UCTにおけるPrior Knowledgeと方策学習を用いた個性の実現

渡辺順哉[1,a)]　金子知適[1,b)]

概要：近年，囲碁においてコンピュータプレイヤ AlphaGo が世界トップ棋士の一人である李セドルに勝利した [11]．囲碁を始めとする様々なゲームにおいてコンピュータプレイヤの強さはプロレベルに達し，個性に関する研究に期待が持たれている．本研究では，広く用いられている探索手法である UCT の囲碁における個性の実現を目標とする．UCT での個性の実現には prior knowledge とプレイアウト方策の調整が必要である．前者についてはどうぶつ将棋を題材とした先行研究がある [1]．この先行研究では prior knowledge を用いることで指し手に特徴を持つプレイヤの実現に成功しているが，本来勝率が低いノードを高評価することでプレイヤが弱くなってしまう問題点がある．強さの調整には様々な手法が考えられるが，方策学習によって強さを調整し個性を実現する研究は行われていない．そこで，本研究では UCT バランシング [2] という学習法と prior knowledge を組み合わせ，探索全体でのバランスを調整し個性を実現することを提案する．また，強さの具体的な調整手法として，学習局面を調整することを提案する．実験結果から，prior knowledge によって囲碁における打ち手に特徴が現れること，方策学習の局面数を調整することで強さが制御できることが確認された．また，提案手法で学習した方策を用いることで，対戦の段階で prior knowledge を用いない場合においても着手が特徴を持つ傾向があることが分かった．

Implementation of Playing Style by Prior Knowledge and Learning of Playout Policy in UCT

Junya Watanabe[1,a)]　Tomoyuki Kaneko[1,b)]

Abstract: Computer player AlphaGo won Lee Sedol, who is one of the world's top player [11]. The strength of the computer player has reached the professional level in variety of games. So, research of playing styles is expected. In this paper, we aim the realization of playing styles by using UCT which is a search method widely used in Go. To realize playing styles in UCT, it is necessary to use prior knowledge and to learn a playout policy. In the previous research on Dobutsu-shogi [1], only prior knowledge is adjusted. Although this previous research realized playing styles by using prior knowledge, there is a problem that the strength of players becomes low by highly evaluating originally low wining percentage nodes. The strength can be adjusted by various methods, but research have not been conducted to realize the adjustment and playing styles by learning a playout policy. In this paper, we propose to realize playing styles to adjust the balance of entire search by combining UCT balancing [2] and prior knowledge. Furthermore, we propose to adjust the number of potisions in the training example to control the strength of computer players. Experimental results shows that it is possible to adjust strength by adjusting the number of learning potisions and to realize playing style by prior knowledge in Go. It is also confirmed that a playing style can be realized by learning of playout policy, without giving prior knowledge toward the style.

1. はじめに

近年，囲碁に置いて世界トップクラスの棋士に勝利するなど，様々なゲームにおいてコンピュータプレイヤの実力

[1] 東京大学大学院総合文化研究科
Department of General Systems Studies, Graduate School of Arts and Sciences, The University of Tokyo
[a)] watanabe@graco.c.u-tokyo.ac.jp
[b)] kaneko@graco.c.u-tokyo.ac.jp

は人間のトップクラスに到達した．その上で、人間のプレイヤが楽しむために、強いだけでなく個性を持ったプレイヤーの実現に期待が持たれている．本研究では，囲碁をはじめとする様々なゲームで広く用いられている探索手法である UCT を用いて，囲碁における個性を実現することを目標とする．UCT での個性の実現には prior knowledge を用いた手法とプレイアウト方策の調整が有効であると考えられる．志水らの研究 [1] では，prior knowledge を用いて個性の持つ手に勝率と訪問回数を優遇して与えることでどうぶつしょうぎにおける個性を実現している．この手法の問題点としては本来勝率が低いノードを高評化することでプレイヤの棋力が下がってしまうことが挙げられる．志水らはこの点に関して，どうぶつしょうぎの完全データベースを用いた前向き枝刈を用いることで強さの調整を行っている．しかし，完全データベースの存在しない多くのゲームにおける強さの調整において課題が残る．強さの調整には様々な手法が考えられるが方策の調整によって強さの保持，または調整を行い，個性を実現する研究は行われていない．そこで本研究ではシミュレーションバランシング [3] という学習法と UCT，prior knowledge を組み合わせ，探索全体でのバランスを調整することで，強さの保持または調整を行っていく．

2. 関連研究

本説ではモンテカルロ木探索，UCT アルゴリズム，progressive widening [8]，prior knowledge [6] 及び UCT における既存研究での個性の実現手法などを紹介する．

2.1 モンテカルロ木探索

モンテカルロ木探索は，プレイアウトのスコアを用いた統計的な局面評価と mini-max 探索を組み合わせた探索手法である．ここで，プレイアウトとはある局面から終局面までランダムに手を進め，終局面からスコアを獲得するという一連の手続きである．基本的なモンテカルロ木探索のアルゴリズムの概要を以下に示す．

(1) ルートノードから，評価値の高いノードを順に選択しリーフノードまで到達.
(2) リーフノードからこれまでに行なわれているプレイアウトの回数が閾値に達しているならば，全ての合法手を展開し 1 回ずつプレイアウトを行なう．ただし，後に述べる progressive widening を用いる場合は,「全ての合法手を展開」するのではなく最も有望な 1 手のみ展開する．そうでなければリーフノードから 1 回プレイアウトを行い，得られたスコアを用いて，手順 (1) で巡ったノードの評価値を更新.
(3) 手順 (1)，(2) を制限時間まで繰り返す
(4) ルートノードからの訪問回数 (または勝率など) が最大のノードを選択

ここで，ノードの評価値として単純に勝率を用いた場合，最善局面を表すノードであっても最初のプレイアウトでたまたま負けてしまうと，上記アルゴリズムの手順 1 で選択されなくなってしまう．この問題点に対して，ノードへの訪問回数を評価値として考慮したのが UCT アルゴリズムである．具体的には，UCT アルゴリズムでは以下の UCB1 値をノードの評価値として用いる．

$$UCB1 = V + C\sqrt{\frac{logN}{n}}$$

ここで，n はノードへの訪問回数，N は親ノードの訪問回数，C は定数である．この様に UCB1 値をノードの評価値として用いることで，最初のプレイアウトでたまたま負けてしまったノードにおいてもステップ 1 で再選択されるようになる．

2.2 Progressive Widening

2.1 節のアルゴリズム手順 1 では全ての合法手を展開しているが，序盤における 1 線などの明らかに悪手と思われるノードを展開しプレイアウトを行うのは非効率である．この問題点に対して，progressive widening [8] では手順 1 のノード展開の際に全てのノードを同時に展開するのではなく，ノードの静的な評価値の順にノードへの訪問回数に応じてノードを徐々に展開する．この様にノードを展開することで，評価値の高いノードを優先して探索することが出来る様になる．なお具体的には，あるノード S に評価値が n 番目のノードが追加されるのは，ノード S への訪問回数が以下で定義される t_{n-1} 回に達したときである [8].

$$t_{n+1} = t_n + 40 \times 1.4^n,$$
$$t_0 = 0.$$

2.3 UCT における prior knowledge を用いた個性の実現

Prior knowledge [6] は UCT において，事前知識や評価関数を用いた探索の効率化手法である．通常の UCT では，ノード展開をする際の勝率と訪問回数は一律に 0 で初期化されるが，prior knoledge を用いた場合では評価関数の評価が高い手に一定の勝率と訪問回数を与える. これによって，有望な局面を中心に探索することができ，プレイヤーの強さが向上することが知られている [6]．志水らの研究 [1] では，prior knowledge を用いて，個性を持つ手に勝率と訪問回数を与えることで個性のあるプレイヤーを実現している. しかし, 個性を持つ手が選択される確率が上昇すると同時に, 本来勝率の低いノードが選択される可能性が高くなり, プレイヤーの強さが損なわれてしまうことが指摘されている．この問題点に対して，既存研究 [1] では完全データベースを用いた枝刈りを UCT と組み合わせることで対応している．しかし，完全データベースースが得られてい

ない多くのゲームにおいて，プレイヤーの強さを以下に調整するかには課題が残る．

3. 方策学習

本説ではプレイアウト方策の学習手法であるシミュレーションバランシング，及び UCT バランシングについて紹介する．

3.1 シミュレーションバランシング

シミュレーションバランシング [3] は方策のバランスを最適化する学習法である．シミュレーションバランシングで学習される方策は，プレイアウト中の手選択において必ずしも良い手を選択しない．シミュレーションバランシングでは以下の関数 L を最小化するように学習を行う．

$$L = E_\rho\Big[(E_{\pi_\theta}[z|s] - V^*(s))^2\Big] \tag{1}$$

ここで，z は報酬，ρ は局面集合，π_θ は方策，$V^*(s)$ は局面 s の minimax 値である．また，$E_{\pi_\theta}[z|s]$ は局面 s において方策 π_θ に従ってプレイアウトした際に得られる報酬 z の期待値である．なお，具体的な方策 π_θ としてはソフトマックス方策を用いる [7]．ソフトマックス方策では局面 s において手 a を選ぶ確率 $\pi_\theta(s,a)$ は

$$\pi_\theta(s,a) = \frac{e^{\phi(s,a)^T\theta}}{\sum_b e^{\phi(s,b)^T\theta}}$$

と定義される．ただし，$\phi(s,a)$ は特徴ベクトル，θ は重みベクトルである．

つまり，この学習法では局面の minimax 値とプレイアウトの平均報酬の平均自乗誤差を最小化するように方策の学習を行なう (本研究では minimax 値は-1, 1 の 2 値のいずれかという立場を取り，報酬 z の期待値は [-1, 1] の範囲の実数値となる)．式 (1) を目的関数としたとき，重み θ の具体的な学習法としては強化学習の 1 手法である方策勾配法を用いる．実際の学習はアルゴリズム 1 によって行なう [3]．なお，アルゴリズム 1 における $\nabla_\theta \log \pi_\theta(s_t,a_t)$ の計算は次の様に行なわれる．

$$\nabla_\theta \log \pi_\theta(s_t,a_t) = \phi(s,a) - \sum_b \pi_\theta(s,b)\phi(s,b)$$

ただし，$\phi(s,a)$ は局面 s における手 a の持つ特徴ベクトル，b は全ての合法手を表す．

ここで，V はプレイアウトの報酬の平均値，α はステップサイズと呼ばれる定数．M, N はプレイアウト数である．なお，minimax 値 $V^*(s)$ の値は実際には求めることが出来ないので，その代わりに十分時間をかけたモンテカルロ木探索による局面の評価値 $\hat{V}^*(s)$ を用いる．

3.2 UCT バランシング

渡辺らの研究 [2] では，simulation barancing と UCT を組み合わせた UCT バランシングという学習法を提案している．この手法では次の目的関数 M を最小化することによって学習を行なう．UCT バランシングでは具体的に次の目的関数 M を用いる手法を提案する．

Algorithm 1 式 (1) を目的関数としたオンライン学習

```
for all s_1 ∈ training set do
  V ← 0, g ← 0
  for i = 1 to M do
    simulate (s_1, a_1, ..., s_T, a_T; z) using π_θ
    V ← V + z/M
  end for
  for j = 1 to N do
    simulate (s_1, a_1, ..., s_T, a_T; z) using π_θ
    g ← g + (z/NT) Σ_{t=1}^{T} ∇_θ log π_θ(s_t, a_t)
  end for
  θ ← θ + α(V̂*(s_1) − V)g
end for
```

$$M = E_\rho\Big[(E_{UCT\pi_\theta}[z|s] - V^*(s))^2\Big] \tag{2}$$

ただし，$E_{UCT\pi_\theta}[z|s]$ は局面 s から方策 π_θ の元で UCT 探索を行い，UCT アルゴリズムが最も訪問したノードで得られる報酬の期待値である．従って，UCT バランシングでは局面の minimax 値と UCT 探索を行なった際に得られる報酬の期待値の平均自乗誤差を最小化するように方策の学習を行なう．なお，具体的な UCT バランシングの学習はアルゴリズム 2 によって行う [2].

Algorithm 2 式 (2) を目的関数としたオンライン学習

```
for all s_1 ∈ training set do
  V ← 0, g ← 0
  for i = 1 to M do
    simulate (s_1, a_1, ..., s_T, a_T; z) using UCT+π_θ
  end for
  select most visited node N.
  V ← N's win rate
  for j = 1 to N do
    simulate (s_1, a_1, ..., s_T, a_T; z) using UCT+π_θ
    g ← g + z/(N(T−T'+1)) Σ_{t=T'}^{T} ∇_θ log π_θ(s_t, a_t)
  end for
  θ ← θ + α(V̂*(s_1) − V)g
end for
```

4. 提案手法

4.1 UCT バランシングと Prior knowlege の組み合わせ

渡辺らの手法 [2] のメリットとして，探索を行いながら学習ができる点が挙げられる．本研究では個性の実現手法として，この学習法に prior kwoledge を組み合わせることを提案する．具体的には，既存研究 [2] の学習アルゴリズムで UCT 探索を行なう際に，prior knowledge を組み合わせる．このように学習の段階で prior knowledge を組み合わせることで，個性を持った手が多く展開された探索木に

おいて行なわれるプレイアウトを学習対象にすることが出来る．その結果，個性のある着手に適したプレイアウト方策が学習されると期待される．加えて，学習後の方策を用いた UCT がより個性を持つ手を選択するようになることも期待される．

4.2 目的関数の調整による強さの制御

志水らは完全データベースを用いた枝刈りを UCT と組み合わせることプレイヤーの強さの制御を行っている．しかし，完全データベースを用いた枝刈りを行うことが出来るゲームは限られているので，より汎用性のある手法でも強さを制御出来ることが望ましい．本手法では UCT での個性における強さの制御手法として，方策学習を行う際の学習局面数を調整することを提案する．具体的には，3.1 節の (1) 式，3.2 節の (2) 式の目的関数を最小化する際の学習局面数を調整することで，ノードの評価値の精度を調整し，その結果強さが制御できると期待される．

5. 実験結果と考察

5.1 実験条件

本研究では 9 路盤での囲碁を対象とし，個性として’相手の直前手へのツケ’，’序盤 10 手以内における相手の直前手の周囲 8 箇所以外’ の 2 つの特徴を用いる．前者の特徴の具体的な定義は’相手の石数 1 の直前手の周辺 4 箇所（この座標を z とする）．ただし z のさらに周辺 4 箇所に相手の直前手以外の手が存在せず，かつ z は 1 線ではない’とした．従って，この特徴が盤面に現れるのは石数の少ない序盤に多く，石数が増える終盤ほど少なくなる．なお，本説では以下この 2 つの特徴をそれぞれ 5.2，5.3 節で分けて取り扱うこととする．

学習局面の生成と対局実験に用いた対局プログラムは参考文献 [5] の実践編に記載されている囲碁プログラムを参考にして作成した．対局プログラムには 2.2 節で解説した progressive widenig を実装した．なお，progressive widening では 3x3 パターンと盤端からの距離，プレイアウト方策では 3x3 パターンに加えて参考文献 [2]，[4] と同様に以下の特徴を用いた．

(1) 直前に相手の打った手の周囲 8 カ所．さらに，以下の特徴 2 から 6 も含む．
(2) 直前にアタリをかけられた自分の連に接する敵連を取る．ただし，特徴 3 に含まれる場合は除く．
(3) 直前にアタリをかけられた自分の連に接する敵連を取る．ただし，助けた連がアタリになる．
(4) 直前にアタリをかけられた自分の連をノビる．
(5) 特徴 4 の条件が成り立ち，かつノビた連がアタリ．
(6) 直前にダメを 3 から 2 に減らされた自分の連に接する敵連の内，ダメ 2 でかつ打てば必ずその連が取れる．

なお，progressive widening で用いる重みの学習は，コンピュータ囲碁の対局サイトである cgos 上で行われたレーティング 2700 台の強いコンピュータ囲碁プログラム同士の対局記録 [10] のうち 1000 局を用いて MM 法 [8] によって行なった．また，参考文献 [5] の実践編に記載されているプログラムを参考にして UCB1 値の定数項の値は 0.31 とした．

学習で用いる棋譜は，プレイアウト数を 1000 回，プレイアウト方策を一様ランダムとした対局プログラムの自己対戦を 5 万局によって生成した．そして 1 局あたり 1 局面をランダムに選択し，Fuego で 20000 プレイアウト思考して得られた評価値をラベル付けて学習局面 5 万局面を生成した．

5.2 相手の直前手へのツケを特徴とした場合

5.2.1 個性率と勝率

UCT バランシングと prior knowledge を組み合わせた学習を行う際の prior knowledge におけるノードの勝率と訪問回数の初期値を定める為に，まず対局プログラムに prior knowledge を実装しプレイアウト回数を 75 回とした Fuego と対戦を行い，結果を以下の図 1，に示した．なお，図の横軸は prior knowledge における訪問回数の初期値である．従って，横軸の値が 0 の場合は prior knowledge を用いずに対戦を行っている．なお，着手に個性を持った手が現れる確率の測定においては，全手数と序盤 10 手以内の 2 通りの場合で測定した．また，対局プログラムのプレイアウト回数は 1000 回，プレイアウト方策には 5 万局面をシミュレーションバランシングで 20 反復学習した重みを用いた． また，’直前の相手の手にツケる’特徴を持

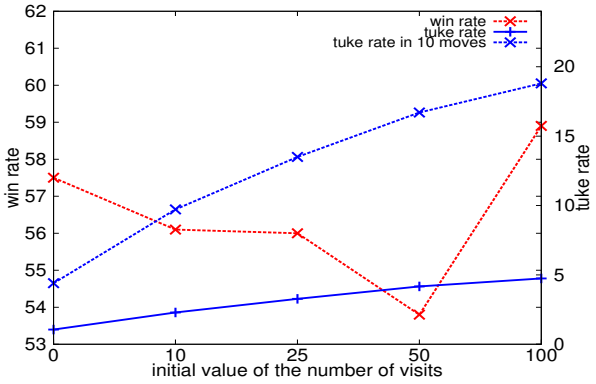

図 1 シミュレーションバランシングで学習した場合

つ手が progressive widening によって枝刈りされている状況では，prior knowledgede で勝率と訪問回数を優遇することが着手選択に影響を与えなくなる可能性があると考えられる．そこで，’直前の相手の手にツケる’特徴を持つ手の progressive widening での評価値に 1000 を足して評価値を強制的に上位にした場合の実験を図 2 で同時に行い，progressive widening の着手の評価値順序が着手の

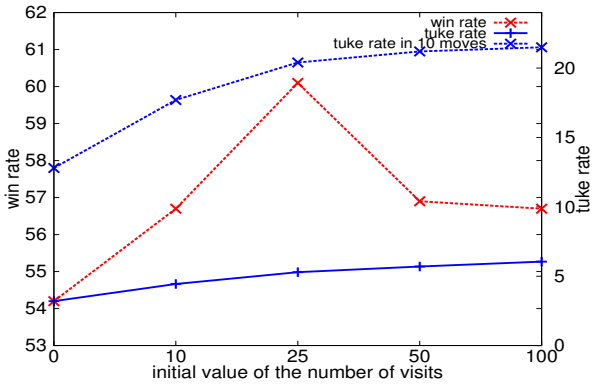

図 2 シミュレーションバランシングで学習した場合，なお対戦実験の際に個性を持った手の progressive widening での評価値に 1000 を追加

個性率に与える影響を調査した．図 1，図 2 の結果から，図 1，図 2 いずれの場合においても着手に個性が現れる確率が，prior knowledge における訪問回数の初期値に対して単調に増加していることが分かる．また，図 1 と図 2 を比較すると，全手数，10 手以内のいずれの場合でも図 2 の方が着手に個性が現れる割合が高く推移していることが分かる．従って，progressive widening の評価値をそのままにしている図 1 の場合では個性を持った手がある程度枝刈りされていると分かる．また，図 2 では訪問回数の初期値が 10，25，50 のときに勝率が上昇していることが分かる．次に，UCTB と prior knowledge を組み合わせた UCTB において，シミュレーションバランシングと同様な条件で学習と対戦実験を行い，結果を以下の図に示した．なお，UCTB におけるノードの展開の閾値は 50 とした．また，UCTB と prior knowledge を組み合わせた学習を行う際に，個性を持った手がなるべく展開されるように prior knowledge におけるノードの訪問回数の初期値は 100 回とし，progressive widening で個性を持つ手のノードの評価値に 1000 を加えた上で学習を行った． この結果から，図 3

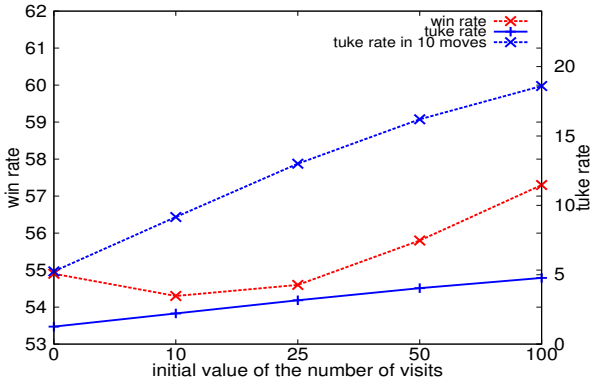

図 3 Prior knowledge を組み合わせていない UCTB で学習した場合

から図 6 の場合においてもシミュレーションバランシングで学習した場合と同様に，着手に個性が現れる確率は prior

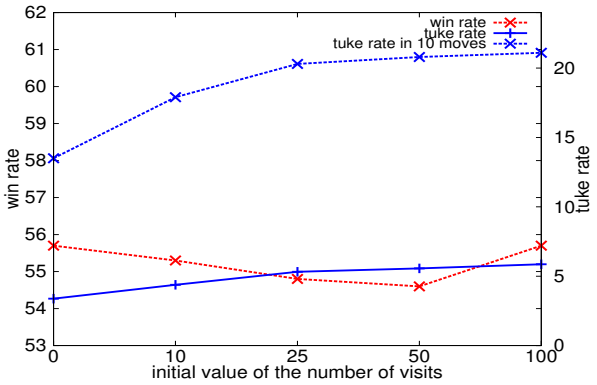

図 4 Prior knowledge を組み合わせていない UCTB で学習した場合，なお対戦実験の際に個性を持った手の progressive widening での評価値に 1000 を追加

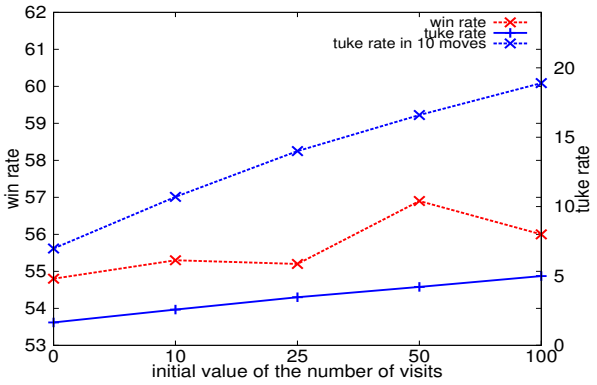

図 5 提案手法で学習した場合

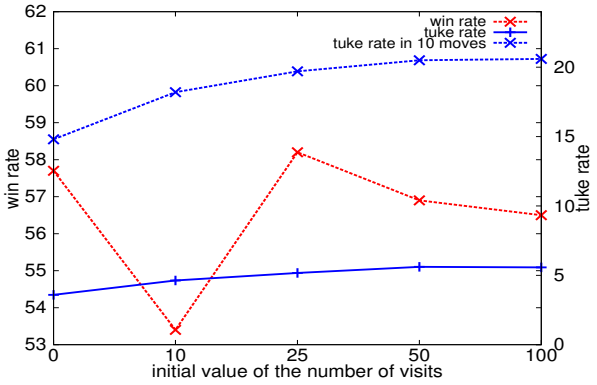

図 6 提案手法で学習した場合，なお対戦実験の際に個性を持った手の progressive widening での評価値に 1000 を追加

knowledge における訪問回数の初期値に対して単調に増加していることが分かる．また，強さに関してはシミュレーションバランシングと提案手法ではっきりとした差が現れていないことが分かる．ここで，図 1 から図 6 の，横軸の値が 0 である場合の全手数における個性率を比較すると，progressive widening の評価値をそのままにしている図 1，図 3，図 5 の場合は，図 1 のシミュレーションバランシングの場合で 1.06%，図 3 の prior Knowledge と組み合わせない UCTB で 1.26%，図 5 の prior knowledge と組み合わせた UCTB で 1.65% であり，同じ図の訪問回数の初期

値が 0 回の場合における 10 手以内の個性率ではそれぞれ 4.4，5.24%，6.98% であった．また，図 2，図 4，図 6 の progressive widening での重みに 1000 を加えた場合においては，訪問回数の初期値が 0 回の場合における全手数での個性率がそれぞれ図 2 のシミュレーションバランシングで 3.20%，図 4 の prior knowledge を含まない UCTB で 3.38，図 6 の prior knowledge と UCTB を組み合わせて学習した場合で 3.59，同じ図の訪問回数の初期値が 0 回の場合における 10 手以内での個性率はそれぞれ 12.8%，13.5%，14.8% であった．従って，progressive widening の評価値のいずれの条件においても，prior knowledge と UCTB を組み合わせて学習した場合が着手に個性が現れる割合が一番高いことが分かった．従って，対局時に prior knowledge を用いない場合においても，事前に提案手法で学習を行うことで着手の個性率が上昇する傾向があることが分かった．

5.2.2 強さの制御

4.2 節で提案した学習局面数の調整による強さの制御についての検証を行う．まずシミュレーションバランシング，UCT バランシング，prior knowledge 入り UCT バランシングの 3 通りの学習法において，それぞれ学習局面数を 1000，5000，10000，50000 とした場合の目的関数の推移を以下の図に示す．なお，学習局面と同様な条件で学習局面とは別にテスト用の局面 1000 局を生成し，目的関数の値を計測した． 図 7 から 9 までの結果から 3 通りの学習法全てにおいて，学習局面を増やすことでテスト局面に対する目的関数の値が低く推移していることが分かる．

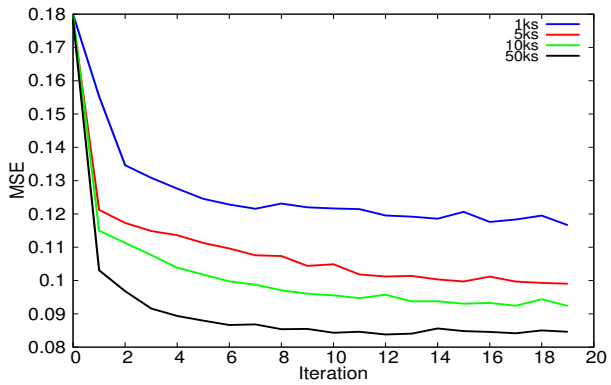

図 7 シミュレーションバランシングの場合

次にそれぞれの目的関数で 10 反復学習した重みを用いて対局実験を行い以下の図に示した．

図 10〜 15 の結果で共通しているのは，学習局面数を増加させることで個性を持つ手の割合を一定程度に保ちながら強さが上昇していることにある．従って，いずれの学習法においても学習局面を調整することで着手に個性を持たせながらも強さを調整できるということが分かった．

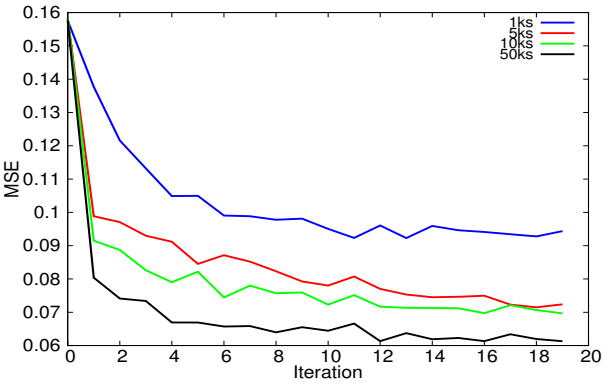

図 8 提案手法で学習した場合

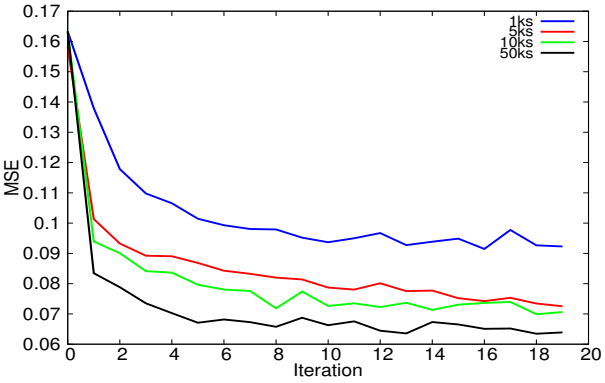

図 9 Prior knowledge を含まない UCTB で学習した場合

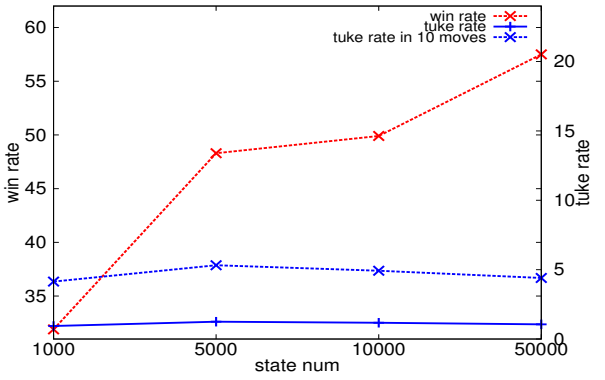

図 10 シミュレーションバランシングで学習した場合，prior knowledge におけるノードの訪問回数の初期値は 0

5.3 序盤 10 手以内における相手の直前手の周囲 8 箇所以外を特徴とした場合

本研究では ‘序盤 10 手以内における相手の直前手の周囲 8 箇所以外’ に該当する手を prior knowledge で優遇するために，‘序盤 10 手以内における相手の直前手の周囲 8 箇所’ のノードの勝率を n 回負けとして初期化する (以下の実験では n は 0 か 100)．

5.3.1 個性率と勝率，及び強さの制御

5.2 節と同様に 3 つの学習法でそれぞれ学習を行い，強さと個性率の測定を行う．学習局面は 1000，50000 の 2 通り，prior knowledge のノードの訪問回数の初期値は 0，100 の 2 通りで実験を行い，結果を表 1， 2 に示した． 表 1，

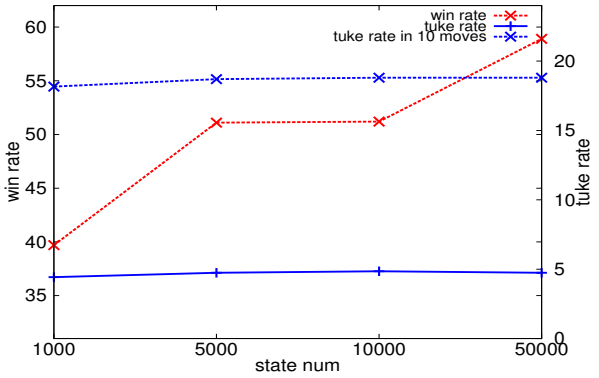

図 11 シミュレーションバランシングで学習した場合，prior knowledge におけるノードの訪問回数の初期値は 100

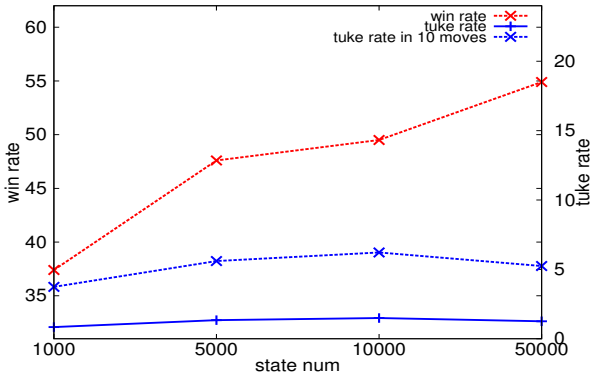

図 12 Prior knowledge を含まない UCTB で学習した場合，prior knowledge におけるノードの訪問回数の初期値は 0

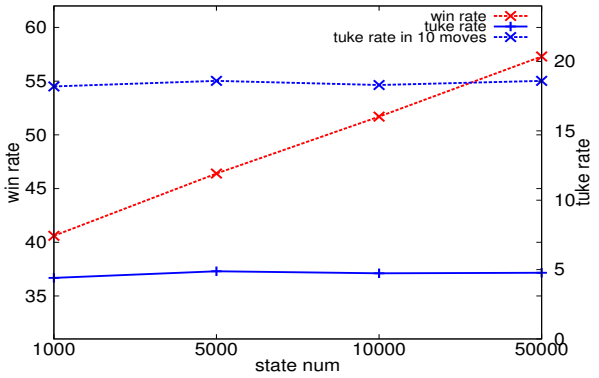

図 13 Prior knowledge を含まない UCTB で学習した場合，prior knowledge におけるノードの訪問回数の初期値は 100

表 1 Prior knowledge のノードの訪問回数を 100 回とした場合

学習法	学習局面数	強さ (%)	個性率 (%)
SB	1000	33.5	100
	50000	55.3	100
UCTB	1000	36.4	100
	50000	49.4	100
提案手法	1000	36.6	100
	50000	53.9	100

2 の結果から，prior knowledge によって個性率が全ての学習法，局面数において上昇していることがわかる．また，

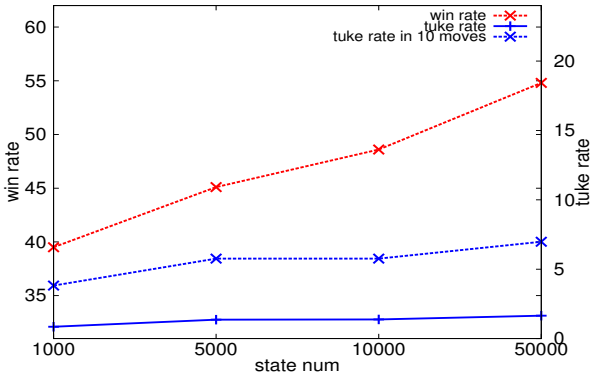

図 14 提案手法で学習した場合，prior knowledge におけるノードの訪問回数の初期値は 0

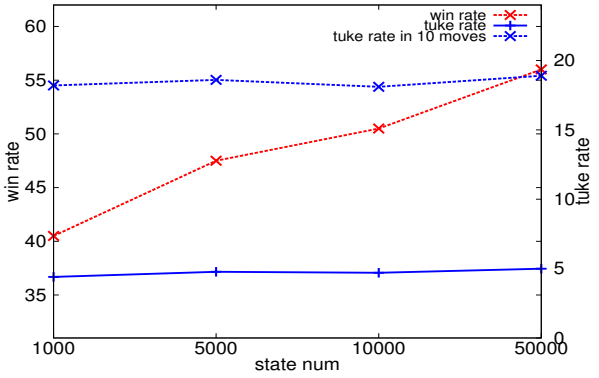

図 15 提案手法で学習した場合，prior knowledge におけるノードの訪問回数の初期値は 100

表 2 Prior knowledge のノードの訪問回数を 0 回とした場合

学習法	学習局面数	強さ (%)	個性率 (%)
SB	1000	31.9	77.2
	50000	57.5	75.3
UCTB	1000	40.2	77.3
	50000	55.7	74.4
提案手法	1000	37.3	78.5
	50000	52.9	73.7

表 1，表 2 いずれの場合においても，学習局面数を 1000 から 50000 に上昇させることで強さが上昇しており，5.2.2 節と同様に目的関数の値を調整することで強さが制御できることが分かった．

6. 結論

本研究では，UCT バランシングと prior knowldege を組み合わせた学習法と，学習局面数を調整し目的関数の値を調整することで強さを制御することを提案した．囲碁における prior knowledge を用いた個性の実現としては，’相手の直前手へのツケ’，’序盤 10 手以内における相手の直前手の周囲 8 箇所以外’ の 2 つの特徴を用い，prior knowledge によって着手が個性を持つ割合を調整出来ることを確認した．また，学習局面の数を調整することで着手が個性を持った割合をある程度保った上で強さを調整できることが分かった．なお，強さの点で提案手法が有望であることは確認出来なかったが，’相手の直前手へのツケ’ を特徴とした場合において提案手法によって学習を行うことで，prior

knowledge を用いない場合でも着手に個性を持った手の割合が上昇することが分かった．今後，19 路盤における実利派や厚み派など，より幅広い個性について検証していく必要があると考えられる．

参考文献

[1] 志水翔，金子知適．二人ゲームプレイヤの Prior Knowledge を用いた UCT による個性の実現手法と評価．ゲームプログラミングワークショップ 2014 論文集，第 2014 巻，pp.188-195，oct 2014.

[2] 渡辺順哉，美添一樹，金子知適．モンテカルロ木探索を統合したプレイアウト方策の最適化．ゲームプログラミングワークショップ 2015 論文集，第 2014 巻，pp. 5-11，oct 2015.

[3] Silver, D., Tesauro, G.: Monte-Carlo Simulation Balancing. In: ICML (2009).

[4] Huang, S., Coulom, R. and Lin, S.: Monte-carlo simulation balancing in practice. In: ICCG (2010).

[5] 松原仁 編, 美添一樹 · 山下宏 著 コンピュータ囲碁 モンテカルロ法の理論と実践. 共立出版 (2012)

[6] Gelly, S., Silver, D.: Combining online and offline knowledge in uct. In ICML (2007).

[7] Sutton, R. J, Barto, A. G.: Reinforcement Learning An Introduction. The MIT Press, Massachusetts (1998).

[8] Coulom, R.: Computing Elo ratings of move patterns in the game of Go. International Computer Games Association Journal, 30(4):198-208, 2007.

[9] Kocsis, L. Szepesvari, C.: Bandit based Monte-Carlo planning. 15th European Conference on Machine Learning, pp.282-293, 2006.

[10] `http://www.yss-aya.com/cgos9_201101_201204_2500over.cab`

[11] `https://deepmind.com/research/alphago/`