将棋に適したデータ構造

3E−3

田中 淳史，久保田 聡，細江 正樹，飯田 弘之，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

一般に、将棋プログラムは指し手の決定に先読みを用いた探索を使うため、非常に多くの局面を生成し、評価することが可能なメカニズムが必要である。探索中での、それぞれの局面の評価や新しい局面の生成などの計算にはその局面を表現している情報を使うので、局面情報の形式の優劣が全体の計算時間に大きく関わってくる。

本稿では、将棋プログラムに適したデータ構造を評価するために、局面情報を保持する情報の違いによる速度の差とメモリ使用量についてと、局面情報の渡し方による速度の差を測定した。そして、より有効と考えられるデータ構造について考察した。

2．将棋の局面を表現する方法

書籍などで将棋の局面は図1のように表される。これは9×9のマスと駒台それぞれの座標にある駒種と状態（表か裏か），所有者（上向きか下向きか）の情報で一つの局面を表現している。

このような将棋の局面をコンピュータプログラム中のデータにするには次の2つの形式が考えられる。

1)盤形式

将棋盤に則した形式の9×9のマス目と駒台それぞれに対応する配列にそのマス目にある駒についての情報を格納する。

2)駒リスト形式

40枚の駒それぞれに対応している配列にその駒についての情報を格納する。

盤形式、駒リスト形式はどちらも局面を完全に表現することができ、相互に変換することも可能である。しかし、これらを将棋プログラムに使った場合の効率は違ってくる。

盤形式、駒リスト形式、および盤形式と駒リスト形式の両形式の併用した形式（併用形式）について、それぞれの局面情報の効率を全探索を行うプログラムで、実行時間とメインメモリへの最大常駐量を測定した。（図2）

実験に使った局面は棋士同志の棋譜の一手目から百十一手めまでの十手おきに取ったもので、測定した値は十局面の平均値である。実験はすべてワークステーション（SUN4）上で GNU C Compiler を使って行った。

実行時間は併用形式が最も短く、盤形式、駒リスト形式の順に実行時間が長くなった。これは将棋プログラムでは座標を使った計算が比較的多いので盤形式のデータを持つ方が計算が速いためと思われる。併用形式は多くの情報を持ち、複雑な計算をする必要が少ないので、局面を更新するときに更新する情報が多いにも関わらず実行時間は最も短かった。メモリの使用量に大きな差はなかったが、局面情報の多い併用形式が最もメモリ使用量が少なかった。これは多くの情報を持つためにプログラムが簡単になったためである。

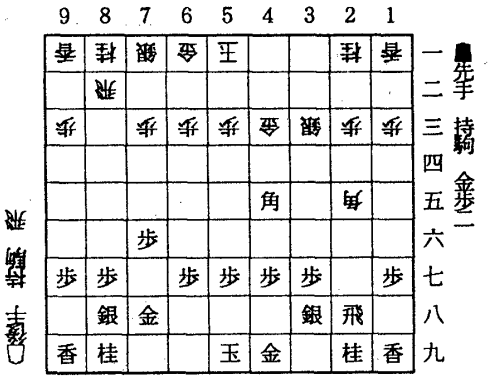

図1 書籍での局面表示

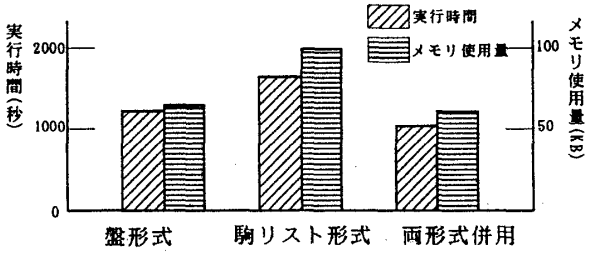

図2 深さ5の全探索の実行時間と必要なメモリー

Data Structure for Shougi

Atsushi TANAKA, Satoshi KUBOTA, Masaki HOSOE, Hiroyuki IIDA, Yoshiyuki KOTANI
Tokyo Univerisity of Agriculture and Technology

3．探索の高速化のために局面情報に付加する情報

実際の探索過程において何回も同じ計算によって求めている情報、あるいは局面を更新しても変化が少ない情報があればそれを保存しておくことによって探索の効率を高めることができる。あらかじめこのような情報を保持するテーブルを作っておき、表引きによって情報を使えば探索の時間を短縮できる。

そのような例として駒の利きがあげられる。探索中、可能な指し手を生成する時に、自分の駒が利いている座標を求める必要がある。駒の利きは局面を更新しなければ変化がなく、一手指した後でも変化は少ない。測定によると一手指した時に利きが変化する駒数の平均は40枚中約3.2枚であった。このように局面を更新しても変化が少ないので、駒の利きをテーブルにしておき局面を更新するときには利きが変化した部分だけテーブルを更新すれば良い。このようなテーブルを持っていれば高速に駒の利きを調べることができる。

駒の利きのテーブルの効果を2の測定に使用した併用形式の局面情報を用いた全探索プログラムとそれに駒の利きのテーブルを持たせたものを作成し比較した。（図３）

この結果駒の利きをテーブルに持つことによって高速な探索が行えることが測定された。

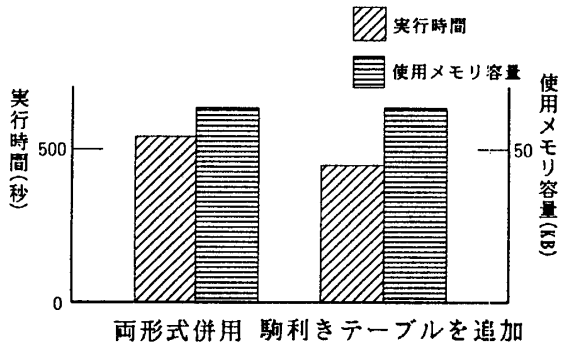

図３ 駒の利きをテーブルにした時の探索速度

4．関数間での局面情報の受け渡し方

将棋プログラムでは局面情報を関数間で共有する必要があるが、Ｃ言語では以下の３種類の方法で構造体を関数間で渡すことができる。

1)グローバル変数に局面の構造体を取り、それを使う。
2) 構造体のアドレスを渡す。
3) 構造体そのものを渡す。ただし、情報を書き換える関数へはアドレスを渡す。

併用形式のデータ構造で、この３種類の渡し方を使った深さ５の全探索プログラムを使い、実行速度を測定した。（図３）

グローバル変数で局面情報を持つものが最も速度が速く、わずかにアドレス渡しが遅く、値渡しは割遅かった。これはプロセッサやコンパイラによって変わる可能性があるが比較的大きいデータを渡す場合には順当な結果だと考えられる。速度の点ではグローバル変数が最も優れていたが、特定の変数と結びついているという欠点があるため、最も一般的に使われるアドレス渡しが優れていると考えられる。

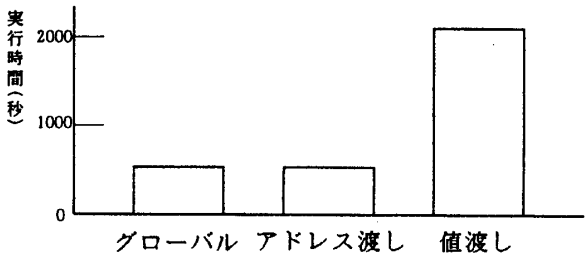

図４ 局面情報の持ち方による速度の差

5．まとめ

今回は全探索についてだけの測定を行ったが、将棋のプログラムには局面の評価が欠かせない。局面の評価要素はプログラムによって様々なものを持っている。その中には、駒の利きのようにテーブルにして局面情報と併せ持つデータ構造にしたほうが探索を高速に行うことができるものも多いと思う。

局面情報は現在のワークステーションやパーソナルコンピュータの記憶容量に比べれば非常に小さい。また、多くの情報を持つことでプログラムが単純で小さくなることもある。そのため、有効な情報はデータ構造に多く含める方が有利になる。

参考文献

[1] 小谷善行，吉川竹四郎，柿木義一，森田和郎：コンピュータ将棋，サイエンス社 (1990).

[2] 飯田弘之，瀬野訓啓，吉田武俊，小谷善行：Prologによる将棋プログラムのデータ構造，記号処理，Vol.55-1 (1990).