

# 不完全情報ゲームにおける並列ゲーム木探索 コンピュータブリッジの高速化

小田和 友仁 野島 健一 小林 紀之 上原 貴夫  
東京工科大学 工学部 情報工学科

## 1. はじめに

チェスや将棋など、相手の手(駒)を含めてすべての情報が取得できるゲームを完全情報ゲームという。一方、トランプゲームのブリッジなど、相手の手が隠されていて情報として取得できないゲームを不完全情報ゲームという。

近年のゲーム研究は、完全情報ゲームから不完全情報ゲームへと移行してきている。不完全情報ゲームでは推測された相手の手札について、さまざまな場合での先読みが必要とされるため、探索量は完全情報ゲームに比べ格段に多い。そのため、単一マシンでの処理には限界がある。

## 2. 目的

不完全情報ゲームであるブリッジのゲーム木探索を、分散し並列に行うことで高速化を図る。

## 3. 完全情報ゲームの並列ゲーム木探索

完全情報ゲームのゲーム木を単純に分散処理するならば、図1のようになる。このゲーム木に法やトランスポジションテーブルを適用しようとした場合、ネットワークを通して共通のメモリ空間あるいは頻繁な通信が必要となる。

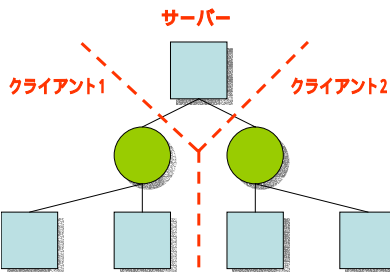


図1 完全情報ゲームの並列ゲーム木探索

Parallel search of incomplete information game tree  
- Improvement in the speed of a computer bridge -  
Tomohito Otawa, Kennichi Nojima, Noriyuki Kobayashi,  
Takao Uehara - Tokyo University of Technology

## 4. 不完全情報ゲームの並列ゲーム木探索

不完全情報ゲームでは、全ての情報が与えられていないため、単純にはゲーム木を探索することができない。そこで、与えられている情報から推測される、いくつかの可能な世界(ディール)を生成する。それぞれの世界においては、仮の情報とはいえ完全な情報がそろったと考えゲーム木を探索することができる。

このように、不完全情報ゲームのモデルは、いくつかの世界と、その世界ごとに一つのゲーム木を持つ形をとる。(図2)

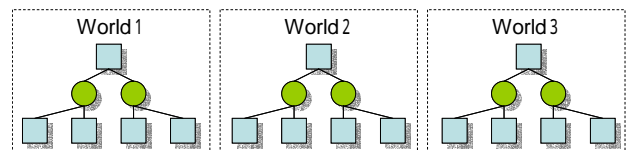


図2 不完全情報ゲームのモデル

このモデルにはモンテカルロシミュレーション適用することができる。現在の可視情報から不可視情報を推測することにより、多数の実現可能な世界をランダムに生成する。それぞれの世界で完全情報ゲームとして探索し、最終的にどの手を採用するかは不完全情報ゲームとして選択する。(図3)

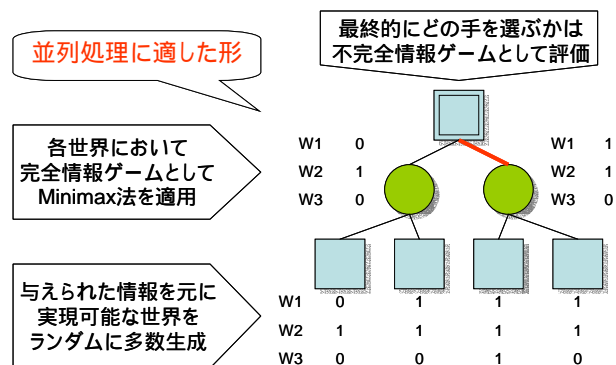


図3 Ginsbergのアルゴリズム

モンテカルロシミュレーションをもちいると、複数の完全情報ゲーム木の探索に分けることができる。各世界をクライアントに分担させることで、クライアントは完全情報ゲームとして探索できるため、法やトランスポジションテーブルも独立して適用することが可能となる。

## 5. 実現方法

クライアントはあらかじめプログラムを実行し、受信待ち状態にしておく。サーバーは得られる情報を元に、実現可能な世界をランダムに生成し、現在の局面と共に各クライアントに送信する。すべて送信したのち、サーバーは結果の受け取り待ちに移る。クライアントは世界と現在の局面を受け取ると、それを元に完全情報ゲームとしてゲーム木探索をおこない、結果を送信する。サーバーは返された結果を受信し、すべてのクライアントから結果を受け取ったのち、集計し次の一手を決定する。(図4)

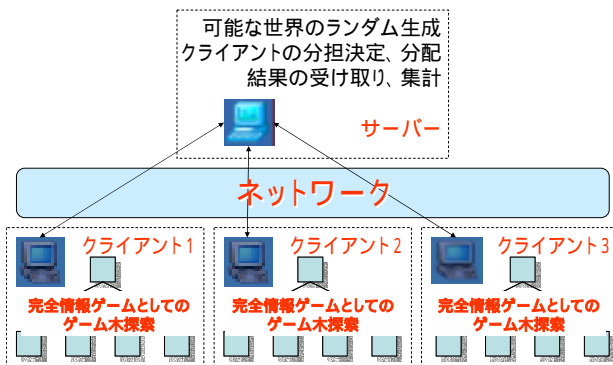


図4 不完全情報ゲームの並列ゲーム木探索

## 6. 実験結果

表1 平均処理時間(秒)

		クライアント台数				
		1	2	4		
探索の深さ	8	ワールド数	4	1.3	1.0	0.8
			12	3.9	2.5	1.5
	16	ワールド数	4	61.1	46.4	39.1
			12	328.6	184.4	119.1

表1は1トリックの探索を、条件を変えてそれぞれ5回測定した処理時間の平均値である。探索の深さ、ワールド数に関わらず、クライアントによる台数効果は適切に得られている。クライアントを2倍4倍と増やしても処理時間が1/2, 1/4とならないのはサーバーでの集計時間や、通信時間が余計にかかるためである。

探索深度の増加は鼠算式に探索量を増やすのに対し、ワールド数の増加では探索量は(ワールド数がクライアント台数で割り切れるのであれば)ほぼ比例して増えると考えられる。しかし、表1では増加に偏りが見受けられる。ワールド数増加による影響はクライアント数が少ないほど増加する。さらに、探索がより深いほどその傾向は顕著に現れる。これは、並列処理にすることで、ワールドごとの処理時間のばらつきが緩和されていると考えられる。

## 7. 使用言語

言語には論理制約言語であるECLiPSeをもちいている。ECLiPSeはSocket機能も提供しているので、サーバー・クライアント間の通信にはこれを利用している。

現在さらなる高速化のため、C言語への移行を開始している。

## 8. 今後の計画

### a. サーバー側での不完全情報ゲーム向処理

最初の1枚だけの選択では問題(Two Way Finess)があるので、少なくとも1トリック(4枚)まで処理する。

### b. クライアント側での不完全情報ゲーム向処理

エージェントモデルに基づくアルゴリズム(自己の推論した世界と他者の推論した世界をペアにして探索)を実装する。

### c. 数100台規模の実験

上級者のプレイと比較評価する。

## 9. まとめ

従来、完全情報ゲームの分野で研究されてきた探索の手法の特長を生かしながら並列処理の台数効果も上げることのできる不完全情報ゲームの並列ゲーム木探索の実装法を立案した。