

コンピュータブリッジにおける他者のモデルとゲーム木探索の高速化

熊谷 幸紀[†] 伊東 一樹[†] 松川 格[†] 小田和 友仁[†] 上原 貴夫[†]

1 はじめに

ゲーム研究の分野では局面の情報が完全に公開されているゲーム(将棋など)を完全情報ゲーム, そうでないゲーム(麻雀など)を不完全情報ゲームと呼ぶ. コンラクトブリッジ(以下ブリッジ)は4人でおこなうカードゲームであり, 不完全情報ゲームに分類される. このような不完全情報ゲームで最善手を求めるには, 従来モンテカルロシミュレーションに基づくアルゴリズムをもちいてきた [1]. 局面の観察によって得られる情報をもとに, それに矛盾しない手札の仮説を立てて完全情報ゲームとして解く方法である. しかしこの方法では, ブリッジにおいては致命的な欠点がある.

我々はこの問題を解決する方法として ABC アルゴリズムを提案し, 従来のコンピュータブリッジでは難しかった上級者のプレイテクニックのいくつか(敵を騙すプレイやパートナーのミスを防ぐなど)が可能となった. ABC アルゴリズムは処理するデータ量が膨大であるため計算時間が長くなる. よって高速化が必要である.

2 モンテカルロシミュレーション

不完全情報ゲームの最善手を求める方法としてモンテカルロシミュレーションに基づくアルゴリズムの動作説明をする.

2.1 局面観察による情報取得

現時点での局面を観察することから得られる部分的な情報から公開されていない情報を推測する. 自分とダミーの手札, 現在場に出されているカードやすでに過去のトリックで使われたカードは公開されている確実な情報として読み取れる. これらのカードは相手の現在の手札には絶対に含まれないものであり, 重要な情報である.

2.2 仮説発生と探索

自分, ダミーを除いた残り2人の手札について先ほど読み取った確実な情報と矛盾しないような仮説をランダムに多数発生させる. それぞれの仮説内では相手の手札がすべて見えた状態なので, 完全情報ゲームとみなし MinMax 法を使いゲーム木探索をして最善手を求める.

2.3 総合評価

候補手ごとにすべての仮説の探索結果で得られた評価値を足し合わせ合計値を求める. その結果もっとも評価値の高い候補手が最善手として選ばれる.

2.4 問題点

完全情報ゲームとして最善手を探索するので, 敵が自分の正確な手札を知っているものとして今後の展開を予測してしまう. よって相手から自分の手札が見えないことを利用し相手に誤った推論をさせるプレイができない.

3 ABC アルゴリズム

ABC アルゴリズムは, 他者の仮説も考慮して探索をする本研究独自のアルゴリズムである [2]. 図 1 に概念図を示す. この節ではアルゴリズムの動作を説明する.

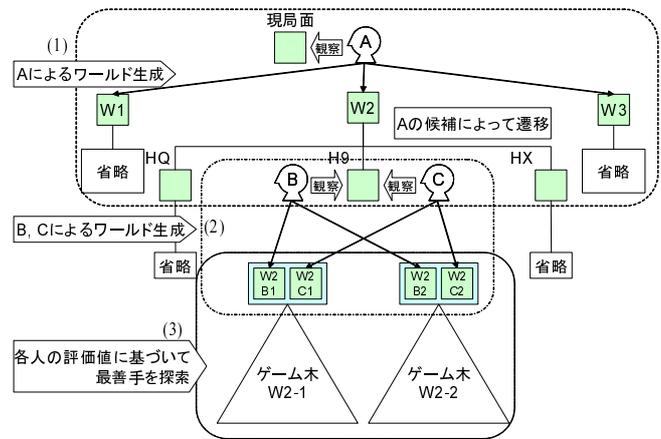


図 1 ABC アルゴリズム

3.1 A の仮説発生

従来のアルゴリズムと同じように局面を観察して得られた情報から複数の仮説を立て, 候補手に応じて遷移先の局面情報を生成する. 図 1 の (1) の部分が該当する.

3.2 B, C の仮説発生

A がカードを出した後の局面を B や C の視点で観察し, 得られた情報に矛盾しないような仮説を複数立てる. B には B の手札とダミーの手札が見え, C には C の手札とダミーの手札が見える. 図 1 の (2) の部分が該当する.

3.3 3つの仮説によるゲーム木探索

図 1 の (3) の部分では3つの仮説によるゲーム木探索をする. 本研究ではゲーム木探索を Probing 方式で実装した. Probing 方式は自分のノードでは MinMax 法同様に自分の出せる手を列挙して枝を作り自分以外のノードでは相手やパートナーの仮説のみを使用しゲーム木探索を新たに始める(これを先行探索という). そこで敵やパートナーが選ぶであろう手を $\alpha\beta$ 法をもちいて計算し, 以降はそれに対応するノードの評価のみを保持する. 従来, 我々が実装した One-Path 方式に比べ高速であることを実験によって確かめた.

4 高速化手法

本研究で実装した以下の高速化手法について述べる.

4.1 パーティションサーチ

パーティションサーチはトランスポジションテーブルを拡張した手法である. トランスポジションテーブルとは, MiniMax 法における高速化手法のひとつである. ゲーム木探索において探索中に同一局面が現れる場合がある. 局面と局面に対する評価値を組にしてテーブルに保存することにより, 同一局面の探索を回避することができる. パーティションサーチはこれに加え, 局面同士の細かい違いを無視することで一般化する. これによりある程度の高速化が期待できるが, 一般化の方法の違いで探索結果に影響がでる可能性がある. 本研究では次の4つの一般化の方法を実装した.

High Speed Game Tree Search for Computer Bridge Using Other Players Model

[†] Takanori Kumagai, Kazuki Ito, Itaru Matsukawa, Tomohito Otawa and Takao Uehara, Tokyo University of Technology

- スモールカードを枚数で管理
- スモールカードを枚数と代表カード(最小)で管理
- スモールカードの境界をトリックの数によって変化
- スモールカードの境目を境目以上のカードの枚数によって変化

4.2 他プレイヤーの探索深度の変更

Probing方式では他プレイヤーの局面で先行探索をおこなう。探索は深度が深くなるにつれ探索時間が増えていく。当然先行探索についてもそれが当てはまる。また先行探索は候補手ごとにおこなうため候補手の数が多いと先行探索の回数も増える。そのため特に深度が深く、候補手の数が多い場合探索時間が膨大になってしまう。この問題を解決するために相手の探索深度を1トリック分に固定する。これは敵や味方プレイヤーが楽観的にプレイすることを意味する。次章で述べるリスク分析には好都合である。これによりこれまで時間のかかっていた先行探索の時間が減り、高速化を図ることができる。

5 実時間リスク分析

敵の誤った推論に期待して選択した手には、敵が誤らなかった場合のリスクが伴う。このリスクを実時間で計算し、避けようとするのが図2の提案である。我々は並列実行の実装にMPIの仕様に準拠した実装ライブラリであるLAM(Local Area Multicomputer)[3]をもちいた。並列実行する部分では、2つのアルゴリズムをそれぞれマスタースレイブ方式を使い並列実行する。ABCアルゴリズムで3トリック分探索する時間と従来のアルゴリズムで6~7トリック分探索する時間が同程度であるとわかっているので同時程度に終わるよう探索深度を設定しておく。そしてその結果ABCアルゴリズムを使った場合は利益とリスクを評価値により比較し、利益が高い場合にはABCアルゴリズムを、リスクが高い場合は従来のアルゴリズムを選択する。

5.1 利益とリスク

ABCアルゴリズムは自分に有利な展開を期待して多く見積もる楽観的なアルゴリズムであり、従来のアルゴリズムは常に最悪の展開を想定する消極的なアルゴリズムであるといえる。これを踏まえABCアルゴリズムで選ばれた手を選んだ場合、予想した理想の展開になり従来のアルゴリズムでの予想よりも高い得点が得られるならば、その差分が利益と換算できる。逆にABCアルゴリズムで選ばれた手を選んだが、従来のアルゴリズムで予想した最悪の展開になってしまい従来のアルゴリズムの

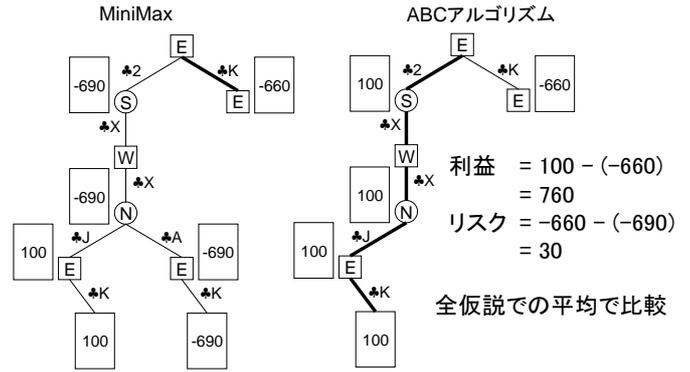


図3 利益とリスクの算出例

予想よりも低い得点となるならば、この差分がリスクと換算できる。利益とリスクは以下の式で算出される。

$$\text{利益} = \text{ABC 最良値} - \text{MiniMax 最良手の ABC での評価値}$$

$$\text{リスク} = \text{MiniMax 最良値}$$

$$- \text{ABC 最良手の MiniMax での評価値}$$

図3はある局面について従来のアルゴリズムとABCアルゴリズムにより1つずつ発生させたゲーム木である。上記の式のABC最良値はABCアルゴリズムで♣2を選んだ場合の評価値100にあたり、MiniMax値は従来のアルゴリズムで♣Kを選んだ場合の-660にあたる。ABC最良手のMiniMaxでの評価値は、ABCアルゴリズムの最良手♣2の従来のアルゴリズムでの評価値-690にあたる。よって利益とリスクは以下ようになる。これは仮説を1つだけ発生させた例である。本来は仮説を複数発生させ、得られる評価値の平均から算出する。

$$\text{利益} = 100 - (-660) = 760$$

$$\text{リスク} = (-660) - (-690) = 30$$

6 おわりに

ABCアルゴリズムを紹介し、その高速化手法と実時間のリスク分析について述べた。ABCアルゴリズムは不完全情報ゲームに特有な騙すテクニックなど様々なプレイテクニックを駆使する上で有効なアルゴリズムである。高速化によってより探索時間が短縮され実用に耐えうる時間で上級者のテクニックが使えるようになった。発表では実験結果として各高速化手法のグラフを示す。

参考文献

- [1] Ginsberg M.L.: 「GIB:Steps toward an expert-level bridge-playing program」, IJCAI-99 (1999)
- [2] Otawa, T. and Uehara, T.: "Model of bridge players who make mistakes", Game Informatics Special Interest Group, IPSJ (2005)
- [3] Trustees of Indiana University, "LAM/MPI Parallel Computing", <http://www.lammpi.org>

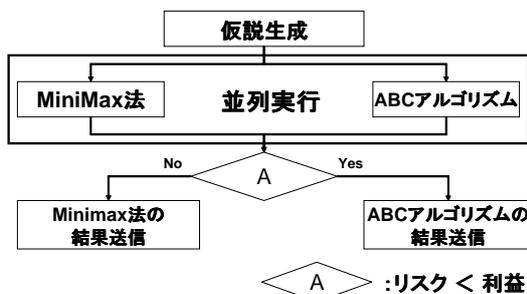


図2 実時間リスク分析