コンピュータブリッジによるディセプティブプレイ

小 林 紀 之† 上 原 貴 夫†

ブリッジは，他のプレイヤーのハンドが見えない不完全情報ゲームであるため，チェス，将棋などの完全情報ゲームとは異なる難しさがある．最近のコンピュータブリッジでは，モンテカルロシミュレーションを基本としたアルゴリズムが主流である．すなわち，見えていないカードの可能な配置（世界）を多数生成し，各世界でダブルダミーブリッジ（すべてのハンドを見て完全情報ゲームとしてプレイをする）として，ゲーム木におけるミニマックス値を求め，次に出すカードを選ぶ．現在，コンピュータブリッジの実力は中級者レベルにとどまっている．その原因の１つは，このアルゴリズムが，見えている自分Ｐのハンドと矛盾しない世界のみを対象としている点にある．本論文では，上級者のプレイテクニックの１つとして知られるディセプティブプレイを発見するアルゴリズムを提案する．敵を騙すプレイを可能にするためには，出されたカードを観察した敵がどのようなハンドを想定するかをモデル化する必要がある．著者らは，経験則に基づいて推論を行うエージェント A を敵のモデルとし，敵から見た世界を生成した．この世界は多くの場合，自分Ｐのハンドと矛盾する．ディセプティブプレイを発見する方法として，敵 A から見た世界では敵に有利であり，かつ，自分Ｐから見た世界では自分に有利なプレイを探索することを提案した．このアルゴリズムを実装して，典型的な例題でその機能を確認した．

Deceptive Play by Computer Bridge Program

Noriyuki Kobayasi† and Takao Uehara†

The Contract Bridge is an incomplete information game, that is, a player cannot look at the other player's hands. Its difficulty is different from that of complete information games such as Chess and Shogi. The Monte Carlo simulation is the most popular technique for modern Computer Bridges. Many random deals (worlds) are generated. The best card to play is selected according to minimax value of a complete information game-tree search as the double dummy Bridge in each world. Computer Bridges have never win the game against top players. It is partially because the algorithm does not generate a deal inconsistent with the hand of the player P. This paper proposes an algorithm to find a deceptive play that is one of the techniques used by top players. An opponent is modeled as an agent A who guesses P's hand by rule-based knowledge. Worlds generated by A may be inconsistent with the P's hand. The proposed algorithm finds a play line that is the best for opponent in the worlds generated by A and also good enough for P in the worlds generated by P. The algorithm is implemented and tested for some typical examples.

1. はじめに

ブリッジは，他のプレイヤーの持っているカードが見えない不完全情報ゲームであるため，チェス，将棋などすべての駒の位置が分かっている完全情報ゲームとは異なる難しさがある．テーブルについた４人のプレイヤー（North，South，East，West）が North/South ペアと East/West ペアに分かれて戦う．まず，ビディング（あるいはオークション）と呼ばれる段階があり，どのスーツ（スペード，ハート，ダイヤモンド，クラブのいずれか）を切札にして（あるいは切札なしのノートランプで）何回取れるかを競り合い，最も高いビッドがコントラクトとなる．次に，カードを１枚ずつテーブル上にプレイする段階に入り，ビディングで競り勝ったペアがコントラクトを達成できるかどうかで得失点が決まる．

コンピュータブリッジプログラムも，この２つの段階に対応して，ビッドのプログラムとプレイのプログラムに分けて研究されてきた[1),2)]．

オークションにおけるビッドの目標は，パートナーと協力して妥当なコントラクトに到達し，プレイの結果として高得点を得ることである．コンピュータビッダーは，ビディングシステムの約束をルールベース

† 東京工科大学
Tokyo University of Technology

（データベース）として持つ[3]．1つのハンドに対して複数のビッドが候補となる場合，モンテカルロシミュレーションの手法を使いビッドの先読みをして決定するコンピュータビッダーが多い[4]．パートナーのハンドを正しく評価するためには，パートナーがこちらのハンドをどのように考えているか知る必要がある．著者らは，ブリッジのプレイヤーをエージェントとしてモデル化することにより，他のプレイヤーの思考を考慮するコンピュータビッダーを提案した[5]．

プレイのプログラムとしては，経験則を特定の状況とそのときとるべき行動からなるルールの形式で記述し，これに従ってプレイする方式が初期に採用された．しかし，このようなルールに従うだけで先読みをしないプレイには限界があった．また，全員のハンドを見える状態（ダブルダミーと呼ばれる）にして，完全情報ゲームとして解くことも研究された[6]．ブリッジの各状態で4つの可能なプレイがあるとすると，52枚のカードをプレイするには $4^{52} \approx 10^{31}$ の可能なプレイの系列がある．ミニマックス法に対する高速化技法（アルファベータ剪定，トランスポジションテーブルなど）を使えば，実効的な分岐の数を（1.4程度に）減らすことができ，Ginsbergの手法（Partition Search）によれば（分岐数1.2程度にして），1秒程度で1つのハンドを分析することができるという[7,8]．不完全情報ゲームとしてのブリッジの探索空間はさらに大きい．この空間内ですべての可能なプレイの列を試すのを避ける1つの方法として，選択肢をあらかじめ決めたタイプのプレイに限る方法も提案された[9]．最近は，モンテカルロシミュレーションが主流である．すなわち，見えていないカードの可能な配置（世界）を多数生成し，各世界でダブルダミーブリッジのミニマックス値を求め，選ばれた候補の中から全体としてよさそうな行動を選ぶ方法である[8]．

現在のコンピュータブリッジの実力は中級者レベルといわれている[10]．上級者のプレイテクニックを研究したものとしては，スクイーズと呼ばれる終盤のプレイを行うエキスパートシステムがある[11]．本論文では，上級者のプレイテクニックの1つとして知られるディセプティブプレイについて検討する．敵を騙すディセプティブプレイを可能にするためには，敵がこちらのプレイを観察して，どのようなハンドを想定するかを知る必要がある．著者らは，ブリッジのプレイヤーをエージェントとしてモデル化し，敵が行う推論をシミュレーションすることにより，敵を騙すプレイを発見する方法を提案する．

2章では，主流となっているモンテカルロシミュレーションを紹介し，3章では，これをディセプティブプレイのアルゴリズムとして拡張する方法について述べる．4章では，試作プログラムの実装の概要と実験結果を報告する．

2. モンテカルロ法によるカード選択

モンテカルロシミュレーションによりプレイすべきカードを選択する方法の具体例として，Ginsbergのアルゴリズムを紹介する[12]．

[Ginsbergのアルゴリズム]

プレイヤーP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．

ステップ1) 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，ディール（4人のプレイヤーに13枚ずつ配られたカード）の集合 D を作る．

ステップ2) 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．

ステップ3) $\sum_d s(m, d)$ が最大となるような行動 m を選ぶ．

このモデルでは，次に出すべき1枚のカードを除き，各プレイヤーが（ダブルダミーで）ベストプレイをすることになる．したがって，敵を騙すプレイを発見することは不可能である．次節では，Ginsbergのアルゴリズムを基本として，ディセプティブプレイを発見できるように改造する方法を検討する．

3. ディセプティブプレイのアルゴリズム

3.1 敵の立場からの推論

Ginsbergのアルゴリズムのステップ1で作られるディールは，プレイヤーPの実際のハンドが見えているものとして，それに矛盾しないように配られる．本論文で提案するモデルでは，Pのハンドは他のプレイヤーには見えないものとして，敵のプレイヤーAの立場から作ったディールの集合をステップ2以降の評価に用いる．

著者らは，ビディングに関して，プレイヤーをエージェントとしてモデル化し，敵が行う推論をシミュレーションする方法を提案した[5]．本論文では，エージェントに，プレイされたカードを観察して，それを説明する仮説を生成するための知識を追加した．このモデルをエージェントAの立場から見たディールの集合を作るためにも使用する．**図1**はその概要を示す．

3.2 敵味方の評価の相違

Ginsbergのアルゴリズムのステップ2では，1つ

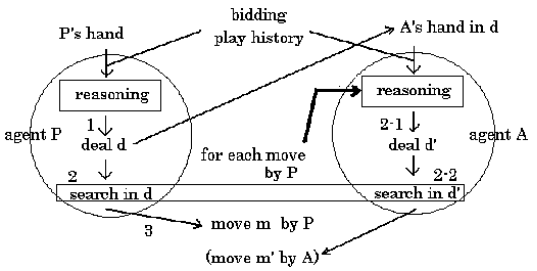

図 1 敵 A によるディールに基づくプレイヤー P による行動の選択
Fig. 1 Selection of move by player P based on deal by opponent A.

のディールにおける各行動に対し，プレイヤー P の立場から一通りの評価を出している．本論文で提案するディセプティブプレイのアルゴリズムでは，各行動ごとに，プレイヤー P の想定したディール d と敵 A の想定したディール d' の双方に対して評価を出す．ただし，プレイヤー P の選択はディール d を想定して行われ，（P から A にいったんコントロールが移った後の）敵 A の選択はディール d' を想定して行われたものとする．

3.3 敵を欺くプレイの発見

Ginsberg のアルゴリズムのステップ 3 では，プレイヤー P の立場からの総合評価の高い行動を選択している．本論文で提案するアルゴリズムでも，プレイヤー P の想定したディールの集合 D において総合評価の高い行動 m を選択する．その行動 m が敵 A の想定したディールの集合 D' において低い（敵に有利な）評価を持つ場合には，ディセプティブプレイが発見されたことになる．

3.4 アルゴリズム

以上の考察より，ディセプティブプレイを発見するためのアルゴリズムとして次の方法を提案する．

[アルゴリズム]

プレイヤー P 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．

ステップ 1) 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 P から見たディールの集合 D を作る．

ステップ 2) 各ディール $d \in D$ ごとに，次の手順で，各行動 $m \in M$ を選んだら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．

2-1) ディール d で行動 m を行った結果，コントロール（カードを出す番）が敵のプレイヤー A に移った場合，プレイヤー A が行動 m を観察して想定するディールの集合 D' を作る．

2-2) 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，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 $s([m, m'], d')$ を計算する．また，敵はディール d' を想定し，味方はディール d を想定してプレイした場合のスコア $s([m, m'], [d, d'])$ を計算する．

2-3) $\sum_{d'} s([m, m'], d')$ が最小となるような行動 m' を選び，$s(m, d)$ を $s([m, m'], [d, d'])$ とする．

ステップ 3) $\sum_{d} s(m, d)$ が最大となるような行動 m を選ぶ．

行動 m は通常 1 枚のカードの選択であるが，その 1 枚を出した結果勝って味方にコントロールが移る場合は，次の 1 枚の選択も合わせて行動 m と考える．また，敵のプレイヤー A は 2 人の敵のうち次にプレイする敵（P がディフェンダーの場合は，A はディクレアラー）である．

3.5 プレイヤーの知識

ビッドに関する知識とオークションの段階での推論については文献 13) に詳しい．ここでは，プレイの段階で用いる知識と推論について考察する．プレイに関する知識としては，次のようなものがある．

(1) プレイのルール
(2) オープニングリードに関する約束
(3) ディフェンスに関する約束（シグナル）

たとえば，次のようなものがある．

(3-1) トランプコントラクトにおける A のオープニングリードに対し，3 人目は K を持っていればカモンシグナル（大きいカード）を出す．

(3-2) トランプコントラクトにおける K のオープニングリードに対し，3 人目はダブルトン（2 枚）ならばカモンシグナルを出す．

パートナー間の約束事項は，敵にも知らせておかなければならない．リードやシグナルを観察し，これらの知識に基づく推論により，敵やパートナーのハンドに関する事実および仮説を生成することができる[14]．

本論文では，さらに，ブリッジプレイヤーが持つ経験則をエージェントの知識として加えた．たとえば，次のようなものがある．

(4-1) 4 人目のプレイヤーは，負けると分かっている場合には，最も弱いカードを出して負ける（ただし，連続して持っているカードは同じ強さと考える）．

(4-2) 4 人目のプレイヤーは，勝つ場合には，勝てるうちで最も弱いカードを出して勝つ．

(4-3) ノートランプにおけるスモールカードのオープニングリードに対して，ダミーにスモールカードし

かない場合，3 人目のプレイヤーは最も強いカードを出す（連続して持っている場合は最低のカード）．

(4-4) トランプコントラクトでディクレアラーがサイドスーツを切ろうとする場合，トランプをかりきらずに負けてくる．

(4-5) ノートランプのオープニングリードに対して，A しかストッパーのないディクレアラーはダックする（すぐにはとらない）．

(4-6) K に対するフィネスが失敗した場合，そのスーツが 2 枚以下ならば K でとる．

本論文では，プレイヤーのモデルとして，このような経験則に従って敵のハンドに関する推論を行うエージェントを採用する．また，騙されるプレイヤーは相手がディセプティブプレイを行うとは考えず，自分もディセプティブプレイは考慮せずに行動を決定するものとした．また，騙す方のプレイヤーも同じ経験則に従って敵のハンドに関する推論を行うが，ディセプティブプレイを発見する能力を持つものとした．

4. 試作プログラムと実験

4.1 制約論理プログラムを用いた実装

[エージェントの実装]

著者らが試作中のコンピュータブリッジは，制約論理プログラミング言語 ECLiPSe を用いている．各プレイヤーが 52 枚のカードのそれぞれを持っているか否かを示す 208 個の変数を基本とし，各スーツの枚数，ハンドの点数などを表す変数を導入した[14]．これら変数を用い，Goren の本[15]に記載されているビッドの約束を記述した[5]．また，プレイに関する知識は，ゲームのルール，パートナー間の約束，経験則に分けて記述した．各プレイヤーは，このような知識と仮説推論機構を持つエージェントとして実装されている．

[ディールの生成]

前記のアルゴリズムにおけるディール生成法は，次の順序で行う．

(1) 各プレイヤーが 13 枚のカードを持っており，同じカードを持っているプレイヤーはいないなどの基本的制約条件の設定を行う．

(2) プレイされたカード，テーブル上のダミーのカード，自分のハンドについて，どのプレイヤーが持っていたかという事実に対応する変数値を設定する．

(3) ビッドの経過から，今までに設定した制約条件と矛盾しない仮説（各ビッドの前提条件の AND）を生成する．

(4) プレイ経過から，経験則に従って推論を行い，新たな制約条件を（AND して）加える．ただし，それまでに設定した仮説と矛盾する場合は加えない．

(5) 上記の制約条件を満たすディールを無作為に生成する．

```
update(MaxMin,Card,Value, [Card1,Value1],[Card1,Value1]):-
  MaxMin=max, Value =< Value1;  MaxMin=min, Value >= Value1.
update(MaxMin,Card,Value,[Card1,Value1],[Card,Value]):-
  MaxMin=max,Value > Value1;  MaxMin=min, Value < Value1.
```

(a) well-known algorithm

```
update2(max,Card,[RValue,IValue],(RRecord,IRecord), (RNewRecord,INewRecord)):-
    update(max,Card,RValue,RRecord,RNewRecord),
    update(max,Card,IValue,IRecord,INewRecord).
update2(min,Card,[RValue,IValue],(RRecord,IRecord), (RNewRecord,INewRecord)):-
    update(min,Card,IValue,IRecord,INewRecord),
% update in the real world d depends on the update in the imaginary world d'
    IRecord=[ _ , IRecordValue],
    (IValue > IRecordValue,   RNewRecord=RRecord;
     IValue < IRecordValue,   RNewRecord = [Card, RValue];
     IValue = IRecordValue,   update(min,Card,RValue,RRecord,RNewRecord)).
```

(b) proposed algorithm

図 2 ゲーム木の節点におけるカードの選択と値の更新
Fig. 2 Selection of cards and update of values at a node of a game-tree.

ただし，3.4 節のアルゴリズムのステップ 2-1 において，エージェント P が次にプレイする敵のエージェント A の立場から見たディールの生成を行う場合には，上記の (2) の「プレイされたカード」および (4) の「プレイ経過」としてエージェント P の行動 m の結果出されるカードも加える．また，上記 (2) の「自分のハンド」としては，エージェント P から見たエージェント A のハンドが対応するが，事実ではなく仮説となる．

[スコアの計算]

ダブルダミー（完全情報ゲーム）としてのスコアの計算は，ミニマックス原理に従って実行する．ただし，敵の選択はディール d' を想定して行われるが，味方の評価はディール d に基づいて決まる．実際のプログラムにおいては，**図 2**（Prolog）のような選択方法をとった．ただし，ディセプティブプレイを行おうとする（ゲーム木の根に対応する）プレイヤーを max とし，敵を min とした．

図 2 (a) は，通常のミニマックスアルゴリズムで用いられるものである．*update* (*MaxMin*, *Card*, *Value*, *Record*, *NewRecord*) は，MaxMin（max か min）節点において，候補となる Card とその評価値 Value が与えられたとき，いままでの記録 Record と比べて，望ましいほうを NewRecord として残す処理を行うものである（プログラム中の；は OR を表す）．ただし，Record および NewRecord はカードと評価値のリストとする．

図 2 (b) のプログラムは，本論文で提案したアルゴリズムにおいて，ゲーム木の min 節点，max 節点でどのようにして Card を選択するかを示している．*update*2 (*MaxMin*, *Card*, *Values*, *Records*, *NewRecords*) は，MaxMin（max か min）節点において，候補となる Card とその評価値 Values が与えられたとき，いままでの記録 Records と比べて，望ましいほうを NewRecods として残す処理を行うものである．ただし，Values は [*RValue*, *IValue*] の 2 つの評価値のリストで，RValue はプレイヤー P の立場から見て生成したディール d に対する評価値，IValue は敵 A の立場から見て生成したディール d' に対する評価値である．Records および NewRecords は，プレイヤー P および敵 A が選択したカードと評価値のリストの対である．各 max 節点では，プレイヤー P の立場ではディール d に対する評価値で選択するが，敵 A はディール d' を想定した評価値に従って選択する．各 min 節点での選択はもう少し複雑になる．敵 A はディール d' を想定した評価値に従って選択し，プレイヤー P はその選択に従ったディール d に対する評価値を得る．

ゲーム木の葉では，d, d' のそれぞれで，残りのハンドにおいて強いものから順にとった場合の回数を数え評価値とした．実際のスコアは，とれた回数（トリック数）がコントラクトを達成しているかどうかで大きく違う．例題の説明では，分かりやすくするために，原則として，スコアをトリック数そのものとしてある．

ある行動が d では可能だが，d' では不可能な場合には，不可能なほうの評価値を未定義とし，プログラムでは未定義の場合には値の更新が起こらないようにした（実験で複数のディールを同時に評価する際には，スコアが未定義のものの数に左右されないように，定義されたものに対する平均値を用いた）．

4.2 実　　験

［例題 1］ **図 3** のハンドを持つ West の 3 ノートランプ（切り札なしで 9 トリックとるというコントラクト）に対して North はスペードのスモールカードをリードし，South からスペード J が出た．West は何をプレイすればよいか[16]．

この問題で試作プログラムのテストをするため，具体的なディールを与え，ビッドとプレイを**図 4** のように進めた（ただし，NS vul. で，IMP 形式で得点を計算する状況を想定した）．

この例題に試作したプログラムを適用した結果を次に示す．

図 4 に示す West のハンドと，ビッドの経過とプレイの経過を（West のエージェントに）与えて生成したディールの 1 つ d が**図 5** (a) である．図 5 (b) は，（North のエージェントに）図 5 (a) における North のハンドを与え，図 4 のビッドおよびプレイ経過の後に West がスペードの Q を出したという仮定のもとで生成したディール d'_1 である．図 5 (c) は，（North のエージェントに）図 5 (a) における North のハンドを与え，West がスペードの A を出したという仮定のもとで生成したディール d'_2 である．

West　　East
S AQx　　S xx
H Jxx　　H xx
D KJxx　　D AQx
C Axx　　C QJ10xxx

図 3 例題 1
Fig. 3 Example 1.

S K10875
H K872
D 87
C K6

S AQ6
H J65
D KJ65
C A54

North
West East
South

S 32
H 109
D AQ10
C QJ10987

S J94
H AQ43
D 9432
C 32

NS Vul.
Dealer South

Bidding	South	West	North	East
	Pass	1D	Pass	2C
	Pass	3C	Pass	3D
	Pass	3NT	Pass	Pass
	Pass			

Play	North	East	South	West
	S7	S2	SJ	?

図 4 例題 1 のディール，ビッドとプレイ経過
Fig. 4 Deal, bidding and play history of Example 1.

図 6 は，アルゴリズム（のステップ 2-2 以降）を試作プログラムで実行したときの要点を説明したものである．プログラムでは，3 トリック（12 枚）先読みを原則とし，敵にコントロールがある間はさらに先まで読んだ．ゲーム木の葉にあたる部分では，コントラクトがダウンすることが明らかになったら評価値を 0 とし，それ以外の場合は大きいカードからプレイする方法でとれるトリック数を評価値とした．

結果を人間に分かりやすく分析すると，次のようになる．スペード Q でとった場合，North は経験則 (4-3)

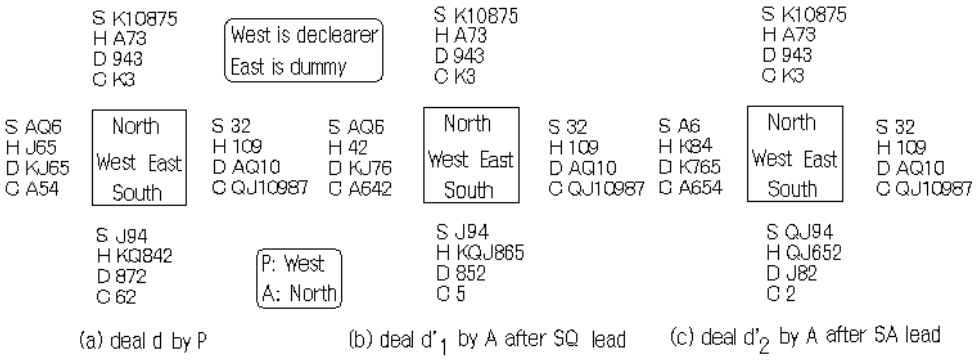

図 5 プレイヤー P および敵のエージェント A が生成したディール
Fig. 5 Deals generated by the player P and opponent's agent A.

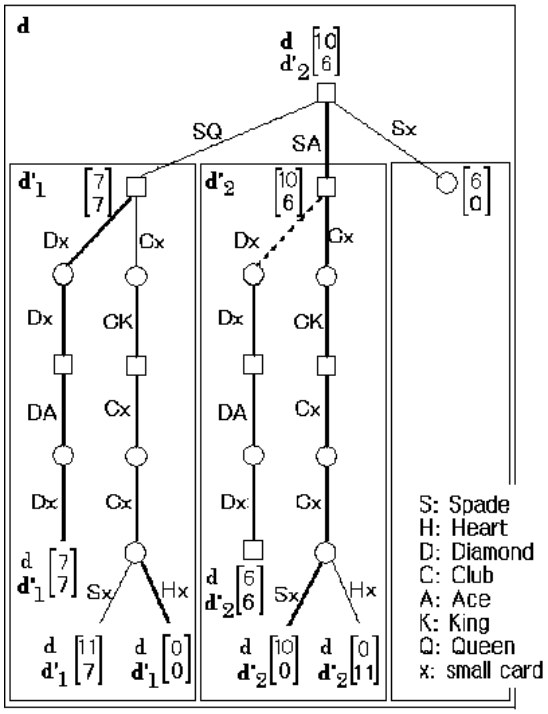

図 6 ディセプティブプレイを発見するためのゲーム木探索
Fig. 6 Game-tree search to find a deceptive play.

により（South のスペードは J が最も強いと推論し）West にスペード AQ があるという推論を行う．この制約のもとで生成したハンド d'_1 を見て，ハートが弱いことを知り，クラブ K でとれたときにハートをリードする．West の想定したディール d でも，このプレイの方針でこられるとダウンするので，7 トリックしかとれないと計算される．(4 人目の）West がスペード A でとれば，North は経験則 (4-2) により（J と同じ強さの）スペード Q が South にあると推論する．この制約のもとで生成したハンド d'_2 を見て，クラブ K でとれたときにスペードをリードするので，West はディール d で 10 トリックとれると計算する（最初に South の J に負ける場合は省略したが，経験則 (4-5) で West はスペードで A しかとれないと推論される．しかし，South からスペードが出されてもハートが出されてもコントラクトは達成できない）．我々の試作プログラムでは，West がスペードの A を出すことを選択し，図 4 に示す実際のハンドに対してコントラクトを達成した．

ちなみに，GIB（バージョン 5.1.7）に図 5 のハンドで 4 人のプレイをさせるとスペードのリードを West が Q でとり，クラブの K で負けたとき North はハートを出しコントラクトはダウンする．すなわち，ディセプティブプレイは行わない．また，GIB にディフェンスをやらせ，ディクレアラーはスペードのリードを A でとりクラブを出すと，North の GIB はクラブ K，スペード K をとり，さらにスペードを続けるのでコントラクトは達成された．

4.3 有効性に関する考察

ビクター・モロー，ニコ・ガードナー著「カードプレイテクニック」[16]に掲載されているディセプティブプレイの練習問題のすべてについて，本論文で提案した方法で正解を発見できるかどうか確認した．「ディフェンダーを惑わす」という章の 8 問については，提案した方法で正解を発見できた．「ディクレアラーを惑わす」という章の 8 問のうちの 4 問も提案した方法で正解を発見できた．トランプに関する 3 問については，ディセプティブプレイが有効であるようなディクレアラーのハンドをサンプルとして与えれば，正解を発見できた．また，1 問は，敵を惑わすためにディセプティブプレイをやったりやらなかったりするのが正解となるもので，本論文で対象とした敵のモデルの範囲を超えた問題である．総合的にみて，サンプルの

生成に改善の余地があるものの，さまざまなタイプのディセプティブプレイを発見できる方法だといえる．

オーバーヘッドとしては，モンテカルロ法の各サンプルに対し，それぞれ M 個のサンプルを発生するので，通常の方法の M 倍程度の時間がかかる．例題 1 のような状況に実戦で出会うことはまれなので（付録の例題 2，例題 3 のタイプはもう少し頻繁にあらわれるが），Ginsberg は与えられた時間をもっと基本的なことに割り当てたほうが強くできると考えているようである．しかし，多くの標準的なハンドを上手くプレイすることとともに，特殊な状況で思いもよらないプレイを発見することは，ブリッジプレイヤーが興味をそそられるところであり，著者は，後者をコンピュータ上で実現する方法を探究している．時間の問題は，M 台のプロセッサで並列処理することにより解決する計画であり，その時点でこのような方法が強さに影響することを実証したい．

5. おわりに

上級者のプレイテクニックの 1 つとして知られるディセプティブプレイを発見するアルゴリズムを提案した．騙される敵のモデルを実装して，典型的な例題を用いて実験を行い，上級者が行うプレイを発見することができた．実験では，それぞれ数個のディールからなる D と D' を同時に評価する方法を試したが，800 MHz のコンピュータで数分かかった．プログラムの高速化とともに並列処理による時間短縮も必要と考えている．また，ディセプティブプレイには，通常のプレイより敵に多くとられるリスクをおかすタイプのものも多いので，状況によって実行すべきかどうか判断することが課題となる．実用上は，通常のアルゴリズムでのコントラクト達成（あるいは阻止）が困難な場合に限り，本アルゴリズムによりディセプティブプレイの発見を試みるのがよいと考えている．

本論文では，ディクレアラーのプレイを例として取り上げたが，付録の例題のような，ディフェンダーによるプレイも同じ原理で発見できることを確認した．今後，経験則の詳細化および敵のモデルの改良（騙されることを警戒するなど）も行いたい．

謝辞 ECLiPSe の使用を許可してくださった Imperial College PARC，論文を読んで改善のためのご意見をくださった多くの方々に感謝いたします．

参考文献

1) イアン・フランク：ブリッジ．松原 仁，竹内郁雄（編）：ゲームプログラミング，pp.96–108, 共立出版 (1997).
2) 上原貴夫：コンピュータブリッジ，人工知能学会誌，Vol.16, No.3, pp.385–392 (2001).
3) Wasserman, A.: Realization of a skillful bridge bidding program, *Fall Joint Computer Conference, AFIPS* (1970).
4) Ginsberg, M.L.: GIB: Steps toward an expert-level bridge-playing program, *IJCAI-99* (1999).
5) 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).
6) Berlekamp, E.R.: Machine solution of no trump double-dummy bridge problems, MS Thesis, Massachusetts Institute of Technology (1962).
7) Ginsberg, M.L.: Partition search, *AAAI-96* (1996).
8) Ginsberg, M.L.: How computers will play bridge, The Bridge World (1996).
9) Frank, I.: Search and planning under incomplete information: A study using Bridge card play, Springer-Verlag in the Distinguished Dissertations Series (1998).
10) 内田富夫：コンピュータブリッジ転戦記（その 1）（その 2），JCBL Bulletin, Vol.47, No.1, pp.20–25, No.2, pp.27–33 (2000).
11) Sterling, L. and Nygate, Y.: Python — An expert squeezer, *Proc. ICLP*, pp.654–674 (1987).
12) Ginsberg, M.L.: GIB: Imperfect information in a computationally challenging game, *Journal of Artificial Intelligence Research* (2001).
13) Ando, T. and Uehara, T.: Reasoning by agent in computer bridge bidding, Computer and Games 2000 (2000).
14) 上原貴夫：コンピュータブリッジにおけるアブダクションの応用，電子情報通信学会論文誌 D-II, Vol.J77-D-II, No.11, pp.2255–2264 (1994).
15) Goren, C.H.: Goren's new bridge complete, Doubleday (1985).
16) 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 幸運を呼ぶ技，日本コントラクトブリッジ連盟 (1998).
17) Kelsey, H.: The tricky game: Deceptive play to winning bridge, Devyn Press (1982).

付　　録

以下のタイプの例題について，例題 1 と同様にハンドを作ってテストし，プログラムの動作を確認した．

［例題 2］　North　A J 10 9 8 7

West　xxx　　East　K x

South　Q x

上のようなディールでディクレアラーの South は Q，West は 3 を出し，ダミーの North からは 7 が出た場合，ディフェンダーの East は何を出すべきか（ただし，コントラクトは 3 ノートランプでこのスーツが 5 トリックとれなければ達成できず，North へのエントリーはほかにはないものとし，South は 2 枚とする）．

2 節の Ginsberg アルゴリズムでは，K を出す．本論文のアルゴリズムでは，経験則 (4-6) が効果を発揮し x を出すプレイを選択した．

［例題 3］　North　A Jxx

West　K x　　East　10 9x

South　Q 8xx

上のようなハンドでディクレアラーの South がトランプの 2 を出し，West の 5 に対して J を出してフィネスをした[17]．本論文のアルゴリズムでは，South のハンドを Q 8xx と仮定した場合には，East が 9 を出すディセプティブプレイを選択した．ここでは，経験則 (4-1) が効果を発揮した．

(平成 14 年 2 月 18 日受付)
(平成 14 年 9 月 5 日採録)

小林　紀之（学生会員）

平成 10 年東京工科大学工学部情報工学科卒業．現在同大学大学院博士後期課程在学中．コンピュータブリッジの研究に従事．

上原　貴夫（正会員）

昭和 40 年早稲田大学理工学部電気通信学部卒業．昭和 45 年同大学大学院博士課程修了．同年（株）富士通研究所入社．平成 6 年東京工科大学情報工学科教授，現在に至る．その間，昭和 52 年より 1 年スタンフォード大学客員研究員．工学博士．CAD，人工知能，ソフトウェア等の研究に従事，日本ソフトウェア科学会，人工知能学会各会員．