

コンピュータブリッジにおける他者のモデルを考慮した ゲーム木探索の提案

小田和 友仁[†] 上原貴夫[†]

これまでのコンピュータブリッジのアルゴリズムでは、敵を惑わすプレイやパートナーを迷わせないプレイなど上級者のプレイテクニックが実現できなかった。これはコントラクトブリッジが不完全情報ゲームであることに由来しており、従来のミニマックス探索ベースの手法では適切に表現できないためである。本論文は完全情報ゲームにおける OM サーチをコントラクトブリッジに拡張した手法を提案する。このアルゴリズムを実装・実験した結果、敵を惑わすプレイやパートナーを迷わせないプレイが可能となることが確かめられた。

Game-tree Search Based on Player Model in Computer Bridge

TOMOHITO OTAWA[†] and TAKAO UEHARA[†]

Any algorithm used by existing Computer Bridges cannot find advanced play techniques such as deceiving the opponent or preventing the partner's mistakes. Since the Contract Bridge is an imperfect information game, those advanced plays are not represented in the game-tree based on the Minimax search. This paper proposes an extension of the OM search, which is developed for the perfect information games, to the Contract Bridge. We implemented it and applied the method successfully to examples deceiving the opponent or preventing the partner's mistakes.

1. はじめに

不完全情報ゲームであるブリッジでは、上級者は敵を惑わすプレイや味方を迷わせないプレイを行う。しかしミニマックス戦略をもちいる現在のコンピュータブリッジでは、これを實現することができない。本研究の動機は、他のプレイヤーの立場からの推論を考慮することにより、これらの上級者のプレイテクニックを實現することである。

現在でも多くのゲームプログラムはフォン・ノイマンが1928年に発表したミニマックス戦略に基づくゲーム木探索アルゴリズムを使っている。すなわち対戦する2人のプレイヤーは、ゲームの局面について同じ評価をするものと仮定している。これに対して Iida らは、同じ局面でも敵が自分と異なる評価をする可能性があると仮定して、2人の評価値に基づくゲーム木探索(オボネントモデルサーチ, 以下 OM サーチ)について研究している^{1),2)}。この方法によれば、敵の先読みの深さが自分より浅い場合、敵の誤りを利用してよ

り良い結果を得ることができる。Iida らの研究は2人によってプレイされる完全情報ゲームを念頭においている。

本論文では4人でプレイされる不完全情報ゲームであるコントラクトブリッジにおいて同じ局面に対する各プレイヤーの評価が異なると仮定し、(ダミー1人を除く)敵、味方、合計3人の評価値に基づく新しいゲーム木探索法を提案する。これは Iida らの OM サーチを不完全情報ゲームに拡張したもので、本研究では不完全情報ゲーム特有の背景に基づく有効な利用法を明らかにした。以下では背景を説明した後、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(敵をだますディセプティブプレイ、味方のミスを防ぐパートナーシップなど)が可能となることを示す。また約120題のディフェンスの例題で実験し、その有効性と問題点について考察する。

なお Sturtevant は多人数ゲームの探索アルゴリズムについて研究しており、完全情報ゲームだけでなく、Spades, Hearts などブリッジに似たゲームへの応用も試みている³⁾。しかしモンテカルロ法をもちい、各サンプルでは全員のカードが見えていると仮定して探索するので、敵を惑わすプレイや味方を迷わせないブ

[†] 東京工科大学大学院工学研究科システム電子工学専攻
Tokyo University of Technology

レイを発見することは不可能である。

また Billings らは不完全情報ゲームとしてよく知られたポーカーの研究で成果をあげている⁴⁾。ポーカーではブラフなど敵を惑わす戦略がブリッジより頻繁に使われる。しかしゲームの性格の相異から、コンピュータポーカーの手法をブリッジに流用することは難しい。

2. オポネントモデルサーチ

完全情報ゲームにおいても戦略や技能の違いによりゲームのある局面に対する評価がプレイヤーにより異なる可能性がある。Iida らはゲーム木探索における *max* プレイヤと *min* プレイヤの評価関数（それぞれ *ma*, *mi* とする）を次のように定義した¹⁾。ここで *i*, *j* は対象の局面に直接接続している子ノードの範囲を表す。

[*ma* の定義]

- *P* が *max* ノードなら
 $ma(P) = \max_i ma(P_i)$
- *P* が *min* ノードなら
 $ma(P) = ma(P_j)$
ただし, *j* は次のようにして決まる。
 $mi(P_j) = \min_i mi(P_i)$
- *P* がターミナルノードなら
 $ma(P) = EV_{ma}(P)$...静的評価

[*mi* の定義]

- *P* が *max* ノードなら
 $mi(P) = \max_i mi(P_i)$
- *P* が *min* ノードなら
 $mi(P) = \min_i mi(P_i)$
- *P* がターミナルノードなら
 $mi(P) = EV_{mi}(P)$...静的評価

図1は、Iida らの論文¹⁾より引用したOMサーチと通常のミニマックス戦略との比較例である。ノード中の数はミニマックス戦略による評価値である。ノード脇にある2つの数値はOMサーチによる評価値であり、上段は *max* プレイヤの、下段は *min* プレイヤの評価値である。ルートでの評価値が2つの方法で異なることが分かる。OMサーチでの *max* プレイヤは、敵がノードGで過大な評価をして木の右側で重大なミスをするのを期待し、(ミニマックス戦略では最善手ではない) ルートから右側のノードへの移動に対応する手を選択する。それによりルートでの評価値が1だけ増える。一般的にOMサーチによって得られる値はミニマックス戦略の場合より小さくなることはないことが証明されている。

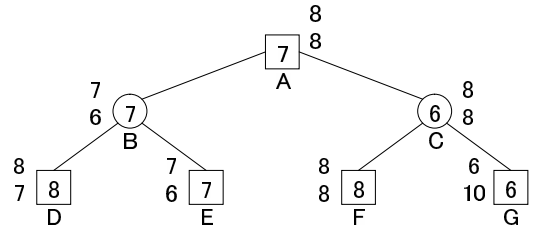


図1 ミニマックス戦略と OM サーチの比較

Fig.1 The OM-search and the minimax strategy compared.

3. 他者の推論を考慮した探索

3.1 ブリッジにおけるゲーム木探索

不完全情報ゲームであるコントラクトブリッジにおいて、次のようにしてゲーム木探索を行う方法が Ginsberg により採用された⁵⁾。見えていない他プレイヤーのカードがどのように配られているかを(ビッドやプレイから)推論し、多数の仮説を生成する。1つの仮説が成り立つ世界を想定し、候補手ごとにダブルダミーブリッジ(4人のハンドを見たうえでプレイ)のミニマックス値を求める。多数の仮説について評価値を計算した後、各候補手の中から全体として良さそうな候補(各世界での評価から求めた期待値が最大の候補)を選ぶ。この方法では敵もパートナーも各世界でミニマックス戦略に従って完全情報ゲームをプレイしていることになる。

3.2 上級者のプレイテクニック

上級者は自分の出すカードによって敵あるいはパートナーがどのように推論するかを考慮する機会が多い。たとえばパートナーがミスしないようなプレイを意識する。またベストディフェンスをされたら負ける場合でも、敵がミスしやすいプレイをしてチャンスを待つ。

3.1節の方法では、自分のハンドが他人に見えていると仮定するので、上記のようなプレイは原理的に不可能である。

3.3 3つの評価値をもちいた探索

我々は上級者のプレイを模擬するため、次に出すカードを見てダミーを除く3人のプレイヤーが推論する仮説の3つ組を生成することとした。各プレイヤーはこの異なる仮説に基づいて局面を評価するものとし、ゲーム木と同じ局面について3つの評価値を対応させた。

自分(ゲーム木のルートに対応するプレイヤー)を *a* とし、ダミーを除く他の2人のプレイヤーを *b*, *c* とする。我々は(*a*を *max* プレイヤと仮定し) *a* の評価関数 *ma* を次のように定義した。

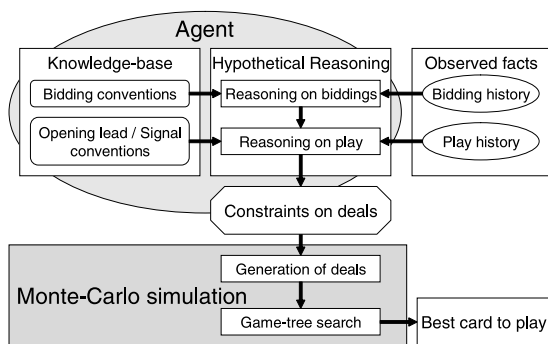


図 2 プレイヤーのモデル
Fig. 2 Model of player.

[*ma* の定義]

- P が a のプレイする max ノードなら
 $ma(P) = \max_i ma(P_i)$
- P が b のプレイするノードなら
 $ma(P) = ma(P_j)$
ただし, j は次のようにして決まる.
 b が a の敵である場合 $mb(P_j) = \min_i mb(P_i)$
 b が a の味方である場合 $mb(P_j) = \max_i mb(P_i)$
- P が c のプレイするノードなら
 $ma(P) = ma(P_j)$
ただし, j は次のようにして決まる.
 c が a の敵である場合 $mc(P_j) = \min_i mc(P_i)$
 c が a の味方である場合 $mc(P_j) = \max_i mc(P_i)$
- P がターミナルノードなら
 $ma(P) = EV_{ma}(P)$...静的評価

プレイヤー b, c の評価関数 mb, mc は, OM サーチの mi (ミニマックス戦略でもちいられる評価関数) と同じ定義であるが, 静的評価関数の値 $EV_{mb}(P)$, $EV_{mc}(P)$ はそれぞれ b, c の視点からの仮説に基づく.

OM サーチからの類推によれば, 評価値は通常 (1つの仮説に基づき1種の評価値をもちいて) ミニマックス戦略で求めた場合と異なる値になる. 具体的には敵のミスにより自分にとっての評価値が上がる場合と, パートナのミスにより自分にとっての評価値が下がる場合が考えられる⁶⁾.

3.4 プレイヤーのモデル

OM サーチでは2人のプレイヤーの評価値が異なる理由として, プレイヤーの技量 (探索の深さの違い) などが考えられた. 不完全情報ゲームであるブリッジでは, プレイヤーの技量が同じでも, 与えられた情報の違いにより同じ局面に対する評価値が異なる可能性がある.

我々はブリッジプレイヤーをビッドやプレイに関する知識と仮説推論機構を持つエージェントとしてモデル化し, 特にパートナーシップの実現に必要なシグナルに

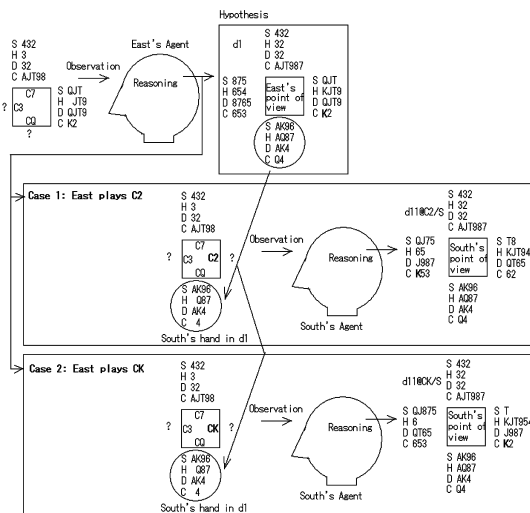


図 3 他プレイヤーのモデル
Fig. 3 Model of other players.

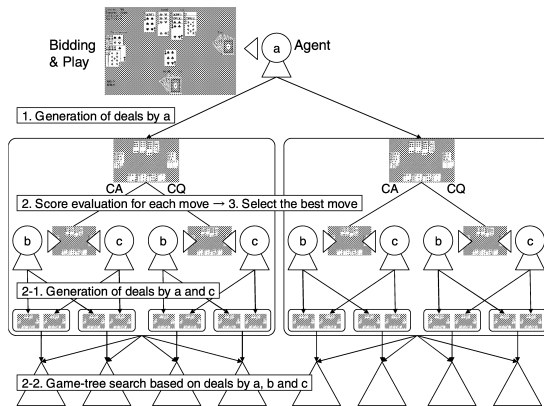


図 4 abc アルゴリズムの流れ
Fig. 4 Illustration of abc algorithm.

関する知識と推論を強化した^{7),8)}. カード選択法としてはモンテカルロシミュレーションの手法を導入した. 図2に示したように, 各プレイヤーのエージェントは観察した事実からディールに関する妥当な制約条件の集合を作る. この制約条件を満たすディールを多数生成し, 各ディールについて完全情報ゲームとしてゲーム木探索を行った結果を集計しカードを選択するのが3.1節の方法である. 本論文で提案する手法では, 図4に示すように現在の局面からさらに1枚のカードを出し, 他のプレイヤー (エージェント) が想定するディールを生成する. この3人のプレイヤーが想定するディールの3つ組をゲーム木探索にもちいる.

図3は4.1節の[例題1]においてEastが♣2と♣Kのどちらを出すか選択する局面を説明している. Eastのエージェントが生成した多数のディールの1つ

が d_1 である。East は d_1 における South のハンド (およびダミーのハンド) を仮定し、 $\clubsuit 2$ を出した場合 South のエージェントが想定する East と West のハンドを生成する。その 1 つが $d_{11}@C2/S$ である。 $d_{11}@CK/S$ は、 $\clubsuit K$ を出した場合に South が想定するディールの 1 つである。同様に、West のエージェントが想定するディールも生成する (図 8 参照)。

エージェントの知識は制約論理プログラムで表現した (付録にその例を示す)。4 人のプレイヤーが 52 枚のカードのそれぞれを持っているかどうかを (0 または 1 の値をとる) 208 個の変数で表現し、観察した事実と知識に基づいて推論を行い、それらの変数間の制約条件を作りだした。この制約条件と矛盾しない変数値の割当てが考慮すべき仮説である。乱数もちいて無作為に値を割り当てることで、モンテカルロシミュレーションのためのサンプルとなるディール (4 人のハンド) を得る。

知識としてはビッドの約束、オープニングリード、シグナルの約束などを実装した。本論文でめざす敵を惑わすプレイなど上級者のプレイテクニックを実現するために追加した知識には、「敵は損をするプレイをするはずがないと推論する」という経験則がある。また初心者が騙されやすい推論のパターンも経験則として実験的に組み込んだ。

3.5 abc アルゴリズム

我々は 3 人のプレイヤー (a, b, c) の推論を考慮した探索法として、3.3 節の評価関数をもちいた方法を提案する。

[アルゴリズム]

プレイヤー a がとりうる行動の候補の集合を M としたとき、次のようにして 1 つの行動 (次に出すカード) 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配り、プレイヤー a から見たディールの集合 $D(a)$ を作る。

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

[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配り、他のプレイヤー 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 $d \in D(a), d_b \in D(b, m, d), d_c \in D(c, m, d)$ の組合せで、行動 $m \in M$ を選んだ結果として得られるスコア $s(m, [d, d_b, d_c])$

をダブルダミーで計算する。ただしプレイヤー a, b, c はそれぞれディール d, d_b, d_c を仮定して (3.3 節の評価関数に従って) 行動を選択するものとする。

[Step2-3] 各ディール $d \in D(a)$ について $[d, d_b, d_c]$ のすべて (n 個) に対する平均値 $\sum s(m, [d, d_b, d_c])/n$ を求め、 $s(m, d)$ とする。

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

3.6 OM サーチとの比較

前節で提案したアルゴリズムは OM サーチを不完全情報ゲームに拡張したものであるが、以下のような違いがある。

- (1) 完全情報ゲームではなく、不完全情報ゲームを対象とする。
- (2) 2 人ゲームではなく、4 人 (ダミーをのぞけば 3 人) ゲームを対象とする。
- (3) 敵のモデルのほかに、味方のモデルを考える。
- (4) プレイヤーごとの評価値の相違は、主に与えられた情報 (から推論したハンド) の違いによると想定する。

上記の (4) より、我々のゲーム木探索においては、ある局面が一部のプレイヤーについては到達できない (探索経過に対応するカードが仮定したハンドに存在しない) 可能性がある。その場合、評価値を * で表した (図 9 参照)。3.3 節の定義における * は、次のように扱う。

v は、* ではないとしたとき、

$$\min(*, v) = v, \min(v, *) = v, \min(*, *) = *$$

$$\max(*, v) = v, \max(v, *) = v, \max(*, *) = *$$

4. 応 用

提案したアルゴリズムがどのような場面で有効に應用されるか例を示す。

4.1 例題 1: ディクレアラを惑わす

図 5 において、 $\heartsuit 6 - \heartsuit 2 - \heartsuit K - \heartsuit A, \clubsuit Q - \clubsuit 3 - \clubsuit 7$ とプレイが進んだとき、East は何を出すべきか。

4.1.1 上級者の考え方

知らないふりして $\clubsuit 2$ をプレイする。 $\clubsuit K$ のガードがなくなってしまうが、South にそれを知る余地はなく、疑う理由もない。South がもう一度フィネスをすればダミーにリード権を渡せなくなり、完全に困らせることができる⁹⁾。

4.1.2 従来のアルゴリズムによる解

3.1 節のアルゴリズムによれば、 $\clubsuit K$ が選択される。East の視点で生成したディールの集合の例を図 6

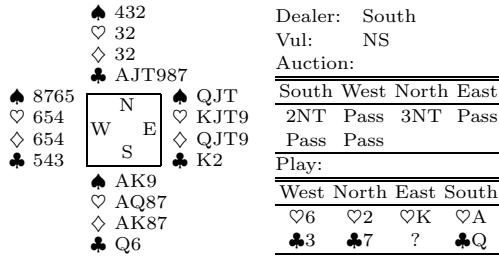


図 5 例題 1 のディールとオークション
Fig. 5 Deal and auction of Ex.1.

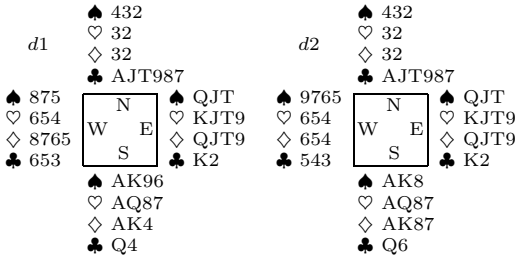


図 6 East の視点で生成したディールの集合
Fig. 6 Deals constructed from East's point of view.

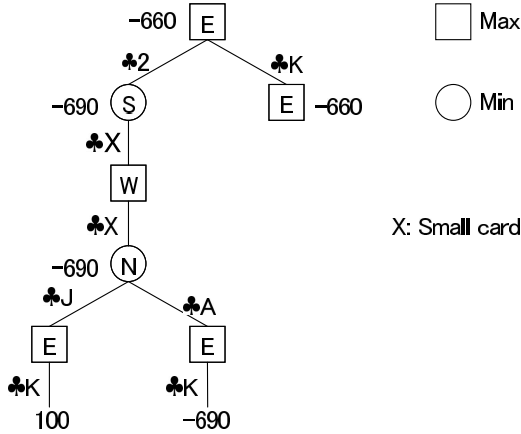
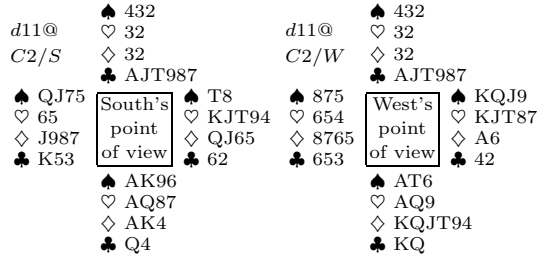


図 7 ミニマックス戦略によるゲーム木探索
Fig. 7 Game-tree search with minimax strategy.

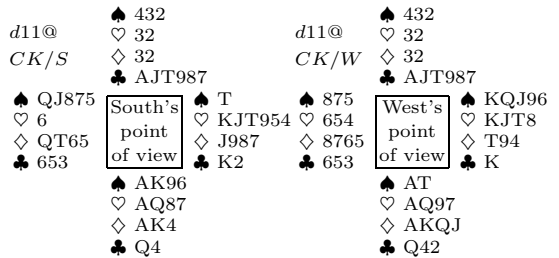
に示す。図 7 はディール $d1$ に対するミニマックス戦略に基づく評価値 (ゲーム木の一部) を表し、♣K をプレイする方が 30 点良いことが分かる。他のディールについても同様の結果になり、3.1 節のアルゴリズムでは (自分のハンドが他のプレイヤーに見えていると仮定しているので) ♣K が選択される。

4.1.3 abc アルゴリズムによる解

abc アルゴリズムをもちいた実験では、♣2 が選択された。その過程を単純化 (ディールを少なく) して説明すると次のようになる。ここでは、プレイヤー a, b, c はそれぞれ East, South, West である。



(a) Deals assuming that East plays ♣2.



(b) Deals assuming that East plays ♣K.

図 8 他者の視点で生成したディールの集合
Fig. 8 Deals constructed from the other player's point of view.

[Step1] East の視点で生成したディールの集合を図 6 とする。

[Step2] 各ディール $d1, d2$ ごとに、行動 $m \in \{\clubsuit 2, \clubsuit K\}$ をとったときのスコアを計算する。

[Step2-1] ディール $d1$ について East が ♣2 を出した場合に、South および West の視点で観察し生成したディールの一組を図 8 (a) に示す。この場合にはディクレアラである South は ♣K が West にあると思っている (エージェントによる推論の概要は付録参照)。またパートナーである West は ♣K が South にあると思っている。図 8 (b) は East が ♣K を出した場合の例である。

[Step2-2] 図 8 のディールに対応したゲーム木の一部を図 9 に示す。ゲーム木のルートから左に移動する部分では、図 8 (a) のディールを仮定している。ラベル N を持つノードで、(ディクレアラである South は) North から ♣J (を出してフィネスする) か ♣A を出すかを選択する。South の視点 (♣K は West と仮定) からの評価値 (3 つ組みの中段) は、♣J に対して -690、♣A に対して 100 である。min プレイヤであるディクレアラは、♣J を選択する。これは max プレイヤであるディフェンスの East にとっては喜ばしい。なぜなら ♣J が選択された場合、East の視点 (♣K は East にある事実を知っている) からの評価値 (3 つ組の上段) は 100 であり、(通常のミニマックス戦略で) ♣A が選択された場合の評価値 -690 を大幅

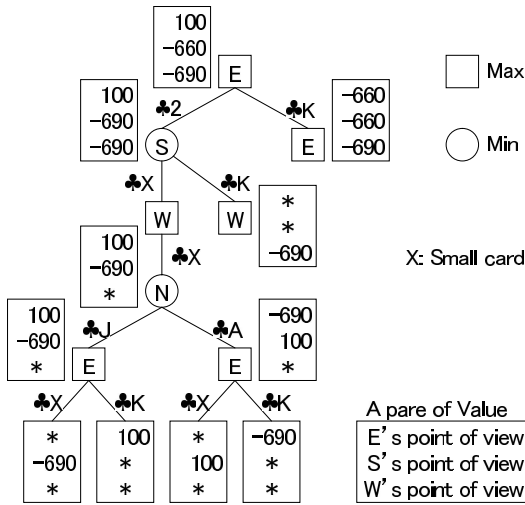


図 9 abc アルゴリズムに基づく 3 人の評価
Fig. 9 Game-tree search with abc algorithm.

に上回るからである。

ゲーム木のルートから右に移動する部分では図 8 (b) のディールを仮定している。こちら側では通常のミニマックス戦略と大差ないので詳細を省略した。

$$s(C2, d1, d11@C2/S, d11@C2/W) = 100$$

$$s(CK, d1, db11@CK/S, d11@CK/W) = -660$$

図 8 と同様な、 $d1$ を基にした他のディールの組については ($d12@C2/S$ において、South の視点から、♣2 を出した East が ♣K を 1 枚で残している場合が生成され) 次のようになった。

$$s(C2, d1, d12@C2/S, d12@C2/W) = -690$$

$$s(CK, d1, d12@CK/S, d12@CK/W) = -660$$

この 2 組で期待値を計算する。

$$s(C2, d1) = (100 - 690)/2 = -295$$

$$s(CK, d1) = (-660 - 660)/2 = -660$$

[Step2-3] $d2$ についても同様にする。

$$s(C2, d2) = (-690 + 200)/2 = -245$$

$$s(CK, d2) = (-660 - 660)/2 = -660$$

[Step3] 各候補の期待値を計算する。

$$s(C2, d1) + s(C2, d2) = -295 - 245 = -540$$

$$s(CK, d1) + s(CK, d2) = -660 - 660 = -1320$$

したがって East は C2 を選択することになる。

4.2 例題 2 : パートナを迷わせない

図 10 において $\diamond A - \diamond 6 - \diamond 9 - \diamond 3, \diamond K - \diamond 7 - \diamond 2 - \diamond 4$ とプレイが進んだとき、West は何を出すべきか (文献 10) より引用)。

4.2.1 上級者の考え方

次に \diamond を出すと、パートナーが切札で切った後 ♠ と ♣ のどちらを出したらよいか、分からないだろう。そ

♠ K4
♥ KQ9
♦ JT876
♣ Q32

Dealer: South
Auction:

South	West	North	East
1♥	1♠	3♥	Pass
4♥	Pass	Pass	Pass

Play:

West	North	East	South
♦A	♦6	♦9	♦3
♦K	♦7	♦2	♦4
?			

♠ J9875
♥ 65
♦ AK5
♣ A74

♠ QT62
♥ 32
♦ 92
♣ JT985

♠ A3
♥ AJT874
♦ Q43
♣ K6

図 10 例題 2 のディールとオークション
Fig. 10 Deal and auction of Ex.2.

♠ AJ32
♥ KQ543
♦ 2
♣ 654

Dealer: South
Vul: NS
Auction:

South	West	North	East
1♦	Pass	1♥	Pass
1♠	Pass	3♠	Pass
4♣	Pass	Pass	Pass

Play:

West	North	East	South
♣K	♣4	♣9	♣A
			?

♠ 654
♥ J987
♦ A87
♣ KQ7

♠ 87
♥ AT6
♦ 6543
♣ JT98

♠ KQT9
♥ 2
♦ KQJT9
♣ A32

図 11 例題 3 のディールとオークション
Fig. 11 Deal and auction of Ex.3.

ここで、先に ♣A を取ってから $\diamond 5$ を出す。

4.2.2 従来のアルゴリズムによる解

3.1 節のアルゴリズムによれば、♣A と $\diamond 5$ は同じに評価される。したがって偶然 ♣A を選ぶ可能性はあるが、パートナーのミスを防ぐ意図はまったくない。

4.2.3 abc アルゴリズムによる解

♣A が選択される。これは West がすぐ $\diamond 5$ を出したと仮定した場合、East の視点から推論した West のハンドに ♠A があるものが生成されるためである (付録参照)。この場合にはパートナーである East が \diamond を切札で切った後、誤って ♠ を出して失点することが [Step2-2] で計算される。West が先に ♣A を出した場合は次に West が $\diamond 5$ を出すので、パートナーのミスによる損はない。

4.3 例題 3 : ディフェンダを惑わす

図 11 において West の ♣ のリードを ♣A で取ったあと、ディクレアラは \diamond の何を出したらよいか (文献 9) より引用。ただし詳細は実験用に設定)。

4.3.1 上級者の考え方

敵に取ってほしかったら連続したカードの 1 番上を出す。取らないでほしかったら、1 番下を出す。この例では $\diamond KQJT9$ が連続しているが、 $\diamond 9$ を出してディフェンスが $\diamond A$ を取りそこなうチャンスをねらう。

4.3.2 従来のアルゴリズムによる解

3.1 節の方法によれば、どのように多数の仮説を生

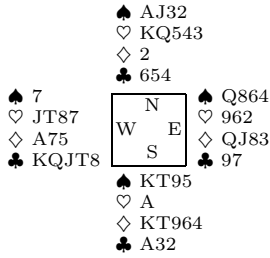


図 12 West の視点によるディールの例
Fig. 12 Deal constructed from West's point of view.

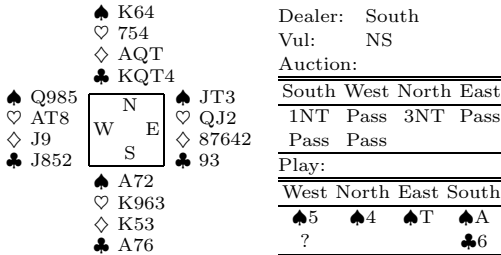


図 13 例題 4 のディールとオークション
Fig. 13 Deal and auction of Ex.4.

成しても South のハンドが見えた状態でゲーム木探索を行うので、♦KQJT9 の評価値は同じになる。

4.3.3 abc アルゴリズムによる解

実験では (9 以下を区別しなかったので) ♦T を選択した [Step2-1] で図 12 のような仮説が生成され (付録参照), [Step2-2] で West は (East が ♦QJ を持っていると思い) ♦A があっても出さない可能性がある と評価された結果である。

4.4 例題 4 : シグナルを出す判断

図 13 において West の ♠ リードを ♠A で取ったあと、ディクレアラは ♣6 を出した。West は何を出したらよいか (文献 11) より引用。ただし詳細は実験用に設定)。

4.4.1 上級者の考え方

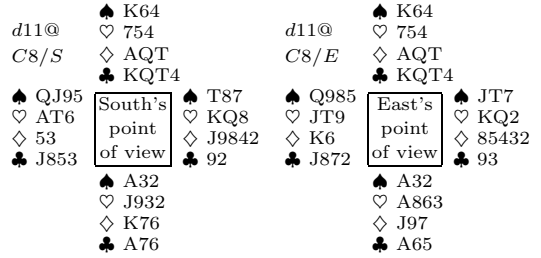
一般にパートナーへの (カウント) シグナルの価値はディクレアラがそれから得られる利益より大きい。しかしこの例題の場合には ♣ が偶数枚であることを ♣8 でシグナルすれば、ディクレアラの推論を助けて ♣ で 4 枚取らせることになる。よって ♣2 を出す。

4.4.2 従来のアルゴリズムによる解

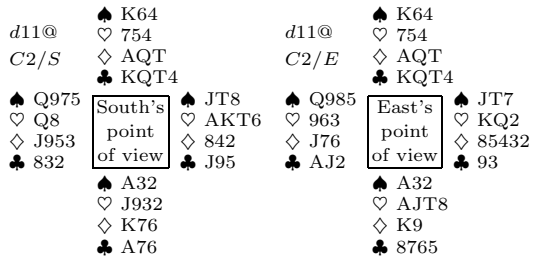
3.1 節の方法ではシグナルの評価はできない。商用コンピュータブリッジではつねにシグナルを出すか、まったく出さないかを事前に設定する。

4.4.3 abc アルゴリズムによる解

パートナーも敵もカウントシグナルが出されることを予期しているモデルで生成されたディールの一例を



(a) Deals assuming that East plays ♣8.



(b) Deals assuming that East plays ♣2.

図 14 各プレイヤーの視点で生成したディール
Fig. 14 Deals constructed from the other player's point of view.

図 14 に示した (詳細は文献 8) 参照)。ゲーム木探索の結果、敵に ♣ が 3 枚だと思わせる ♣2 が選択された。パートナーも騙されたが悪影響はなかった⁸⁾。

5. 実装

OM サーチの実装法には One-Pass 方式と Probing 方式がある¹²⁾。我々はこの 2 つの方式により abc アルゴリズムを実装した。検証した結果 Probing 方式は OnePass 方式に比べ探索速度が 10 倍以上速く、結果も妥当であると判明した。そこで今回は Probing 方式を採用した、64 台の並列処理により実験した。

5.1 One-pass 方式

One-pass 方式とは各ノードを 1 度しか訪れないで計算を進める方法である。OM サーチにおいて Iida らは「Max ノードでの枝刈はできず、Min ノードで ($\alpha\beta$ 法でいう) β カットが可能である」ことを示している。abc アルゴリズムでも a のプレイするノードでの枝刈は勝利にはできない。さらにプレイヤー b , c はそれぞれ独立にミニマックス戦略で探索しているので、One-pass 方式において片方のプレイヤーの判断でカットすることはできない。一方、我々の設定では 3.6 節で説明したように、一部のプレイヤーが到達できない (評価値が * で示される) ノードがある。そこで表 1 のような $\alpha\beta$ 法のカットと * の組合せで、全プレイヤーにとって unnecessary な場合に枝刈を行った。4 トリック (カード 16 枚) の先読みで、枝刈をしない場

表 1 カットの可能な組合せ
Table 1 Conditions to be cut.

a の判断	b の判断	c の判断
カット or *	カット or *	カット or *

合の 1/4 程度のノード探索で済んだ¹³⁾。

5.2 Probing 方式

Probing 方式では他人のプレイするノードにおいてミニマックス戦略 ($\alpha\beta$ 法) でどの枝が選ばれるか予測し、選ばれた枝のみ自分の評価値を返す。さらに自分のプレイするノードでは最大の評価値を返す枝を選び、末端のノードでは自分の静的評価値を返す。One-pass 方式とちがいで、ゲーム木の同じ部分を繰り返したるので計算時間が長くなる可能性もある。しかし $\alpha\beta$ 法が使えるので、OM サーチでは One-pass 方式より速いと報告されている¹²⁾。abc アルゴリズムにおける我々の実装でも、10 倍ほど速くなった。

5.3 並列処理

我々は先にコンピュータブリッジの並列処理方式を提案し、64 台のプロセッサで 40 倍以上の高速化を実現した¹⁴⁾。この方式に基づき abc アルゴリズムの並列処理を実装した。3.5 節で述べた abc アルゴリズムの [Step2-2] におけるスコア計算をタスクの単位とし複数のプロセッサに割り振り並列に実行する。実験では 64 台のプロセッサで 30 倍以上の効果を得た。4 章の検証では、例題 1 で 4.83s、例題 2 で 250.53s、例題 3 で 258.67s、例題 4 で 48.75s かかった。

6. リスク

敵のミスに期待する場合には一般的にリスクがともなう。例題 1 ではミニマックス戦略と比較して大きな利得を得る可能性を見つけている。敵のミスでコントラクトがダウンすれば利得は大きいですが、ミスしなければ少し損をする。ランダムに多くのディールを生成すれば確率的な期待値があるので、それに従ってカード選択を行う。

例題 2 ではミニマックス戦略と比較しての利得はまったくない。パートナーがミスをして損をする可能性を見つけている。むしろリスクを減らしているといえる。

例題 3 ではミニマックス戦略と比較して、大きな利得を得る可能性を見つけている。しかしリスクは犯していない ($\diamond A$ に取られてもともとである)。

例題 4 ではミニマックス戦略と比較して、大きな利得を得る可能性を見つけている。この例ではリスクはないが、一般的には敵と一緒にパートナーも騙す場合は

リスクをともなう。

従来のアルゴリズムは敵のベストプレイを仮定しているのでディクレアラのプレイではリスクがない。しかしディフェンスではパートナーのベストプレイを仮定しているのでリスクがある。abc アルゴリズムはリスクをともなう可能性があるが、それに見合う利得があるプレイを選択する。実装上では他のプレイヤー (b または c) の選択において同じ評価値の行動が複数個ある場合にもリスクの問題が起きる。max プレイヤである a の評価値が最小のものを選択することにより、過大評価になるリスクをさけることができる。

実戦的には、他者のハンドの正しい推論と期待値を計算するに足るディールの生成がないと、誤った判断をするリスクが大きくなる。我々は特に難しいディフェンダの推論を強化するために、シグナルによるパートナーシップを確立した¹⁵⁾。

7. ディフェンスの実験と評価

7.1 シグナル

パートナー間のコミュニケーションの手段としてシグナルが知られている。

たとえば例題 1 で West の出した $\clubsuit 3$ は自分の \clubsuit の中で一番小さいカードなので、 \clubsuit を奇数枚持っていることをパートナーである East に伝えている。このカウントシグナルにより、図 6 の East の視点から見たディールでは West の \clubsuit が 3 枚、South の \clubsuit が 2 枚と正確に推論されている。これを裏付けとして、 $\clubsuit K$ を 2 回目取るプレイを発見する。

実験では準備段階として従来のアルゴリズムにシグナルの機能を組み込み、桜井恒夫著「ディフェンスのシグナルとカードプレイ」¹⁰⁾ の第 1 部にある約 100 題の問題を解き、90%以上の正解率を得た¹⁵⁾。

7.2 ディフェンスの技法

前節の準備の後、abc アルゴリズムがどの程度有効か評価するために文献 10) の第 2 部にあるディフェンスの技法に関する問題をすべて解かせた¹⁵⁾。同じ深さ (16) で従来のアルゴリズム (シグナルの機能について強化した版) と abc アルゴリズムとで解かせた結果を表 2 にしめす。

「パートナーを迷わせない」、「(パートナーからリードしにくいスートをリードするために) リードを取る」という分類に入る問題では、予想どおりの改善が見られた。一方、「(相手方の) エントリをなくす、(味方のエントリを) 作る」では正解数が半減した。主な原因は、敵がこちらのハンドを知らないで誤ったプレイをすると考え、(ミニマックス戦略に基づく) 敵の

表 2 従来手法と abc アルゴリズムの比較実験
Table 2 Number of problems solved by minimax and abc algorithm.

分類	問題数	正解数		
		両者正解	minimax のみ	abc のみ
パートナーを迷わせない	11	1	0	4
リードを取る	6	3	0	2
スローイン, スクイズをさせる	8	1	0	0
絵札, 長さを知る	5	0	0	0
コントロールを残す	1	0	0	0
トランプトリック, ラフ	23	8	5	4
ダックする, しない, させない	25	12	4	0
2, 3, 4 枚カード	18	1	4	1
エントリをなくす, 作る	22	3	8	3

ベストプレイを予想した正解とは異なる解を出したためである。「2, 3, 4 枚カード (の注意すべきディフェンス)」でも, やはり敵のミス进行を期待して, 次善の解になったものが多い。「ダックする, しない, させない」の分類中, ダックする問題については, 従来方式に組み込んだヒューリスティックな知識が正解数を左右した。これは「候補の評価値が同じならば, 小さいカードを選択せよ」という知識であり, ダックする (そのトリックで勝てるのにわざと負ける) 真の理由によらずに正解を出す原因となった。abc アルゴリズムではパートナーのミスも考慮するので, 各候補の評価値がまったく同じになることは少なく, この知識はほとんど利用されない。以上の結果をふまえて, 次のような考察を行った。

- (1) abc アルゴリズムはパートナーシップの改善には有効である。
- (2) 従来アルゴリズムの方が有効な場合が多いので, 2つの方法の融合をはかるのが実践的である。
- (3) 融合方法として次のような案が考えられる。
 - ミニマックス戦略で探索したとき, 相手のコントロールをダウンさせられる確率が低い場合には, abc アルゴリズムを適用してみる。評価値の大きな改善が見られれば, その解を選ぶ。
 - ミニマックス戦略で探索したとき, 評価値の同じ候補が複数ある場合には, それらの候補について abc アルゴリズムを適用する。評価値の大きな改善が見られれば, その解を選ぶ。

従来のアルゴリズムでは並列処理をもちいれれば深さ 28 まで探索できる可能性がある。表 2 の問題について深さ 28 で実験したところ深さ 16 の場合に比べ, 正解数が 12 増えた。一方 abc アルゴリズムは深さ 16 での実験において, 短いもので 3s, 長い場合は 3000s 以上かかる場合がある。そこで従来方式の深い探索を基本とし, 場合を限って候補を絞り, 時間制限付で abc アルゴリズムを適用するのがよいと考える。

8. おわりに

コンピュータブリッジにおけるプレイヤーのモデルとそれに基づくゲーム木探索法を提案した。それは Iida らの OM サーチを不完全情報ゲームに拡張したものであり, 不完全情報ゲーム特有の背景と特徴を持っている。応用面では, 従来のコンピュータブリッジでは不可能であった上級者のプレイテクニックのいくつかを可能とした。一方, ゲーム木探索の複雑化にともない計算時間が長くなりリアルタイムのプレイが難しくなったが, 並列処理の導入によりある程度解決した。解説書の例題を解く実験により, 今まで不可能であったプレイができるようになった。しかし問題点も見つかり, 従来方式との実践的な融合が必要であることが判明した。

参考文献

- 1) Iida, H., Uiterwijk, J.W.H.M., van den Herick, H.J. and Herschberg, I.S.: Potential application of opponent-model search, Part1, *ICCA Journal*, Vol.16, No.4, pp.201-208 (1993).
- 2) Iida, H., Uiterwijk, J.W.H.M., van den Herick, H.J. and Herschberg, I.S.: Potential application of opponent-model search, Part2, *ICCA Journal*, Vol.17, No.1, pp.10-14 (1994).
- 3) Sturtevant, N.: Current challenges in multi-player game search, *Proc. Computers and Games* (2004).
- 4) Billings, D., Davidson, A., Schaeffer, J. and Szafron, D.: The challenge of poker, *Artificial Intelligence*, Vol.134, No.1-2, pp.201-240 (2002).
- 5) Ginsberg, M.L.: GIB: Steps toward an expert-level bridge-playing program, *IJCAI-99* (1999).
- 6) Otawa, T. and Uehara, T.: Model of bridge players who make mistakes, *IPJS SIG Technical Reports*, Vol.2005, No.87, pp.25-32 (2005).
- 7) 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).
- 8) 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).
- 9) ビクター・モロー, ニコ・ガードナー(著), 難波田愈(訳): カードプレイテクニック, 日本コントラクトブリッジ連盟 (1998).
- 10) 桜井恒夫: ディフェンスのシグナルとカードプレイ, エスアイビー・アクセス (1998), 改訂版 (2002).
- 11) Bird, D. and Smith, M.: *Defensive Signal, Bridge Technique Series*, Master Point Press (2000).
- 12) Donkers, H.H.L.M.: Searching with opponent models, *SIKS Dissertation Series*, No.2003-13 (2003)
- 13) 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探索アルゴリズムと高速化, 情報処理学会ゲーム情報学研究会, *The 9th Game Programming Workshop 2004*, pp.64-70 (2004).
- 14) 小田和友仁, 上原貴夫: コンピュータブリッジの並列処理, 情報処理学会論文誌, Vol.47, No.5, pp.1538-1545 (2006).
- 15) 村上隆志, 小田和友仁, 上原貴夫: コンピュータブリッジにおける各種シグナルの実装, 情報処理学会第 67 回全国大会, 6L-8 (2005).
- 16) 小田和友仁, 村上隆志, 上原貴夫: コンピュータブリッジにおけるディフェンス, 情報処理学会ゲーム情報学研究会, *The 10th Game Programming Workshop 2005*, pp.24-31 (2005).

付 録

1. エージェントの知識

[基本的知識の例]

絵札点 (HCP) の合計は 40 点

$$\text{SouthHCP} + \text{WestHCP} + \text{NorthHCP} \\ + \text{EastHCP} = 40$$

プレイヤーの HCP は持っている各スーツの絵札点の合計 (他のプレイヤーも同様)

$$\text{SouthHCP} = \text{SouthSpadeHCP} \\ + \text{SouthHeartHCP} \\ + \text{SouthDiamondHCP} \\ + \text{SouthClubHCP}$$

スーツの絵札点は Ace が 4 点, King が 3 点, Queen が 2 点, Jack が 1 点として合計 (他のプレイヤー, スーツも同様)

$$\text{SouthSpadeHCP} = 4 * \text{SouthSpadeAce} \\ + 3 * \text{SouthSpadeKing} \\ + 2 * \text{SouthSpadeQueen}$$

$$+ 1 * \text{SouthSpadeJack}$$

スーツは 13 枚 (他のスーツも同様)

$$\text{SouthSpade} + \text{WestSpade} + \text{NorthSpade} \\ + \text{EastSpade} = 13$$

ハンド中のスーツの枚数は, そのスーツのカードの合計枚数 (他のプレイヤー, スーツも同様)

$$\text{SouthSpade} = \text{SouthSpadeAce} \\ + \text{SouthSpadeKing} \\ + \dots + \text{SouthSpade2}$$

プレイヤーのハンド中の 52 枚のカードの有 (1) 無 (0) を 208 個の変数で表現 (他のプレイヤー, スーツも同様)

$$\text{SouthSpadeAce} = 0 \text{ or } 1, \\ \text{SouthSpadeKing} = 0 \text{ or } 1, \\ \dots \text{SouthSpade2} = 0 \text{ or } 1$$

あるカードを複数のプレイヤーが持っていることはない (他のカードも同様)

$$\text{SouthSpadeAce} + \text{WestSpadeAce} \\ + \text{NorthSpadeAce} \\ + \text{EastSpadeAce} = 1, \\ \text{SouthSpadeKing} + \text{WestSpadeKing} \\ + \text{NorthSpadeKing} \\ + \text{EastSpadeKing} = 1, \\ \dots \text{SouthSpade2} + \text{WestSpade2} \\ + \text{NorthSpade2} \\ + \text{EastSpade2} = 1$$

[ビッドに関する知識の例]

1NT オープニングビッド:

$$\text{HCP} \geq 15, \text{HCP} \leq 17, \text{バランスハンド}$$

2NT オープニングビッド:

$$\text{HCP} \geq 21, \text{HCP} \leq 22, \text{バランスハンド}$$

1H のオープニングビッド: 13 点以上で Heart が 5 枚以上

1S のオーバーコール: Spade が 5 枚以上

[オープニングリードに関する知識の例]

KQx は K からリードする

[シグナルに関する知識の例]

ローカードは奇数枚, ハイカードは偶数枚

[経験則の例]

敵が A を持っているとき, K を 1 枚だけにするプレイは避ける

2. 例題におけるエージェントの推論

[例題 1]

East の推論

観察:

- (1) EastClub = 2 (自分のハンド),
NorthClub = 6 (ダミーのハンド)

- (2) EastHCP = 13 (自分のハンド),
NorthClub = 5 (ダミーのハンド)
- (3) South の 2NT ビッド
- (4) West の ♣3 (ローカードのプレイ)
- (5) 4 枚の Jack は East と North にある
(West に Jack はない)

推論:

- (1) WestClub + SouthClub = 13 - 6 - 2 = 5
(基本的知識, 観察 1 より)
- (2) WestHCP = 0 (基本的知識, 観察 2, 3, 5 より)
- (3) Club の枚数は West が 3 枚で South が 2 枚,
あるいは West が 1 枚で South が 4 枚
(シグナルに関する知識, 観察 4, 推論 1 より)

仮説生成の例 (図 6)

East が ♣2 を出した場合の South の推論

観察:

- (1) SouthClub = 2 (自分のハンド),
NorthClub = 6 (ダミーのハンド)
- (2) West の ♣3 (ローカードのプレイ)
- (3) East の ♣2 (ローカードのプレイ)

推論:

- (1) WestClub + EastClub = 5
- (2) West, East が両方奇数枚という仮説は推論 1
と矛盾するので, どちらかはシグナルを出して
いない
- (3) 前記の経験則によれば, West の Spade は K2
ではない

仮説生成の例 (図 8(a) 左)

[例題 2]

West の推論

観察:

- (1) WestDiamond = 3 (自分のハンド),
NorthDiamond = 5 (ダミーのハンド)
- (2) East の ◇9 (ハイカードのプレイ)
- (3) South は Diamond を 2 回出した

推論:

- (1) EastDiamond = 2 で, 次にトランプで切れる
West が ◇5 を出した場合の East の推論

観察:

- (1) South が 1H でオープニングビッド
- (2) West が 1S でオーバーコール
- (3) EastHCP = 3 (自分のハンド),
NorthHCP = 11 (ダミーのハンド)
- (4) West が ◇A, ◇K (合計 7 点) をプレイ

推論:

- (1) SouthHCP \geq 13

- (2) SouthHeart \geq 5
- (3) WestSpade \geq 5
- (4) WestHCP \leq 40 - 13 - 11 - 3 = 13
- (5) WestHCP - 7 \leq 6 で, West はまだ Spade
か Club の Ace (4 点) を持っている可能性が
ある. しかしどちらかは判明しない

仮説生成: West が ♠A を持っているディールが ♣A
を持っているディールと同じように生成される

[例題 3]

South の推論

観察:

- (1) NorthDiamond = 1, SouthDiamond = 5
- (2) SouthHCP = 15, NorthHCP = 10

推論:

- (1) EastDiamond + WestDiamond = 13 - 1 - 5 = 7
- (2) EastHCP + WestHCP = 40 - 10 - 15 = 15

仮説生成: 図 12 の West のハンドは一例

South が ◇10 を出した場合の East の推論

観察:

- (1) NorthDiamond = 1, WestDiamond = 3
- (2) South がオープン
- (3) WestHCP = 11, NorthHCP = 10

推論:

- (1) EastDiamond + SouthDiamond = 13 - 1 - 3 = 9
- (2) SouthHCP \geq 13
- (3) EastHCP \leq 40 - 11 - 10 - 13 = 6

仮説生成: 図 12 のハンドは一例 (West は East に
◇Q, ◇J がある可能性を信じた)

(平成 18 年 2 月 28 日受付)

(平成 18 年 9 月 14 日採録)



小田和友仁 (正会員)

1979 年生. 2002 年東京工科大学
工学部情報工学科卒業後, 同大学大
学院博士課程進学. 2006 年現在, 博
士後期課程 3 年目にあたる. ゲーム
プログラミングを題材に並列処理の

研究に従事.



上原 貴夫（正会員）

1942年生．1965年早稲田大学理工学部電気通信学科卒業．1970年同大学大学院博士課程修了．工学博士．同年（株）富士通研究所入社．1994年東京工科大学工学部情報工学科教授，2004年同大学コンピュータサイエンス学部教授．現在に至る．その間，1977年より1年間Stanford大学客員研究員．CAD，人工知能，ゲームプログラミング等の研究に従事．人工知能学会，電子情報通信学会，日本ソフトウェア科学会，日本コントラクトブリッジ連盟各会員．
