Vol.2013-GI-30 No.6
2013/6/28

方策勾配法による局面評価関数とシミュレーション方策の学習

五十嵐治一[†1]，森岡祐一，山本一将[†2]

本論文では強化学習の一手法である方策勾配法をコンピュータ将棋に適用する方法を考察した．方策勾配法は，報酬や方策にマルコフ性の制限なく自由に設計することができるという大きなメリットがある．本論文では，最初に全 leaf 局面の局面評価値をその局面への遷移確率値で重み付けた期待値を用いた指し手評価方式を提案する．これをベースに，探索木の各ノードにおける指し手の選択法として Boltzmann 分布に基づくソフトマックス戦略を採用した場合の局面評価関数に含まれるパラメータの学習則を導出した．しかし，探索や学習時の計算量が膨大となるため，３つの近似計算法を考案した．次に，探索時にシミュレーション方策を用いてモンテカルロ探索を行う場合や，探索の深さを制御する場合のために，局面評価関数とシミュレーション方策の両者を同時に学習する学習則を方策勾配法により導出した．さらに，この方策勾配の計算法を利用すると，局面ごとに正解手が既知の場合の教師付学習も可能であることを示し，実際に学習則を導出した．

Learning Positional Evaluation Functions and Simulation Policies by Policy Gradient Algorithm

HARUKAZU IGARASHI[†1]　YUICHI MORIOKA　KAZUMASA YAMAMOTO[†2]

This paper applies policy gradient reinforcement learning to *shogi*, a traditional Japanese board game that resembles chess. First, we propose a move evaluation function, which is defined by the expectation of the values of all leaf nodes produced by the move in a search tree that is weighted by the transition probabilities to the leaf nodes from the root node produced by the move. Since policy gradient reinforcement learning does not require Markovian properties of reward functions and policies, system designers can create the rewards functions and policies more freely than when using other reinforcement learning methods that must be applied in Markov decision processes. The learning rules of the parameters in the positional evaluation function can be calculated recursively when the Boltzmann distribution function gives the probabilities of taking branches in a search tree. We also consider three approximation methods to reduce the computation time for tree searching and parameter learning. Second, we derived the learning rules for both positional evaluation functions and simulation policies for Monte-Carlo simulation search and controlling the search depth by the policy gradient algorithm. This approach can also be applied to supervised learning problems of a teacher's moves in a given position.

1. はじめに

近年，コンピュータ将棋の実力はプロ棋士に迫るものがある[1]．例えば，2013 年に行われた第 2 回電王戦では，プロ棋士 5 名とコンピュータ将棋ソフトのトップ 5 との対局が行われ，コンピュータ側の 3 勝 1 敗 1 分けという結果であった[2]．この一因となっているのが，将棋ソフトBonanzaで提案された評価関数の自動学習である[3]．現在では，評価関数をプロ棋士の棋譜データベースを利用した教師付き学習により構築することが主流となっている．

一方，教師付き学習ではなく強化学習により評価関数を学習する方法も考えられている．その代表的な強化学習法としてはTD(λ)法とTDLeaf(λ)法がある．TD(λ)法はバックギャモンでは大成功を収めており[4]，TDLeaf(λ)法はチェスにおいて棋力向上への有効性が確認されている[5]．しかし，将棋ではまだそれほど良い適用結果が報告されてはいない．

そこで本論文ではこれまでのTD(λ)法やTDLeaf(λ)法ではなく，“方策勾配法”と呼ばれる別の強化学習法を適用することを考えた．方策勾配法は報酬を自由に設定することができるので，棋力向上だけでなく棋風の学習など様々な学習目的に対して幅広く適用できる．

しかしながら，将棋のように着手決定の際に探索(読み)を要する問題では，方策関数の中に探索処理を取り入れるための工夫が必要である．そこで，本論文ではまず，着手決定の方策として，“指し手評価の期待値”を用いた確率的方策を提案する．この方策は，探索木のleaf局面の局面評価関数を用いて再帰的に表現されるので，最適方策の学習は局面評価関数中の特徴量パラメータの学習に帰着する．この学習則を導出した後，近似計算法としてPGLeaf法を初めとするいくつかの学習法を提案する．

次に，指し手評価の期待値を計算する際に，leaf局面への遷移確率の値を着手選択の方策とは別の“シミュレーション方策”(simulation policy)を用いて計算する場合を考える．これは，モンテカルロ探索のように局面や指し手の評価値を何らかのシミュレーションで決定する場合である．また，この遷移確率の値は“激指”[6]の“実現確率”[a]

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

a) 激指の実現確率の計算においてベースとなっている「指し手の遷移確率」は，その手を指すか指さないかの 2 値選択の選択確率であり，合法手の間での選択確率とは異なる．本論文では後者の場合を考えている．

に相当し，探索木の前向きの枝刈り処理に用いることも可能である．本論文では，このシミュレーション方策中のパラメータと局面評価関数中の特徴量パラメータの両方を同時に学習できる強化学習則を導出する．さらに，強化学習ではなく，その局面での正解手を与える教師付き学習の場合の学習則も同様な方法で導出できることを示す．

2. 方策勾配法による強化学習

2.1 方策勾配法とは

強化学習では，Q学習のように行動価値関数を通して，あるいはTD法のように状態価値関数を通して，間接的に方策を学習する価値ベースの強化学習法(value-based algorithm)がよく知られている[4]．一方，方策中にパラメータを入れておき，パラメータ空間内での期待報酬関数の最急勾配を計算することにより，方策を直接学習する強化学習法がある．WilliamsのREINFORCEアルゴリズム[7]や，部分観測マルコフ決定過程POMDP（Partially Observable Markov Decision Processes）環境における木村らの確率的傾斜法[8]などである．また，Q値を用いて上記の勾配関数を表現する方式[9][10][11]や，自然勾配を利用する方式[12]も考案されている．これら一連の方策ベースの強化学習法は，“方策勾配法”(policy gradient method)と呼ばれ，例えば，Petersらの文献[13]中に簡潔にまとめられている．

本研究では，五十嵐らが提案している方策勾配法[14]を用いる．この方式は，Williamsのエピソード単位の学習方式（episodic REINFORCE algorithm）[7]に基づいており，環境モデル（状態遷移確率と報酬）と方策に関する単純マルコフ性を必要としない．これまでにマルチエージェント学習の標準問題として知られている追跡問題や，粒子群を用いた最適化手法であるPSO (Particle Swarm Optimization)等へ適用され，有効性が確認されている[15][16].

2.2 方策勾配法による学習

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

$$\pi_a\left(a_t\middle|u_t;\omega\right)=\exp\left(E_a\left(a_t,u_t;\omega\right)\middle/T_a\right)\middle/Z_a \tag{1}$$

$$Z_a\equiv\sum_{x\in A(u_t)}\exp\left(E_a\left(x,u_t;\omega\right)\middle/T_a\right) \tag{2}$$

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，$A(u_t)$は A の手番局面 u_t における全ての合法手の集合である．(1)の右辺は Boltzmann 分布と呼ばれる確率分布関数であり，$E_a(a_t,u_t;\omega)$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”目的関数”と呼ぶ．一方，対戦エージェント B の方策 $\pi_b(b_t|v_t)$は既知であるとする．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．エピソード終了後に勝敗等を考慮して学習エージェントに報酬 r を与える．一般に，両対局者の指し手の決定は確率的方策によるものとする．したがって，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 の観測値もエピソードごとに変動する．

文献[14]の方策勾配法を適用して，一局当たりの期待報酬値 $E[r]$を極大化することを考える．そこでは，

$$\partial E[r]/\partial\omega=E\left[r\sum_{t=1}^{L_a}e_\omega(t)\right] \tag{3}$$

$$e_\omega(t)\equiv\partial\ln\pi_a\left(a_t\middle|u_t;\omega\right)\middle/\partial\omega \tag{4}$$

と表されることから，学習則として

$$\Delta\omega=\varepsilon\cdot r\sum_{t=1}^{L_a}e_\omega(t) \tag{5}$$

が用いられている．ε は学習係数で小さな正数にとる．(5)は報酬と実現手の選択確率の勾配との積（相関）に比例させて各パラメータを更新（強化）している．今，方策が(1)の場合，(4)の特徴的適正度 $e_\omega(t)$は，

$$e_\omega(t)=\left(1/T_a\right)\Big[\partial E_a\left(a_t,u_t;\omega\right)\middle/\partial\omega - \sum_{x\in A(u_t)}\pi_a\left(x\middle|u_t;\omega\right)\partial E_a\left(x,u_t;\omega\right)\middle/\partial\omega\Big] \tag{6}$$

と表される．本論文で用いる方策勾配法のアルゴリズムをまとめると次の様になる．

【方策勾配法の学習アルゴリズム】

step1: 各時刻 t において方策 $\pi_a(a_t|u_t;\omega)$により指し手 a_t を選択し，特徴的適正度 $e_\omega(t)$を計算する．

step2: エピソード終了後に報酬 r を与える．

step3: 学習則(5)により更新量$\Delta\omega$ を計算する．

step4: ω を $\omega+\Delta\omega$ と変更し，step1 から繰り返す．ただし，終了条件を満たせば終了する．

3. 探索と局面評価関数による指し手の評価

指し手の評価は，読み（探索木の展開）を伴う方がより正確と考えられる．そこで，(1)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 $G_D(a_t,u_t)$の末端の局面（以下，leaf 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 $v(a_t,u_t)$をルートとする深さ D の探索木である．ただし，学習エージェント A の手番から次の A の手番までを深さの 1 単位とし，出現局面 u_t を深さ 0 の，$G_D(a_t,u_t)$の leaf 局面を深さ D の A の手番局面とする．

本研究では，以下の“指し手評価の期待値”

$$E_a^*\left(a_t,u_t;\omega\right)\equiv\sum_{u\in U_D(a_t,u_t)}P\left(u\middle|a_t,u_t;\omega\right)E_a^s\left(u;\omega\right) \tag{7}$$

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する．

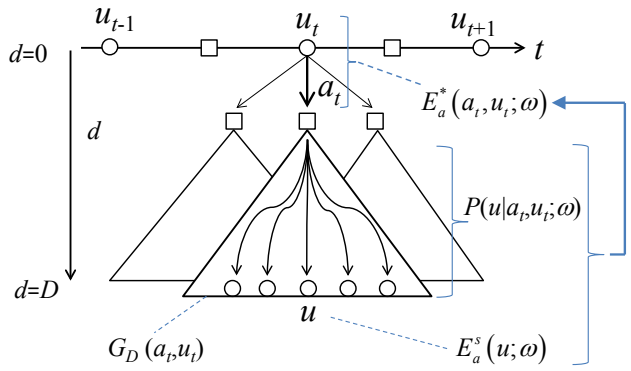


図 1　指し手評価の期待値 $E^*_a(a_t,u_t;\omega)$と leaf 局面での局面評価値 $E^s_a(u;\omega)$，遷移確率 $P(u|a_t,u_t;\omega)$の関係を表す．

Figure 1　Expected evaluation function $E^*_a(a_t,u_t;\omega)$ of move a_t, positional evaluation function $E^s_a(u;\omega)$ of leaf node u, and transition probability $P(u|a_t,u_t;\omega)$.

ただし，$U_D(a_t,u_t)$は探索木 $G_D(a_t,u_t)$の全 leaf 局面の集合を，$E^s_a(u;\omega)$は leaf 局面 u での静的局面評価関数の値を表している．$P(u|a_t,u_t;\omega)$ は確率的な方策により leaf 局面 u へ遷移する確率である．これらの概念図を図 1 に示す．

また，着手決定のためのアルゴリズムは以下のように表される：

【着手決定のアルゴリズム】

step1: 手番局面 u_t において全ての合法手 a_t を生成する．

step2: 各 a_t に対して $E^*_a(a_t,u_t;\omega)$を計算する．

step3: 方策 $\pi_a(a_t|u_t;\omega)$により着手を選択する．

ただし，(7)の $E^*_a(a_t,u_t;\omega)$は，4.2 で後述するように再帰により厳密に計算できる（(12)参照）．ただし，再帰の最下層では leaf 局面の評価値が呼び出される．また，方策 $\pi_a(a_t|u_t;\omega)$は，(1)の Boltzmann 分布を念頭に置いている．

通常，ゲーム木探索における指し手評価では，探索木 $G_D(a_t,u_t)$に対して min-max 探索法や $\alpha\beta$ 探索法により得られた leaf 局面での局面評価値を指し手 a_t の評価値とする．これは，(7)の右辺の計算において期待値計算を厳密に行わないで，探索木の最善応手手順(principal variation)の leaf 局面 (principal leaf) $u^*_D(a_t,u_t)$の局面評価値 $E^s_a(u^*_D(a_t,u_t);\omega)$で代表するという一種の近似計算に相当する．

また，ヒューリスティクスを用いた探索木の枝刈りも，(7)の右辺の探索過程において leaf 局面への遷移確率をゼロとおくことに相当する．例えば，激指チームの“実現確率”（＝“親の実現確率”×“指し手の遷移確率”）による枝刈り[5]も同様である．これについては 5.2 で考察する．

さらに，近年，囲碁などで盛んなモンテカルロ探索[17]は，局面評価のために多数回のプレイアウトを行う．これは，(7)の右辺の期待値操作を，あるシミュレーション方策により生成した leaf 局面の評価値の単純平均操作で置き換えたと見なすことができる．シミュレーション方策を用いた近似計算法については，改めて 6.で詳細に検討する．

本論文で指し手の評価として(7)のような期待値を提案した理由は次の２つである．まず，上記のように様々な指し手探索法を特殊ケースとして導くことが可能で理論的な見通しが良いことである．次に，最善応手手順だけを利用すると，読みの深さや評価関数の精度に限界がある場合には，最善応手手順以外の変化手順をも十分考慮して指し手の評価を行う方が，読みの深さが有限であることと評価値の誤差に起因する探索の揺らぎに対して頑健な評価法を与えるのではないかと考えたからである．

4. 探索と方策勾配法による評価関数の学習

4.1 学習則

3.では(1)の目的関数として出現局面 u_t における指し手 a_t の直接的な評価値 $E_a(a_t,u_t;\omega)$ではなく，(7)に示した a_t 以下の全 leaf 局面の評価値 $\{E^s_a(u;\omega)\}(u\in U_D(a_t,u_t))$と各 leaf 局面への遷移確率 $P(u|a_t,u_t;\omega)$とを用いて計算することを提案した．したがって，学習エージェント A の方策(1),(2)は，

$$\pi_a(a_t|u_t;\omega)=\exp\left(E^*_a(a_t,u_t;\omega)/T_a\right)/Z_a \tag{8}$$

$$Z_a\equiv\sum_{x\in A(u_t)}\exp\left(E^*_a(x,u_t;\omega)/T_a\right) \tag{9}$$

と表される．このときの学習則は，(5),(6)より，

$$\Delta\omega=\varepsilon\cdot r\sum_{t=1}^{L_a}e_\omega(t) \tag{10}$$

$$e_\omega(t)=(1/T_a)\Big[\partial E^*_a(a_t,u_t;\omega)/\partial\omega - \sum_{x\in A(u_t)}\pi_a(x|u_t;\omega)\partial E^*_a(x,u_t;\omega)/\partial\omega\ \Big] \tag{11}$$

と表される．

(8)～(11)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計算できる．ただし，これらの値は局面 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 leaf 局面 $u\in U_D(a,u_t)$に依存する．したがって，2.2 で述べた通常の方策勾配法の適用方式では，出現局面 u_t 以下の深さ 1 の局面に含まれる特徴量パラメータのみが更新対象となるが，探索を伴う本方式では全 leaf 局面に含まれる特徴量パラメータすべてが更新対象となり，一対局あたりの学習の効率化が期待できる．

4.2 指し手評価の期待値とその勾配の再帰計算

(8)～(11)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を示す．まず，深さ d ($0\leqq d\leqq D$-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により生成される対戦相手 B の手番局面を $v^d_t=v(a^d_t,\ u^d_t)$，その局面から B が指し手 b^d_tを指して得られた学習エージェント A

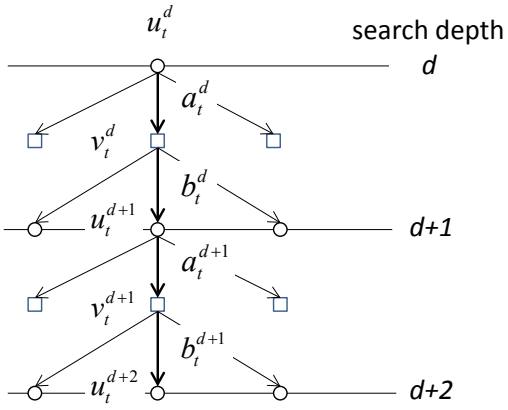


図 2　探索の深さ，手番局面，指し手の関係．
Figure 2　Search depth, positions and moves.

の手番局面を $u^{d+1}{}_t=u(b^d{}_t, v^d{}_t)$とする（図 2）．ただし，対戦相手のエージェント B の方策 π_b は既知とする．

この時，探索の深さ d における $E^*{}_a(a_t{}^d,u_t{}^d;\omega)$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．

$$E_a^*\left(a_t^d,u_t^d;\omega\right)=\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\cdot \sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)\cdot E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right) \tag{12}$$

$$\partial E_a^*\left(a_t^d,u_t^d;\omega\right)\Big/\partial\omega=(\partial/\partial\omega)\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\cdot \sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)\cdot E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right) \tag{13}$$

$$=\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\sum_{a_t^{d+1}}\partial\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)\Big/\partial\omega\cdot E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right) +\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)\cdot\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)\Big/\partial\omega \tag{14}$$

$$=\sum_{b_t^d}\pi_b\left(b_t^d\middle|v_t^d\right)\sum_{a_t^{d+1}}\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right) \cdot\left[e_\omega(t,d+1)E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)+\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)\Big/\partial\omega\right] \tag{15}$$

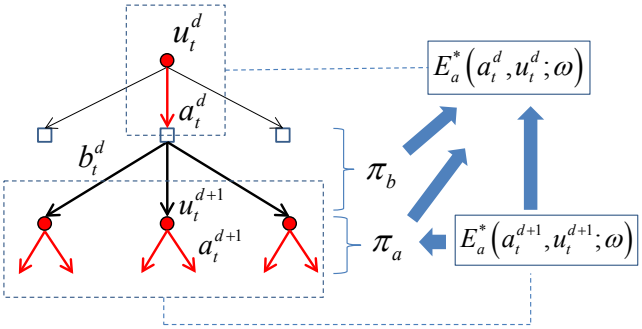
ただし，

$$e_\omega(t,d+1)\equiv\partial\ln\pi_a\left(a_t^{d+1}\middle|u_t^{d+1};\omega\right)\Big/\partial\omega \tag{16}$$

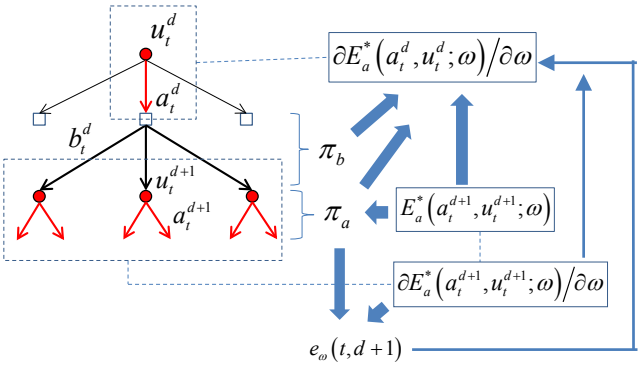
$$=(1/T_a)\left[\partial E_a^*\left(a_t^{d+1},u_t^{d+1};\omega\right)\Big/\partial\omega - \sum_{x\in A\left(u_t^{d+1}\right)}\pi_a\left(x\middle|u_t^{d+1};\omega\right)\partial E_a^*\left(x,u_t^{d+1};\omega\right)\Big/\partial\omega\right] \tag{17}$$

である．

また，(12),(13)における再帰の終端は，もし，$u^{d+1}{}_t$ が leaf 局面，すなわち，d=D-1 ならば，



(a) 指し手評価の期待値 $E_a^*\left(a_t^d,u_t^d;\omega\right)$



(b) 1 階微係数 $\partial E_a^*\left(a_t^d,u_t^d;\omega\right)\Big/\partial\omega$

図 3　PG 行動期待値法の再帰計算における依存関係：(a)指し手評価の期待値，(b) 1 階微係数の値．
Figure 3　Recursive calculation in “PG expectation algorithm”: (a) Expected evaluation function $E^*{}_a(a_t{}^d,u_t{}^d;\omega)$, and (b) its first derivative $\partial E^*{}_a(a_t{}^d,u_t{}^d;\omega)/\partial\omega$.

$$E_a^*\left(a_t^d,u_t^d;\omega\right)=\sum_{b_t^{D-1}}\pi_b\left(b_t^{D-1}\middle|v_t^{D-1}\right)E_a^s\left(u_t^D;\omega\right) \tag{18}$$

$$\partial E_a^*\left(a_t^d,u_t^d;\omega\right)\Big/\partial\omega=\sum_{b_t^{D-1}}\pi_b\left(b_t^{D-1}\middle|v_t^{D-1}\right)\cdot\partial E_a^s\left(u_t^D;\omega\right)\Big/\partial\omega \tag{19}$$

と書ける．図 3 に上記の依存関係を表した模式図を示す．

なお，本論文では 2.2 で述べた出現局面における指し手評価値を用いた方策勾配法を“PG 法”または単に方策勾配法，4.1 と 4.2 で提案した全 leaf 局面に基づく指し手評価の期待値を用いた方策勾配法を“PG 行動期待値法”(PG expectation algorithm)と呼んで区別することにする．

5. 計算量削減のための近似手法のアイデア

5.1 min-max 探索または$\alpha\beta$探索の適用：PGLeaf 法

3.の指し手評価には，(12)の再帰計算で探索木の最下段での全 leaf 局面の局面評価値を知る必要がある．さらに，4.の学習においては，全 leaf 局面での勾配値も必要である．したがって，指し手決定と学習にかかる計算時間は膨大となる可能性が予想される．そこで，計算量を削減するための近似手法に関するアイデアを本章では述べる．

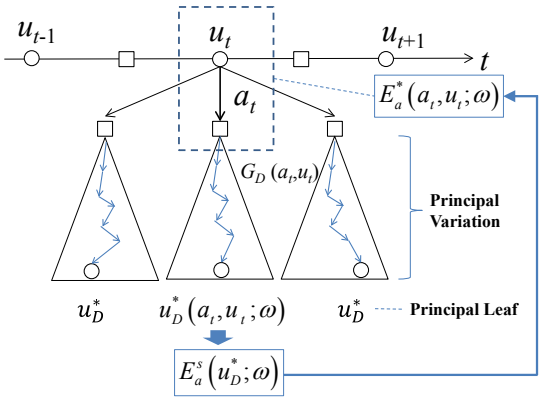


図 4　PGLeaf 法
Figure 4　PGLeaf algorithm.

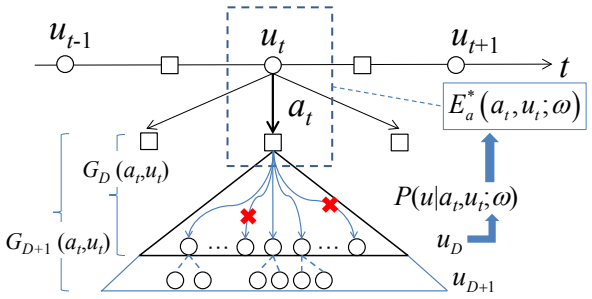


図 5　PG 行動期待値法への反復深化法の適用
Figure 5　An iterative-deepening search applied to PG expectation algorithm.

まず，対戦相手 B の方策 π_b として min 探索を用いる．この近似に加えて，学習エージェント A の方策として max 探索を行う（(8)で $T_a \to 0$ と置くことに相当する）．すなわち，min-max 探索，あるいは $\alpha\beta$ 探索を行い，最善応手手順だけを考える．これは(7)の遷移確率において，

$$P(u|a_t,u_t;\omega)=\begin{cases}1 & \text{if} \quad u=u^*_D(a_t,u_t;\omega)\\ 0 & \text{otherwise}\end{cases} \qquad (20)$$

と置いたと解釈できる．このように指し手評価の期待値 $E^*_a(a_t,u_t;\omega)$を principal leaf $u^*_D(a_t,u_t;\omega)$の局面評価値 $E^s_a(u^*_D(a_t,u_t;\omega);\omega)$で置き換えた指し手決定法と学習法を“PGLeaf 法”と呼ぶことにする．PGLeaf 法では学習時に(12)~(17)のような再帰計算は不要で，通常の $\alpha\beta$ 探索アルゴリズムをそのまま利用できる．PGLeaf 法の概念を図 4 に示す．

5.2 反復深化法の適用

探索時に反復深化法を適用する方法が考えられる．ある深さDを設定し，leaf局面u_Dの集合とそれらの局面評価値$E^s_a(u_D;\omega)$を用いてleaf局面までの遷移確率$P(u_D|\,a_t,u_t;\omega)$と指し手評価の期待値$E^*_a(a_t,u_t;\omega)$を計算する．ただし，遷移確率$P(u_D|\,a_t,u_t;\omega)$が閾値以下であればそれ以下の部分木はカットする．次にDを1だけ増やしてこの操作を繰り返す．

(12)~(19)での指し手評価の期待値の再帰的計算や学習時には，カットされないで残った枝の leaf 局面だけを用いる．この場合，残った枝の leaf 局面に含まれる特徴量パラメータすべてが更新される．図 5 に模式図を示す．

5.3 異なる評価関数の principal leaf による期待値の計算法

この近似法は5.1で述べたPGLeaf法の合議制バージョンに相当する．まず，N個の異なる評価関数を持った探索アルゴリズムkによりそれぞれmin-max探索を行い，N個の principal leaf $\{u^*_{D,k}\}(k=1,2,\ldots,N)$を求める．次に，各principal leafにおける局面評価値$E^s_a(u^*_{D,k})$を計算し，信頼度α_kを重み係数とする線形和により，指し手評価の期待値を

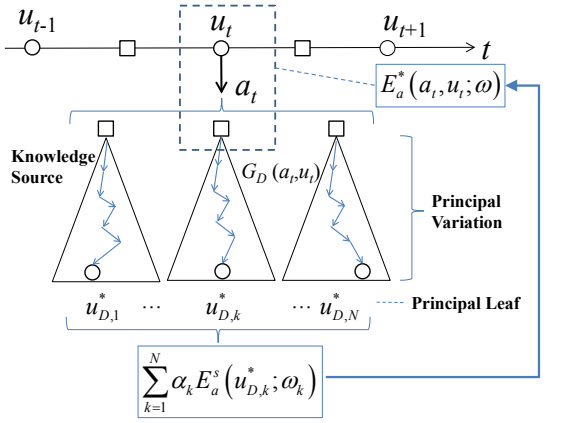


図 6　異なる評価関数による期待値操作
Figure 6　Expectation with different positional evaluation functions.

$$E^*_a(a_t,u_t;\omega)\approx\sum_{k=1}^{N}\alpha_k E^s_a(u^*_{D,k}(a_t,u_t;\omega_k);\omega_k) \qquad (21)$$

と近似する．学習時には(21)を(11)へ代入して得られる特徴的適正度を用いる．探索アルゴリズムkは自らが探索した principal leaf $u^*_{D,k}(a_t,u_t;\omega_k)$に含まれている特徴量に関するパラメータを更新する．これは，複数の探索アルゴリズム（知識エージェント）によるある種の“合議”による指し手決定[18]と，各探索アルゴリズムの評価関数の学習方法を与えており並列処理向きである．この際，異なる探索アルゴリズムの生成法として，評価関数にランダムノイズを付加する方法も考えられる．図 6 にこれらの考えをまとめた説明図を示す．なお，(21)の信頼度α_kは学習パラメータと考えて本学習方式の枠組みで学習することも可能である．

6. 探索時にシミュレーション方策を用いる場合の学習

6.1 着手選択時の方策とシミュレーション方策

(8)で定義した方策 $\pi_a(a_t|u_t;\omega)=\exp(E^*_a(a_t,u_t;\omega)/T_a)/Z_a$ は，エ

ージェント A が手番で着手を選択する際の方策（以下，“着手決定方策”）である．その際には，(7)で定義された”指し手評価の期待値” $E^*_a(a_t,u_t;\omega)$の値が必要であった．この期待値は，深さ D での leaf 局面 $u \in U_D(a_t,u_t)$の局面評価値とその leaf局面 u への遷移確率 $P(u|a_t,u_t)$とから(7)で計算される．4．で述べた”PG 行動期待値法“では，この $P(u|a_t,u_t)$の計算にも着手決定方策(8)を用いていた．したがって，深さ D での全ての leaf局面 $u \in U_D(a_t,u_t)$の評価値を知る必要があり，厳密に求めようとすると計算量が膨大となる．

そこで，本章では着手決定方策 $\pi_a(a_t|u_t;\omega)$とは別に，leaf 局面の評価値を使用しない方策を探索用として用意する．通常，このような探索木生成のための方策は，“シミュレーション方策”(simulation policy)と呼ばれており，モンテカルロ探索や，実現確率を用いて前向きの枝刈りを行う場合の遷移確率の計算に用いられている[6].

本章では，シミュレーション方策が局面評価関数 $E^s_a(u;\omega)$中の ω に依存しない場合を考える．つまり，激指のように，探索中の指し手選択の方法と leaf 局面の評価法とは独立であるとする．さらに，探索木中の現在のノード局面の情報だけを用いて指し手を選択する場合を考える．つまり，シミュレーション方策がマルコフ性を持っているとする．

今，シミュレーション方策を $\pi'_a(a|u;\theta)$で表す．ただし，u は探索木中のノード局面（エージェント A の手番），θ はシミュレーション方策に含まれるパラメータの総称であり ω とは異なる．これらを用いると，手番局面 u_t から指し手 a を経由した leaf 局面 u への遷移確率 $P(u|a,u_t)$は，

$$P\left(u\middle|a,u_t;\theta\right)=\pi'_a\left(a\middle|u_t;\theta\right)\pi_b\left(b_t^0\middle|v_t^0\right)\pi'_a\left(a^1\middle|u_t^1;\theta\right)\pi_b\left(b_t^1\middle|v_t^1\right)$$

$$\cdots\pi'_a\left(a^{D-1}\middle|u_t^{D-1};\theta\right)\pi_b\left(b_t^{D-1}\middle|v_t^{D-1}\right) \quad (22)$$

$$=\prod_{d=0}^{D-1}\pi'_a\left(a^d\middle|u_t^d;\theta\right)\pi_b\left(b_t^d\middle|v_t^d\right) \quad (23)$$

と表すことができる．ただし，上付きの添え字は探索木の深さを表し，$a^0=a$，$u_t^0=u_t$，$u_t^D=u$ とする．したがって，(7)の指し手評価の期待値は，

$$E^*_a\left(a_t,u_t;\omega,\theta\right)\equiv\sum_{u\in U_D(a_t,u_t)}P\left(u\middle|a_t,u_t;\theta\right)E^s_a\left(u;\omega\right) \quad (24)$$

と 2 種類のパラメータ ω, θ を含む．着手決定方策 π_a も同様であり，以下本章では $\pi_a=\pi_a(a|u;\omega,\theta)$と記す．

次に，上記 2 種類のパラメータに関する学習則は以下のようにまとめることができる．

$$\Delta\omega=\varepsilon\cdot r\sum_{t=1}^{L_a}e_\omega\left(t\right) \quad (25)$$

$$e_\omega\left(t\right)\equiv\partial\ln\pi_a\left(a_t\middle|u_t;\omega,\theta\right)\Big/\partial\omega \quad (26)$$

$$=\left(1/T_a\right)\left\{E_{\pi'_a}\left[\partial E^s_a\left(u;\omega\right)\Big/\partial\omega\middle|a_t,u_t\right]\right.$$

$$\left.-E_{\pi_a}\cdot E_{\pi'_a}\left[\partial E^s_a\left(u;\omega\right)\Big/\partial\omega\middle|x,u_t\right]\right\} \quad (27)$$

$$\Delta\theta=\varepsilon\cdot r\sum_{t=1}^{L_a}e_\theta\left(t\right) \quad (28)$$

$$e_\theta\left(t\right)\equiv\partial\ln\pi_a\left(a_t\middle|u_t;\omega,\theta\right)\Big/\partial\theta \quad (29)$$

$$=\left(1/T_a\right)\left\{E_{\pi'_a}\left[E^s_a\left(u;\omega\right)\sum_{d=0}^{D-1}e'_\theta\left(d\right)\middle|a_t,u_t\right]\right.$$

$$\left.-E_{\pi_a}\cdot E_{\pi'_a}\left[E^s_a\left(u;\omega\right)\sum_{d=0}^{D-1}e'_\theta\left(d\right)\middle|x,u_t\right]\right\} \quad (30)$$

ただし，

$$E_{\pi_a}\left[z\middle|u_t\right]\equiv\sum_{x\in A(u_t)}z\cdot\pi_a\left(x\middle|u_t;\omega,\theta\right) \quad (31)$$

$$E_{\pi'_a}\left[z\middle|a,u_t\right]\equiv\sum_{u\in U_D(a,u_t)}z\cdot P\left(u\middle|a,u_t;\theta\right) \quad (32)$$

$$E_{\pi_a}\cdot E_{\pi'_a}\left[z\middle|x,u_t\right]\equiv E_{\pi_a}\left[E_{\pi'_a}\left[z\middle|x,u_t\right]\middle|u_t\right] \quad (33)$$

$$=\sum_{x\in A(u_t)}\pi_a\left(x\middle|u_t;\omega,\theta\right)\left[\sum_{u\in U_D(x,u_t)}z\cdot P\left(u\middle|x,u_t;\theta\right)\right] \quad (34)$$

$$e'_\theta\left(d\right)\equiv\partial\ln\pi'_a\left(a^d\middle|u_t^d;\theta\right)\Big/\partial\theta \quad (35)$$

と定義した．なお，$E_\pi[\cdot|u]$は局面 u において着手決定方策 $\pi_a(a|u;\omega,\theta)$による期待値操作を，$E_{\pi'}[\cdot|a,u]$は局面 u で a を選択し，それ以降はシミュレーション方策 $\pi'_a(a|u;\theta)$を用いて指し手選択を行ってシミュレーションした場合の期待値計算を表している．

(27)は，高報酬を得た対局での着手の選択確率を高めるためにその手の評価値を高めたい．そこで，シミュレーション方策は固定しておいて，その手から始まるシミュレーションにより得られる leaf 局面 u の局面評価値 $E^s_a(u;\omega)$を高めるように ω を $E^s_a(u;\omega)$の増加する勾配方向へ動かすと解釈できる．

一方，(30)では，同様の目的ではあるが，今度は局面評価関数を固定しておいてシミュレーション方策の方を調整する．すなわち，シミュレーション方策中のパラメータ θ の更新量$\Delta\theta$ を，高評価の leaf 局面への遷移確率 $P(u|a_t,u_t;\theta)$の値を増加させるように調節している．このときの調整法は，u_t を初期状態，a_t を初期行動とする方策勾配法による強化学習を行っていると見なすことができる．実際，(30)の π'_aによる期待値 $E_{\pi'}[E^s_a(u)\cdots|a_t,u_t]$は，一般的な方策勾配法の学習則(3)～(5)において，$\pi'_a(a|u;\theta)$を方策 π，$E^s_a(u)$を報酬 r とする期待報酬値の勾配$\partial E_\pi[r]/\partial\theta=\partial E_{\pi'}[E^s_a(u)|a_t,u_t]/\partial\theta$になっている．つまり，シミュレーション方策の強化学習を方策勾配法を用いて指し手の探索シミュレーション内で行

うことができることを表している．

なお，シミュレーション方策がマルコフ性を持っておらず，探索のルート局面 u_t から現ノード局面 u_t^d までの状態行動履歴 $h_t^d \equiv \{u_t,a,u_t^1,a^1,..,u_t^{d-1},a^{d-1}\}$ に依存する場合も，$\pi'_a(a|u_t^d;\theta)$ を $\pi'_a(a|u_t^d, h_t^d;\theta)$ と置き換えると，(25)〜(35)はそのまま成り立つ．例えば，直前の手に応じて指し手を変化させる場合などがこれにあたる．

シミュレーション方策の例としては，次の Boltzmann 分布を考える．

$$\pi'_a\left(a_t^d \middle| u_t^d, h_t^d;\theta\right) = \exp\left(E'_a\left(a_t^d, u_t^d, h_t^d;\theta\right)/T'_a\right)/Z'_a \quad (36)$$

$$Z'_a \equiv \sum_{x\in A\left(u_t^d\right)} \exp\left(E'_a\left(x, u_t^d, h_t^d;\theta\right)/T'_a\right) \quad (37)$$

ただし，目的関数 $E'_a(a_t^d,u_t^d, h_t^d;\theta)$中の特徴量は，囲碁では石の局所的な配置パターンなどであり，将棋では激指などでの指し手選択のための特徴量，例えば，「王手かどうか」「ひもをつける手かどうか」[6]などである．これらは，人間の将棋では，「手筋」「型」のような経験的で断片的なミニ知識による指し手，あるいは，「直観」，「第一感」などの「深い読み」を伴わない処理で指される手と考えられる．

実際の対局における探索時には，このようなシミュレーション方策を用いて，探索のルートノードから探索木の各ノードへの遷移確率の値を次のように近似的に計算できる．

$$P\left(u_t^d \middle| a,u_t;\theta\right) = \pi'_a\left(a^{d-1}\middle|u_t^{d-1};\theta\right)\pi_b\left(b_t^{d-1}\middle|v_t^{d-1}\right)P\left(u_t^{d-1}\middle|a,u_t;\theta\right) \quad (38)$$

上記の遷移確率値は，モンテカルロ探索による指し手評価や，$\alpha\beta$探索の際の前向き枝刈り処理に用いることができる．例えば，モンテカルロ探索においては，leaf局面に対して遷移確率値を計算すれば，(24)の指し手評価の期待値$E^*_a(a_t, u_t;\omega,\theta)$を近似的に計算することができる．したがって，膨大な回数の試行を行って局面評価値の平均操作を行う必要がなく，試行回数の削減に役立つ．また，$\alpha\beta$探索においては，探索木の途中のノードへの遷移確率値が閾値以下になればそのノード以下の探索を打ち切るなどの処理が考えられる．この方法には，激指の実現確率による枝刈り処理とは異なり，兄弟手の良し悪しの度合いにより探索の深さを制御できる利点がある．例えば，飛車を取る手があれば端歩を突く手は殆ど読む必要はないが，他に有力な手がない局面では深く読む必要があると言うような場合に有効であると考えられる[19].

6.2 シミュレーション方策と局面評価関数の教師付き学習

6.1 ではシミュレーション方策を用いた方策勾配法による強化学習を考察した．本節では強化学習ではなく，ある局面において正解手が与えられた場合の学習，すなわち，教師付き学習を考える．6.1 で用いた方策勾配法では，エピソードごとの期待報酬値の勾配を計算したが，学習システムの正解手に対する選択確率値の勾配を同じように計算することができる．なお，簡単のために本節ではシミュレーション方策がルート局面からそのノード局面までの指し手履歴によらないマルコフ性のある場合を考える．そうでない場合も全く同様に導出できる．

通常，局面評価関数に教師付き学習を適用する際は，プロ棋士の棋譜データベース等から，局面とそこで指された指し手を唯一の正解手として局面・指し手ペアの訓練データを作成する．しかし，ここではより一般的な場合を扱う．すなわち，正解手を１つに限定せずに，正解と思われる複数の指し手に対してそれらを選択する確率分布を学習させることにする．そこで，今，正解の着手決定方策を π^*，学習システムの着手決定方策を π_a とし，シミュレーション方策 π'_a は(36)を仮定する．次の誤差関数 U_{err} を考える．

$$U_{err}(\pi^*,\pi_a) \equiv \sum_{a\in A(s)} \pi^*(a|s)\ln\left[\pi^*(a|s)/\pi_a(a|s;\omega,\theta)\right] \quad (39)$$

$U_{err}(\geq 0)$は，正解の方策 π^*と学習システムの方策 π_a との距離を表すカルバック・ライブラー情報量（Kullback–Leibler divergence）である． ただし，6.1 と同じく，

$$\pi_a\left(a|s;\omega,\theta\right) = \exp\left(E^*_a\left(a,s;\omega,\theta\right)/T_a\right)/Z_a \quad (40)$$

$$Z_a \equiv \sum_{x\in A(s)} \exp\left(E^*_a\left(x,s;\omega,\theta\right)/T_a\right) \quad (41)$$

$$E^*_a\left(a,s;\omega,\theta\right) = \sum_{u\in U_D(a,s)} P\left(u|a,s;\theta\right)E^s_a\left(u;\omega\right) \quad (42)$$

と仮定する．このとき，ω に関する勾配ベクトルは，

$$\partial U_{err}/\partial\omega = -\sum_{a\in A(s)} \pi^*(a|s)\cdot\partial\ln\pi_a\left(a|s;\omega,\theta\right)/\partial\omega \quad (43)$$

と表されるが，右辺中の対数微分の項は，(26),(27)において $a_t=a$，$u_t=s$ と置き換えた式で表される．すなわち，(43)を用いて局面評価関数 $E^s_a(u;\omega)$中の ω の更新量は，

$$\Delta\omega = -\varepsilon\cdot\partial U_{err}/\partial\omega \quad (44)$$

と計算すればよい．

また，θ に関する勾配ベクトルは，

$$\partial U_{err}/\partial\theta = -\sum_{a\in A(s)} \pi^*(a|s)\cdot\partial\ln\pi_a\left(a|s;\omega,\theta\right)/\partial\theta \quad (45)$$

と表されるが，右辺の対数微分の項は，(29),(30)において $a_t=a$，$u_t=s$ と置き換えた式で表される． したがって，(45)の値を用いて，シミュレーション方策 $\pi'_a(a|s;\theta)$中のパラメータ θ の更新量は次のように計算すればよい．

$$\Delta\theta = -\varepsilon\cdot\partial U_{err}/\partial\theta \quad (46)$$

なお，探索（読み）にシミュレーション方策を用いないで着手決定方策を用いる場合でも局面評価関数中の ω の教師付き学習は可能である．その場合は，(43)の右辺の対数微分は(11)と同一であり，4.2 や 5.での手法が使える．

6.3 シミュレーション方策の教師付き学習に関する先行研究との関係

激指は探索時にその局面における指し手の選びやすさである実現確率を用いた枝刈りを行い，探索の深さの制御を積極的に行っている．この実現確率の計算はシミュレーション方策を用いた状態遷移確率の近似計算の一種とみなすことができる．激指ではシミュレーション方策中のパラメータ θ を，人間の棋譜データベースから統計的処理により求めている．さらに，Bonanza の学習法をベースにした局面評価関数中の ω の教師付学習も行っているが，これら２つの学習は全く独立しており直接的な関係はない．

また，方策勾配を利用したシミュレーション方策の教師付き学習の先行研究として，Policy-gradient simulation balancing法が囲碁の場合に提案されている[20]．そこでは，訓練局面sに対して正解となる局面評価値$V^*(s)$が必要となり，かつ，着手決定方策として局面評価関数を利用していないので局面評価関数の学習は行っていない．

さらに，着手方策中の局面評価関数をTD(λ)法で求めておいて，その学習結果として得られたヒューリスティクスをUCT探索におけるシミュレーション方策へ組み込む将棋の研究がある[21]．しかし，そこでの着手決定方策の学習は探索のない学習であり，かつ，シミュレーション方策の学習結果が着手決定方策の学習に反映されることはない．

本方式では，正解の方策 π^*に対して，着手決定方策で用いられる評価関数 $E^s_a(u;\omega)$中のパラメータ ω と，シミュレーション方策 $\pi'_a(a|u;\theta)$中のパラメータ θ とが同時に連携して(39)の誤差を減らすように学習を行うことができる．

7. おわりに

本論文では，これまでに強化学習の手法としてコンピュータ将棋に用いられてきた TD(λ)法や TDLeaf(λ)法ではなく，方策勾配法を適用する手法についての理論的な検討を行った．その結果，最初に，将棋のように着手決定に探索を要する場合については，“指し手評価の期待値”による確率的方策を用いた“PG 行動期待値法”と呼ぶ着手決定方式を提案した．次に，その近似計算法として“PGLeaf 法”など３つの方法を提案した．

さらに，この着手決定のための方策とは別に，探索時にシミュレーション方策を用いる場合への方策勾配法の適用についても考察し，両方の方策に含まれるパラメータの学習則を導出することができた．学習後のシミュレーション方策は，モンテカルロ探索における期待値の計算や，探索木中のノード局面への遷移確率値の計算に用いることができる．したがって，モンテカルロ探索の試行回数の削減や，探索時の深さ制御のための前向き枝刈り処理の精度向上に役立つと考えられる．

最後に，ここまでに用いた方策勾配の計算は，強化学習だけでなく，局面ごとに正解手の方策を確率分布の形で与える教師付き学習問題へも適用することができることを示した．この場合のシミュレーション方策と局面評価関数に関する学習則も導出することが出来た．

今後は，本論文で展開した学習則や探索法を実装し，実験により有効性を検証，評価して行く予定である．

参考文献

1) 松原仁 編著：コンピュータ将棋の進歩⑥プロ棋士に並ぶ，共立出版(2012).
2) 第 2 回電王戦の公式ページ：http://ex.nicovideo.jp/denousen2013/
3) 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).
4) Sutton, R. S. and Barto A. G. : Reinforcement Learning, The MIT Press, Massachusetts (1998).
5) Baxter, J., Tridgell, A., and Weaver, L., : KnightCap: A chess program that learns by combining TD(λ) with game-tree search, Proceedings of the Fifteenth International Conference (ICML '98), pp.28-36 (1998).
6) 鶴岡慶雅：「激指」の最近の改良について，松原仁 編著：コンピュータ将棋の進歩⑥プロ棋士に並ぶ，第 4 章，共立出版(2012).
7) Williams, R. J. : Simple Statistical Gradient- Following Algorithms for Connectionist Reinforcement Learning, Machine Learning, Vol.8, pp.229-256 (1992).
8) 木村元，山村雅幸，小林重信：部分観測マルコフ決定過程下での強化学習-確率的傾斜法による接近，人工知能学会誌，Vol.11, No.5，pp761-768 (1996).
9) Sutton, R.S., McAllester, D., Singh, S. and Mansour, Y. : Policy Gradient Methods for Reinforcement Learning with Function Approximation, Proc. of Advances in Neural Information Processing Systems 12 (NIPS’99), pp.1057-1063 (2000).
10) Konda, V. R. and Tsitsiklis, J. N.: Actor-Critic Algorithms, Proc. of Advances in Neural Information Processing Systems 12 (NIPS’99), pp.1008-1014 (2000).
11) 阿部健一：強化学習―価値関数推定と政策探索”，計測と制御，第 41 巻，第 9 号，pp.680-685 (2002).
12) Kakade, S.: A natural policy gradient, Proc. of Advances in Neural Information Processing Systems 14 (NIPS’01), pp.1531- 1538 (2002).
13) Peters, J., and Schaal, S.: Policy Gradient Meth-ods for Robotics, Proc.of the IEEE International Conference on Intelligent Robotics Systems (IROS 2006), pp.2219-2225(2006).
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, Vol.J90-D, No.9, pp.2271-2280 (2007).
15) 石原聖司, 五十嵐治一 ：マルチエージェント系における行動学習への方策こう配法の適用-追跡問題-, 電子情報通信学会論文誌 D-I, Vol.J87-D1, No.3, pp.390-397 (2004).
16) 五十嵐 治一，半田 雅人，石原 聖司，篠埜 功：マルチエージェントシステムにおける行動制御―PSO における重み係数の強化学習―，電子情報通信学会論文誌 D, Vol. J94-D, No. 10, pp. 1612-1621 (2011).
17) 美添一樹：モンテカルロ木探索-コンピュータ囲碁に革命を起こした新手法-，情報処理，Vol.49, No.6, pp.686-693 (2008).
18) 伊藤毅志：コンピュータ将棋における合議アルゴリズム，人工知能学会誌，Vol.26，No.5，pp.525-539 (2011).
19) 一丸貴則：WCSC21 ツツカナアピール文書，http://www.computer-shogi.org/wcsc21/appeal/tsutsukana/WCSC21_tsutsukana_20110327.pdf
20) Silver, D., Tesauro, G.: Monte-Carlo simulation balancing. In: Bottou, L., Littman, M. (eds.) Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, pp. 945-952. Omnipress (June 2009).
21) 燧 暁彦，三輪 誠，鶴岡慶雅，近山 隆：TD(λ)学習を用いた Ms. Pac-Man AI のモンテカルロ木探索の方策の学習，情報処理学会研究報告，Vol.2013-GI-29, No.2, pp.1-8 (2013).