ゲーム情報学 2－8
(2000. 3. 17)

不完全情報ゲームにおける推論とプレーのアルゴリズム

小林 紀之　安藤 剛寿　上原 貴夫
東京工科大学

あらまし 不完全情報ゲームであるブリッジにおけるヒューリスティックな戦略決定法を提案する．これは，Ginsberg が採用したモンテカルロ法を用い繰返し完全情報ゲームを解いて近似する方法ほど楽観的ではなく，Frank 等が提案するディフェンス側のみ完全情報を持つと仮定する方法ほど悲観的ではない．我々の方法では，敵の過去の行動を観察して隠されたハンドの可能性について推論し，現ラウンドにおける不完全情報に基づく行動の選択に反映させる．ただし，次ラウンドからは双方が相手のベストプレー（完全情報）を仮定して行動を選択するものとした．これにより，Frank 等の指摘した問題点に対処しつつ，完全情報ゲームの技術を応用した高速なゲーム木探索が可能となった．

Inference and Search in Games with Incomplete Information

Noriyuki Kobayashi　Takahisa Ando　Takao Uehara
Tokyo University of Technology

Abstract: This paper proposes a heuristic search algorithm in games with incomplete information. This algorithm is not so optimistic as the Monte-Carlo sampling method, and is not so pessimistic as the best defense model proposed by I. Frank. We design the architecture of computer bridge using this algorithm. Our algorithm assumes that defender has complete information but player does not have it during one round. The algorithm overcomes problem of strategy fusion at least in one round, and high-speed search technique is applicable after the one round. We also proposed a new method how to analyze defender's strategy.

1．まえがき

ゲームをコンピュータにプレーさせることは，人工知能の課題として研究が進められてきた．１９９７年には，IBM の Deep Blue［１］というプログラムがチェスの世界チャンピオンである Kasparov に勝ち，完全情報ゲームに関する研究が一段落した（碁，将棋の研究に重点が移った）．しかし，不完全情報ゲームであるブリッジでは，全ての駒の位置がわかっているチェスと異なり，他のプレイヤーの持っているカードが見えないため，完全情報ゲームで使われた一般的な方法を適用することが難しい．

［２］によれば，コンピュータによるブリッジのプレーに関する研究としては，Carley によるルールベース的アプローチが古い．これは，経験則を集めルールベースとし，これに従ってプレーするもので先読み探索は行わない．このアプローチは，プレーの決定に要する時間は短いものの，強いプレイヤーには勝てなかったといわれる．Berlekamp はダブルダミーとよばれる全員のハンドが見えた状態で，完全情報ゲームとしてプレーの手順を求めている［３］．初期の商用コ

ンピュータブリッジでは，人間はシングルダミー（自分とダミーのハンドのみ見えて，敵のハンドは見えない状態）でプレーをし，コンピュータはダブルダミーでプレーするようになっていた．Frank は，フィネス，キャッシュなどの基本的な戦術の組合せとして，プレーのプランをたてる方法を提案している［４］．シングルスーツのプランには成功したが，４つのスーツからなる実際の場合は実現していない．

過去に２度コンピュータブリッジのチャンピオンになった Bridge Baron の開発者 Throop は，自分自身が試合で試した経験則に従ってプランをたて，敵のカードの確率的分布を考慮して戦略を決めているようである． Ginsberg は，モンテカルロ法を用いて，完全情報ゲームの問題を繰返し解くことにより不完全情報ゲームの解の近似を試みた［５］．この方法は，すでに日本の内田夫妻が開発した Micro Bridge などに使われていたものである．Ginsberg は，ゲーム木探索の高速化をはかり，モンテカルロ法のサンプルとして５０以上のハンドを用いた．この結果，１９９８年，彼の GIB（Goren In Box）は Bridge Baron を破ってチャンピオンになった．Frank 等は，Best Defense Model を提案し，GIB のアルゴリズムには理論的な問題点があることを指摘している［６］［７］．

我々は，不完全情報ゲームであるブリッジにおけるヒューリスティックな戦略決定法を提案する．これは，Ginsberg が採用したモンテカルロ法を用い繰返し完全情報ゲームを解いて近似する方法ほど楽観的ではなく，Frank が提案するディフェンス側のみ完全情報を持つと仮定する方法ほど悲観的ではない．以下，第２章では不完全情報ゲームの木表現，第３章では Ginsberg のモンテカルロ法，第４章では Frank のベストディフェンスモデル，第５章では我々の提案する方法，第６章では各方法の比較について述べる．

2．不完全情報ゲーム

2.1 完全情報ゲームの探索木

チェスに代表される完全情報ゲームは，図１のようなゲーム木で表される．ゲーム木では，節点はゲームの状態，枝はプレイヤーの指手に対応する．ただし，異なる指手の系列によって得られる状態が同じでも，それらは同じ節点では表さない．ゲーム木の中で同じ深さにある節点の集合を層（ply）とよぶ．

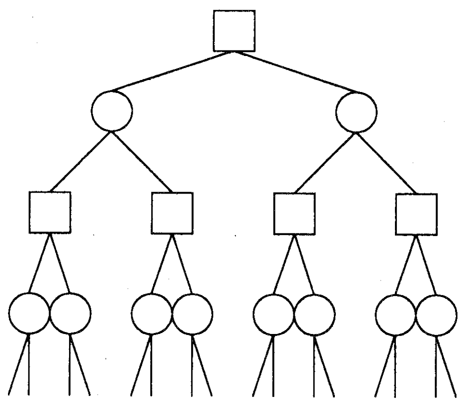


図１　完全情報ゲームの探索木

MinMax 法は，ゲーム木の何層か先まで探索して指手を選択する標準の方法である．この方法は，当方にとって良いことは相手にとって悪く，相手にとって良いことは当方にとって悪いという，零和（zero-sum）仮説に支えられている．当方の目標は，その状態の値を最大になるような指手を選ぶことであり，相手の目標はその値を最小にするような指手を選ぶことである． MinMax 法には，$\alpha\beta$ 剪定など，チェスの研究［８］を通して得られた数多くの高速化技術がある．

2.2　不完全情報ゲームの探索木

図２に，ブリッジに代表される不完全情報ゲームの木表現の一例を示す［６］［７］．葉の部分における値は，世界（可能なゲーム状態，ブリッジでは見えないハンドの状態）ごとに異なる．

世界を $W_1, W_2, ..., W_n$ とすれば，節点の値はｎ次元のベクトルで表される．ある節点における枝（指手）を $M_1, M_2, ...$ とし，M_i の先の節点ｉにおける値を $K_i=[K_{i1}, K_{i2}, ..., K_{in}]$ としたとき，スコア関数 $f(M_i)$ を定義し，スコア関数の値によって指手を選択することができる．

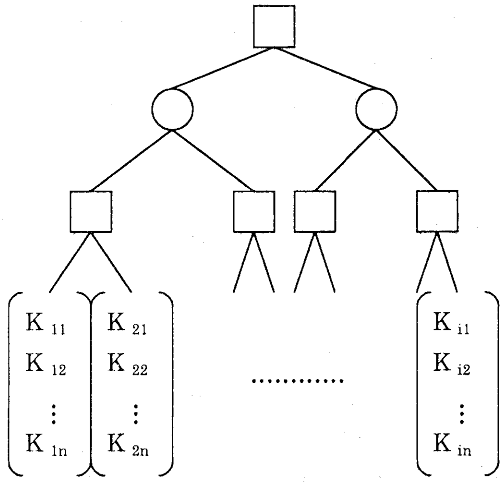


図２　不完全情報ゲームの探索

スコア関数のとしては，

$$f(M_i) = \sum_{j=1}^{n} K_{ij}\mathrm{prob}(W_j)$$

がよく使われる．

不完全情報ゲームの戦略決定方法の一つとして次のような方法が考えられる．これは，Max 側も Min 側も同じように不完全情報のもとで選択をする場合に相当する．

［方法１］(以下 VectorMin VectorMax 法とよぶ. 図５(a)参照)

Max 節点ではスコア関数の値が最大の指手，Min 節点ではスコア関数の値が最小の指手を選び，選ばれた子節点のベクトル値を親節点のベクトル値とする．

３．モンテカルロ法

Ginsberg が採用した方法は，モンテカルロ法を用い繰返し完全情報ゲームを解くことにより，不完全情報ゲームの解を近似しようと意図するものである．前章の表現を用いれば，モンテカルロ法で生成された世界を $W_1, W_2, ..., W_n$ としたとき，根節点以外では，完全情報ゲームとして扱う．ある親節点における指手を $M_1, M_2, ...$ とし，M_i の先の子節点ｉにおける値を $K_i=[K_{i1}, K_{i2}, ..., K_{in}]$ とすれば，つぎのようになる．

［方法２］(以下モンテカルロ法とよぶ. 図５(b)参照)

根節点では，Max 節点ならばスコア関数の値が最大の指手，Min 節点ならばスコア関数の値が最小の指手をぶ．

根節点以外では，

1）親節点が Max 節点の場合，親のベクトル値は

$[\max K_{i1}, \max K_{i2}, \ldots, \max K_{in}]$.

2）親節点が Min 節点の場合，親のベクトル値は

$[\min K_{i1}, \min K_{i2}, \ldots, \min K_{in}]$.

4. Frank の Best Defense Model

Frank 等は，ブリッジの本でエキスパートが問題を解析する際の仮定を次のようにまとめ，Best Defense Model を提案した．

仮定１： Min は完全情報をもつ．

仮定２： Min は Max の後で戦略を決める．

仮定３： Max の戦略は pure strategy である．

彼等は，Best Defense Model のもとでの最適戦略を求める問題は，NP－complete であることを示した．

また，Ginsberg が採用したモンテカルロ法（方法２）には，次のような問題点があることを指摘した．

1）strategy fusion

根節点以外での指手選択において，不完全情報しかもたないはずの Max が，世界ごとに異なる選択をすることが起きる．

2）non-locality

最適な戦略はゲーム木の局所的な情報だけでは決定できない場合があるが，それが考慮されていない．

3）delay

着手を遅らせた方が良い指し手を今指してしまう可能性がある（原因は strategy fusion）

そして，これらの問題に対処するため数種のヒューリスティックなアルゴリズムを提案している［６］．

［方法３］（以下 Vector Max 法とよぶ．［６］では Vector minmaxing と名づけているが，本論文では方法１との区別を示すため別名にした．図５(c)参照）

Max 節点ではスコア関数の値が最大の指手を選び，選ばれた子節点のベクトル値を親節点のベクトル値とする．親節点が Min 節点の場合，子節点 i における値を $K_i=[K_{i1}, K_{i2}, \ldots, K_{in}]$ とすれば，親のベクトル値は

$[\min K_{i1}, \min K_{i2}, \ldots, \min K_{in}]$.

すなわち，Min は世界ごとに最小値を選ぶ．

［方法４］（以下 Payoff-reduction 法とよぶ．図５(d)参照）

1）通常の MinMax 法を各世界 W_i について行い，各 Min 節点における値 m_i を求めておく．

2）各世界 W_i に対応する葉節点の値 p_i を，p_i とその葉節点の先祖にあたる全ての Min 節点

における値 m_i の最小値に書き直す．

3） 以上の結果できた木に Vector Max 法を適用する．

その他，β-reduction 法，Iterative Biasing 法を提案し，これらのアルゴリズムの組合せについても試している．

5．我々の提案する方法

我々の方法では，敵(Min)の過去の行動を観察して隠されたハンドの可能性について推論し，現ラウンドにおける不完全情報に基づく行動の選択に反映させる．ただし，次ラウンドからは双方が相手のベストプレー（完全情報）を仮定して行動を選択するものとした．これにより，Frank の指摘した問題点に対処しつつ，完全情報ゲームの技術を応用した高速なゲーム木探索が可能となった．

［方法５］（以下では One-round Vector Max 法とよぶ．図５(e)参照）

1） 次に選択すべき指手を含む１ラウンドは Vector Max 法，次のラウンド以降は通常の MinMax 法を適用する．

1） Min が実際に指手 M_i を選択した場合には，１）の探索木から M_i の親節点（根）での世界 W_j における値を B_j，M_i の子節点（現在の状態）での世界 W_j における値を K_j とし，$K_j < B_j$ ならば世界 W_j は存在しないという仮説をたてる．

2） Max が Vector Max 法で指手を選択する場合，２）の仮説のもとで行う．

（プログラムの実装に際しては，スコア関数の計算において，その世界の確率を極めて低くすることにより実現している）．

［例題］図３において West がハート× を出したとき，North は４回とることが目標の場合ならば何を出すべきか．

繰返し MinMax 法によれば，表１に示すように Q を出せば 25/55，A を出せば 10/55 の確率で４回とれるので，ハート Q を選択する．

North
S　AK
H　AQ

East, West の持っているカード
S　×××××
H　K××

South
S　××
H　××

図３　ハンドの例
West が H×を出したとき North は何を出すべきか

表１　ハンドの分布および確率

	West	East	確率	Q で取れるか	A で取れるか
W1	S：×　H：K××	S：××××	５／５５	○	
W2	S：××　H：K×	S：×××　H：×	２０／５５	○	
W3	S：××　H：××	S：×××　H：K	１０／５５		○
W4	S：×××　H：×	S：××　H：K×	２０／５５		

しかし，本当にハートのＱでいいのだろうか．なぜなら，この決定には，相手がなぜハートの×を出したのかが考えられていないからである．

そこで我々が提案するのは，なぜ West がハートの×を出したか分析し，可能性の低い世界を特定し，可能性の高い世界に重点をおいて戦略を決定する方法である．

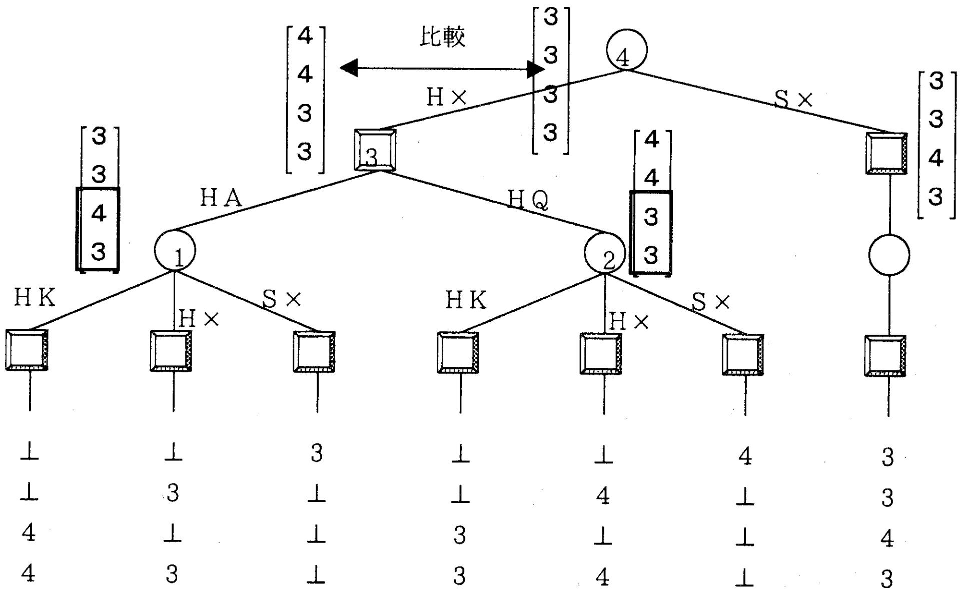


図４　例題の One Round Vector Max 法による解

図４のような例のとき，我々の提案の通りに解いていくとどのように評価されていくかを以下に示す．

（１）まず，節１の評価値を求める．このとき，節１は敵（Min）なので，ベストディフェンスをする．つまり，各世界で最もよい値を取るので，この節の値は［３３４３］となる．

（２）次に節２の評価値を求める．この節も敵（Min）なので，ベストディフェンスをし，［４４３３］という値になる．

（３）節３の評価値を求める．ここは味方（Max）なので，より多い世界でトリックが達成できる節２を選択し，［４４３３］という値になる．

（４）節４の評価を求める．ここは敵（Min）なので，［３３３３］という値になる．

（５）ここで，節４の値と節３の値を比較する．すると，W１と W２の値がハートの×を選ぶことによって上がっている（敵にとって不利になっている）ので，これら世界である可能性が低いと考えられるので，これらの世界を削除する．

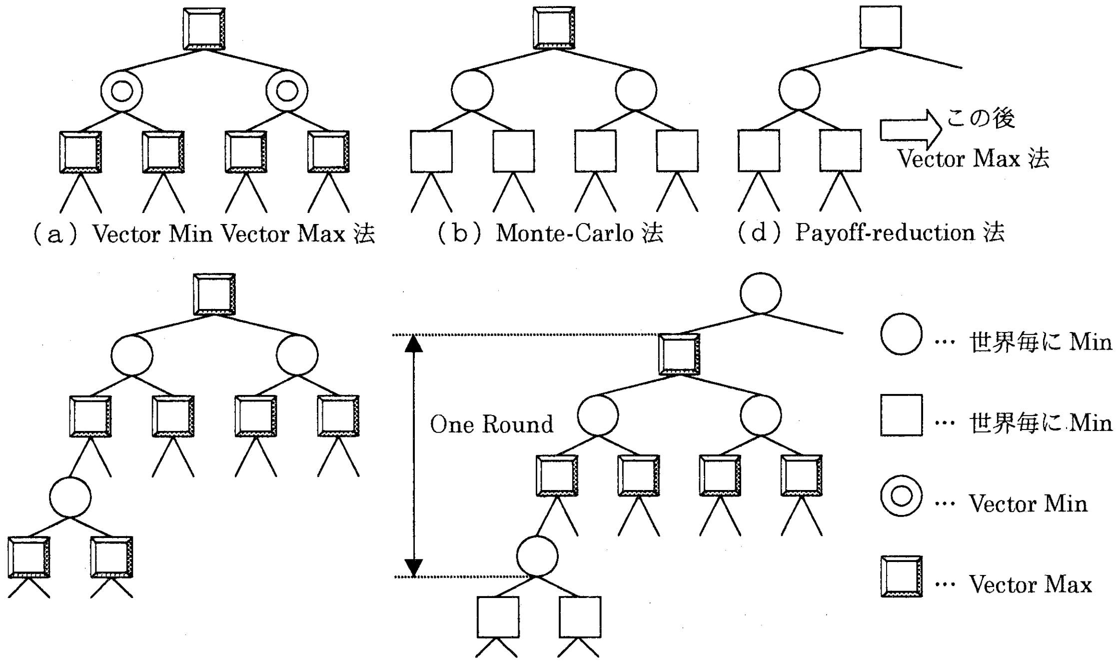
（６）次に節１と節２のスコア関数の値を求める．この結果，

　　ｆ（ＨＡ）＝４＋３＝７

　　ｆ（ＨＱ）＝３＋３＝６

　　となり，ハートのＡを出すほうがスコア関数の値が高くなる．

よって，ハートＡを選択することが正しいという結論に達する．



（a） Vector Min Vector Max 法　（b） Monte-Carlo 法　（d） Payoff-reduction 法

（c） Vector Max 法　（e） One-round Vector Max 法

図５　各方法の概略

6. 各方法の比較

図５に各方法の Max，Min の指手選択方法の概略を図示した．

方法１（VectorMin VectorMax 法）は，双方にとって常識的な選択であり，初心者どうしのプレーに近いと思われるが，記載した文献は見当たらない．方法２と比較して高速化が困難である．

方法２（モンテカルロ法）は，次に出す一枚のカードをのぞいて，双方が完全情報ゲームを行うと仮定している．そこで，$\alpha\beta$ 剪定など完全情報ゲームで用いられた高速化手法が利用可能である．しかし，Frank 等によって指摘された Strategy Fusion 等の問題点があり，楽観的すぎる解がでる可能性がある．特に，次にパートナーのハンドから出すカードを敵のハンドを覗いて決めることができる設定になっているのは問題である．

方法３（Vector Max 法）は，Best Defense Model の近似解を求める方法であり，Max 側の Strategy Fusion の問題を解決している．しかし，Non-locality の問題は残る．モンテカルロ法ほとの高速化は望めない．

方法４（Payoff-reduction 法）は Non-locality の問題をなくす試みの一つである．Vector Max 法より正しい解を出すが，準備に MinMax 法を解く必要があるため，より時間がかかる．

β-reduction 法など高速化が試みられており，幾つかの手法の組合せによりより正確な解がもとめられている［６］．しかし，モンテカルロ法に匹敵する高速化は困難と思われる．

我々の提案する方法５（One-round Vector Max 法）は，モンテカルロ法の高速性を保ったまま，その問題点を実用レベルで解決する試みである．

7． むすび

不完全情報ゲームにおいて強力な敵と対戦する場合のヒューリスティックな意志決定アルゴリズムを提案した．この方法では，モンテカルロ法では正しい答えが出ない場合でも，正しい答えをだすことが出来る．１ラウンドの間の不完全情報を仮定したことにより，モンテカルロ法では困難であった決定を遅らせて情報獲得を待つ効果（嫌な決定を遅らせる水平線効果の有効利用）も出た．２ラウンド以降はモンテカルロ法とにしたために，従来の高速化の技法が利用できる．

今後の課題として，まずは高速プログラムの実装があげられる．現在のプログラムは，あまり高速化について突き詰めた作りをしていないので，高速化を考えた改良をしていく必要がある．また，処理の並列化も考えている．次に，世界の生成方法をあげることができる．今は，ただ確率から生成しているだけであるが，今後はやさしいプレーにおいて安全性を求めるためにもプレーの難度から生成方法を変えることを考えている．

参考文献

[1] 松原仁：1997年IBM Chess Rematch 報告，pp．39－43，ｹﾞｰﾑﾌﾟﾛｸﾞﾗﾐﾝｸﾞ，共立出版（1997）．

[2] Ian Frank: ブリッジ，pp．96－108，ｹﾞｰﾑﾌﾟﾛｸﾞﾗﾐﾝｸﾞ，共立出版（1997）．

[3] E. R. Berlekamp: Program for double-dummy bridge problems-a nem strategy for mechanical game playing, Jounal of the ACM, 10(4): 357-364, 1963.

[4] Ian Frank: Search and Planning Under Incomplete Information, A study using Bridge Card Play, Ph. D thesis, Department of Artificial Intelligence, Edinburgh, 1996

[5] Matthew L. Ginsberg: GIB: Step Toward an Expert-Level Bridge-Playing Program, http://www.cirl.uoregon.edu/~ginsberg/

[6] Ian Frank, David Basin: Optimal Play Against Best Defence: Complexity and Heuristics, First International Conference on Computers and Games, 1998

[7] Ian Frank, David Basin: Search in games with incomplete information: A case study using bridge card play, Artificial Intelligence, 100(1-2):pp.87-123,1998.

[8] D.リービ，M.ニューボーン著：コンピュータチェス　世界チャンピオンへの挑戦，サイエンス社（1994）

[9] 上原　貴夫，安藤　剛寿，関谷　好之，小林　紀之：コンピュータブリッジのアーキテクチャ，Proceedings of 4th Game Programming Workshop，箱根，pp.179-186（1997）

[10] 小林紀之，安藤剛寿，関谷好之，上原貴夫：コンピュータブリッジのプレーの研究，電子情報通信学会，情報システムソサイエティ大会，D-8-4，p.75（1998）

[11] 小林 紀之，安藤 剛寿，上原 貴夫：コンピュータブリッジにおけるプレーのプランニング，電子情報通信学会総合大会，D-8-8， p.168（1999）

[12] 小林紀之，安藤剛寿，上原貴夫：不完全情報ゲームにおける仮説生成，情報処理学会，第59回全国大会，2J-6，p.61(1999)