

ハッシュテーブルを用いた
コンピュータ将棋の探索の効率化 (2)

3E-2

細江正樹 久保田聡 瀧口伸雄 小谷善行
(東京農工大学 工学部 電子情報工学科)

1. はじめに

筆者らは、人工知能研究の一環として将棋プレーシステム「shouchan」を作成し、現在も研究を続けている。

今までの shouchan は Prolog 言語で書かれていたのだが、スピードが遅く、実験を行うにもたくさんの時間がかかり、よい環境ではなかった。そこで、本年度システムをC言語で書き換え、スピードアップをはかるとともに、昨年度筆者が作成した探索システムを、拡張した。

本稿では、作成したシステムの概略を示すとともに、[5]で発表したシステムからの拡張点などについて述べる。

2. システムの概略

コンピュータ将棋におけるゲーム木探索は、チェスと比べると生成される可能手の数がかかなり多く、全探索は難しい。そのため、探索の結果を変えなくゲーム木の大きさをできるだけ小さくする必要がある。そこで、 $\alpha\beta$ 法による枝刈りに次に述べるハッシュテーブルを加え、さらに効果的に枝刈りを行えるようにした。

探索中に生成された局面の情報をハッシュテーブルに格納する。ハッシュテーブルに格納する情報は、その局面における最善手・その局面の評価値・評価値の属性(4章参照)・評価値を得るために探索した木の深さ・その局面のハッシュコードである。もし同じ局面が再度出現したときに、格納されている情報が十分(何をもち十分とみなすかは後述)であるならば、その局面を末端局面として、それ以降の再探索を省略することができる。つまり本方式は、同じ局面の探索を繰り返す手間を省いてゲーム木の大きさを小さくしようというものである。

ハッシュキーは、駒の種類やマス目などからなる Piece-Square テーブル(以下 ps_table)を用いて計算する。将棋盤上の40枚の駒について、先後・置いてあるマス(または持駒)を求め、それらの値の排他的論理和をとることによってキーを求める。ps_tableには、適当な整数が格納されている。

探索中に局面はどんどん変わっていく。その度にいちいちハッシュコードの計算をするのは効率が悪いので、指し手に応じてハッシュコードを変更していく必要がある。まず、動かす駒の種類とマス目を元に ps_

table から値を取り出し、元のハッシュコードと排他的論理和をとる。これを移動させるマス目に対しても行う。排他的論理和の計算を用いる理由は、同じ数同志の計算結果が0になるという点に由来する。つまり、元のハッシュコードと動かす駒の現在座標における値の排他的論理和をとることによってその駒をいったん盤上から打ち消し、移動させる座標における値との排他的論理和を計算することによって盤上から打ち消した駒をその座標に置くことになる。このように、排他的論理和の計算を用いることによってハッシュコードの変更作業を簡単に行うことができる。駒の移動の場合は二回の計算で局面の変更を行えるが、駒を打ったり取ったりする手になると、持ち駒の枚数についても考えなくてはならないため、計算の回数はおのおの三回・五回になる。

3. 同一局面の判定

ある局面にくると、与えられたキーからテーブルのアドレスを求める。そのアドレスに情報が何もなければ、その局面にきたのは初めてなので何もせずに探索を行う。もしなにか情報が格納されていても、その局面についての情報でない可能性がある。そこで、現在局面と同一局面の情報かを判定する必要がある。

まず、テーブルに格納されているハッシュコードと現在局面のハッシュコードを比較する。たいていの場合がこの場合なので、この判定を行うことでほとんどの誤りを検出できる。次に、手順前後などの原因で、ハッシュコードが同じ値(駒の並び方が同じ)のときでも手番が異なる可能性があるためそれを比較する。本システムでは、盤面の比較を行わずに、テーブルの中に格納されている最善手が実際に着手可能かを調べることで手番の判定を行う。

この判定を行うことによって、異なった局面の情報が誤って使用される可能性はかなり低くなる。

4. 拡張された点

今回のシステムで拡張された点は、次の四点である。

・手の順序づけを行う

ハッシュテーブルの中に現在探索中の局面と同一の局面の情報があっても、その局面を末端局面とみなすのに十分でないときがある。その場合には、ハッシュテーブルに格納されている最善手を第一候補手として

最初に探索するようにする。一般に $\alpha\beta$ 法による探索では、評価値の高いものから順番に探索した方が効率よく探索できることが知られている。したがって、その局面では枝刈りはできないけれども、手の順序づけを行うことによってその後の探索での枝刈り効果が期待できる。

・前の探索で利用した情報を使用できる。

[5]で発表したシステムでは、ハッシュテーブルの情報の使用は一回の探索の間しかできなかった。今回のシステムでは、以前の探索で格納された情報も使用できるようにした。[5]のシステムでは、ある程度探索を行わないとハッシュテーブルの情報が格納されている局面にたどり着かなかつたが、今回のシステムでは、ハッシュテーブルの情報が探索の開始と同時に使用することができる。主に前の探索で使用した情報は、手の順序づけに利用される。

・評価値の属性

[5]のシステムでは、評価値のみを格納していた。しかし実際の探索中の評価値は枝刈りが起きると、その評価値をある値に特定することはできない。そこで、確定値(子局面を全探索した結果得られた値)・上限値(枝刈りの結果得られたMINノードの評価値)・下限値(枝刈りの結果得られたMAXノードの評価値)という三つの属性を用意して不確定な値も表現できるようにした。

・テーブルの内容の判定

テーブルに格納されている情報が、現在局面に適用可能と判定したとき、次にその情報が探索を省略するのに十分な情報であるかを判定する。

格納されている情報が十分な情報であるための条件は次の二つである。

(i) (評価値を得るために行った探索の深さ) \geq (その局面から末端局面までの深さ)

一般的に、コンピュータ将棋のように終了局面までの全探索が不可能なゲームでは、より深く(より終了局面に近く)探索すればするほど正確な評価値が得られるといわれている。したがって、先ほど示した条件が成り立たなければハッシュテーブルの情報を使用するよりも再探索を行ったほうがより正確な評価値を得られるといえる。したがって、この条件が成り立たなければ、より正確な評価値を得ることができなくなるので、探索の省略はできない。

評価値を得るために行った探索の深さをハッシュテーブルに格納することによって、前回の探索の結果を利用することが可能になる。

(ii) 評価値が確定値、または、確定値ではないが枝刈り可能な値

評価値が確定値ならば、その局面の探索は完了して

いるので、これ以上探索を続ける必要はない。また、枝刈りなどで評価値が確定していない場合は、その評価値で枝刈りが可能な場合のみ探索が省略される。枝刈りができない場合は、再探索を行い評価値を計算し直す。

(i), (ii)の条件が満たされなければ再探索を行うのだが、その場合、テーブルにそれまでの探索で得られた最善手が格納されているので、その最善手を最初に探索することによって探索の効率化をはかる。

5. 評価

本システムと、[5]で作成したシステムの比較を行った。比較の対象は一回の探索に要した局面の数である。ゲーム木の大きさは、深さ3で可能手の全読みである。深さ3の場合、ハッシュテーブルの情報から枝刈り可能と判定される局面はほとんど末端局面なので、さらに効率化をはかるためには、探索開始局面などの、深さが浅い局面での手の順序づけが重要になると思われる。

結果は、[5]で作成したシステムよりも、さらに5%ほど探索局面数を減らすことができた。前回の探索の情報から手の順序づけを行ったことで、さらに効率的な探索ができるようになった。特に探索開始局面での手の順序づけの効果が大きい。探索の深さをさらに深くすれば、手の順序づけの効果はさらに大きくなるものと思われる。

6. まとめ

前回作成したシステムを改良し、以前の探索で得られた情報も利用できるように拡張した。その結果、[5]で作成したシステムよりもさらに5%ほど探索を効率化することができた。

7. 謝辞

本稿を執筆するにあたり、ご指導いただいた飯田弘之氏に感謝の意を表します。

・参考文献

- [1] Levy D., Newborn M.: How Computer Play Chess, COMPUTER SCIENCE PRESS, pp. 153-190, 1991
- [2] 実近憲昭: ゲームと探索, 電子通信学会誌, Vol. 65 No. 4, 1982, pp. 405-412
- [3] 小谷・吉川・柿木・森田: コンピュータ将棋, サイエンス社, 1990
- [4] Zobrist A.L., A New Hashing Method with Application for Game Playing. ICCA Journal, Vol. 13, No. 2, pp. 69-73, 1990
- [5] 細江・久保田・瀬野・飯田・小谷: ハッシュテーブルを用いたコンピュータ将棋の探索の効率化, 情報処理学会第44回全国大会論文集, 1R-7, 1992