Vol.2012-GI-27 No.3
2012/3/2

Cross-Entropy Method を用いたコンピュータ将棋における探索パラメータの学習

松原徹[†]　古宮嘉那子[††]　小谷善行[††]

将棋やチェスなどの思考ゲームが強いプログラムを作るためには，評価関数の精度と効率的な局面探索が必要であるとされている．複雑な評価関数のパラメータが機械学習に成功していることに比べ，静止探索の深さや Futility Pruning のマージンなどのパラメータは手動で決められていることが多く，最適な値であるとは言えない．そこで本研究ではこれらのパラメータを Cross-Entropy Method を用いて学習する手法を提案した．実験の結果，静止探索の深さや Aspiration 探索のウインドウ幅などのパラメータが良いと思われる値に収束した．また，学習したパラメータは人手による調整のパラメータに対して大きく勝ち越した．

Learning of Search Parameters using Cross-Entropy Method in Computer Shogi

TORU MATSUBARA[†]　KANAKO KOMIYA[††]
YOSHIYUKI KOTANI[††]

It is considered that the accuracy of a evaluation function and efficient game-tree search are necessary to make a strong program of shogi and chess. Compared with the parameters of a complicated evaluation function, which have succeeded in machine learning, the searching parameters, such as the depth of quiescence search and a margin of Futility Pruning, are decided manually in many cases, and are not the optimal values.
In this paper, we propose to apply the Cross-Entropy Method for learning these parameters. From the result of the experiment, some parameters converged as the value that seems to be good.Moreover, the learned parameters-based greatly won to the manual parameters-based.

1. はじめに

人工知能の研究として，いかに強いゲームプログラムを作るかという課題がある．「二人零和有限確定完全情報ゲーム」と呼ばれる将棋やチェスなどのゲームは，ゲームの終了条件や勝ち負けなどが明らかであり，偶然の要素を含まず論理的に知識を表現できるため，人工知能研究において注目される分野となっている．局面の評価を行う評価関数の精度と局面の探索の効率が強さにかかわる要素であると考えられており，様々な研究が行われてきた．

局面探索の効率向上のため，$\alpha\ \beta$ 探索 1)を基本とした様々なゲーム木探索の手法が研究されている．例えば，静止探索 2)，NegaScout 探索 3)，Extended Futility Pruning4)などが提案されており，それぞれ将棋やチェスで有効性を示している．さらに，これら探索手法のパラメータ学習として，伊藤らが動的に Futility Pruning のマージンを変化させる手法 5)や小幡らがムーブオーダリングの評価関数を機械学習する手法 6)を提案している．また，鈴木らは探索手法の有効な組み合わせ方の研究を行った 7)．しかし，複数の探索手法の最適なパラメータや組み合わせについては十分研究がされておらず，人手によって調整されていることが多い．そこで本研究では，機械学習を用いてこれらの探索手法の最適なパラメータや組み合わせを得ることを目的として，Cross-Entropy Method8)を用いた学習手法を提案する．Cross-Entropy Method はパラメータの学習手法の一つで，Guillaume らはコンピュータ囲碁で主に用いられている Monte-Carlo 探索のパラメータを Cross-Entropy Method を用いて機械学習することに成功している 9)．この Cross-Entropy Method を用いて探索パラメータの学習を行い，実験によって有効であることを示す．

2. 関連研究

本節では，コンピュータ将棋で用いられている局面探索手法のうち，本研究で学習するパラメータに関連する手法について述べる．

2.1 静止探索

将棋やチェスのようなゲームの局面の評価は駒価値による評価値が大部分を占めているので，駒の取り合いが発生すると評価値が大きく変化することになる．
このとき，局面の評価が駒の取り合いの途中の局面で行われると，不正確な評価値を得てしまうことがある．これを防ぐために，駒の取り合いがなくなり，静かな局面になるまで駒を取る手のみを探索する手法のことを静止探索という．この静止探索が浅

[†] 東京農工大学大学院工学府
[††] 東京農工大学工学研究院

いと局面が静止しきらないことがあり，反対に静止探索が深いとあまり重要でない駒の取り合い等により，探索に無駄な時間がかかることがあるため，静止探索の最大深さをパラメータQ_dとして学習する．

2.2 Extended Futility Pruning

探索の末端の親局面(frontier node)において，局面の駒割のみの評価値が探索ウインドウからF_1以上外れていたら，枝刈りを行う手法のことを Futility Pruning という．このF_1のことをマージンと呼ぶ．
コンピュータ将棋では，局面の駒割のみの評価値は，通常の評価値よりも計算するコストが少なく済み，かつ通常の評価値に近い値を得ることが出来る．
この駒割のみの評価値が，末端の親局面において十分探索ウインドウから外れていたならば，末端においても外れたままであるという考えにより枝刈りを行う．
この枝刈りが発生する条件を式にすると次の式(2.1)のようになる．ただし，αは$\alpha\ \beta$探索のα値，$M(p)$は frontier node の駒割のみの評価値を示しているものとする．

$$\alpha \geq M(p) + F_1 \quad (2.1)$$

さらに，末端の親局面のさらに親局面，そのさらに親局面において駒割のみの評価値が探索ウインドウからF_2，F_3以上外れていたら枝刈りを行うように拡張したものを Extended Futility Pruning¥cite{futility}¥cite{extended futility}という．この Extended Futility Pruning のマージンをパラメータF_1，F_2，F_3として学習する．

2.3 Aspiration 探索

$\alpha\ \beta$探索ではα値とβ値の幅である探索ウインドウ（α，β）が小さいほど枝刈りが発生し，探索ノード数が小さくなる．
この探索ウインドウを，一つ前の反復深化で得られた評価値を中心としたウインドウ幅A_wにして探索を行い，その探索が失敗したときのみ再探索を行う手法のことを Aspiration 探索という 10)．このパラメータA_wが大きければ探索の失敗は少ないが探索ノード数はあまり減らず，A_wが小さければ探索ノード数は減るが探索の失敗が多くなるので，Aspiration 探索を有効に行うために，このウインドウ幅A_wをパラメータとして学習する．

2.4 PVS 探索

PVS 探索 11)は 1 つ目の子ノードの探索を行い，その評価値vを用いて 2 つ目以降の子ノードの探索を効率的に行う手法である．2 つ目以降のノードの探索は探索ウインドウ幅$(v,\ v+1)$で行い，探索が失敗した場合のみ，そのときの情報を用いて再探索を行えば，$\alpha\ \beta$探索と同等の結果が得られる．再探索を行うことになり，探索の無駄になる可能性もあるが，ムーブオーダリングの精度が高い場合，$\alpha\ \beta$探索よりも高い性能を示すとされている．この PVS 探索を使用するかを学習する．

2.5 NegaScout 探索

NegaScout 探索は 2.2.4 で述べた PVS 探索と似た手法である．まず 1 つ目の子ノードの探索を行い，その評価値v_1を用いて 2 つ目以降の子ノードの探索を探索ウインドウ幅（v_1，v_1+1）で行なっていく．
この評価値v_1を使った探索が失敗した場合，PVS 探索と同様に，そのときの情報を用いて再探索を行い評価値v_2を得る．
それ以降の子ノードの探索では新たに得られた評価値v_2を用いて探索を行なっていく．
このように NegaScout 探索では探索ウインドウに用いる評価値を変えていく点が PVS 探索とは異なっている．この NegaScout 探索を使用するかを学習する．

2.6 ムーブオーダリング

$\alpha\ \beta$探索では最も良い手から探索を行った場合に探索する局面数が最も少なく済む．このとき，探索順序を着手の評価によって並び替えることで探索する局面数を減らすことが出来る．この着手の並び替えのことをムーブオーダリングという 12)．着手の評価として，本論文で用いた着手の評価関数の項目を以下に示す．

- 王手の評価値O_1
- 駒を取る手の評価値O_2
- 駒に利かす手の評価値O_3
- 駒を逃がす手の評価値O_4
- 焦点(利きが複数あるマス)への手の評価値O_5

これら 5 つの評価値をパラメータとして学習する．

3. Cross-Entropy Method によるパラメータの学習方法

本節では，Cross-Entropy Method を用いてパラメータを学習する方法について述べる．

3.1 Cross-Entropy Method によるパラメータの学習手法の概要

Cross-Entropy Method は分布推定アルゴリズムに類似したパラメータ学習法であり，その方法の概要を図 1 に示す．まず，パラメータを持ったサンプルを$\rho \cdot N$個作り，分

布U_γとする．次に，ガウス分布や二項分布やベルヌーイ分布などの分布の集合Gの中からクロスエントロピー距離D_{CE}が最小となる分布gを式(3.1)により選択する．

$$D_{CE}(g \| U_\gamma) = \int g(x) \log \frac{U_\gamma(x)}{g(x)} dx \quad (3.1)$$

この分布gを元に，パラメータを持ったサンプルをN個作成する．そしてこの各サンプルを持ったプログラムと予め用意しておいたプログラムをB回対局させる．このときの対局回数をバッチサイズとする．最後に，このサンプルの中から勝率の良いサンプル$\rho \cdot N$個をエリートサンプルとして取り出し，新たな分布U_γとする．これを繰り返すことにより，パラメータを学習する．ただし，本論文では分布の集合Gにはガウス分布のみ用いているため,パラメータの学習は図 2 に示す擬似コードのように行われる．

3.2 出力が二値のパラメータの学習手法

分布から出力されるパラメータの取りうる値が二値のみの場合，学習が不十分にも関わらず，一つの値に固定されてしまう可能性が高いと考えられる．そこで，サンプルの多様性を損なわないため，少なくともn [%]は生起確率を持つようにする．

3.3 パラメータの正規化

出力されるパラメータの値の範囲が大きい場合，値の変動幅も大きく，小さな値の学習が上手く行われないことが考えられる．そこで，パラメータの値の対数を分布のパラメータとすることにより，サンプルの範囲の広さと学習効率を保つようにする．

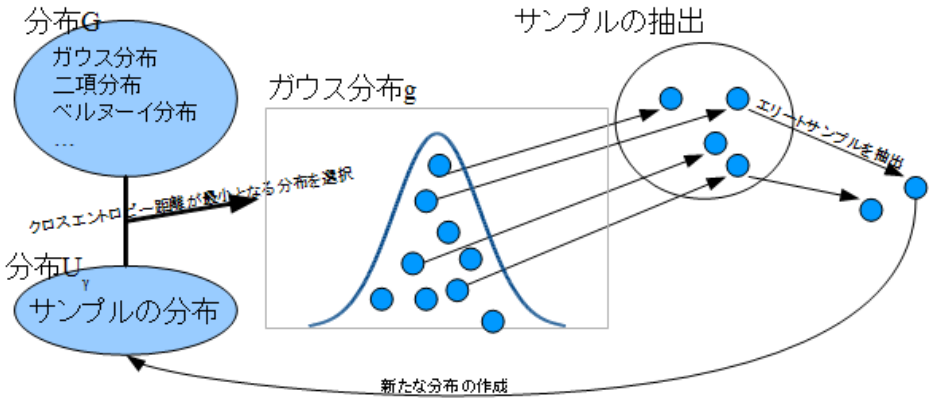

図 1 Cross-Entropy Method の概要図

- サンプルの個数N，エリートサンプルρの比率を決定する
- 学習回数T，バッチサイズBを決定する
- 初期分布を作成する

for(t from 0 to $T-1$){
  for(i from 1 to N){
  - 分布(μ_t, σ_t^2)を元にサンプル$x^{(i)}$を作成する
  - サンプル$x^{(i)}$のプログラムと予め用意したプログラムで先手後手を入れ替えながら対局をB回行う
  - 勝率の高い順にサンプル$x^{(1)}$から並び替える
  - 勝率の高い$\rho \cdot N$個をエリートサンプルとする
  - エリートサンプルから新たな分布$(\mu_{t+1}, \sigma_{t+1}^2)$を作成する

  }
}

図 2 Cross-Entropy Method による学習の擬似コード

表 1 実験で学習したパラメータ

探索パラメータ	範囲	初期平均	初期標準偏差	手調整の結果
Q_d	[0.0;10.0]	5.0	2.5	5.0
$\log F_1$	[1.00;3.00]	2.30	0.50	2.30
$\log F_2$	[1.30;3.30]	2.60	0.50	2.60
$\log F_3$	[1.48;3.48]	2.78	0.50	2.78
$\log A_w$	[1.30;3.30]	2.30	0.50	2.30
PVS	[0.0;1.0]	0.7		1.0
NS	[0.0;1.0]	0.7		1.0
O_1	[0.0;16.0]	8.0	4.0	16.0
O_2	[0.0;16.0]	8.0	4.0	8.0
O_3	[0.0;16.0]	8.0	4.0	4.0
O_4	[0.0;16.0]	8.0	4.0	2.0
O_5	[0.0;16.0]	8.0	4.0	1.0

4. パラメータの学習実験

本節では，Cross-Entropy Method による探索パラメータの学習実験を行い，その結果について述べる．

4.1 実験で用いたプログラム

実験では，本研究用に開発した，HIMMEL と名付けたプログラムを用いた．このプログラムには実験に必要な探索アルゴリズム，反復深化法，トランスポジションテーブル 13)を実装した．また序盤の定跡として約 1200 局面を用いた．

評価関数の項目には駒(14 種類)の価値，王と 2 駒(歩，王以外の 8 種類)の位置関係による価値，先手王と後手王と 1 駒(歩，王以外の 8 種類)の位置関係による価値を用いた．ただし，位置関係を見る場合，と金，杏(成香)，圭(成桂)，全(成銀)は金として扱う．また，王と 2 駒の関係は王と 1 駒の関係も内包している．評価関数の値には Bonanza14)のものを用いた．

4.2 実験内容

探索パラメータとして学習するパラメータを表 1 に示す．ただし，F_1，F_2，F_3，A_w に関してはそれぞれ$\log F_1$，$\log F_2$，$\log F_3$，$\log A_w$として正規化を行った．また，PVS 探索と NegaScout 探索の不使用/使用[0/1]の 2 値を出力に持つパラメータPVS，NSは，最小生起確率nは 5[%]とした．ただし，PVS 探索と NegaScout 探索を両方使用する場合，NegaScout 探索を優先して使用した．

これらのパラメータを Cross-Entropy Method を用いて学習する．それぞれのパラメータの取りうる範囲と初期平均と初期標準偏差を表 1 に加えた．対局相手は筆者が手調整を施した探索パラメータの HIMMEL とし，その値を表 1 に用いた探索パラメータ}に加えた．サンプルの数は 100 個，エリートサンプル率は 10[%]とした．対局時の持ち時間は 1 手 10 秒とし，バッチサイズは 10，20，50 で実験を行った．

4.3 実験結果

探索パラメータを Cross-Entropy Method で学習したときの結果を表 2，表 3，表 4 に示す．ただし，表 2 はバッチサイズ 10 で学習したもの，表 3 はバッチサイズ 20 で学習したもの，表 4 はバッチサイズ 50 で学習したものとする．

この表 2，表 3，表 4 より，サンプルの平均勝率の推移をグラフにしたものを図 3 に示す．ただし，(a)は横軸を更新回数としたもの，(b)は横軸を実験での対局総数としたものである．図 3 より更新を重ねるごとに勝率が良くなっていくことが分かる．

同様に，静止探索の深さの学習の推移をグラフにしたものを図 4 に，の学習の推移をグラフにしたものを図 5 に示す．

ただし，(a)は横軸を更新回数としたもの，(b)は横軸を実験での対局総数としたものである．また，(a)にはバッチサイズ 20 のみの標準偏差を，(b)には全ての標準偏差をエラーバーを用いて表した．

表 2，3，4 と図 4 より，Q_d，$\log F_1$，$\log A_w$については良い値と思われる数値に収束したことが分かる．反対に，表 2，3，4 と図 5 より，着手の評価値はほとんど収束しておらず，また各バッチサイズでの値もばらばらになった．また，NegaScout 探索は使用しない方が良い結果が得られた．

表 2　バッチサイズ 10 の学習結果

学習回数	0	1	2	3	4
Q_d	5.0	2.1	2.0	1.9	1.8
$\log F_1$	2.30	2.10	2.20	2.16	2.13
$\log F_2$	2.60	2.80	2.86	2.78	2.77
$\log F_3$	2.78	2.90	2.93	2.98	2.99
$\log A_w$	2.30	2.54	2.63	2.76	3.19
PVS	0.7	0.4	0.4	0.3	0.3
NS	0.7	0.3	0.3	0.0	0.0
O_1	8.0	8.0	5.5	4.9	5.7
O_2	8.0	6.4	7.2	8.1	8.2
O_3	8.0	8.4	7.8	4.7	3.3
O_4	8.0	9.1	10.7	10.6	10.1
O_5	8.0	10.2	9.6	7.6	8.6
サンプルの平均勝率[%]	40.5	59.4	68.7	78.8	
エリートの平均勝率[%]	88.0	98.5	100.0	100.0	

表 3　バッチサイズ 20 の学習結果

学習回数	0	1	2	3	4
Q_d	5.0	2.2	1.9	2.0	1.7
$\log F_1$	2.30	2.10	2.27	2.33	2.36
$\log F_2$	2.60	2.84	2.99	2.97	2.98
$\log F_3$	2.78	2.94	2.68	2.60	2.81
$\log A_w$	2.30	2.46	2.73	2.87	2.77
PVS	0.7	0.5	0.4	0.5	0.5
NS	0.7	0.3	0.0	0.0	0.0
O_1	8.0	7.6	7.4	8.9	7.4
O_2	8.0	6.4	6.7	7.1	7.5
O_3	8.0	7.3	7.5	7.4	9.2
O_4	8.0	9.1	7.8	8.2	10.9
O_5	8.0	9.4	10.0	10.9	9.6
サンプルの平均勝率[%]	43.4	61.3	73.0	75.9	
エリートの平均勝率[%]	87.3	95.0	97.0	98.3	

表 4　バッチサイズ 50 の学習結果

学習回数	0	1	2	3	4
Q_d	5.0	2.0	1.8	1.5	1.2
$\log F_1$	2.30	2.23	2.39	2.59	2.60
$\log F_2$	2.60	2.79	2.99	3.05	3.09
$\log F_3$	2.78	2.94	2.98	3.10	3.24
$\log A_w$	2.30	2.48	2.84	2.96	3.08
PVS	0.7	0.5	0.5	0.5	0.7
NS	0.7	0.2	0.0	0.0	0.0
O_1	8.0	7.1	7.8	8.9	9.3
O_2	8.0	7.4	8.1	8.4	8.9
O_3	8.0	9.7	11.1	11.5	12.1
O_4	8.0	7.6	8.2	8.9	9.6
O_5	8.0	8.6	7.9	9.6	9.8
サンプルの平均勝率[%]	42.9	64.8	75.6	84.4	
エリートの平均勝率[%]	84.6	94.6	97.1	99.6	

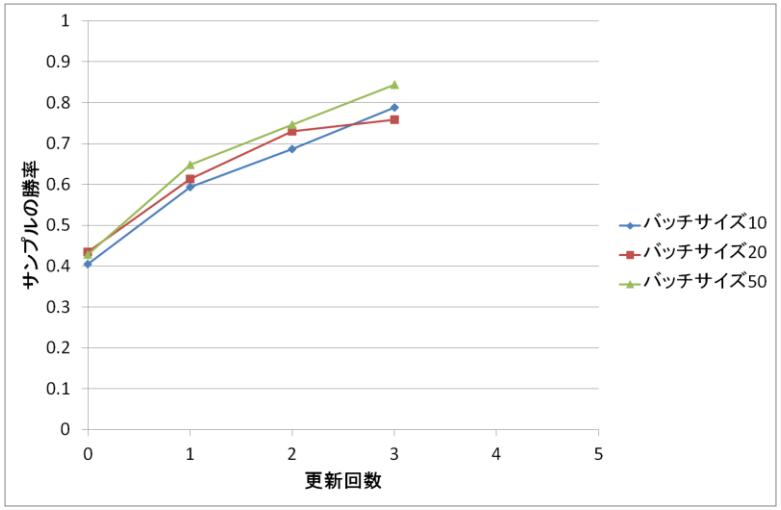

図 3(a) サンプルの平均勝率(横軸:更新回数)

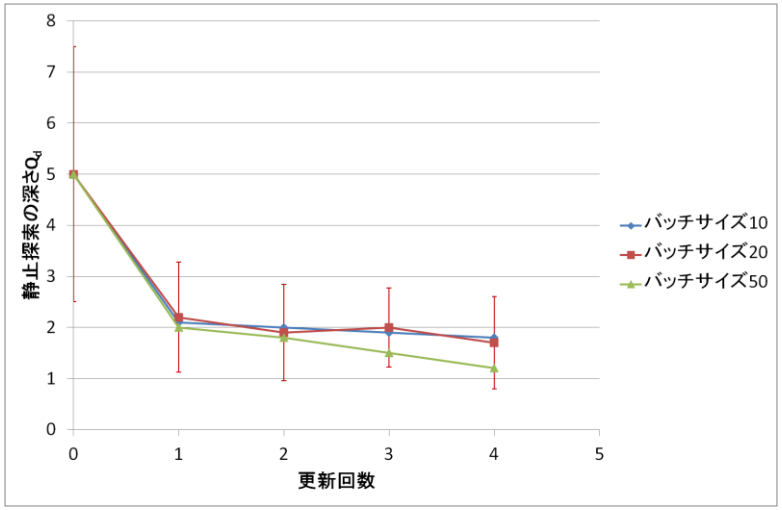

図 4(a) 静止探索の深さ Q_d の学習結果(横軸:更新回数)

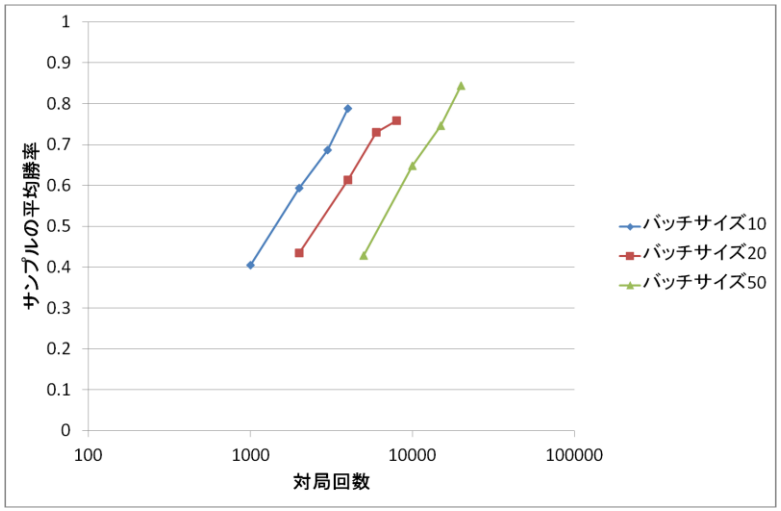

図 3(b) サンプルの平均勝率(横軸:対局数)

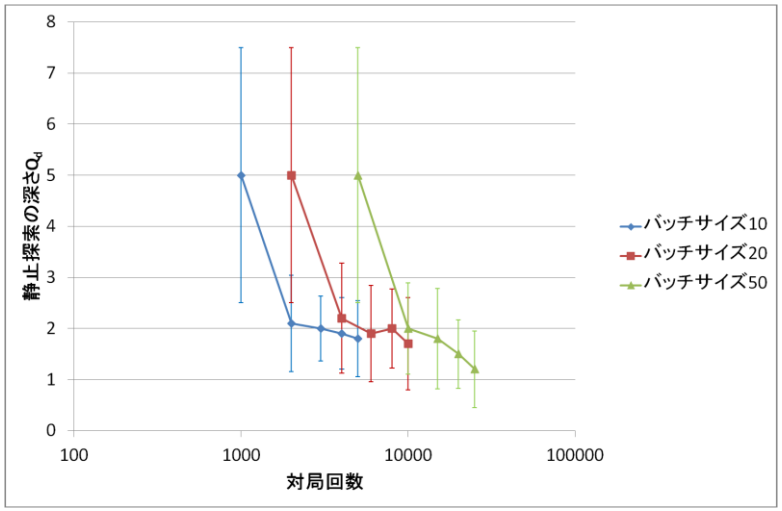

図 4(b) 静止探索の深さ Q_d の学習結果(横軸:対局数)

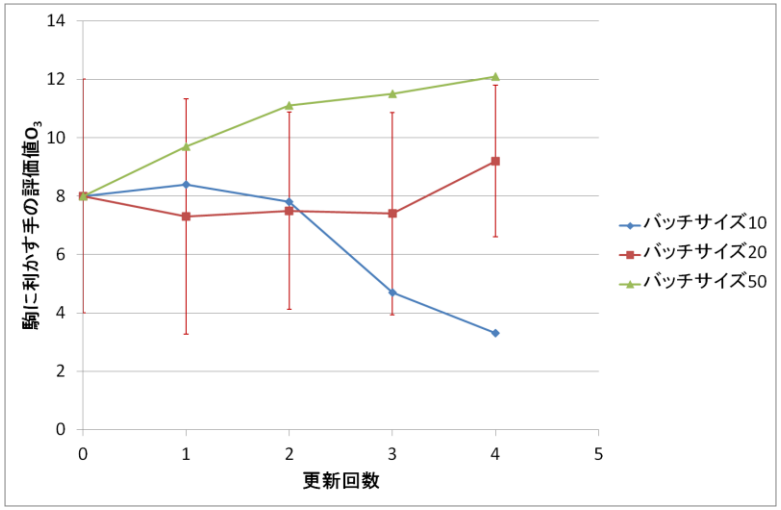

図 5(a) 駒に利かす手の評価値 O_3 の学習結果(横軸:更新回数)

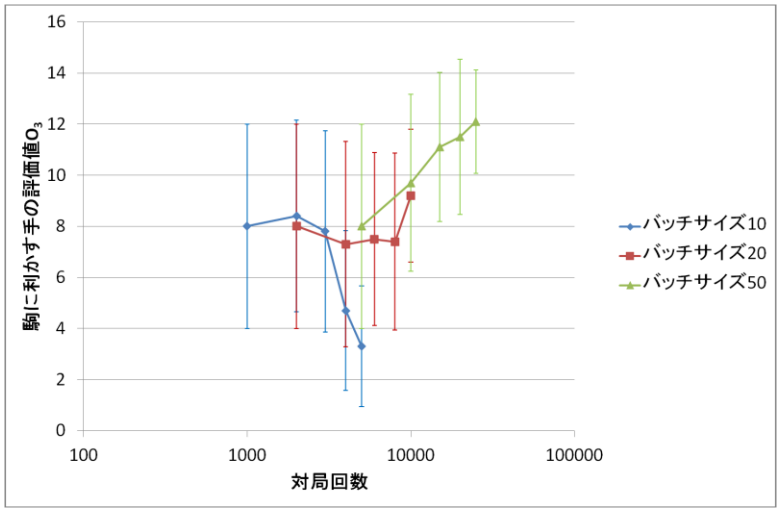

図 5(b) 駒に利かす手の評価値 O_3 の学習結果(横軸:対局数)

5. 結論

本研究では，様々な探索手法のパラメータを Cross-Entropy Method により学習した．学習実験の結果，静止探索の深さQ_d, Futility Pruning の深さ 1 のマージンF_1, Aspiration 探索のウインドウ幅A_wについては良い値と思われる数値に収束した．

一方で，着手の種類ごとの評価値はほとんど収束しておらず，課題が残る結果となった．手調整のパラメータに対し，4 回目の更新後のバッチサイズ 10 のサンプルは 771 勝 195 敗 34 分け，バッチサイズ 20 のサンプルは 1474 勝 440 敗 86 分け，バッチサイズ 50 のサンプルは 4136 勝 683 敗 181 分けと有意に勝ち越すことが出来た．これは，対局相手である手調整のプログラムの静止探索の深さの調整が不十分だったことが主な原因だと考えられる．静止探索が浅い方がよい結果を得られた理由として，結果的により深く探索出来ることや，静止探索内でα β探索を行っているので大まかな評価値の変化は浅い静止探索でも計算出来ることが考えられる．

また，NegaScout 探索はあまり効果がないという結果だったが，PVS 探索と NegaScout 探索の効率にはムーブオーダリングの精度が関わっているため，学習を繰り返して様子を見る必要がある．その他のパラメータもまだ標準偏差が大きく，バッチサイズごとの値もばらばらなため，学習回数がまだ不十分であったといえるだろう．

今回の実験で用いた対局相手である手調整パラメータの HIMMEL は静止探索の深さQ_dが 5 であったが，実験で静止探索が 1 から 3 のものに対して大きく負け越していたことから手調整のパラメータの強さが不十分であったことが考えられる．対局相手の強さを改善するために，静止探索の深さQ_dのパラメータを 1 から 3 にして実験を行えば，他のパラメータの学習も更に進むことが期待できる．また，多くのパラメータが収束しておらず，パラメータの学習回数が不十分だったと思われるので，もっと学習回数を増やす必要がある．

参考文献

1) Knuth, D.E. Moore, R.W.: An Analysis of Alpha-beta Pruning, Artificial Intell, Vol.6, pp.293-326 (1975).
2) Hermann Kaindl: Quiescence Search in Computer Chess, in Computer-Game-Playing; Theory and Practice, pp.39-52 (1983).
3) Reinefeld, A. : An Improement of the Scout Tree-Search Algorithm, ICCA Journal6(4), pp4-14 (1983).
4) 保木邦仁: コンピュータ将棋における全幅探索と futility pruning の応用, 情報処理学会誌, Vol.47, No.8, pp.884-892 (2006).
5) 伊藤裕，橋本剛，橋本隼一: 動的なマージンを用いる Futility Pruning, 第 12 回ゲームプログラミングワークショップ, pp.1-8 (2007).
6) 小幡拓弥，伊藤毅志: α β探索における探索順序の自動学習，情報処理学会研究報告，

Vol.2009-GI-21, pp.49--54 (2009).
7) 鈴木豪, 乾伸雄, 小谷善行: 将棋におけるゲーム木探索アルゴリズムの比較, 情報処理学会研究報告 Vol.1999-GI-1, pp.79-84 (1999).
8) Rubinstein, R. Y. : Cross-Entropy for combinatorial and continuous optimization, Methodology and Computing in Applied Probability, Vol.1 No.2, pp.127-190 (1999).
9) Guillaume M.J-B. Chaslot, Mark H.M. Winands, Istvan Szita, H. Jaap van den Herik: Cross-Entropy for Monte-Carlo Tree Search, ICCA Journal Vol.31 No.3, pp.145-156 (2008).
10) Reza Shams, Hermann Kaindl, Helmut Horacek: Using Aspiration Windows for Minimax Algorithms, IJCAI 1991, Vol.1, pp.192-197 (1991).
11) Campbell, M.S. Marsland, T.A.: Parallel Search of Strongly Ordered Game Tree, Computing Surveys 14(4), pp533-551 (1982).
12) 小谷善行: コンピュータ将棋の頭脳, サイエンス社 (2007).
13) 石川裕之, 乾伸雄, 小谷善行: 将棋におけるトランスポジションテーブルを用いた探索の効率化, 全国大会講演論文集第 54 回, pp.97-98 (1997).
14) 保木邦仁: 局面評価の学習を目指した探索結果の最適制御, 第 11 回ゲームプログラミングワークショップ, pp.78-83 (2006).