方策勾配法による探索制御の一考察

五十嵐治一[†1]　　森岡祐一　山本一将[†2]

コンピュータ将棋において探索木の枝を成長させる際に，その枝までの探索経路に沿った指し手の累積的な選択確率の値を基に探索制御を行う方法を提案する．このときの指し手の選択には，将棋の指し手に関するヒューリスティクスを組み込んだシミュレーション方策を使用する．この際，枝成長を決定論的に行う場合と確率的に行う２つの場合を考えた．さらに，本手法ではこのシミュレーション方策中のパラメータを強化学習の一手法である方策勾配法により学習する．

Learning Search Control by Policy Gradient Algorithm

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

This paper proposes a method based on the policy gradient learning algorithm for search control in computer shogi. In this method, whether every arc in a search tree should be extended is determined by the accumulated move-selection probability from the root node to the arc. Moves are selected by a simulation policy that includes heuristics for evaluating shogi moves. We consider two types of arc extension: deterministic and stochastic. In both cases, the parameters in the simulation policy can be learned by the policy gradient algorithm, which is a method of reinforcement learning.

1. はじめに

近年，コンピュータ将棋の棋力向上がめざましい．この大きな一因としては機械学習による局面評価関数の構築が挙げられる．「Bonanza メソッド」[1]と称されるプロ棋士の棋譜を用いた教師付き学習法がこの代表例である．教師付学習ではなく，プロ棋士の棋譜を用いない強化学習による学習方式[2),3)]も提案されているが，将棋においては教師付き学習ほどの棋力向上を得るまでには至っていない．しかし，人間の知識の結晶であるプロ棋士の棋譜を全く用いないでコンピュータが自ら戦法や戦型を獲得し，プロ棋士に匹敵する棋力に到達できるかという点では大変興味深い試みであると考えられる．我々もこれまでに，強化学習の一手法である方策勾配法を用いた局面評価関数の学習法を提案してきた[4),5)]．

一方，探索の戦略の面においては，将棋には合法手が多いことからさまざまな前向き枝刈りや探索延長の手法が試みられてきた．これらの選択戦略においては，ヒューリスティックな方法が多い中，「激指」は実現確率による枝刈り方法を提案した[6)]．これはプロ棋士の棋譜からよく指されやすい手かどうかを評価する関数を学習し，探索中の局面が実現される度合いを見積り，枝刈りに利用する方法である．さらに，この実現確率探索に探索中の動的な情報を有効に用いる方法[7)]や，プロ棋士が指した手をその兄弟手よりも深く探索することを目標とした「ツツカナ」の方法[8)]も提案されている．

†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.

しかしながら，これらの探索制御に対する機械学習の応用例は，いずれもプロ棋士の棋譜を利用した教師付き学習である．そこで我々は方策勾配法を用いた探索制御法を考案した．本手法は先に文献 9)で提案した学習方法をベースにしている．その学習方法は，局面評価関数と探索制御の両者の学習に適用できるが，本論文では後者の探索制御の学習法に関する詳細な理論を述べる．

2. PG 行動期待値法

2.1 指し手評価の期待値と着手方策

我々は文献 9)において，方策勾配法を用いた局面評価関数中のパラメータの学習法として「PG 行動期待値法」を提案してきた[9)]．そこでは，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 の指し手 a_t の評価値として，次の「指し手評価の期待値」

$$E_a^*(a_t,u_t;\omega) \equiv \sum_{u\in U_D(a_t,u_t)} P(u|a_t,u_t;\omega)V(u;\omega) \qquad (1)$$

を定義した．(1)において，$U_D(a_t,u_t)$は深さ D の探索木 $G_D(a_t,u_t)$の全 leaf 局面の集合，$V(u;\omega)$は leaf 局面 u での静的局面評価関数の値，$P(u|a_t,u_t;\omega)$ は leaf 局面 u へ遷移する確率である．さらに，指し手評価の期待値を基にした指し手選択法を「着手方策」と定義した後，A の着手方策として，(1)を目的関数とする Boltzmann 分布

$$\pi_a(a_t|u_t;\omega) = \exp\left(E_a^*(a_t,u_t;\omega)/T_a\right)/Z_a \qquad (2)$$

$$Z_a \equiv \sum_{x\in A(u_t)} \exp\left(E_a^*(x,u_t;\omega)/T_a\right) \qquad (3)$$

を用いることを提案した．ただし，ω は評価関数中のパラメータ，T_aは温度パラメータ，$A(u_t)$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．このとき，方策勾配法によるパラメータ ω の学習則は次のように表される．

$$\Delta\omega = \varepsilon \cdot r \sum_{t=1}^{L_a} e_\omega(t) \tag{4}$$

$$e_\omega(t) = (1/T_a)\Big[\partial E_a^*(a_t,u_t;\omega)/\partial\omega - \sum_{x\in A(u_t)} \pi_a(x|u_t;\omega)\partial E_a^*(x,u_t;\omega)/\partial\omega \ \Big] \tag{5}$$

文献 9)では，$\partial E_a^*(x,u_t;\omega)/\partial\omega$ と $E_a^*(a_t,u_t;\omega)$ が再帰的に計算できることを示した．また，(5)は

$$e_\omega(t) = (1/T_a)\Big[\big(1-\pi_a(a_t|u_t;\omega)\big)\partial E_a^*(a_t,u_t;\omega)/\partial\omega - \sum_{x\neq a_t} \pi_a(x|u_t;\omega)\partial E_a^*(x,u_t;\omega)/\partial\omega\Big] \tag{6}$$

と表される．(6)において，実現手 a_t の $\partial E_a^*(a_t,u_t;\omega)/\partial\omega$ の係数は正であり，その兄弟手 x の $\partial E_a^*(x,u_t;\omega)/\partial\omega$ の係数は負であることから，高報酬が得られたエピソード（一局のゲーム）では，そのときの実現手がより出現しやすくなるように実現手の評価値を高め，逆にその兄弟手に対しては評価値を下げる方向にパラメータベクトル ω を更新しようとしているのがわかる．

2.2 PGLeaf 法：最善応手手順による指し手評価

さらに，文献 9)では 2.1 で述べた PG 行動期待値法の近似計算として PGLeaf 法を提案した．この方法は，Min-Max 探索(αβ 探索)により最善応手手順(principal variation, PV)の leaf 局面 u^*_D を求めて，(1)の遷移確率 $P(u|a_t,u_t;\omega)$ を

$$P(u|a_t,u_t;\omega) = \begin{cases} 1 & \text{if} \quad u = u_D^*(a_t,u_t;\omega) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

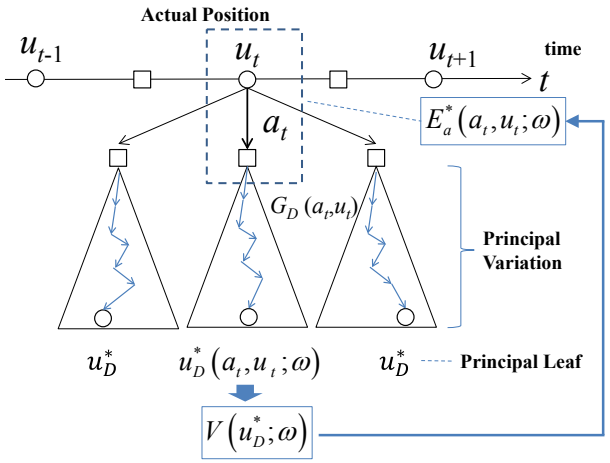

図 1　PGLeaf 法の概念図

Figure 1　The schematic diagram of the PGLeaf algorithm.

と近似する方法である．したがって，学習エージェント A の手番局面 u_t において，各合法手 a_t の評価値 $E_a^*(a_t,u_t;\omega)$ を a_t 以下の PV 上の leaf 局面(Principal position/leaf) u^*_D の局面評価値 $V(u^*_D;\omega)$で代用する，図 1 に PGLeaf 法の概念図を示す．

3. シミュレーション方策と探索制御

3.1 シミュレーション方策を用いた枝の成長確率

最近のコンピュータ囲碁では，探索木の leaf 局面の評価値ではなく，指し手自身の評価値を用いたモンテカルロ計算による探索法がよく用いられている [10]．その際，探索時においては読みを伴わない手自身の直接評価による選択方法が用いられる．これを(2)の着手方策と区別して，「シミュレーション方策」と称する．将棋の場合もプロ棋士は指し手に関する経験的な知識に基づいて探索範囲を絞っていると言われている．そこで，本研究ではシミュレーション方策を探索時の枝の成長または枝刈りに利用することを考えた．

まず，エージェント A の手番のルート局面 u_t から深さ j における A の手番局面 u_t^j における指し手 a_t^j までの遷移確率 $P'(a_t^j|a,u_t;\theta)$ を「累積選択確率」と呼び，シミュレーション方策 π'_a を用いて以下のように定義する．

$$P'(a_t^j|a_t,u_t;\theta) = \pi'_a(a_t^j|u_t^j,h_t^j;\theta)\prod_{d=0}^{j-1}\pi'_a(a_t^d|u_t^d,h_t^d;\theta)\pi'_b(b_t^d|v_t^d) \tag{8}$$

$$= \pi'_a(a_t^j|u_t^j,h_t^j;\theta)\pi'_b(b_t^{j-1}|v_t^{j-1})P'(a_t^{j-1}|a_t,u_t;\theta) \tag{9}$$

ただし，(8)では対局相手 B のシミュレーション方策 π'_b は既知の関数に固定しておく．また，シミュレーション方策 π'_a 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^j までの局面と指し手の履歴 $h_t^j \equiv \{u_t,a_t,u_t^1,a_t^1,..,u_t^{j-1},a_t^{j-1}\}$に依存する場合も考慮している．

上記の累積選択確率は，コンピュータチェスの fractional-ply extensions における *depth*[11]や，激指の「実現確率」[6]と同様な概念である．ただし，探索木中の手番局面で累積選択確率を指し手の選択確率 π'_a や π'_b に比例して子ノードへ割り振り，各ノードにおける合法手の選択確率の和が 1 になるように正規化されている点が depth や「実現確率」と異なっている．これにより，指し手選択における兄弟手間の競合が枝刈りに反映されている．

次に，探索中において，A が指し手 a_t^d を選択後，その枝を刈らないで成長させる確率（以下，成長確率または生き残り確率）を $P_{sv}(a_t^d;\theta)$とする．手番局面 u_t から指し手 a_t を経由した深さ $D(u)$の leaf 局面 u への遷移確率 $P(u|a_t,u_t)$は，

$$P(u|a_t,u_t;\omega,\theta) = \prod_{d=0}^{D(u)-1} \pi_a(a_t^d|u_t^d;\omega)P_{sv}(a_t^d;\theta)\pi_b(b_t^d|v_t^d) \tag{10}$$

と表される．本研究では，枝の成長確率 $P_{sv}(a_t^d;\theta)$のために(8)の累積選択確率 $P'(a_t^j|a_t,u_t;\theta)$ を次のように用いる．

$$P_{sv}\left(a_t^d;\theta\right)=P_{sv}\left(a_t^d\left|u_t^d,h_t^d;\theta\right.\right)\equiv g\left(P'\left(a_t^{\,d}\left|a_t,u_t;\theta\right.\right);\mu,\tau\right) \quad (11)$$

$$g\left(x;\mu,\tau\right)\equiv\frac{1}{1+e^{-(x-\mu)/\tau}} \quad (12)$$

(8),(9)の累積選択確率と，それを用いた(11)の成長確率の定義，さらに，その成長確率と着手方策とを用いて計算される leaf 局面への遷移確率の三者の関係を表した模式図を図 2 に示す．(10)~(12)から分かるように，シミュレーション方策により選択される確率が高い枝は枝刈りされにくく成長しやすい．

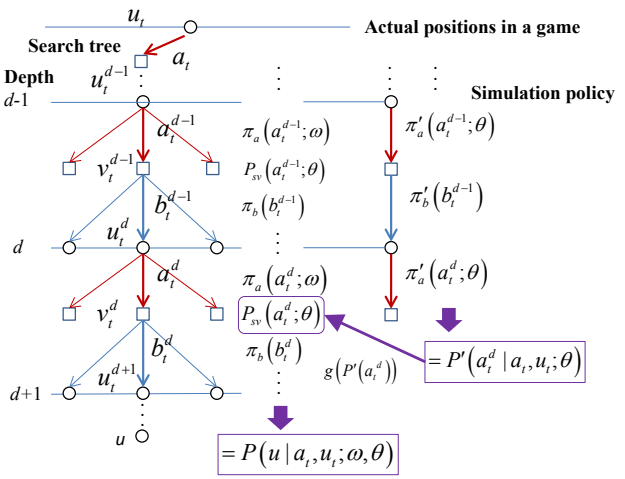

図 2　累積選択確率 $P'\left(a_t^{\,j}\left|a_t,u_t;\theta\right.\right)$ により成長確率 $P_{sv}(a_t^{\,d};\theta)$ と遷移確率 $P(u|a_t,u_t;\omega)$ を計算する本モデルの説明．

Figure 2　Illustration of a model where extention probability $P_{sv}(a_t^{\,d};\theta)$ and transition probability $P(u|a_t,u_t;\omega)$ are calculated from cumulative selection probability $P'\left(a_t^{\,j}\left|a_t,u_t;\theta\right.\right)$.

図 2 の左側には leaf 局面 u への遷移確率が探索 path 上の着手方策による指し手の選択確率と枝の成長確率の積により計算されることが示されている．一方，右側には，枝の成長確率がそこまでのシミュレーション方策による指し手の選択確率の積である累積選択確率により計算されることが表されている．

3.2　シミュレーション方策の学習

シミュレーション方策が次の Boltzmann 分布である場合を考える．

$$\pi'_a\left(a_t^j\left|u_t^j,h_t^j;\theta\right.\right)=\exp\left(E'_a\left(a_t^j,u_t^j,h_t^j;\theta\right)/T'_a\right)/Z'_a \quad (13)$$

$$Z'_a\equiv\sum_{x\in A\left(u_t^j\right)}\exp\left(E'_a\left(x,u_t^j,h_t^j;\theta\right)/T'_a\right) \quad (14)$$

$$E'_a\left(x,u_t^j,h_t^j;\theta\right)\equiv\sum_{i=1}^{N_f}\theta_i f_i\left(x;u_t^j,h_t^j\right) \quad (15)$$

ここで，$f_i\left(a;u,h\right)$ は指し手 a の良さを評価する特徴量(feature)であり，激指の事例 6)では「王手かどうか」「ひもをつける手かどうか」などを表している．θ_i はその重み係数である．この場合，θ の学習則は(4),(5)と同様に，

$$\Delta\theta=\varepsilon\cdot r\sum_{t=1}^{L_a}e_\theta\left(t\right) \quad (16)$$

$$e_\theta\left(t\right)=\left(1/T_a\right)\left[\partial E_a^*\left(a_t,u_t;\omega,\theta\right)/\partial\theta-\sum_{x\in A\left(u_t\right)}\pi_a\left(x\left|u_t;\omega,\theta\right.\right)\partial E_a^*\left(x,u_t;\omega,\theta\right)/\partial\theta\ \right] \quad (17)$$

と表される [4]．ただし，

$$\partial E_a^*\left(a_t,u_t;\omega,\theta\right)/\partial\theta=\sum_{u\in U_D(a,u_t)}\partial P\left(u\left|a_t,u_t;\omega,\theta\right.\right)/\partial\theta\cdot V\left(u;\omega\right) \quad (18)$$

である．ここで(10)より，

$$\partial P\left(u\left|a_t,u_t;\omega,\theta\right.\right)/\partial\theta=\sum_{d=0}^{D-1}e''_\theta\left(a_t^d;\theta\right)\cdot P\left(u\left|a_t,u_t;\omega,\theta\right.\right) \quad (19)$$

となる．ただし，

$$e''_\theta\left(a_t^d;\theta\right)\equiv\partial\ln P_{sv}\left(a_t^d;\theta\right)/\partial\theta \quad (20)$$

$$=\frac{P'\left(a_t^{\,d}\left|a_t,u_t;\theta\right.\right)}{P_{sv}\left(a_t^d;\theta\right)}\cdot\frac{d}{dx}g\left(x;\mu,\tau\right)\Big|_{x=P'\left(a_t^{\,d}\left|a,u_t;\theta\right.\right)}\cdot\sum_{j=0}^{d}e'_\theta\left(a_t^{\,j}\left|a_t,u_t\right.\right) \quad (21)$$

は，シミュレーション方策の対数微分

$$e'_\theta\left(a_t^j\left|a_t,u_t\right.\right)\equiv\partial\ln\pi'_a\left(a_t^j\left|u_t^j,h_t^j;\theta\right.\right)/\partial\theta \quad (22)$$

から計算できる．

4.　PGLeaf 法を用いた学習の高速化

4.1　決定論的な枝成長の場合

3.2 で述べたシミュレーション方策の学習において，2.2 の PGLeaf 法を適用する．PGLeaf 法では $\alpha\beta$ 探索を用いるので探索木中の最善応手手順の path だけを考える，このとき，PV leaf 局面 u^* への遷移確率 $P(u^*|a,u_t)$ は，(10)から

$$P\left(u^*\left|a_t,u_t;\theta\right.\right)=\prod_{d=0}^{D\left(u^*\right)-1}P_{sv}\left(a_t^d;\theta\right)\Big|_{PV} \quad (23)$$

と表される．ただし，上式の右辺では PV に沿った path 上での演算を記号 $|_{PV}$ で表してある．このとき，(1)の指し手評価の期待値は，

$$E_a^*\left(a_t,u_t;\omega,\theta\right)\approx P\left(u^*\left|a_t,u_t;\theta\right.\right)V\left(u^*;\omega\right) \quad (24)$$

$$=V\left(u^*;\omega\right)\prod_{d=0}^{D\left(u^*\right)-1}P_{sv}\left(a_t^d;\theta\right)\Big|_{PV} \quad (25)$$

と近似される．(2)の着手方策においても(25)を用いる．また，θ の学習則中の表式(18),(19)は，それぞれ，

$$\partial E_a^*\left(a_t,u_t;\omega,\theta\right)/\partial\theta\approx\partial P\left(u^*\left|a_t,u_t;\theta\right.\right)/\partial\theta\cdot V\left(u^*;\omega\right) \quad (26)$$

$$\partial P\left(u^*|a_t,u_t;\omega,\theta\right)/\partial\theta=\partial/\partial\theta\cdot\prod_{d=0}^{D\left(u^*\right)-1}P_{sv}\left(a_t^d;\theta\right)|_{PV} \quad (27)$$

$$=\sum_{d=0}^{D\left(u^*\right)-1}e_\theta''\left(a_t^d;\theta\right)|_{PV}P\left(u^*|a_t,u_t;\theta\right) \quad (28)$$

と表される．

今，探索木の枝を成長させる際に，ある閾値 K_{th} を設定し，枝の成長確率がこれを上回った場合，すなわち $P_{sv}(a^d;\theta)>K_{th}$ であれば枝を成長させ，下回った場合には枝刈りを行う決定論的な枝成長を考える．この場合の(16),(17)の学習則は，(26)からわかるように実現手以下の PV 上の遷移確率 $P\left(u^*|a_t,u_t;\theta\right)$ を強化し，逆に，その兄弟手以下の PV 上の遷移確率を抑制している．これらの強化と抑制は(23)の枝の成長確率に影響を与えることがわかる． 例えば，実現手以下の PV 上の枝の成長確率を増加させ，逆に，その兄弟手については PV 上の枝の成長確率を減少させる．したがって，探索時間を一定とする条件下では，実現手以下の最善応手手順を深く読み，逆に，その兄弟手については最善応手手順の読みの深さを抑制するように枝成長を行って探索深さの制御を行う．

また，(9)~(11)のように枝の成長確率 $P_{sv}(a^d;\theta)$はシミュレーション方策の指し手選択確率 π'_a や π'_b により計算されると仮定した．この選択確率は実現確率とは異なり，各ノードにおいて総和が 1 になるように各合法手に確率値を割り振っているので，PV 上の枝を優先的に成長させることは PV 以外の枝を抑制することにつながる．したがって，実現手以下の探索木では PV 以外の枝は早めにカットされるように，逆に，その兄弟手については PV 上の枝の成長を抑制し，PV 以外の枝を成長させて指し手の評価値を下げる方向に θ の学習が進むことになる．

4.2 確率的な枝成長の場合

また，探索木の枝を成長させる際に，実際に乱数を使って成長確率 $P_{sv}(a;\theta)$の確率で枝を成長させる確率的な枝成長を考える．この場合，乱数の値によって出来上がる探索木は異なり，それぞれの探索木の PV や PV 上の leaf 局面も異なる．その結果，指し手の評価値は 4.1 で述べた決定論的に枝成長を行う方法とは異なり一意的には定まらない．そこで，N_{try} 本の探索木を作成し，その平均値により指し手の評価を行うこととする．すなわち，

$$\partial E_a^*\left(a_t,u_t;\omega,\theta\right)/\partial\theta\approx\frac{1}{N_{try}}\sum_{i=1}^{N_{try}}\partial P\left(u^*(i)|a_t,u_t;\omega,\theta\right)/\partial\theta\cdot V\left(u^*(i);\omega\right) \quad (29)$$

$$\approx\frac{1}{N_{try}}\sum_{i=1}^{N_{try}}\left[\sum_{d=0}^{D\left(u^*(i)\right)-1}e_\theta''\left(a_t^d;\theta\right)|_{PV(i)}\right]\cdot V\left(u^*(i);\omega\right) \quad (30)$$

により計算する．ただし，$u^*(i)$, $PV(i)$は i 番目の探索木の PV 上の leaf 局面 と PV を，$D(u^*(i))$は $u^*(i)$の深さを表している．この方法の概念を図 3 に示す．

実際に乱数を用いて確率的に枝成長をさせることは，成長確率が探索木の成長を通して PV 上の leaf 局面の評価値へどういう影響を与えるかをシミュレーションにより定量的に計算してくれる．シミュレーション方策中のパラメータの変動により，枝刈り後の探索木の形が大きく変動してしまうような局面では，4.1 の決定論的な枝成長を行う方法よりも向いているのではないかと期待される．

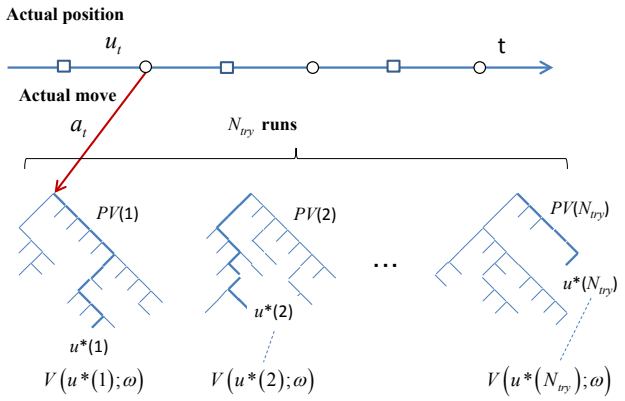

図 3　確率的な枝成長の下での PGLeaf 法

Figure 3　PGLeaf algorithm with stochastic search extention.

5. おわりに

本論文では，コンピュータ将棋において探索木の枝を成長させる際に，その枝までの探索経路に沿った指し手の累積的な選択確率の値を基に探索制御を行う方法を提案した．さらに，このときの指し手の選択に用いたシミュレーション方策を強化学習の一手法である方策勾配法により学習する学習法を提案した．今後は，実際のプログラムに実装し，学習実験を行っていく予定である．また，本手法は将棋以外のゲームにも適用できる一般性があると考えており，将棋以外にも適用を試みて行きたい．

謝辞　本研究は JSPS 科研費 26330419 の助成を受けた．謝意を表する．

参考文献

1) 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).
2) Beal, D. F. and Smith, M. C.: Temporal difference learning applied to game playing and the results of application to shogi, Theoretical Computer Science, Vol. 252, pp.105-119 (2001).
3) 薄井克俊，鈴木豪，小谷善行：TD 法を用いた評価関数の学習，第 4 回ゲーム・プログラミングワークショップ, pp.31-38 (1999).
4) 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, pp.118-121(2012).
5) 森岡祐一，五十嵐治一：方策勾配法と α β 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, pp.122-125 (2012).
6) 松原仁 編著：コンピュータ将棋の進歩⑥，第 4 章，共立出版 (2012).

7) 佐藤佳州，高橋大介：探索結果を利用した実現確率探索，情報処理学会論文誌，Vol. 51, No.11, pp.2021-2030 (2010).
8) ツツカナのアピール文書， http://www.computer-shogi.org/wcsc21/appeal/tsutsukana/WCSC21_tsutsukana_20110327.pdf
9) 五十嵐治一，森岡祐一，山本一将：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，情報処理学会研究報告，Vol. 2013-GI-30, No. 6, pp.1-8 (2013).
10) 松原仁 編：コンピュータ囲碁ーモンテカルロ法の理論と実践ー，共立出版 (2012).
11) Bjornsson, Y. and Marsland, T.: Learning Serach Control in Adversary Games, Advances in Computer Games9 (eds. H. van den Heirk and B. Monien), pp.157-174 (2001).