将棋プログラム作成

2003MM116 山田 浩徳
指導教員: 杉浦 洋

1　はじめに

現在，計算機能力の向上，あるいはアルゴリズムの発達に伴い，チェスや，オセロ，バックギャモンといったいくつかのコンピュータエージェントの技術は人間の世界チャンピオンか，あるいはそれ以上のレベルにまで到達している．また五目並べのようないくつかのゲームにおいては，コンピュータにより必勝法が発見されている．しかし，探索空間が大きかったり，あるいは知識や判断基準の定義が分かりにくく，また数値化が難しいことなどにより，人間のトップの領域に達していないゲームもある．そして，その中の一つが将棋である．

2　アプローチ

本研究では，C++で実際にルールどおり将棋を指すプログラムを作成することを目指した．全幅探索(しらみつぶし的に指し手を読んで行く方法)で，探索アルゴリズムにはα-β法を用いた．本研究では，将棋プログラムを作成することにより，そのデータ構造とアルゴリズムを学び，コンピュータに将棋を指させるということがどういうことかを研究する．その過程で私が実際に将棋を指して体得して来た価値観，経験を生かす．

3　指し手のリストアップ

将棋のルールの詳しい説明は紙面の都合で省略させて頂く．さて，将棋の「局面」は盤面の駒の配置と持駒で定義される．ある局面における指し手の選択にはその局面で指せる手をリストアップすることが必要である．それには以下に示す将棋のルールに従う合法的な手をリストアップすることが基礎となる．

3.1　駒の移動

将棋の駒は8種類あり，それぞれ異なる動き方をする．それぞれの駒に共通する原則は以下の通りである．

- 自分の駒のある所には進めない．
- 相手の駒のある所に進むと，その駒を取る事ができる．取った駒は「持ち駒」になる．
- 持ち駒は，原則として好きな所に置くことができる．ただし，駒のあるところ，行き所のないところには打つことはできない．
- 相手の陣地に進むと，駒を裏返して，動き方を変える事ができる．(「玉」と「金」は除く．)なお，一度成った駒を元に戻すことはできない．

3.2　歩による反則

- 「二歩」の反則
  味方の歩がある筋に，もう一枚自分の歩を打つこと．なお，歩が成った「と」と同じ筋には歩が打てる．
- 「打ち歩詰め」の反則
  持ち駒の歩を打って相手の玉を詰ませること．

3.3　王手と詰み

将棋は，先手と後手が交互に指し，最終的に相手の玉将を取ったほうが勝ちとなる．相手が何を指しても次に玉を取られる状態を「詰み」という．なお，次に玉を取ろうとする手の事を「王手」という.「詰み」は防ぎようのない「王手」とも言うことができる.「詰み」以外の王手は放置してはいけない．

以上のルールにそって，その局面で合法的な指し手を全てリストアップするルーチンを作成した．

4　指し手リストの圧縮

王手に関連して指し手のリストの圧縮ができる．王手とは次に何か防がなければ玉が取られる状態であるので，必ず防ぐ必要がある．つまり，王手された際の指し手のリストアップは王手を防ぐ手だけで良い．王手を防ぐ手は大きく分けて三つある．

1. 玉をその場所から逃げる．
2. 王手をかけている駒を取る．
3. 合い駒をする(敵の飛，角などの駒と玉の間に持駒を打つ)．

の三つである．また，まれに図1の左の様に二つの駒から王手をされる場合がある．これを両王手という．両王手とは違う二つの方向から王手をされることであるから，3の合い駒をするや，王手をかけている駒を玉以外で取るといった手は，二つの王手のうちどちらか片方しか防ぐことができないので無効である．つまり，両王手をかけられたときは，周囲八方向のどこかに玉を移動するしかない．また，ゲームのルール上，三つ以上の駒から王手をされることはない．

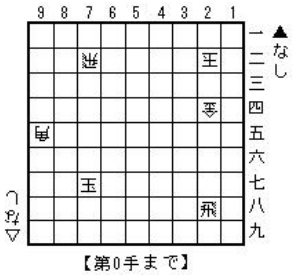

図 1: 局面例

4.1　移動すると玉を取られる駒

王手以外にも，指し手リストが圧縮される場合がある．将棋には局面によっては，移動すると玉が取られてしまう駒が存在する．図1の右を見て欲しい．金は本来五ケ所

動く場所があるが，図のような局面では前と後の二箇所しか動くことができない．

5　指し手の探索

5.1　評価関数

将棋の変化は10の220乗以上とも言われており，ミニマックス法やα-β法を使っても全ての局面変化を探索することは不可能である．そこで，局面をなんらかの方法で評価してやり，n手後に最大価値の局面に到達しようとする戦略を採用する．主に局面の評価は，駒の価値の総和で行う．駒の価値に付いてはいくつかの参考文献の中で様々な値が紹介されているが，ここでは私が実際に将棋を指してきた経験を生かして駒の価値を決めた．注意するところは，将棋は持駒の方が盤上にいる駒より使いやすいことが多いので，持駒に少し高い価値を与える．

5.2　α-β法

保木[2]にそってα-β法について説明する．この枝刈方法はミニマックス法を効率化したアルゴリズムである．図2.4を見て欲しい．左から順に評価関数の値を計算しチェックしていくとする．互いが相手の局面を決めることができ，局面の値が大きい程，先手が良く，値が小さい程，後手が良いとする．つまり先手は最小値が最大になること(マックスミニ法)を目指し，後手は最大値が最小になること(ミニマックス法)を目指す．

図2でIから始まりMまでチェックすると，すでにMはそれまでの中で最大値であることが分かるので，相手はAの段階で，決してEを選ばないと判断できる．つまりCのMAXは6，DのMAXは5，EのMAXは9以上であるからだ．そこでNを計算する必要がなくAの値は5 となる．この種の枝刈をbetaカットと呼ぶ．

次にBの指し手に移り，OとPを計算したところで，FのMAXは3であることが分かる．すると，まだ探索していないGやHの値が何であろうと，Bの値は3より大きくならないことが分かる．この時点で，もうそれ以上計算する必要がなく，BよりもAが優れていることが明らかになる．この種の枝刈をalphaカットと呼ぶ．

このα-β法を使うことにより，実際のプログラムでは平均して50%から90%の探索を省くことができる．

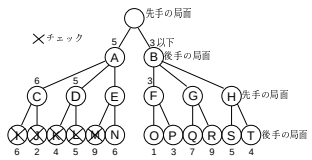

図 2: α-β法の例

6　実験結果

評価関数が駒の損得だけであるので，強さの方は全く満足いくものにはならなかった．それでも5手先まで読ませると，一応将棋らしい手は指す．ただし，かなり時間がかかってしまう．具体的には，三手の読みで序盤三十手指させても，４秒程度などに対して，五手の読みだと，同じ条件で600秒程かかる．現在のプログラムでは，現実的な持ち時間では，七手読みは不可能である．

3手先まで読ませると，スピードは速いが，対局を壊してしまうほどの悪手を高い確率で指してしまう．駒の損得以外評価をしていないので，当然ではあるが序盤は全くダメで，持久戦に誘導しても将棋にならなくなる．逆に，序盤から駒を交換するような展開(例えば角交換)は持久戦に比べればかなりまともに戦う．駒を直接取ったり交換したりする序盤の方が目的(主に駒得)がはっきりしているため，プログラムには理解しやすいと言える．

7　研究の成果と今後の課題

今回の研究の成果として次のことが上げられる．

- 将棋をルールどおり指すプログラムの完成(プログラムはC++で2000行弱)．
- 人との対戦モードとプログラム同士の対戦モードを持つ．
- 実時間で五手先まで読める．
- ときどき鋭い手を指す．

将棋をルールどおりに指させるプログラム，特に合法手の生成の部分はかなり手間がかかる．データ構造のイメージと，どのようなアルゴリズムで指し手を生成していくのかの流れをつかむまでが大変であったが，こちらの指し手にプログラムが答えてくれるようになってからは大いに研究を進めることができたと思う．

また，今回の研究でルールどおりに将棋を指すプログラムが完成したがまだまだ満足のいく強さにはならなかった．これからの課題としては次のことが上げられる．

- データ構造の改良とプログラムのさらなる効率化．
- より良い評価関数の作成．
- 詰将棋を解くプログラムの作成．
- 定跡データベースの作成と活用．

8　感想

私は，今このプログラムに将棋を覚えたばかりの子供に似たイメージを持っている．全体の指し手はまだまだ幼稚であるが，テーマの局面で目を見張る手を指すこともある．研究のしがいのある，大いなる可能性を秘めているように思う．ここで研究の区切りとしなければならないのが，大変に惜しい．

参考文献

[1] 日本将棋連盟，
http://www.shogi.or.jp/.

[2] 保木邦仁，渡辺明：ボナンザVS勝負脳，角川書店(2007).