7K-2

# コンピュータブリッジにおける他者のモデルを用いた並列探索

小田和 友仁 †　　村上 隆志 †　　上原 貴夫 †
東京工科大学 †

## 1 はじめに

コントラクトブリッジ（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．我々はコンピュータブリッジの着手選択法として Ginsberg のアルゴリズム [1] を採用した．しかし Ginsberg のアルゴリズムはすべての手札が公開されていると想定して探索をおこなうため，他者が自分の正確な手札を把握しベストプレイすると過信し悲観的に振舞う傾向にある．

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 [2][3]．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 Ginsberg の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．そこで並列処理により高速化を図る．

## 2 Ginsberg のアルゴリズム

見えていないカードの可能な分布（ディール）を多数生成し，各ディールでダブルダミー（4 人の手札を見た状態でプレイ）の MinMax 値を求め，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る [1]．Ginsberg は出しうるカード候補の集合を $M$ としたとき，つぎのようにして最善手を決定している．

**[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 $m \in M$ を選択した場合どのような結果になるかをダブルダミーで評価し，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 他者のモデル

著者はブリッジ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 [2]．プレイに関する知識は主にハンドの推論に使われる [3]．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．

## 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．図 1 に ABC アルゴリズムの流れを示す．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．

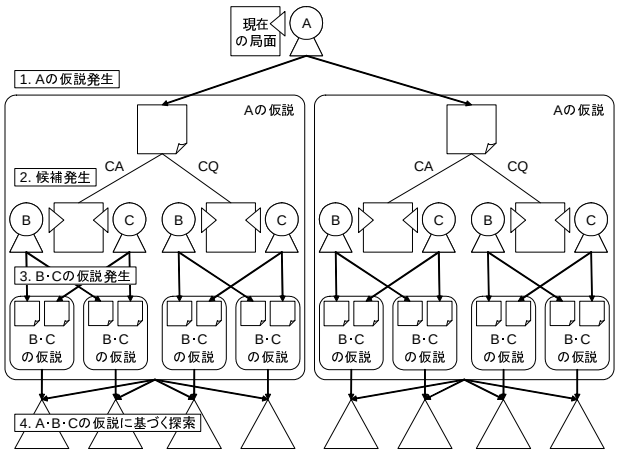


図 1　ABC アルゴリズムの流れ

**[Step1]** プレイヤ $A$ の視点で，それまでのビッドおよびプレイと矛盾しないディールの集合 $Da$ を作る

**[Step2]** 各ディール $d \in Da$ ごとに，各行動 $m \in M$ を選んだ場合のスコア $s(m,da)$ を [Step2-1]〜[Step2-3] の手順で計算する

**[Step2-1]** 各ディール $d \in Da$ ごとに他のプレイヤ $B$, $C$ の視点で，それまでのビッドおよび $m$ を含むプレイと矛盾しないディールの集合 $D(b,m,d)$，$D(c,m,d)$ を作る

**[Step2-2]** 3 つのディール $d \in Da$, $db \in D(b,m,d)$，$dc \in D(c,m,d)$ の組合わせで，行動 $m \in M$ を選んだ結果として得られるスコア $s(m,[d,db,dc])$ をダブルダミーで計算する

**[Step2-3]** 各ディール $d \in Da$ について $[d,db,dc]$ のすべて（$n$ 個）に対する平均値 $\sum s(m,[d,db,dc])/n$ 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] のスコア計算では 3 組のディールを扱うために MinMax 法を拡張したゲーム木探索をおこなう [4]．各プレイヤが自分の仮説に基づきプレイすると仮定し探索するため，枝刈りによる高速化が難しく処理時間がかかる．

探索すべきゲーム木の数 $N$ は以下の式で求められる．プレイヤ $A$ の仮説発生数を $N_A$，プレイヤ $B$,$C$ の仮説発生数を $N_{BC}$，候補数を $N_M$ とする．

$$N = N_A * N_M * N_{BC}$$

候補数 $N_M$ は手札からの算出値であるが，仮説の発生数である $N_A$ と $N_{BC}$ は任意に決定することができる．モンテカルロ法に基づくとシミュレーションの回数を十分に取らなければならないため，$N_A$ と $N_{BC}$ を大きく設定するのが望ましいが，比例して多くのゲーム木探索を必要とするため，時間とのトレードオフを考慮しなければならない．

---

Parallel Game Tree Search for Computer Bridge Using Other Players Model

† Tomohito Otawa, Takashi Murakami, Takao Uehara, Tokyo University of Technology

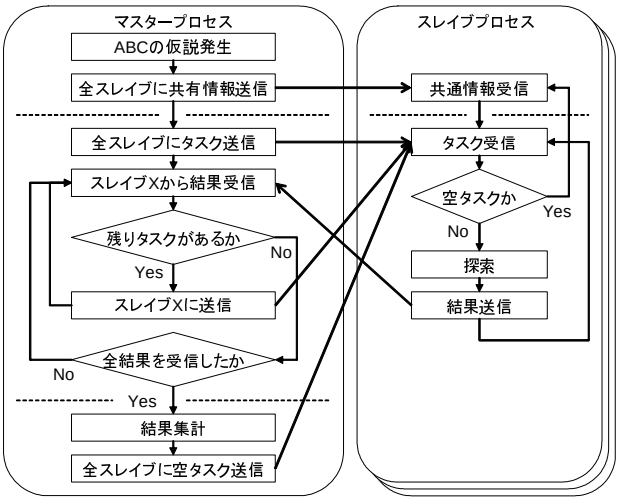


図 2　フローチャート

## 5　ABC アルゴリズムの並列処理

4 章で述べたアルゴリズムの手順における [Step2-2] のスコア計算は 3 者の仮説をもちいるが MinMax 法による探索であり，一つの完全情報ゲームとして探索することを意味する．すなわち探索するための情報は独立しており，タスク並列性が高い．本研究では 3 者の仮説に基づく探索をタスクの単位とし，並列処理を実装した．

探索部はタスクプールによる動的スケジューリングをマスター／スレイブモデルとして実装する．図 2 に最善手決定の流れを表したフローチャートを示す．マスターはタスクの生成，管理と処理結果の集計を担当し，スレイブはゲーム木探索に専念する．

並列処理を実装するためにはクラスタの管理や，クラスタ間のメッセージ通信が必要である．これらを実装するために MPI ライブラリである LAM[5] をもちいる．

### 5.1　MPI(Message Passing Interface)

MPI は並列プログラミング用の API 仕様である．仕様に準拠した実装ライブラリに LAM がある [5]．LAM は UNIX ベースの OS のみをサポートし，Windows 系列の OS では動作しない．UNIX のデーモンとして常駐し，これを介して通信するため非常に高速な通信が可能である．

### 5.2　SPMD(Single Program Multi Data stream)

MPI の提供するプログラミングモデルは各ノードで異なるプログラムを実行する MPMD(Multi Program Multi Data stream) ではなく SPMD である．すなわち全ノードが一つのプログラムを読み込み実行する．ただし各ノードに割り当てられた識別 ID により処理を分岐することは可能である．マスター／スレイブモデルを実現するために，この識別 ID を利用した．

### 5.3　マスタープロセス

マスターは主に 4 章の [Step2-2] 以外を担当する．図 2 をもちいてマスタープロセスの流れを説明する．

はじめに 4 章の [Step1]〜[Step2-1] に従いプレイヤ $A$ と $B$，$C$ の仮説を発生する．さらに全スレイブに対し共有情報を送信する．共有情報とはすべてのタスク処理において共通な情報である．コントラクトやディクレアラ，現在のプレイヤ，プレイの履歴が含まれる．

続いてタスク分配部に移る．タスクとして送る情報は $A$ と $B$，$C$ の仮説の組 $[d, db, dc]$ と $A$ の候補のうち 1 枚のカード $m$ とする．マスターは全スレイブに対して 1 つずつタスクを送信したのちにループに移る．ループでは処理が終了し結果を送信してきたスレイブに対して，タスクが残っていれば次のタスクを渡す．すべてのタスクを送り出し，その結果が帰ってくるまでこれを繰り返す．

スレイブから受け取る結果は候補のカード $m$ と，そのスコア $s(m, [d, db, dc])$ とする．マスターは結果集計用のテーブルを持つ．スコアの総計を格納するためのテーブルであり，要素はプレイヤ $A$ の候補手に対応する．マスターは受け取った候補のカード情報に対応する要素にスコアを足し合わせていく．

全タスクの結果が揃ったところで，4 章の [Step3] に従い集計に移る．結果集計用のテーブルに格納されたスコアの総和を比較し，最大となる要素に対応するカードを最善手として決定する．

最後に次の局面での探索に備えて全スレイブを共有情報受け取り待ちに移行させるため，全スレイブに対し空のタスクデータを送信する．

### 5.4　スレイブプロセス

スレイブは主に 4 章の [Step2-2] を担当する．マスタから共有情報を受け取ると，タスク処理のループに移行する．マスターからタスクデータを受信する．タスクデータが空の情報であった場合，次の局面での探索に備えるため共通情報受信待ちへと移行する．タスクデータが空でない場合，含まれる 3 者のディールからプレイの経過を参照してすでに出されたカードを取り除き，現在の局面まで更新する．さらにタスクに付随する候補のカード $m$ をプレイした場合のスコア $s(m, [d, db, dc])$ を完全情報ゲームとして探索する．結果として得られたスコア $s(m, [d, db, dc])$ と候補のカード $m$ をマスターに送信し，次のタスク受信待ちに移行する．

## 6　おわりに

他者の推論を考慮するアルゴリズムである ABC アルゴリズムを紹介し，その並列化手法を述べた．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．並列処理が実現すれば実用に耐えうる時間内で様々なテクニックが再現可能となり，コンピュータブリッジの性能向上につながるであろう．発表では実験結果として台数効果を示し，並列処理の有効性を示す．

## 参考文献

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

[2] 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)

[3] 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)

[4] 小田和友仁, 上原貴夫, ”コンピュータブリッジにおける新しいアルゴリズムと高速化”, 第 9 回ゲーム・プログラミング ワークショップ 2004, pp.64-70 (2004)

[5] Trustees of Indiana University, ”LAM/MPI Parallel Computing”, `http://www.lam-mpi.org`