52 枚のカードと 4 人のプレイヤを表す合計 56 個の乱数を用意する．それまでにプレイされたカードに対応するすべての乱数とつぎのプレイヤに対応する乱数の排他的論理和をとりハッシュ値とする．ブリッジの場合，1 つのディールを対象とする場合にもこの方法を適用できる．盤を使うチェスや将棋，碁などのゲームには使えないが，トランプゲームには広く適用できる．

同じとみなせる局面をまとめて保存するパーティションサーチにおいては，区別しないカードの集合について，プレイされたカードの総数ではなく誰がプレイしたかという情報も必要になるが，容易に拡張できる．

5　おわりに

abc アルゴリズムの高速化手法として $\alpha\beta$ 法とトランスポジションテーブルを拡張する方法を述べた．発表ではこれらの評価についても述べる．

参考文献

[1] 小林紀之, 小田和友仁, 上原貴夫: コンピュータブリッジによるカウントシグナル, 情報処理学会論文誌, Vol.45, No.6, pp.1704-1714 (2004)

[2] Ginsberg M. L.: GIB: Steps toward an expert-level bridge-playing program, IJCAI-99 (1999)

[3] 安藤剛寿, 小林紀之, 上原貴夫: コンピュータブリッジのビッドにおける協調と競合, 電子情報通信学会論文誌, Vol. J83-D-I, No.7, pp.759-769 (2000)

[4] 小林紀之, 上原貴夫: コンピュータブリッジによるディセプティブプレイ, 情報処理学会論文誌, Vol.43, No.10, pp.3056-3063 (2002)