

# PC クラスタを用いた HEX における最善応手手順の生成

東京農工大学 \*中川誠一 NAKAGAWA Seiichi

01207140 東京農工大学 品野勇治 SHINANO Yuji

## 1. はじめに

一般に組み合わせゲームと呼ばれるものは、探索空間の全探索によってその最善応手手順（必勝手順）を求めることが出来る。しかし、多くのゲームでは探索空間が巨大であり、最善応手手順の探索に長い時間を必要とする。

本研究では HEX という組み合わせゲームを対象とし、PC クラスタを用いた並列化を用いることで、巨大な探索空間をより高速に探索する手法を考案する。

## 2. HEX [1]

HEX とは、図 1 のようなひし形の盤面上で行われるゲームである。競技者は白黒二人で、互いに盤上に石を置き合い、自らの陣地を繋いだものが勝利となる（図 1 は白の勝利盤面）。また、HEX には引き分けがなく、先手の白に必ず最善応手手順が存在する。

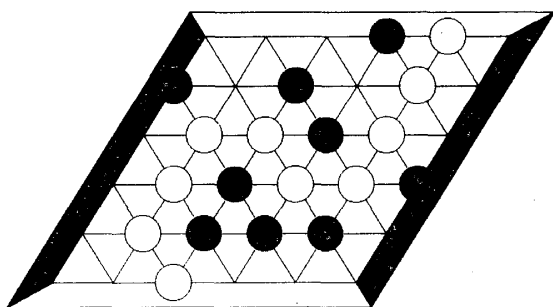


図 1 HEX の盤面

### 2.1 先手必勝性の証明の概略 [2]

HEX の先手必勝性は次のように「戦略盗み」と呼ばれる手法を用いて証明できる。

まず、後手の黒に必勝戦略が存在し、後手が必勝であると仮定する。

これに対し先手である白は、まず第 1 手目を任意のマスに置き、そのことを忘れてしまう。そして以後、先手が黒であると考え、自分は後手の必勝手順に従って石を置き続ける。

このとき、必勝手順に従っている白が、「忘れ

てしまった」第 1 手目のマスに石を置かなければならないことがある。しかしそのことは問題にならない。なぜならそのマスには既に白石が置かれているのだから、白は必勝手順に従ってそのマスに白石を置くつもりで、他の任意の場所に石を置く。そして新たに置いたその手をまた「忘れてしまった」ことにする。

このようにゲームを進めていくと、必勝手順に従って打ち続ける白が必然的に勝利することとなる。しかしこれは後手の黒が必勝であるという仮定と矛盾する。

よって、背理法によって仮定が誤っていることとなり、先手の白が必勝であるということが示される。つまり、このゲームは先手必勝である。

### 2.2 HEX における解の探索

HEX の探索空間は図 2 のように、ひとつも石が置かれていない盤面を根とした木で表される。但し、図中の白丸は白手、黒丸は黒手を表す。

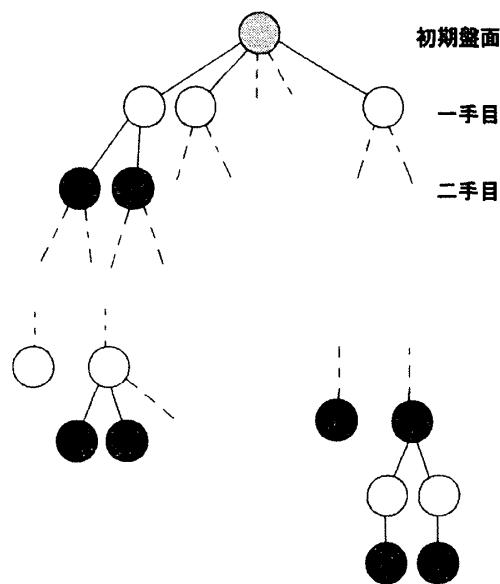


図 2 探索空間

先手の最善応手手順は、この木を全探索することによって発見することができる部分木として

得られる。探索空間が巨大であること、及び並列化の容易さから、探索には深さ優先探索が適している。

### 2.3 枝刈り

巨大な探索空間を少しでも減少させるために、本研究では「最善応手の発見による枝刈り」「後手の勝利による枝刈り」「飛び石による勝敗判定の緩和」「同一盤面の削除」という手法を用い、不要な探索空間の削減を行っている。

## 3. 並列化

本研究では PVM (Parallel Virtual Machine) を用いて、PC クラスタによる探索空間の並列探索を行う。探索空間の部分空間を独立して探索する PVM のタスクを探索タスクと呼ぶ。つまり、探索タスクが CPU 数以下であれば、探索タスク数分並列に探索空間が探索される。

### 3.1 タスクによる並列化 [3]

問題の性質上、ある部分探索空間の結果が他の部分探索空間の結果に依存するといった状況が生じる。この場合、一部の探索タスクが他の探索タスクの結果を待つことになり、計算機の処理に無駄が生じる。

この問題を回避し、より効率よく CPU 資源を利用するために、「待ち状態」という概念を導入する。この「待ち状態」になった探索タスクは、次に実行権が与えられるまで動作を停止する。また、これら探索タスクの状態を管理するためのタスク（管理タスク）をひとつ用意し、実行権の付与や新しい探索タスクの作成権などを処理する。

実行権が与えられるタスクは 1 計算機あたり 1 探索タスクに制限されているが、「待ち状態」にある探索タスク数は動的に変化する。

### 3.2 並列ハッシュ

枝刈りのひとつである「同一盤面の削除」はハッシュを用いて実現されている。このハッシュ空間をより大きなものとし、また処理の効率化を行うべく、PC クラスタ内の計算機間をまたがるひとつの巨大なハッシュ（並列ハッシュ）テーブルを構築する。

## 4. 実験結果

実験は 21 台までの計算機を用いて行った。並列化による速度向上を確認するために、以下の概

念を採用する。

$$speedup = \frac{T(1)}{T(n)}$$

ここで、 $T(n)$  は  $n$  台の計算機における実行時間を表す。つまりこの式は、 $n$  台の計算機を利用することにより、処理速度が何倍向上したかを表している。

PC クラスタに用いた計算機数と speedup の関係を表したものが図 3 である。

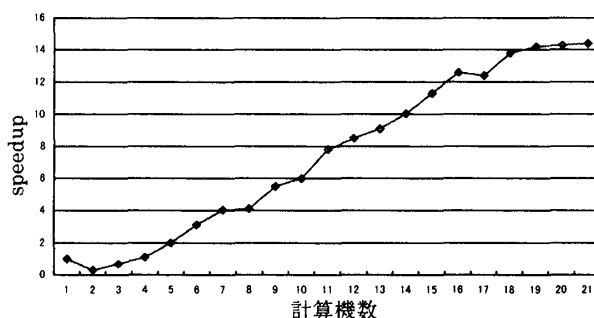


図 3 計算機数と speedup

計算機数が少ないうちは並列化特有の処理によって速度低下が生じているが、計算機数が増えることにより、ほぼ単調な速度向上傾向が見られる。

## 5. おわりに

本研究では HEX の最善応手手順を PC クラスタを用いた並列探索によって生成し、考案した並列化手法が組み合わせゲームの探索に対して効果的であることを示した。また、本手法は様々な列挙法に適用可能である。

### 参考文献

- [1] 山崎洋平：「組み合わせゲームの裏表」, シュプリンガー・フェアラーク東京株式会社, 1989
- [2] I. スチュアート：「数学レクリエーションヘックス必勝法」, 日本経済新聞社, 日経サイエンス 2000 年 12 月号, pp.136-138
- [3] J. Libano Alonso, H. Schmidt and V.N. Alexandrov : "Parallel Branch and Bound Algorithms for Integer and Mixed Integer Linear Programming Problems under PVM", Recent Advances in Parallel Virtual Machine and Message Passing Interface, LNCS, 1332, pp. 313-320, 1997