指し手の種類別探索深さの学習

岩井麻子* 鈴木豪† 小谷善行† 堤正義*

チェスや将棋などのゲームプログラムでは、可能な指し手を列挙したゲーム木をより深く探索すれば強くなると考えられるが、ゲーム木全てを現実的な時間内に探索することは難しい。そこで有力な指し手のみを深く探索する手法が必要となる。筆者らは「延長パラメータ」と呼ぶ探索延長のための指標を導入し、指し手の種類に注目してこの延長パラメータの学習を行なった。その結果を報告する。

Search Extension by Kind of Move in Shogi

IWAI Asako* SUZUKI Tsuyoshi† KOTANI Yoshiyuki†
TSUTSUMI Masayoshi*

In Shogi it seems that a program is stronger when it searches deeper the game tree. Although it is hard to search all of the nodes in the tree in the realistic time. So we consider a kind of move, such as capture or put, then introduce a parameter which called "Extension parameter" for each move. The parameter is adjusted through the game by a certain method based on Temporal Difference learning. We will show the experimental details and results in this paper.

1 はじめに

将棋プログラムでは最善手のみ更に探索を延長する 0.5 手延長アルゴリズム [4] や Singular Extension [3] が提案されている。しかし指し手の種類に基づいた探索の延長については殆んど行われていなかった。

そこで筆者らは指し手を「駒を取る手」「駒を取らない手」といったいくつかの種類に分類し、「延長パラメータ」と呼ぶ探索延長の指標を導入して、どのような種類の指し手を延長して探索するべきかを調査した [7]。

先の研究 [7] では、プログラムを対戦させることで指し手の種類毎に延長パラメータを調整する実験を行った。この実験の結果、延長パラメータが特定の値に収束するといった数値的な結果は得られなかったが、どの種類の指し手は深くまたは浅く探索するという傾向が得られた。さらに得られた延長パラメータを用いて実験に用いたプログラム同士の対戦実験を行い勝率を調べた。延長パラメータの利用により勝率が向上した場合もあったが低下したものもあり、延長パラメータの利用のみで勝率が向上したとはいえない結果となった。

以上の研究を踏まえて指し手の種類を駒種などに着目してさらに分類した場合について、延長パラメータの学習を行なった。

*早稲田大学大学院理工学研究科
†東京農工大学大学院工学研究科
†東京農工大学工学部
*早稲田大学理工学部
*Waseda Univ.
†Tokyo Agri.and Tech. Univ.

2 探索の延長

指し手の種類による探索の延長は通常の $\alpha\beta$ アルゴリズムの改良により実現した。$\alpha\beta$ アルゴリズムでは、探索の深さの指定はルート局面で指定した探索の深さ $depth$ を 1 ずつ減らして計算する (式 (1))。$depth$ が 0 となった時末端に到達したとしてルート局面の評価値を計算する。

$$\mathbf{alphabeta}(depth-1) \tag{1}$$

例えばルート局面で探索の深さ 2 を指定した場合、減算が 2 回行われ、2 手先まで探索を行なうこと

になる。

今回の指し手の種類による延長アルゴリズムでは、*depth* から減算する値を1で固定せず指し手の種類による値 *extension* で減算している (式 (2))。以下この減算する値 *extension* を「延長パラメータ」と呼ぶ。延長パラメータが1の場合が通常の $\alpha\beta$ アルゴリズムとなる。

$$\mathbf{search}(depth - extention) \qquad (2)$$

図1に指し手の種類による探索延長アルゴリズムを示す。

```
search(position,alpha,beta,depth)
{
  if( depth ≤ 0 )
    return(evaluate(position) );
  width = generate(position);
  if( width == 0 )
    return(evaluate(position) );
  score = alpha;
  for(child=0;child<width;child++){
    extension = getextension
                (position.child);
    value=-search(position.child,
     -beta,-score,depth-extension);
    if(value > score)
      score = value;
    if(value ≤ beta)
      return(score);
  }
  return(score);
}
```

図 1: 指し手の種類による探索延長アルゴリズム

ただし generate は局面 *position* の子局面を生成する関数、*position.child* は局面 *position* の子局面、getextension は指し手の延長パラメータを計算する関数、evaluate は局面 *position* の静的評価値を計算する関数である。

3 延長パラメータの学習

先にのべた延長パラメータを実際の対戦を通して学習させる。延長パラメータの学習には次式を用いる。

$$w \leftarrow w + \sum_{t=1}^{n} \alpha(P_{t+1} - P_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w P_k$$

ただし

$x_1, x_2, ..., x_t$	:	局面
α	:	学習率 (実数定数)
$P_t = P_t(w; x)$	:	予想確率 $(0 \le P_t \le 1)$
$w = (w_1, w_2, ...w_n)$	:	延長パラメータ
$e(w; x)$	:	局面の評価値
λ	:	予想確率更新の重み $(0 \le \lambda \le 1)$

である。予想確率 $P_t(w; x)$ はシグモイド関数

$$f(x) = \frac{1}{1 + \exp(-x)}$$

を用いて $P_t(w; x) = f(e(w; x))$ として計算する。また、$\nabla_w P_t(w; x)$ は

$$\begin{aligned} \nabla_w P_t(w; x) &= \left(\frac{\partial P_t(w; x)}{\partial w_1}, ... \frac{\partial P_t(w; x)}{\partial w_n}\right) \\ \frac{\partial P_t}{\partial w_j} &= f(e(w; x))\{1 - f(e(w; x))\}\frac{\partial e}{\partial w_j}(w; x) \\ \frac{\partial e}{\partial w_i}(w; x) &\approx \frac{e(w_1, ..., w_i + \delta_i, ..., w_n; x) - e(w; x)}{\delta_i} \end{aligned}$$

とする。

本実験では先手プログラムについて延長パラメータの学習を行った。λ は 0.95 で固定している。また δ_i は $0, 0.5, -0.5$ のいずれかの値を取るようにした。

また深く読めば読むほど強くなるので、このままでは延長パラメータがどんどん小さくなってしまう (深く読む方向)。そこで、全体の深さを一定に保つため

$$w_1 + w_2 + ... + w_n = \beta$$

(β は実数定数) として w_i を

$$w_i = \beta \frac{w_i}{\sum_j^n w_j}$$

で補正している。

4 実験の概要

先の研究で着目した指し手の種類をさらに以下のように分類した:

- 「盤上の駒を動かす手」について、駒の種類「歩、香、桂、銀、金、角、飛、王、と、成香、成桂、成銀、馬、竜」(計 15 種類)
- 「駒を取る手」について、取れる相手の駒の種類「歩、香、桂、銀、金、角、飛、と、成香、成桂、成銀、馬、竜」(計 14 種類)
- 「移動先と自分の王との距離」について、距離 1 から 8 まで (計 8 種類)
- 「移動先と敵の王との距離」について、距離 1 から 8 まで (計 8 種類)

(x_1, y_1) にある駒 K_1 と (x_2, y_2) にある駒 K_2 の距離は $dist(K_1, K_2) = \max(|x_1 - x_2|, |y_1 - y_2|)$ で定義する。例えば 4 三にいる駒と 2 七にいる駒の距離は 3 である。

上記の指し手の種類それぞれが延長パラメータを持つ。また本実験では「盤上の歩を動かす手」と「自分の歩で相手の銀を取る手」というような重複する指し手は考えない。それぞれ上記の着目する指し手について、別々のプログラムで実験を行う。

実験では延長パラメータの初期値 w を 1 とし、対戦を行う。その対戦の結果と棋譜を利用して自分の手番毎に延長パラメータの学習を行い、その学習結果を次の対戦で利用する。このように対局を繰り返し行い、その結果を用いて延長パラメータの学習を行う。

実験に用いた将棋プログラムは探索の基本深さを 2 として指し手の種類による探索延長アルゴリズムを適用した。またゲーム木末端で駒の取り合いが起こっている場合は、その取り合いを延長して探索して評価している。200 手までで勝負がつかない場合は引き分けとして対局を打ちきる。

5 結果

上記の指し手の種類について延長パラメータの変化の様子を以下の図 3〜図 5に表した。ただし縦軸の延長パラメータは 100 倍して表示している。

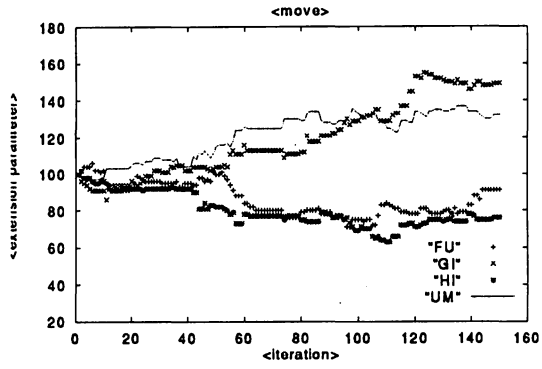

図 2: 盤上の駒を動かす手 (歩、銀、飛、馬)

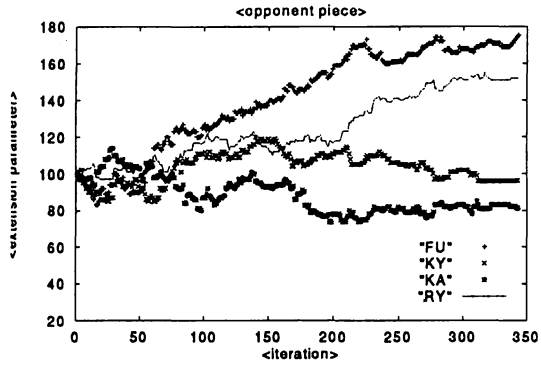

図 3: 相手の駒を取る手 (歩、香、角、竜)

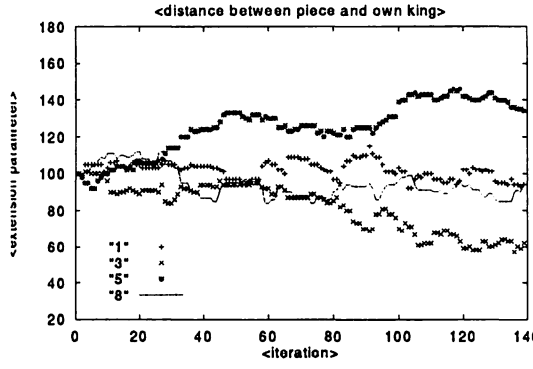

図 4: 移動先と自分の王との距離 (距離 1,3,5,8)

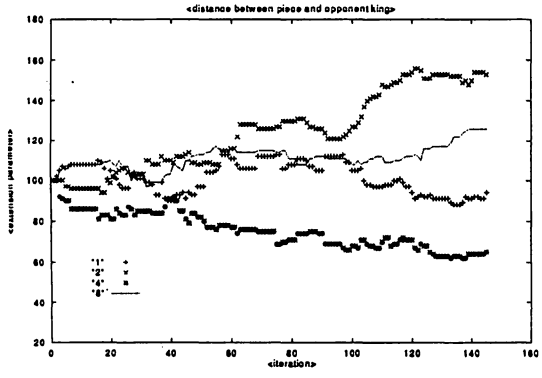

図 5: 移動先と敵の王との距離 (距離 1,2,4,8)

6 考察

図 2の「盤上の駒を動かす手」では、歩はあまり探索の深さを変えず、銀は浅めに探索するような傾向が見られた。飛車は深めに探索する傾向が見られた。また馬は浅めに探索する傾向が見られたが、成り駒は出現回数が少ないため学習が進まなかった可能性があると考えられる。

図 3の「相手の駒を取る手」では、香車はあまり探索の深さを変えず、歩は浅く探索し、各は少し深く探索するような傾向が見られた。また竜は浅めに探索する傾向が見られたが、「盤上の駒を動かす手」同様成り駒の出現回数が少ないため学習が進まなかった可能性があると考えられる。歩や香車を取ってもあまり駒得にならないため、このような結果になったと考えられる。

図 4の「移動先と自分の王との距離」では、自分の王に近い距離 1 と王から遠い距離 8 はあまり探索の深さを変えず、少し離れた距離 3 で深く探索する傾向が見られた。

図 4の「移動先と敵の王との距離」では、敵王に近い距離 1 と少し距離がある距離 4 は少し深く探索する傾向が見られる。一方で距離 2 と距離 8 は浅く探索する傾向が見られる。これは敵の王を攻めている距離が関係していると考えられる。

先の実験で着目した指し手の種類を更に細かく分類すれば、「人間がどのような駒を取る時に深く読んでいるのか」「どのような駒を進める時に浅く読んでいるのか」などがわかるかと考えたが、本実験を通しても特に明らかな規則性は見つけられなかった。

7 おわりに

本研究では将棋プログラムにおいて指し手の種類に着目してゲーム木の探索を延長する実験を行った。指し手を駒種に着目するなどして細かく分類し、それぞれに延長パラメータを持たせ、対戦を通してそれらの学習を行った。

今後の課題として、放っておくとより深く探索する傾向に学習が進んでしまうのをどのように制約して調整するかが挙げられる。今回は延長パラメータの総和を用いたがその他の条件も検討したい。また探索の基本深さを変えた場合に、どのように延長パラメータの効果があるのか実験を重ねたい。

参考文献

[1] 小谷善行, 飯田弘之: なにを刈るべきか–指し手の分類と指した手の割合–, ゲームプログラミングワークショップ'95,pp.148–166 (1995)

[2] 中山義久, 小谷善行: singular extension の将棋への適応, ゲームプログラミングワークショップ'96,pp.210–217 (1996)

[3] 中山義久, 小谷善行: singular extension の着手の性質, ゲームプログラミングワークショップ'97,pp.38–45 (1997)

[4] 山下宏: 0.5 手延長アルゴリズム, ゲームプログラミングワークショップ'97,pp.46–54 (1997)

[5] D.F.Beal and M.C.Smith: Quantification of Search-Extension Benefits, *ICCA Journal*, Vol 18, No.4, pp.205–218 (1995)

[6] D.F.Beal and M.C.Smith: First Results from Using Temporal Difference Learning in Shogi, *Computer and Games* pp.113–125 (1998)

[7] 岩井麻子, 鈴木豪, 小谷善行, 堤正義:ゲーム情報学研究会,99-GI-1,1-12,pp.85–89(1999)