確率的選択探索の将棋への適用

桐井杏樹[†1]　原悠一[†1]　五十嵐治一[†1]　森岡祐一　山本一将[†2]

概要：本論文では，Boltzmann 分布を用いた Softmax 探索に基づく選択探索の一方式を提案する．本探索方式はモンテカルロ木探索の一種と見なすこともできるが，葉局面の評価に playout ではなく局面評価関数のみを用いる．本探索方式ではルートノードからの確率的選択によるモンテカルロ・サンプリングの反復により探索木を生長させる．また，サンプリングごとにスレッドを割り当てることにより，非同期な並列化が容易に実現できる．さらに，局面評価関数の学習も探索時に生成された探索木に対して探索時と同様なモンテカルロ・サンプリングの反復により並列的に実現できる．本探索方式の有効性については，将棋において実際に予備的な探索実験を行うことにより検証した．

キーワード：コンピュータ将棋，選択探索，モンテカルロ木探索，Softmax 探索，ボルツマン分布

Stochastic Selective Search Applied to Shogi

AZUKI KIRII[†1]　YUICHI HARA[†1]　HARUKAZU IGARASHI[†1]
YUICHI MORIOKA　KAZUMASA YAMAMOTO[†2]

Abstract: This paper proposes a stochastic selective-search algorithm based on softmax search using a Boltzmann distribution function that is classified to the Monte Carlo Tree Search algorithm. It does not use a playout; it uses a positional evaluation function for evaluating leaf nodes. We developed a search tree from a root node by iterative Monte Carlo sampling. Asynchronous parallel processing can be easily realized by assigning a thread to each sampling procedure. Moreover, evaluation functions can be learned by applying the same sampling method as in looking for search trees. In this paper, in our preliminary experiments we applied our search algorithm to *shogi*, a Japanese version of chess, and reported the results.

Keywords: Computer shogi, Selective search, Monte Carlo Tree Search, Softmax search, Boltzmann distribution

1. はじめに

現在のコンピュータ将棋においては，探索木を一定の深さまで全幅探索(full-width tree search)することを基本とする探索方式が主流である．そこでは葉局面(leaf node)の局面評価値を Minimax 演算により上位ノードへ伝搬させ，最善応手手順を計算する Minimax 探索を基本とする．しかし，局面評価関数中のパラメータを学習しようと勾配を計算する際には，Minimax 演算やα(またはβ)カットなどの枝刈り処理は妨げになってしまう．そこで利用されたのが Softmax 探索である[1]．文献[1]では探索木内部のノードの評価値は，そのノードの全ての子ノードの評価値を選択確率の値で重み付き平均を取った期待値で定義されている．このときの子ノード選択確率は評価値を用いた Boltzmann 分布を用いる．

したがって，途中で枝刈を行うのではなく，ルートノード(root node)から確率的に子ノードを選択しながら探索木を成長させて行く「選択探索」の方が Softmax 探索には向いている．この考えに従って木を成長させ，最良優先探索を行う探索方法が試みられた[2]．ただし，この研究ではルートノードから対象ノードへ至る経路上での選択確率の積を「実現確率」[a]と定義し，実現確率による探索深さの制御を行っていた．

しかし，この方法では子ノードの選択は決定論的であり，最良でない子ノードは探索が後回しになり，浅い探索でも発見可能な最善手を逃してしまう場合も多かった．また，探索処理の並列化にも Softmax 探索の利点を特別に生かすことはなく，従来のα β探索で用いられている並列処理をそのまま適用しただけであった．例えば，「GPS 将棋」で提案された兄弟ノードにスレッドを割り当てる P-GPP 法(pipeline-game position parallerization)[3][4]をベースにした並列探索方式[2]や，深さごとにスレッドを割り当てる Lazy SMP 法をベースに実現確率の閾値ごとにスレッドを割り当てる並列探索方式である[5].

そこで，本研究では Softmax 探索において，ルートノードから子ノードの選択を確率的に行いながら末端のノードを一段階だけ展開する操作の反復により，探索木を徐々に成長させて行く方式を提案する．このルートノードから末端ノードへの探索は一種のサンプリングとみなすことができる．この確率的な選択探索を用いたサンプリングは，囲

†1 芝浦工業大学工学部情報工学科
Shibaura Institute of Technology
†2 (株) コスモ・ウェブ
Cosmoweb Co., Ltd.

a) 「激指」で古くから提案されている実現確率とは定義が若干異なり，実現確率を構成する子ノード選択確率は局面評価関数だけから計算する．

碁で大成功を収めたモンテカルロ木探索(Monte Carlo Tree Search, MCTS)と類似性があり，探索処理の並列化も複数スレッドがルートノードからのサンプリングを非同期に行えば容易に実現できる．

本論文の構成は以下の通りである．2 章では提案手法と MCTS の代表例である UCT(Upper Confidence bounds applied to Trees)やその変形手法である UCT_H や $UCTMAX_H$ との関連の他，これまでに MCTS を将棋へ適用した研究報告との関連を述べる．3 章では提案手法の詳細を，4 章では提案手法の評価実験の結果と考察を述べる．さらに 5 章では，本探索手法で生成された探索木に対して，探索時と同様なサンプリングに基づく局面評価関数の学習方式を提案する．

2. 関連研究

2.1 選択探索において確率分布を用いた関連研究

将棋への適用ではないが，選択探索にランダム性を導入して並列処理に役立てた方法として RBFM(Randomized Best-First Minimax Search)[6]がある．これは子ノードの評価値を確率変数とみなし，それらの実現値に対して Minimax 探索を行う．これ以外にも子ノードの値に確率分布を仮定する研究はいくつかあるが，3 章で述べる提案手法では，子ノードの評価値の推定に確率分布を仮定するのではなく，子ノードを選択する確率的方策として確率分布を用いる点が異なっている[1][2].

2.2 モンテカルロ木探索との関連

囲碁で大成功したモンテカルロ木探索(MCTS)は，探索木を選択的に降りて行き，葉局面(leaf node)を評価・展開しながら徐々に木を成長させる[7]．この点は本研究での提案手法と共通点がある．モンテカルロ木探索は将棋へも適用が試みられているが，殆どはノードの選択に UCT を用いている[8][9][10]．UCT でのノード選択は決定論的であり，本提案手法の確率的探索とは異なる．さらに，通常の MCTS においては，葉局面での評価には playout を行うが，本提案手法では静止探索とそれで得られた局面評価関数だけを用いる．また，将棋では並列処理をも組み込んだ MCTS の研究例は少ない．

また，UCT の改良法として UCT_H と $UCTMAX_H$ とが Ramanujan & Selman により提案されている[11]．前者は UCT において，探索木の葉局面の評価のために playout ではなくヒューリスティックな局面評価関数を用いる方法である．後者は UCT_H に加えて，葉局面からルートノードへの経路上のノード評価値を書き換えるバックアップ操作[b]の際に，ノードの評価値をその子ノードの評価値に対して Minimax 演算により決定する方法である[c]．彼らはこれら 2 つの方法を Mancala と呼ばれる 2 人零和ゲームへ適用し，$UCTMAX_H$ が Minimax 探索や UCT，UCT_H よりも効果的な探索法であったことを報告している[11].

3 章で述べる提案手法は，葉局面の評価に playout の代わりに局面評価関数を用いる点は UCT_H と同じである．しかし，子ノードの選択が確率的である点が異なる．また，本提案方法において温度パラメータを低く設定すると，確率的なノード選択方式とバックアップ操作は $UCTMAX_H$ の Minimax 演算に近づく．

3. 本研究で提案する探索手法

3.1 全体の流れ

本研究で提案する探索手法はノード評価値の Minimax 演算ではなく，確率分布に基づいて確率的なノード選択を行う Softmax 探索を使用する．ルートノードから確率的に子ノードを選択して末端ノードまで降りて行く過程は，確率過程におけるモンテカルロサンプリングと見なすことができる．そこで，本研究で提案する手法を「モンテカルロ Softmax 探索」(Monte Carlo Softmax Search, MCSS)と呼ぶ．この探索の流れを以下に示す（図 1 参照）[12][d].

①初期設定：現在の探索ノード v にルートノードを設定
②ノード選択：ノード v から選択確率に従って子ノード child(v)を選択
③ノード展開：child(v)が展開済みであれば v を child(v)に置き換えて②へ，未展開であれば一段階だけ子ノードを全て展開し，静止探索で得られたそれぞれの局面の評価値を局面評価関数により計算
④バックアップ：ルートノードから v への経路を逆にたどり，経路上のノード評価値を更新

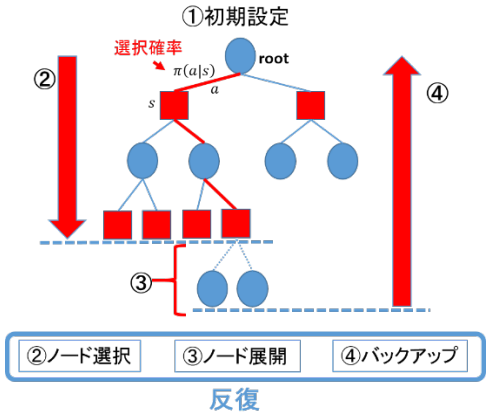

図 1　提案する探索アルゴリズムの流れ
Figure 1　Flow of the proposed search algorithm.

b)　Value Back-up または Backpropagation とも呼ばれる．
c)　これは Minimax 探索における探索木内部のノード評価値の計算法であり，Minimax 探索とのハイブリッドな方法だと考案者は称している[10].

d) 2017 年の第 27 回世界コンピュータ将棋選手権に初出場した「芝浦将棋 Softmax」は MCSS を採用していた．

この①～④を繰返して探索木を徐々に成長させる．探索終了後に本探索手法における最善応手手順を決定するには，ルートノードから確率値の最も高い子ノードを順々に選択すればよい．また，①～④の一連の処理を一つのスレッドに割当てて，次々にスレッドを実行させれば簡単に非同期な並列化が可能である．

3.2 Boltzmann 分布による確率的なノード選択

本提案手法では，探索木中のノードを選択する際に使用する確率分布として，以下の Boltzmann 分布を用いる[1].

$$\pi(a|s) = exp(E_s(v(a;s))/T)/Z \quad (1)$$

$$Z \equiv \sum_{x \in A(s)} exp(E_s(v(x;s))/T) \quad (2)$$

ただし，$E_s(v(a;s))$は局面 s において指し手 a により生成される子ノード$v(a;s)$の評価値, $A(s)$は局面 s における合法手の集合，T は温度パラメータである．また，ノードの評価値$E_s(s)$は, s が葉局面であれば局面評価関数が呼ばれる．葉局面でなければ, $E_s(s)$は子ノード$v(x;s)$の評価値$E_s(v)$を用いて次のように再帰的に定義されている[1][2],

$$E_s(s) = \sum_{x \in A(s)} \pi(x|s)\, E_s(v(x;s)). \quad (3)$$

ここで，子ノードの確率的選択に(1)の Boltzmann 分布を用いる理由は次の通りである．Boltzmann 分布に従う選択は, それまでに得た探索結果の情報（子ノードの評価結果）の下で，それら以外の情報，知識，予断などを用いない最も偏りのない公平なノード選択を与えるからである．このときの偏りが少ない基準としてエントロピー（の最大化）を用いている．Boltzmann 分布の導出とその意味については付録に記した．

3.3 温度パラメータによる探索深さと広がり

(1)(2)の温度パラメータ T の値により選択確率の分布が変化し，探索木の形が変化する．T の値が高いと全幅探索に，低くなると最良優先探索に近づく．したがって，T の値を高く設定すると，探索の幅は広がるが探索の深さは浅くなる．逆に T の値を低く設定すると，探索は深くなるが広がりはなくなる．

前者の場合，読みが浅くなってしまい，後者の場合は，読みは深いが浅いレベルでの有望な指し手を見逃しやすくなってしまう．温度の設定や調整により得られる探索木の形や棋力が変化すると考えられる．

4. 探索手法の評価実験

4.1 温度と棋力

温度と棋力の関係を調べるための対局実験を行った．実験は将棋と５五将棋で行った．将棋の場合には，いくつかの値に T を固定し，同じ相手と下記の条件で対局させ棋力を比較した．実験では，提案チーム，対局相手ともに静的局面評価関数として Bonanza (ver.6.0.0)の評価関数を使用した．ただし，末端ノードでの局面評価には静止探索を用いている．対局実験の結果を表 1 に示す.

【将棋での実験条件】
- 対局相手 : 芝浦将棋 Jr.（2015 年第 3 回将棋電王トーナメント 28 チーム中 16 位，$\alpha\beta$ 探索，スレッド数 1）
- 思考時間：1 手 10 秒に固定
- ponder：なし
- 対局数：300 局

表 1　将棋における対局実験の結果
Table 1　Game results in shogi.

スレッド数	温度 T	勝	負	分	勝率[e]
1	120	69	231	0	23.0%
	80	124	171	5	42.0%
	40	100	194	6	34.0%
6	80	162	129	9	55.7%

表 1 において，T=80 という値は，0.92×(歩 1 枚の値)[f]と等価である．表 1 の結果から温度 T の値によって棋力が影響を受けることがわかる．よって棋力を向上させるには適切な温度設定が必要である．また，並列化によって勝率が向上しており，並列化の効果は表れていると言える．

また，５五将棋への適用については，提案手法である MCSS を「GA 将 ver.9」[g]へ実装したプログラムをフリーソフトの ssp[h]と対局させた．局面評価関数は GA 将 ver.9 の評価関数[i]をそのまま用いた．さらに参考のために，実装前の GA 将 ver.9 と ssp の対局実験も行った．

【５五将棋での実験条件】
- 対局相手：ssp （スレッド数 1）
- 思考時間：1 手 1 秒に固定
- ponder：なし
- 対局数：1000 局
- 使用 CPU：Core i7 5960X

e) 勝率の計算には引き分け数は除いてある．
f) Bonanza ver.6.0 では歩 1 枚の価値は 87 である．
g) αβ 探索で，反復深化・Extended Futility Pruning・Null Move Pruning・LMR・Move Count Based Pruning・Scout 及び静止探索を実装している．
h) http://www.geocities.jp/shogi_depot/ssp.zip で公開されている．
i) 駒価値，2 駒・3 駒関係，手番，王将の移動可能範囲から構成．重みは PGLeaf の改良版の強化学習法を 57 万局の自己対局に適用して得た．

表 2　5 五将棋における対局実験の結果[j]

Table 2　Game results in 5×5 shogi.

スレッド数	温度 T'	勝	負	勝率
1	120	564	380	59.7%
	80	635	132	64.9%
	60	651	316	67.3%
	40	647	296	68.6%
	20	527	388	57.6%
16	120	740	220	77.3%
	60	773	193	80.0%
	40	737	234	73.1%
GA 将($\alpha\beta$ 探索)	―	838	132	86.4%

なお，表中の温度 T'=80 は，実際の温度 T ではなく，将棋の場合と同じく，0.92×（歩 1 枚の値）となる温度を表している．GA 将 ver.9 では歩 1 枚の価値は 3.25×10^{-7} としているので，実際の温度 T と T'との関係は，

$$T = 0.92\times3.25\times10^{-7}\times T'/80 \qquad (4)$$

である．表 2 からも，温度設定の重要性と並列化の効果を見てとることができる．

4.2 温度パラメータによる探索木の深さと広がりの変化

本節では温度パラメータ T と探索終了後の探索木の深さと広がりを間接的に知るために，深さ d ごとのノード数分布のグラフを作成した．図 2 は，将棋において T=120，80，40，1 の場合の探索木におけるノード数分布，図 3 は 5 五将棋において T'=120，80，60，40，20 の場合のノード数分布である．横軸が探索深さを表し，縦軸がそれぞれの深さごとのノード数を表している．グラフは 4.1 の 1 スレッドでの対局実験で作成された棋譜から，次の手順で作成した．

【深さごとのノード数分布グラフの作成手順】
①無作為に 1 局の棋譜を選択[k]
②各手番局面において温度ごとに探索木を生成[l]
③深さごとのノード数を全ての手番局面について集計
④集計数を局面数で割り，局面あたりの平均分布を計算[m]

図 2 と図 3 を見ると，温度が高い場合は探索木のノード数は浅い場所に多く分布し，温度が低い場合はその分布が深い場所へシフトしていることがわかる．したがって，温度が高い場合は探索木が浅い場所で広がり，低い場合はその広がりを抑えて枝が深く伸びて行く傾向があると言える．この温度による生成された探索木の形状の違いが，4.1 の対局実験で示されたように棋力に影響したと考えられる．

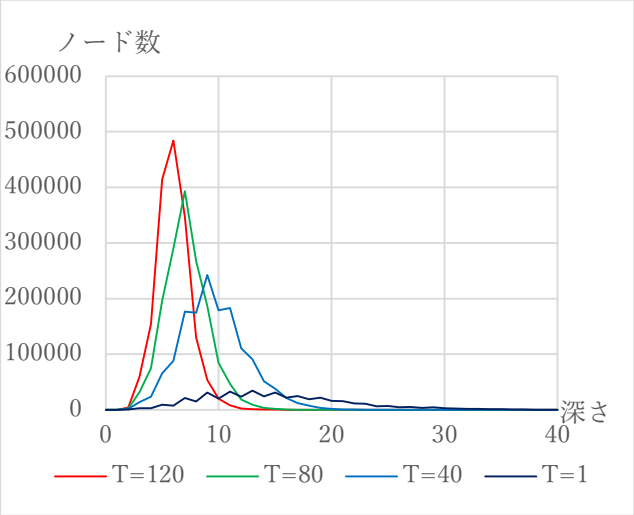

図 2　深さごとのノード数分布（将棋）

Figure 2 Distribution of nodes over search depth in shogi.

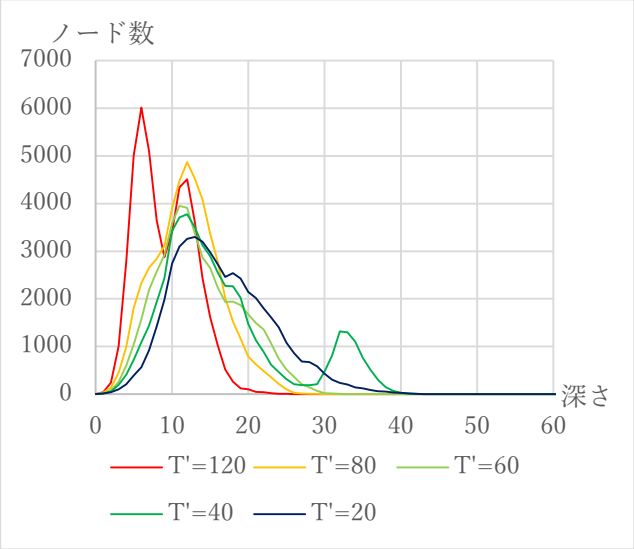

図 3　深さごとのノード数分布（5 五将棋）

Figure 3 Distribution of nodes over search depth in 5×5 shogi.

5. 本探索手法におけるサンプルベースの学習アルゴリズム

5.1 PG 期待値法

文献[1]では，本探索手法(MCSS)で用いた確率的選択探索により，固定した深さ D の範囲ですべてのノードを展開した全幅木を生成し，葉局面における静的局面評価関数中のパラメータ ω に関する方策勾配$\partial\pi/\partial\omega$を計算する強化学習法（PG 期待値法）を提案した[n]．その概要は以下のとおりである．

j) 表 2 の実験では千日手は引き分けとし，勝率の計算には除いてある．
k) 将棋は T=80，5 五将棋は T'=80 での対局実験の棋譜から選んだ．
l) 将棋は 1 手 10 秒，5 五将棋は 1 手 1.7 秒の探索とした．
m) 5 五将棋では手番局面数(=32)で割ってない値が図 3 に示されている．
n) 文献[1]では「PG 行動期待値法」と呼んでいた．

t 回目(t=1,2,…,L_a)の手番局面 u_t において学習エージェント A が指し手 a_t を選択する確率(方策)を

$$\pi_a\left(a_t\middle|u_t;\omega\right)=\exp\left(E_a\left(a_t,u_t;\omega\right)/T_a\right)/Z_a \tag{5}$$

$$Z_a\equiv\sum_{a'}\exp\left(E_a\left(a',u_t;\omega\right)/T_a\right) \tag{6}$$

とする．ただし，ωは評価関数中の学習パラメータ，T_aは温度パラメータである．$E_a(a_t,u_t;\omega)$は手番局面 u_tにおける指し手 a_tの評価を表す指標であり「指し手の評価値」と呼び，Boltzmann 分布における「目的関数」[o]とも呼ぶ．一方，対戦エージェント B の方策は$\pi_b(b_t|v_t)$で与えられ，既知の関数に固定されているとする．ただし，v_t は対戦エージェントの t 回目(t=1,2,…,L_b)の手番局面であり，b_t はそのときの指し手を表している．

一局の指し手と出現局面との時系列データ（棋譜）を「エピソード」と定義する．エピソード終了後，学習エージェントに報酬 r を与える．両対局者の指し手の決定は確率的方策によるので，学習エージェント A の指し手数（≡エピソード長 L_a）や報酬 r の値もエピソードごとに変動する．

方策勾配法[13][14]では一局当たりの期待報酬値 $E[r]$を極大化するように学習パラメータ ωを学習する．それによれば，$E[r]$の勾配ベクトルが

$$\partial E\left[r\right]/\partial\omega=E\left[r\sum_{t=1}^{L_a}e_\omega\left(t\right)\right] \tag{7}$$

$$e_\omega\left(t\right)\equiv\partial\ln\pi_a\left(a_t\middle|u_t;\omega\right)/\partial\omega \tag{8}$$

と表されることから，学習則として

$$\Delta\omega=\varepsilon\cdot r\sum_{t=1}^{L_a}e_\omega\left(t\right) \tag{9}$$

を用いる．ただし，ε は学習係数で小さな正数にとる．今，方策が(5)である場合，(8)の特徴的適正度 $e_\omega(t)$は

$$e_\omega\left(t\right)=\left(1/T_a\right)\Big[\partial E_a\left(a_t,u_t;\omega\right)/\partial\omega-\sum_{a'}\pi_a\left(a'\middle|u_t;\omega\right)\partial E_a\left(a',u_t;\omega\right)/\partial\omega\Big] \tag{10}$$

と表される．

指し手の評価は，読み（探索木の展開）を伴う方が精度が高いと考えられる．そこで，(5)の目的関数 $E_a(a_t,u_t;\omega)$ を，着手後の局面 v=$v(a_t,u_t)$ではなく，探索木 $G_D(a_t,u_t)$の葉局面の評価値を用いた関数とする．ここで，$G_D(a_t,u_t)$は局面 $v(a_t,u_t)$をルートノードとする深さ D の探索木で，学習エージェント A の手番から次の手番までを深さの 1 単位とし，

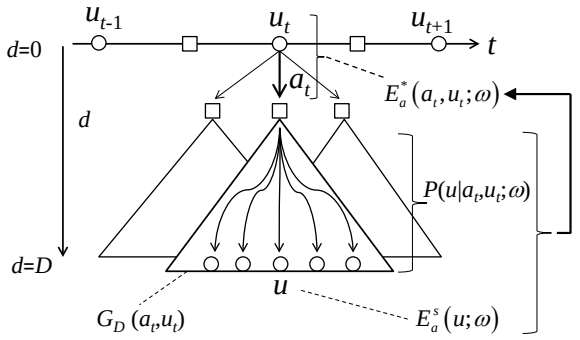

図 4　指し手評価の期待値$E_a^*(a_t,u_t;\omega)$と葉局面での静的局面評価値$E_a^s(u;\omega)$，遷移確率 $P(u|a_t,u_t;\omega)$の関係を表す[1].
Figure 4 Expected value of move a_t, $E_a^*(a_t,u_t;\omega)$, static evaluation function of state u, $E_a^s(u;\omega)$, and transition probability from u_t to u, $P(u|a_t,u_t;\omega)$[1].

出現局面 u_t を深さ 0 の，$G_D(a_t,u_t)$の葉局面を深さ D の A の手番局面とする．図 4 に模式図を示す．図中，t は時間順序，d は読みの深さ，$G_D(a_t,u_t)$は深さ D の部分探索木を表している．また，○印のノードは学習エージェント A の手番局面，□印のノードは対戦相手 B の手番局面を表している．

文献[1]では，以下の「指し手評価の期待値」，

$$E_a^*\left(a_t,u_t;\omega\right)\equiv\sum_{u\in U_D\left(a_t,u_t\right)}P\left(u\middle|a_t,u_t;\omega\right)E_a^s\left(u;\omega\right) \tag{11}$$

を(1)の目的関数 $E_a(a_t,u_t;\omega)$として用いることを提案している[p]．ただし，$U_D(a_t,u_t)$は探索木 G_D (a_t,u_t)の全葉局面の集合を，$E^s{}_a(u;\omega)$は葉局面 u での静的局面評価関数である（図 4）．

よって，学習エージェント A の方策(1),(2)は，

$$\pi_a\left(a_t\middle|u_t;\omega\right)=\exp\left(E_a^*\left(a_t,u_t;\omega\right)/T_a\right)/Z_a \tag{12}$$

$$Z_a\equiv\sum_{a'}\exp\left(E_a^*\left(a',u_t;\omega\right)/T_a\right) \tag{13}$$

と表される．このときの学習則は，(9),(10)より，

$$\Delta\omega=\varepsilon\cdot r\sum_{t=1}^{L_a}e_\omega\left(t\right) \tag{14}$$

$$e_\omega\left(t\right)=\left(1/T_a\right)\Big[\partial E_a^*\left(a_t,u_t;\omega\right)/\partial\omega-\sum_{a'}\pi_a\left(a'\middle|u_t;\omega\right)\partial E_a^*\left(a',u_t;\omega\right)/\partial\omega\Big] \tag{15}$$

(12)～(15)は，$E_a^*(a_t,u_t;\omega)$と$\partial E_a^*(a_t,u_t;\omega)/\partial\omega$の値が局面 u_t における合法な指し手 a についてすべて分かれば厳密な値が計算できる．

o) 統計物理学の分野では，この関数の符号を逆にした関数は体系のエネルギーを表す「エネルギー関数」であり，パラメータ T は温度である．
p) ここでは静止探索は考えていない． また，親ノードと子ノードの評価値の関係は(3)であることを仮定している．

5.2 PG 期待値法における再帰的な厳密解法アルゴリズム

文献[1]では(12)～(15)の$E_a^*(a_t,u_t;\omega)$と$\partial E_a^*(a_t,u_t;\omega)/\partial\omega$は再帰的に計算できることが示されている．まず，深さ d(0≦d≦D-1)における学習エージェント A の手番局面 $u^d{}_t$において，指し手 $a^d{}_t$が生成する相手の手番局面を $v^d{}_t$=v($a^d{}_t$,$u^d{}_t$)，その局面から相手が指し手 $b^d{}_t$ を指して得られた学習エージェントの手番局面を $u^{d+1}{}_t$=u($b^d{}_t$, $v^d{}_t$)とする（図 2）．

この時，探索深さ d の$E_a^*(a_t^d,u_t^d;\omega)$と$\partial E_a^*(a_t^d,u_t^d;\omega)/\partial\omega$は次のように再帰的に書ける[1]．

$$E_a^*(a_t^d,u_t^d;\omega)=\sum_{b_t^d}\pi_b(b_t^d|v_t^d)\cdot\sum_{a_t^{d+1}}\pi_a(a_t^{d+1}|u_t^{d+1};\omega)\cdot E_a^*(a_t^{d+1},u_t^{d+1};\omega) \tag{16}$$

$$\partial E_a^*(a_t^d,u_t^d;\omega)/\partial\omega=(\partial/\partial\omega)\sum_{b_t^d}\pi_b(b_t^d|v_t^d)\cdot\sum_{a_t^{d+1}}\pi_a(a_t^{d+1}|u_t^{d+1};\omega)\cdot E_a^*(a_t^{d+1},u_t^{d+1};\omega) \tag{17}$$

$$=\sum_{b_t^d}\pi_b(b_t^d|v_t^d)\sum_{a_t^{d+1}}\partial\pi_a(a_t^{d+1}|u_t^{d+1};\omega)/\partial\omega\cdot E_a^*(a_t^{d+1},u_t^{d+1};\omega)+\sum_{b_t^d}\pi_b(b_t^d|v_t^d)\sum_{a_t^{d+1}}\pi_a(a_t^{d+1}|u_t^{d+1};\omega)\cdot\partial E_a^*(a_t^{d+1},u_t^{d+1};\omega)/\partial\omega \tag{18}$$

$$=\sum_{b_t^d}\pi_b(b_t^d|v_t^d)\sum_{a_t^{d+1}}\pi_a(a_t^{d+1}|u_t^{d+1};\omega)\cdot\left[e_\omega(t,d+1)E_a^*(a_t^{d+1},u_t^{d+1};\omega)+\partial E_a^*(a_t^{d+1},u_t^{d+1};\omega)/\partial\omega\right] \tag{19}$$

ただし，

$$e_\omega(t,d+1)\equiv\partial\ln\pi_a(a_t^{d+1}|u_t^{d+1};\omega)/\partial\omega \tag{20}$$

$$=(1/T_a)\left[\partial E_a^*(a_t^{d+1},u_t^{d+1};\omega)/\partial\omega-\sum_{a'}\pi_a(a'|u_t^{d+1};\omega)\partial E_a^*(a',u_t^{d+1};\omega)/\partial\omega\right]. \tag{21}$$

また，(16),(17)における再帰の終端は，もし，$u^{d+1}{}_t$が葉局面，すなわち，d=D-1 ならば，

$$E_a^*(a_t^d,u_t^d;\omega)=\sum_{b_t^{D-1}}\pi_b(b_t^{D-1}|v_t^{D-1})E_a^s(u_t^D;\omega) \tag{22}$$

$$\partial E_a^*(a_t^d,u_t^d;\omega)/\partial\omega=\sum_{b_t^{D-1}}\pi_b(b_t^{D-1}|v_t^{D-1})\cdot\partial E_a^s(u_t^D;\omega)/\partial\omega \tag{23}$$

と書ける．図 5 に上記の依存関係を表した模式図を示す．しかし，上記の学習法(16)～(23)では，$E_a^*(a_t,u_t;\omega)$と$\partial E_a^*(a_t,u_t;\omega)/\partial\omega$の値は局面 u_t において指し手 a を指した局面以下の部分木 $G_D(a,u_t)$の全葉局面 $u\in U_D(a,u_t)$に依存する．つまり，一定の深さ D までのすべてのノードを展開する必要があり，計算量の点で難がある．そこで，(12)～(15)における期待値計算を一切行わないで，探索木の最善応手手順(Principal Variation, PV)の葉局面の静的局面評価値で近似したのが「PGLeaf 法」(Policy Gradient Leaf Method) [15]である[q]．ただし，PGLeaf 法でも，ルートノードからの全合法手に対する PV を計算する必要がある[r].

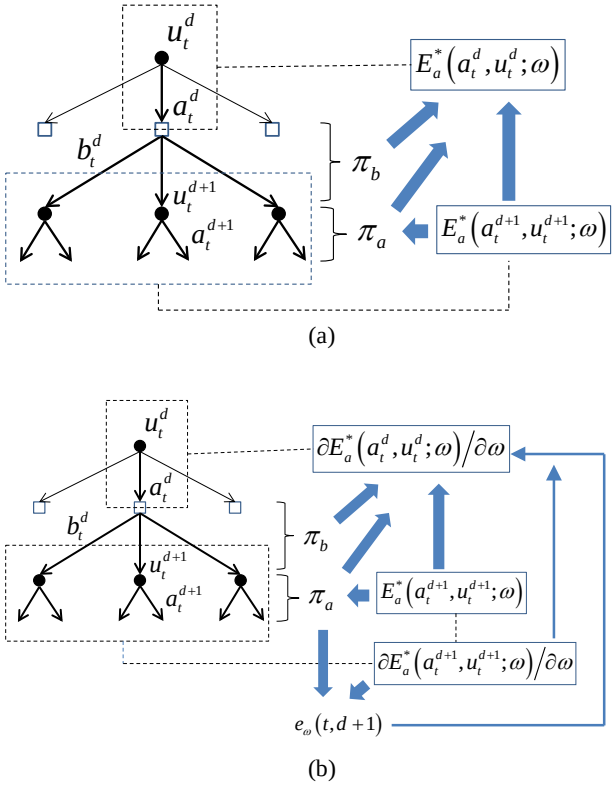

図 5　PG 期待値法の再帰計算における依存関係[1]：(a)指し手評価の期待値$E_a^*(a_t^d,u_t^d;\omega)$，(b) 1 階微係数の値$\partial E_a^*(a_t^d,u_t^d;\omega)/\partial\omega$.

Figure 5　Recursive relations in “PG expectation method”[1] for (a)the expectation values of moves, and (b) the first derivatives.

5.3 MCSS におけるサンプルベースの学習アルゴリズムの導出

本節では前節で示した再帰的な計算式(16)～(21)をそのまま計算するのではなく，探索木に対するサンプリングにより計算する学習アルゴリズムを提案する．そこで，(19)の右辺に(21)を代入し，さらに次のように変形する．

$$\sum_{b_t^d}\pi_b(b_t^d|v_t^d)\sum_{a_t^{d+1}}\pi_a(a_t^{d+1}|u_t^{d+1};\omega)\cdot\left[e_\omega(t,d+1)E_a^*(a_t^{d+1},u_t^{d+1};\omega)+\partial E_a^*(a_t^{d+1},u_t^{d+1};\omega)/\partial\omega\right] \tag{24}$$

$$=\sum_{b_t^d}\pi_b(b_t^d|v_t^d)\sum_{a_t^{d+1}}\pi_a(a_t^{d+1}|u_t^{d+1};\omega)\cdot \tag{25}$$

q) (7)の右辺で $P(u^*{}_D(a_t,u_t)|a_t,u_t)$=1 とし，他の遷移確率 $P(u|a_t,u_t)$は 0 と置いたことに相当する．

r) この事情は「Bonanza メソッド」でも同じである．

$$\left[(1/T_a)\left[\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)/\partial\omega-\sum_{a'}\pi_a\left(a'\middle|u_t^{d+1};\omega\right)\partial E_a^*\left(a',u_t^{d+1};\omega\right)/\partial\omega\right]\right.$$
$$\left.\cdot E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)+\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)/\partial\omega\right]$$
$$=\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)\cdot \quad (26)$$

$$\left[(1/T_a)E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)/\partial\omega+\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)/\partial\omega\right]$$
$$-\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)\cdot$$
$$(1/T_a)\sum_{a'}\pi_a\left(a'\middle|u_t^{d+1};\omega\right)\partial E_a^*\left(a',u_t^{d+1};\omega\right)/\partial\omega$$
$$=\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)\cdot \quad (27)$$

$$\left[(1/T_a)E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)/\partial\omega+\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)/\partial\omega\right]$$
$$-\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)E_a^*\left(u_t^{d+1};\omega\right)\cdot(1/T_a)\sum_{a'}\pi_a\left(a'\middle|u_t^{d+1};\omega\right)\partial E_a^*\left(a',u_t^{d+1};\omega\right)/\partial\omega$$
$$\left[(\because)\sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)=E_a^*\left(u_t^{d+1};\omega\right)\right]$$
$$=\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)\cdot \quad (28)$$

$$\left[(1/T_a)\left[E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)-E_a^*\left(u_t^{d+1};\omega\right)\right]+1\right]\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)/\partial\omega$$

したがって，(19)は以下のように表すことができる．

$$\partial E_a^*\left(a_t^d,u_t^d;\omega\right)/\partial\omega$$
$$=\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)\cdot \quad (29)$$
$$\left[(1/T_a)\left[E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)-E_a^*\left(u_t^{d+1};\omega\right)\right]+1\right]\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)/\partial\omega$$

(16)と(29)より，$E_a^*(a_t^d,u_t^d;\omega)$と$\partial E_a^*(a_t^d,u_t^d;\omega)/\partial\omega$はモンテカルロ・サンプリングにより得られたサンプル（ルートノードから葉局面までの経路上）の情報[s]から，数値的に計算できることが分かる．また，探索木の生成を 3 章で提案した MCSS アルゴリズムにおいては，探索時に各ノードの評価値は計算済であるので，上記の学習は高速に計算できる可能性がある．

6. おわりに

本論文では，Boltzmann 分布を用いた Softmax 探索に基づく選択探索の一方式を提案した．本探索方式はモンテカルロ木探索の一種で，葉局面の評価に playout ではなく局面評価関数のみを用いる．本探索方式ではルートノードからの確率的選択によるモンテカルロ・サンプリングの反復により探索木を生長させる．また，サンプリングごとにスレッドを割り当てることにより，非同期な並列化が容易に実現できる．実際に，並列化したプログラムを作成し，対局実験を行い，効果を検証することができた．

さらに，本探索方式であれば，局面評価関数の学習も探索時に生成された探索木に対して探索時と同様なモンテカルロ・サンプリングの反復により並列的に実現できる．本論文では具体的にその学習則を導出した．

今後は，探索深さによる温度調整の方法を研究するとともに，最良優先探索と組み合わせたより深い探索の方法を研究していく予定である．

s) (29)の右辺中の[(1/T)[…]+1]の値を指す．leaf ノードから root ノードへ向けての経路上でこの値を次々に掛けて行けば良い．

参考文献

[1] 五十嵐治一，森岡祐一，山本一将．方策勾配法による静的局面評価関数の強化学習についての一考察．第 17 回ゲーム・プログラミング・ワークショップ(GPW2012) 論文集，pp.118-121.

[2] 原悠一，五十嵐治一，森岡祐一，山本一将．ソフトマックス戦略と実現確率による深さ制御を用いたシンプルなゲーム木探索方式．第 21 回ゲーム・プログラミング・ワークショップ(GPW20 16)論文集, pp.108-111.

[3] Yokoyama, S., Kaneko and Tanaka, T. Parameter-Free Tree Style Pipeline in Asynchronous Parallel Game-Tree Search. Advances in Computer Games. Vol. 9525 of the Series Lecture Notes in Computer Science, 2015, p. 210-222.

[4] 横山秀，金子知適．評価値を用いて展開制御したゲーム木に基づく並列探索． 第 20 回ゲーム・プログラミングワークショップ(GPW2015) 論文集, pp. 46-53.

[5] 原悠一．ソフトマックス戦略と実現確率による深さ制御を用いたシンプルなゲーム木探索．2016 年度芝浦工業大学理工学研究科修士論文.

[6] Shoham, Y. and Toledo, S. Parallel Randomized Best-First Minimax Search. Aritificial Intelligence, 2002, vol.137, pp.165-196.

[7] Browne, C. B., et al. A Survey of Monte Carlo Tree Search Methods. IEEE Trans. of Comp. Intell. and AI in Games, 2012, vol. 4, no. 1, pp.1-43.

[8] 佐藤佳州，高橋大介．モンテカルロ木探索によるコンピュータ将棋．情報処理学会論文誌，2009, vol. 50, no. 11, pp.2740-2751.

[9] 竹内聖悟，金子知適，山口和紀．将棋における評価関数を用いたモンテカルロ木探索．第 15 回ゲームプログラミングワークショップ(GPW2010) 論文集, pp. 86-89.

[10] 横山大作．モンテカルロ木探索アルゴリズムの将棋への適用，第 17 回ゲームプログラミングワークショップ (GPW2012)論文集，pp.76-83.

[11] Ramanujan, R and Selman, B. Trade-Offs in Sampling-Based Adversarial Planning. Proc. 21st Int. Conf. Automated Planning and Scheduling, 2011, pp.202-209.

[12] 芝浦将棋 Softmax のアピール文書，http://www2. com puter-shogi.org/wcsc27/appeal/Shibaura_Shougi_Softmax/ appeal.pdf, (参照 2017-10-02).

[13] Williams, R. J. Simple Statistical Gradient- Following Algorithms for Connectionist Reinforcement Learning. Machine Learning,

1992, vol.8, pp.229-256.
[14] 五十嵐治一，石原聖司，木村昌臣．非マルコフ決定過程における強化学習―特徴的適正度の統計的性質―．電子情報通信学会論文誌 D, 2007, vol.J90-D, no.9, pp.2271-2280.
[15] 森岡祐一，五十嵐治一．方策勾配法と $\alpha\beta$ 探索を組み合わせた強化学習アルゴリズムの提案．第 17 回ゲームプログラミングワークショップ(GPW2012)論文集，pp. 122-125.
[16] 小野 周．熱力学．岩波書店，1974，第 2 章，pp.67-71.

付録

付録 A.1 Boltzmann 分布の導出

熱力学や統計物理学の分野では，E. T. Jaynes が提唱した「最大エントロピー原理[t]」(Principle of maximum entropy)に基づいて，体系のエネルギーなどの物理量の計算に必要な確率分布関数が導出されている[16]．ここではエージェントが行動選択を行う際の確率的方策を導出するための原理として用いる．

今，エージェントがその状態で最適と思われる行動を選択する問題を考える．通常，最適な行動をあらかじめ知ることはできない．しかし，行動の良さに関する何らかの知識（E とする）はあるものとする．そこで，行動 i (i=1,2,…,n)の良さを E_i で表すことにする．E_i が大きければ大きいほど，行動 i は良いとする．この知識を使用して行動 i を p_i の確率で選択するものとする．

次に，エントロピーf を次の式で定義する．エントロピーは対象に関する知識の不確かさを表している．

$$f\left(p_1, p_2, ..., p_n\right) \equiv -\sum_{i=1}^{n} p_i \ln p_i \tag{A.1}$$

また，p_i は確率なので次の式を満たす．

$$\sum_{i=1}^{n} p_i = 1 \tag{A.2}$$

さらに，行動は確率的に選ぶが，ある程度の質は保証したい．そこで，$\{E_i\}$の期待値<E>をこちらで E_T と指定することにする．すなわち，

$$\left\langle E \right\rangle \equiv \sum_{i=1}^{n} p_i E_i = E_T \tag{A.3}$$

とする．ただし，E_Tは$\sum_{i=1}^{n} E_i/n \le E_T \le \max\{E_1, E_2, \ldots, E_n\}$の範囲で選ぶ．

ここで，(A.2)と(A.3)の制約条件の下で(A.1)で定義したエントロピーを最大にする$\{p_i\}$の分布を求めることを考える．すなわち，最も選択の不確定さが大きい分布が，最も偏見や予断の少ない公平な選択のための確率分布を与えると考える[u]．この分布を求めるためにラグランジュの未定乗数法(method of Lagrange multiplier)を用いる．

まず，次の目的関数 ϕ を定義する．

$$\phi\left(\{p_i\}, \lambda_1, \lambda_2\right) \equiv f\left(p_1, p_2, ..., p_n\right) + \lambda_1\left(\sum_{i=1}^{n} p_i - 1\right) + \lambda_2\left(\sum_{i=1}^{n} p_i E_i - E_T\right) \tag{A.4}$$

(A.4)の右辺における λ_1 と λ_2 は未定乗数である．次に，目的関数 ϕ の極値を与える次の方程式を考える[v].

$$\frac{\partial \phi}{\partial p_i} = \frac{\partial f}{\partial p_i} + \lambda_1 + \lambda_2 E_i = 0 \tag{A.5}$$

$$\frac{\partial \phi}{\partial \lambda_1} = \sum_{i=1}^{n} p_i - 1 = 0 \tag{A.6}$$

$$\frac{\partial \phi}{\partial \lambda_2} = \sum_{i=1}^{n} p_i E_i - E_T = 0 \tag{A.7}$$

(A.5)に(A.1)の定義を代入すると，

$$p_i = e^{\lambda_1 - 1 + \lambda_2 E_i} \tag{A.8}$$

を得る．これを(A.6)へ代入すると，

$$e^{\lambda_1 - 1} \sum_{i=1}^{n} e^{\lambda_2 E_i} = 1 \tag{A.9}$$

となる．ここで，$\lambda_2 \equiv 1/T$ とおくと，

$$e^{\lambda_1 - 1} = 1 \Big/ \sum_{i=1}^{n} e^{E_i/T} \tag{A.10}$$

と表される．これと $\lambda_2 = 1/T$ を(A.8)へ代入すると，Boltzmann 分布

$$p_i = e^{E_i/T} \Big/ \sum_{i=1}^{n} e^{E_i/T} \tag{A.11}$$

を得る[w]．また，温度パラメータ T は(A.11)を(A.7)へ代入した式から数値的に計算することができる．すなわち，分布$\{p_i\}$による E_i の期待値が与えられた値 E_T になるように温度パラメータ T が定まる．

t) 「最小偏見の原理」とも言われている．
u) 予断や偏見を加えることは何らかの知識を与えることであり，選択の不確定さ（＝エントロピー）が減ってしまう．ここでは$\{E_i\}$と２つの制約条件だけを判断のための知識として用いるものとする．
v) (A.6)と(A.7)は最初に仮定した(A.2)と(A.3)の制約条件を表している．
w) 統計物理学では E は体系のエネルギーを表し，低い方がエネルギー的に安定な状態を表す．その場合，$\lambda_2 \equiv -1/T$ と置くので，(A.11)の E_i にはマイナスの符号が付く．