

コンピュータブリッジにおけるマルチモードプレイ

小林紀之 山下桂治 上原貴夫

東京工科大学

あらまし

ブリッジのエキスパートは、難しい場合は楽観的に考えて大胆にプレイし、易しい場合は悲観的に考え安全にプレイをするよう教えている。現在のコンピュータブリッジには、問題の難易に従って、プレイのモードを切替える機能はない。本論文では、マルチモードプレイの実装方法を提案する。

Multi-mode Play for Computer Bridge

Noriyuki Kobayashi, Keiji Yamashita, Takao Uehara
Tokyo University of Technology

Abstract: An expert of the contract bridge selects the optimistic play if the contract is difficult to achieve, and he selects the safety play if the contract seems to be easy. No existing computer bridge programs have the function to switch the playing mode according to the difficulty of the contract. This paper proposes a method to implement the multi-mode play.

1. はじめに

現在のコンピュータブリッジにおけるプレイのアルゴリズムは、モンテカルロシミュレーションが主流である。見えていないカードの可能な分布(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良さそうな行動を選ぶ方法である。内田夫妻の Micro Bridge[内田 00], Ginsberg の GIB[Ginsberg 99]もこの方法を用いている。

Ginsberg は、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)を決定している。

- 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くばり、ディールの集合 D を作る。
- 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。

3) $s(m,d)$ が最大となるような行動 m を選ぶ。

ディールの集合の生成は，モンテカルロシミュレーションの原理によれば，ランダムであることが望ましいとされている。

ブリッジのエキスパートは，難しい場合は楽観的に考えて大胆にプレイし，易しい場合は悲観的に考え安全にプレイをするよう教えている．現在のコンピュータブリッジには，コントラクトの難易に従って，プレイのモードを切替える機能はない．本論文では，コントラクトの難易に従って，生成するディールの集合を恣意的に変えることにより，状況に応じたマルチモードプレイを可能にする方法を提案する．

一般のゲームにおいて，相手に応じて最善手が変わるプログラムについては，飯田によって論じられている[飯田 97]．[小林他 00]では，ブリッジに関して，上記のGinsberg[Ginsberg 99]のアルゴリズムは楽観的であり，Frank[Frank and Basin 98]のベストディフェンスモデルは悲観的であると評価し，その中間に位置付けられる方法を提案している．これらのアルゴリズムをコントラクトの難易に応じて切替える方法も考えられる．例えば，易しい場合は，敵がベストディフェンスをすとして指手を選択する．敵がベストディフェンスをすれば難しいコントラクトでは，敵がまちがえる可能性もあるというモデルを用いて指手を選択する．

上記の2つのタイプのモード切替を組み合わせることにより，エキスパートに近い機能を実現できると考えられるが，本論文では恣意的なディールの生成を中心に述べ，相手モデルの切替について別途報告する．

2. 不完全情報ゲームの探索木

図1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[Frank and Basin 98]．葉の部分における値は，世界(可能なゲームの状態，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, \dots, 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, \dots とし， M_i の先の節点 i における値を $K_{i1}, K_{i2}, \dots, K_{in}$ 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

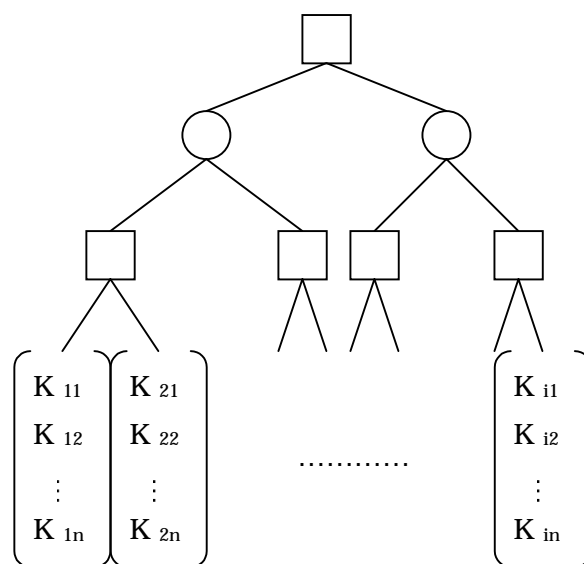


図1 不完全情報ゲームの探索

選択することができる .

スコア関数の例としては,

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

がよく使われる .

ブリッジの場合にはコントラクトが達成できたかどうかでスコアが大きく異なるので, K_{ij} を獲得したトリック数としコントラクト達成に必要なトリックを C としたとき, つぎのようなスコア関数の例も考えられる .

$$f(M_i) = \sum_{j=1}^n g(K_{ij}, C) \text{prob}(W_j)$$

ただし, $g(K_{ij}, C)$ は $K_{ij} < C$ ならば 0 , そうでなければ 1 .

本論文では, 議論を簡単にするために, 各世界の確率を等しいとみなしつぎのスコア関数を用いた .

$$f(M_i) = 1/n \sum_{j=1}^n g(K_{ij}, C)$$

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)

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 .

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 . モンテカルロ法で生成された世界を W_1, W_2, \dots, 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, \dots とし, M_i の先の子節点 i における値を $K_i = [K_{i1}, K_{i2}, \dots, K_{in}]$ 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)

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えらぶ .

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

$$[\max_i K_{i1}, \max_i K_{i2}, \dots, \max_i K_{in}] .$$

i i i

2) 親節点が Min 節点の場合, 親のベクトル値は

$$[\min_{i} K_{i1}, \min_{i} K_{i2}, \dots, \min_{i} K_{in}] .$$

その他, [小林他 00] では, いくつかのモデルについて述べているが, 以下ではモデルに依存しないように説明を進める.

3. 難易度の指標

与えられた状況で, コントラクト達成が易しいか難しいか判断する方法を提案する.

- 1) それまでのビッドおよびプレイと矛盾しない, ランダムに生成した N 個のディールの集合 D を作る.
- 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評価してコントラクトの達成率 $mc(m,d)$ を計算する. ただし, 達成できる場合 $mc(m,d) = 1$, 達成できない場合 $mc(m,d) = 0$ とする.
- 3) $\sum_d mc(m,d) / N$ が最大となるような行動 m^* を選ぶ.

$\sum_d mc(m^*,d) / N$ を難易度の指標に用いる. 以下では, その値が 1 の場合に「易しい」と言い, 値が 0 の場合に「難しい」と言う.

4. 楽観的プレイ

つぎに示す例題は, コントラクトを達成できる確率が低く, ランダムに N 個のディールを生成した場合, 難易度の指標 $\sum_d mc(m^*,d) / N$ が 0 になり「難しい」と判定される可能性が高い (ディール生成の実験結果については後に示す).

[例 1] [Lawrence 86] より引用
East が 1 Club とオープニングビッドし, South が 1 Spade とオーバーコールした後, East と West にビッドはなく, North と South で 4 Spades のコントラクトに達した. クラブを 2 つ取られた後, East からダイヤモンドが出てきた. どのようにプレイしたらよいだろう

	North	
	S A 6 4	
	H K 8 7 6 5 3	
	D 8 3	
	C Q 5	
West		East
S 5 3 2		S K
H A 10 9		H Q J
D 9 6 5 4		D J 10 7 2
C 10 9 7		C A K J 6 4 3
	South	
	S Q J 10 9 8 7	
	H 4 2	
	D A K Q	
	C 8 2	

か .Lawrence の本では ,スペードをフィネスせずに A で取ってコントラクトを作るプレイの考え方 (card play by assumption または wishful thinking とよんでいる)が述べてある . 簡単にいえば , あるカードが特定の場所になればコントラクトができないならば , そこにあると仮定してプレイするという方法である . この例では , West にハート A がないとコントラクト達成は難しく , West にハート A があると仮定するとスペード K は East にあると推論でき , スペード K が 1 枚でいると仮定しないと 4 Spades はできない .

5 . 安全なプレイ

つぎに示す例題は , コントラクトを達成できる確率が高く , ランダムに N 個のディールを生成した場合 , 難易度の指標 $\mu_{mc}(m^*,d)/N$ が 1 になり「易しい」と判定される可能性が高い (ディール生成の実験結果については後に示す) .

[例 2] [Goren 85]より引用

コントラクトは 3 No Trump で West はハートの 2 をリードした . どのようにプレイしたら良いか .

エキスパートは , ダイヤモンドがアンバランスでもできる方法を検討し , 安全なプレイの手順を発見することができる .

North
S 8 6
H A
D A K Q 10 9 5
C 9 7 5 4

South
S A Q 5 4 3
H K 8 7 3
D 6 3
C A Q

6 . マルチモードプレイ

具体的に状況によりディールの生成を切替える方法について述べる .

(1) 難しい場合 ($mc=0$)

易しいハンド (ディール) を生成する .

例えばトランプのバランス , ロングスーツのバランス , フィネスの成功など .

目安として , $\mu_{mc}(m^*,d)/N > 0$ となるまで生成を続ける .

例 1 では , ハートのフィネスと , スペードのフィネスあるいはドロップの両方の条件を満たす場合を生成するため , つぎの制約条件をヒューリスティックなアルゴリズムによって作成した後 , 制約条件を満たすディールを生成する .

`has(west,ha),(has(west,sk);has(east,sk),ENS#=1)`

ただし , 制約論理プログラム言語 Eclipse を用いて実装しており , ENS は East の Spade の枚数をあらわし , has(Player,Card) は , Player が Card (例えば ha,sk はそれぞれ Hart の Ace, Spade の King) を持っていることを表す (# は Eclipse 特有の記法であるが , これを無視して読めば , 通常の意味が理解できる) .

(2) 易しい場合 (mc=1)

難しいハンド(ディール)を生成する.

例えばトランプのアンバランス, ロングスーツのアンバランス, フィネスの不成功など.

目安として, $\sum mc(m^*, d) / N < 1$ となるまで生成を続ける.

例2では, 敵の5枚のダイヤモンドが4-1あるいは5-0, スペードとクラブのフィネスは成功しない場合を生成するため, つぎの制約条件をヒューリスティックなアルゴリズムによって作成した後, 制約条件を満たすディールを生成する.

(END# ≥ 4 ; WND# ≥ 4), has(west, sk), has(west, ck)

ただし, END, WND は, それぞれ East および West の Diamond の枚数を表し, ck は Club の King を示す.

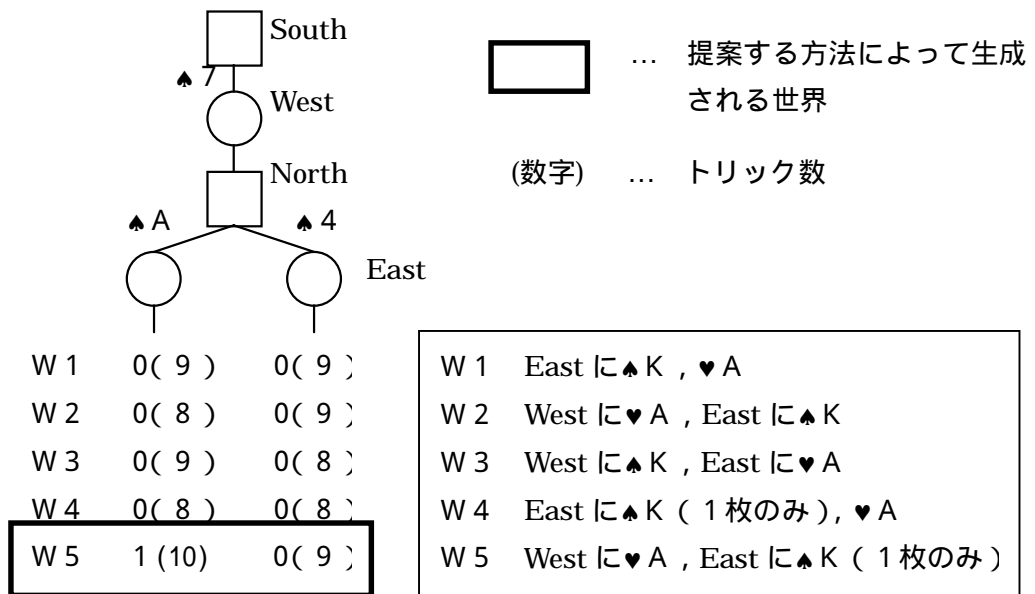


図2 例1における探索木

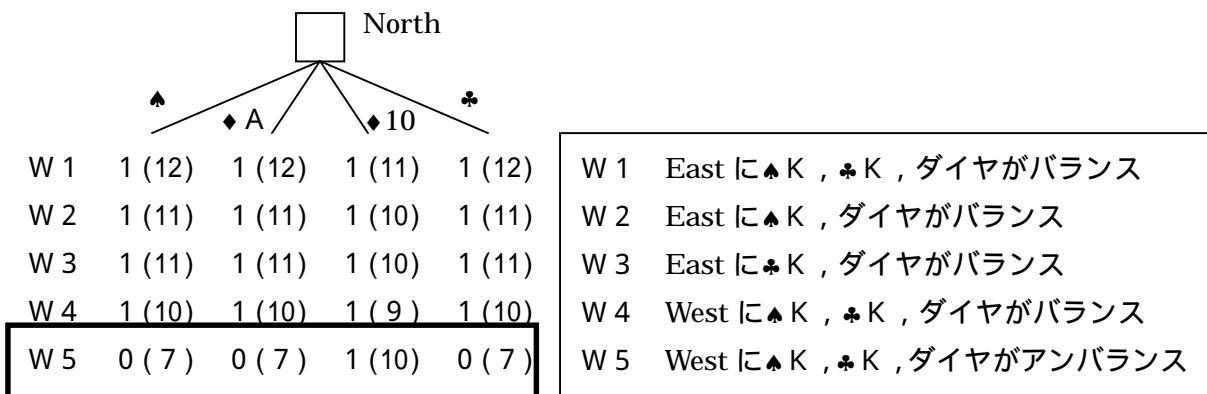


図3 例2における探索木

7. 実験

先に報告した制約条件を満たすハンド(ディール)を高速に生成する方法を用い実験を行った [安藤他 97] [安藤他 99].

例1のハンドにおいてビッドとプレイに矛盾しないという条件のみで2000回のハンド生成を行った結果, East に♥A、West に♠K が一枚という Deal は71回生成された.

例2のハンドにおいてビッドとプレイに矛盾しないという条件のみで2000回のハンド生成を行った結果、East または West にダイヤが4枚以上で♠K と♣K がどちらも West にある、という Deal は149回生成された.

これらの結果から、この例1、例2では、生成するディール数を制限していると、コントラクトを達成するために、あるいはコントラクトを落とさないために気付く必要のある世界を生成することは難しい事がわかる。よって、本論文で提案する方法が有効であるということが出来る。

North
S S64
H K87653
D 83
C Q5

West		East
S xxx		S K
H AJ		H Q10x
D Jxxxx		D 10xx
C xxx		C AKJ10xx

South
S QJ10987
H 42
D AKQ
C 82

例1において生成されたハンドの例

North
S 86
H A
D AKQ1095
C 9754

West		East
S K10x		S Jxx
H xxxx		H QJ10x
D x		D Jxxx
C KJxxx		C 10x

South
S AQ543
H K873
D 63
C AQ

例2において生成されたハンドの例

8 . おわりに

難しい場合は楽観的に考えて大胆にプレイし，易しい場合は悲観的に考え安全にプレイをするよう，プレイのモードを切替える機能の実装方法を提案した．従来のモンテカルロシミュレーションによる方法ではランダムにディールを生成していたが，本提案では恣意的にディールを生成する．すなわち，コントラクトが難しければ楽観的なディール，易しければ悲観的なディールを生成する．これにより，難しくてもあきらめず，易しくても油断しないプログラムが実現できる．生成するディールの難易度を制御するヒューリスティックなアルゴリズムを完成し，エキスパートに近づけることが今後の課題である．

参考文献

- [安藤他 97]安藤 剛寿，関谷 好之，上原 貴夫：コンピュータブリッジにおけるハンドの生成，電子情報通信学会，情報・システムソサイエティ大会，D-8-6，p.69（1997）
- [安藤他 99]安藤 剛寿，小林 紀之，上原 貴夫：コンピュータブリッジにおけるハンド生成条件の表現，電子情報通信学会総合大会，D-8-7，p.167（1999）
- [飯田 97] 飯田弘之：あなたにも作れるゲームプログラム 3目並べを題材として，bit別冊「ゲームプログラミング」，共立出版，pp.12-24（1997）.
- [内田 00] 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）.
- [小林他 00] 小林紀之，安藤剛寿，上原貴夫：不完全情報ゲームにおける推論とプレイのアルゴリズム，情報処理学会，ゲーム情報学研究会，2-8, pp.55-62（2000）.
- [Frank and Basin 98] I. Frank and D. Basin: Optimal play against best defence – Complexity and Heuristics, Computers and Games 98, Lecture Notes in Computer Science 1558, pp.50-73（1998）.
- [Goren 85] C. H. Goren: Goren's new bridge complete, Doubleday（1985）.
- [Ginsberg 99] M. L. Ginsberg: GIB: Steps toward an expert-level bridge-playing program, IJCAI-99（1999）.
- [Lawrence 86] M. Lawrence: How to read your opponents' cards, Devyn Press（1986）.