情報処理学会論文誌 Vol.55 No.11 2399–2409 (Nov. 2014)

対局に基づいた教師データの重要度の学習

佐藤 佳州[1,†1,a)] 高橋 大介[2]

受付日 2014年2月21日, 採録日 2014年9月12日

概要：近年，ゲームプログラミングの分野では機械学習が大きな注目を集めており，評価関数，探索深さ，モンテカルロ木探索の playout の方策等，多くのパラメータの学習で成功を収めている．現在のゲームプログラミングにおける機械学習では，人間のエキスパートの棋譜を教師として，その指し手に近づけるようにパラメータの調整を行っている．しかし，将棋等のゲームでは，コンピュータはすでに人間のトッププレイヤに迫る強さとなっており，単純に人間の指し手を再現することが必ずしも「強い」プレイヤの生成に結び付くとは限らない．本論文では，このような課題を改善するため，教師データに重要度を導入した学習手法を提案する．提案手法では，勝率を適応度とした進化的計算による重要度の学習と，重要度に従ったパラメータ学習を組み合わせた学習を行う．提案手法を将棋の評価関数，実現確率，playout の方策の学習へ適用した結果，従来手法との対局実験において有意に勝ち越すことに成功し，その有効性を示した．また，実験結果から局面の進行度や戦術等によって教師データの重要度に違いが生じることが分かり，教師データの効果的な利用により，より強いプログラムを実現する知識の獲得が可能となることを示した．

キーワード：ゲーム，人工知能，将棋

Learning Weights of Training Data by Game Results

Yoshikuni Sato[1,†1,a)] Daisuke Takahashi[2]

Received: February 21, 2014, Accepted: September 12, 2014

Abstract: Recently, machine learning is attracting much attention in the field of game programming, and it has succeeded in tuning evaluation functions, search depth, playout policies in Monte-Carlo Tree Search, etc. Existing machine learning methods in game programming tune parameters by using game records of human expert players. However, computer programs have almost the same strength as human professional players in some games such as shogi. Thus, learning by simply using human records is not necessarily good for generating strong computer players. In this paper, we propose a new learning method that estimates the importance of each training record by playing many games and tunes parameters according to the importance. The experimental results show the effectiveness of our method for learning evaluation functions, realization probability search, and playout policies. Moreover, the results show that features of training data such as progress of games or tactics affects their importance.

Keywords: game playing, artificial intelligence, shogi

1. はじめに

近年，ゲームプログラミングの分野では，機械学習が大きな注目を集めている．現在のゲームプログラミングで用いられている学習手法の多くは，人間のエキスパートの指し手を教師とし，その指し手に近づけるように各特徴のパラメータを調整している．この方法は，評価関数，探索深さの決定等，多くの場面で成功し，プログラムの性能向上

1 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
2 筑波大学システム情報系
Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
†1 現在，パナソニック株式会社先端技術研究所
Presently with Advanced Technology Research Laboratories, Panasonic Corporation
a) ysato@hpcs.cs.tsukuba.ac.jp

© 2014 Information Processing Society of Japan

に大きく貢献してきた．

一方で，このような人間の棋譜を教師とした学習によって獲得されるパラメータは，教師データの性質に大きく依存する．人間の棋譜は教師として完全に理想的なものとはいえず，悪手が含まれる場合が存在する，人間の手を真似ることが必ずしもコンピュータにとって最善とは限らない，といった問題点が存在する．従来は，将棋，囲碁等の複雑なゲームでは，コンピュータの強さは人間のプレイヤに大きく劣っていたため，このような問題を含む棋譜も理想的な教師データとして近似することができた．しかし，現在では，将棋等の複雑なゲームにおいても，コンピュータが人間のトッププレイヤに迫る強さとなっており，単純に人間の指し手を教師とする手法では，性能に限界が生じると考えられる．

このような問題は，本来「強い」パラメータを学習したいのに対して，それを人間の指し手との一致率で近似しているために生じているといえる．人間の棋譜を教師としない学習も研究されているものの，現在までのところ将棋等の複雑なゲームでは人間の棋譜を教師とした学習を明らかに上回る手法は存在していない．

本論文ではこのような現在のゲームにおける機械学習の問題点を改善するため，教師データに重要度（その学習局面が強いプレイヤの学習にどの程度寄与するか）を導入し，対局による重要度の学習と重要度に従った重み付き学習を組み合わせた学習手法を提案する．また，コンピュータ将棋を題材に，評価関数の学習，実現確率の学習，モンテカルロ木探索における playout の方策の学習，に提案手法を適用し，その有効性を検証する．

以下に本論文の構成を示す．2 章では，現在成功を収めている，棋譜を教師とした機械学習を中心に，ゲームプログラミングで用いられる機械学習の手法について述べる．3 章では，現在のゲームプログラミングにおける機械学習の問題点を述べ，その問題を改善する手法として，重要度を導入した学習を提案する．4 章では，提案手法を，(1) 評価関数，(2) 実現確率，(3) playout の方策の学習に適用し，その効果を検証する．その際，本論文では，学習するパラメータ数を抑えるため，棋譜単位，進行度単位の重要度を学習する．実験では，重要度を導入した場合の性能の向上，および学習された重要度の持つ意味合いについて検証する．最後に，5 章で今後の課題，6 章で本論文のまとめを述べる．

2.　関連研究

現在，ゲームの分野では様々なパラメータ調整において，人間の棋譜を教師とした教師あり学習が成功を収めている．本章では，現在，ゲームで成功を収めている棋譜を教師とした学習として，評価関数の学習，探索深さの学習（実現確率探索），モンテカルロ木探索における playout の方策の学習，について概要を説明する．また，提案手法と関連するその他の学習手法についても述べる．

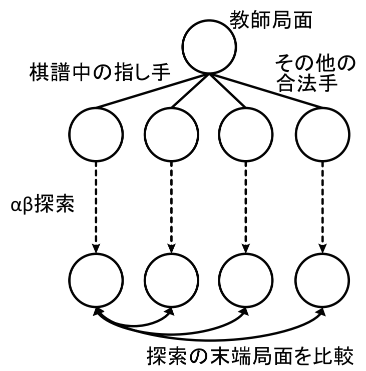

図 1　Comparison Training の概要
Fig. 1　Procedure of the Comparison Training.

2.1　評価関数の学習

ゲームの分野において，評価関数の学習は古くから研究されていた課題である．近年では，将棋において，機械学習による評価関数の学習が成功を収め，人間のトッププレイヤに迫る強さを得ている [1]．文献 [1] の学習は Comparison Training [2] と呼ばれる手法の一種であり，特にコンピュータ将棋では Bonanza メソッドとも呼ばれる．Comparison Training の概要を図 **1** に示す．

この手法では，棋譜中で実際の指し手と，それ以外の合法手との比較によって学習を行う．具体的には，棋譜中で指された手を正解手とし，それ以外の合法手の評価値を上回るように評価関数のパラメータを学習する．

パラメータ学習時には，PV（Principal Variation，最善手順）の生成と評価関数のパラメータの学習を繰り返し行うことを特徴としている．文献 [1] の学習手法では，具体的には式 (1) の目的関数の最適化を行っている．

$$J_1(P,\boldsymbol{v}) = \sum_{p \in P} \sum_{m=2}^{M_p} T\bigl(\xi(p_m,\boldsymbol{v}) - \xi(p_1,\boldsymbol{v})\bigr) \tag{1}$$

ここで，P は学習対象の局面集合，M_p は局面 p における合法手数，p_1 は棋譜中で実際に指された手，p_m はそれ以外の手，$\xi(p_m,\boldsymbol{v})$ は p_m において探索を行った場合の末端局面の特徴 $\boldsymbol{v}$ によって算出された評価値を示す．また $T(x)$ はシグモイド関数を表す．実際の学習時には，パラメータの値を安定させるために，駒の価値の総和等で拘束条件を課していることが多い．また，評価関数の学習では，数万個以上の非常に多くのパラメータを学習するため，L1 正則化を行うのが一般的であるが，式 (1) ではこれらの項は省略して表記している．

評価関数の学習は，コンピュータ将棋の強さを大きく向上させることに成功し，現在ではトップレベルのプログラムの多くが，Bonanza の学習をベースとした学習を用いている [3], [4], [5].

© 2014 Information Processing Society of Japan

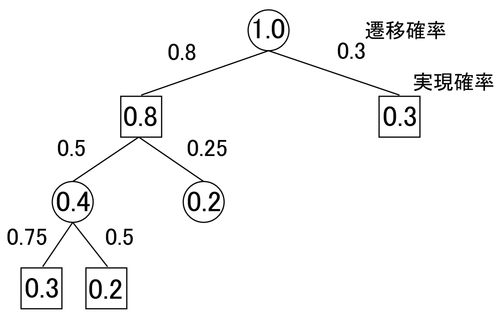

図 2　実現確率探索の例
Fig. 2　Example of the realization probability search.

2.2　実現確率探索

実現確率による探索打ち切りアルゴリズム [6]（以下，実現確率探索）は，知識に基づいた代表的な探索手法である．この探索手法は将棋を中心にさかんに研究が行われており [7], [8], [9], [10]，いくつかのトップレベルのプログラムでも採用されている．一般的な探索手法では，深さを探索の打ち切り条件とするのに対して，実現確率探索では深さの代わりに局面の実現確率を利用する．実現確率探索の探索例を**図 2** に示す．局面の実現確率は以下の式によって再帰的に定義される値である．

$$（実現確率）=（親局面の実現確率）\times（遷移確率） \quad (2)$$

ルート局面の実現確率は 1.0 とし，遷移確率には指し手が選択される確率を用いる．指し手の選択確率は，指し手を特徴ごとに分類し，その特徴に対応する指し手がプロの棋譜中でどれくらいの割合で指されたかを算出することにより求める．このように局面の実現確率を探索の閾値とすることで，実際に起こりやすい展開を重点的に深く探索できる点が実現確率探索の大きな特徴である．

遷移確率の学習は，従来は単純に実際の棋譜における選択確率を用いていたが，近年はロジスティック回帰による予測が良い結果を得ており [11]，本論文でもこの手法を用いる．ロジスティック回帰による実現確率の学習の手順を以下に示す．この手法では，n 個の特徴が存在し，i $(1 \leq i \leq n)$ 番目の特徴の値を x_i とすると，特徴の値 $(x_1, x_2, \ldots, x_n)$ を持つ指し手の遷移確率 p は式 (3) で表される．

$$p(x_1, x_2, \ldots, x_n) = \frac{1}{1 + \exp(-\sum_{i=1}^{n} w_i x_i)} \quad (3)$$

学習の際には，プロの棋譜において実際に指された手を正例，指されなかった手を負例とすることで，各特徴の重み w_i を推定する．この重み w_i の算出には，L1 正則化ロジスティック回帰が用いられることが多い．具体的には式 (4) の最適化によって w_i を算出する．

$$\min_{\boldsymbol{w}} \quad |\boldsymbol{w}| + C \sum \log(1 + e^{-y_j \boldsymbol{w}^T \boldsymbol{x}_j}) \quad (4)$$

ここで，y_j は正例，負例のラベル，C は正則化の重みを制御する定数である．実現確率探索も将棋において成功を収めている探索手法であり，評価関数の学習と同様，多くのプログラムが採用している手法となっている．以降，本論文中で，実現確率探索における遷移確率の学習を指す場合，単に「実現確率の学習」と表現する．

2.3　モンテカルロ木探索中の playout の方策の学習

モンテカルロ木探索では，乱数を利用したゲームのシミュレーション（playout）を何度も行い，その勝率によって局面を評価する．Playout 中の指し手の選択（playout の方策）は，完全なランダムでは良い結果を得ることは難しく，ゲームの知識を取り入れ，確率的に選択することが有効であることが知られている．Playout の方策にも機械学習が用いられており，棋譜を教師とした代表的な手法としては，指し手の Elo レーティングを用いた方策が存在する [12]．モンテカルロ木探索中の playout の方策には前節で述べた実現確率を用いることも可能であり，同等の意味合いを持つ．

2.4　学習棋譜がプログラムの強さに与える影響

学習棋譜が学習された評価関数の強さに与える影響についても，これまでにいくつかの研究が行われている [13], [14]．文献 [13] では，プロ，コンピュータ，アマチュアの棋譜を学習棋譜としたときの評価関数の強さの比較を行っており，アマチュアよりもプロ，コンピュータの棋譜を教師としたほうがやや強い評価関数を学習できるといった結果が報告されている．

また，一部のトップレベルのプログラムでは，学習棋譜の取捨選択，強いプレイヤの棋譜の重みを大きくするといったことも行われている [15]．ただし，これらの調整は，開発者が手作業で行ったものであり，学習棋譜の性質を十分に考慮した学習が行われているとはいえず，どの程度の効果が得られているかも明らかになっていない．

2.5　その他の学習手法

ゲームプログラミングの分野では，前述の手法に限らず，従来から様々な学習手法が提案されている．たとえば，文献 [16] では，強化学習の一種である TD 法を用いた駒の価値の学習が行われており，文献 [17] では進化的計算を用いた評価関数の学習が行われている．

また，これらの学習手法を組み合わせた学習手法も提案されている．文献 [18] では，局所探索を得意とする TD 学習と大域的探索を得意とする GA を組み合わせたハイブリッド GA により評価関数のパラメータ学習を行っている．この方法により，オセロの実験では従来手法（単一の学習手法を用いた場合）を上回る結果を得ている．

しかし，これらの強化学習や進化的計算によるパラメータ学習は，現在までのところ将棋のような複雑なゲームで

© 2014 Information Processing Society of Japan

は，人間の棋譜を教師とした学習を明確に上回る成果は得られていない．

3. 提案手法

3.1 人間の棋譜を教師とした学習の問題点

人間の棋譜を教師とした学習は，現在のところ多くの課題に対して良い結果を得ているものの，以下に示すような問題点が存在する．

第 1 に，人間の棋譜を絶対的に信頼しているため，教師とするのにふさわしくない局面も学習対象とすることがある．人間の棋譜を教師とした場合，様々な強さの対局者の棋譜が混ざったものを用いることになる．一般的に，学習棋譜としては，プロやアマチュア高段者等ある程度の強さの対局者のものを利用するが，対局者の強さにばらつきが存在するという点は本質的には解決できていない．また，人間のプレイヤの場合，棋譜中にミスによる悪手が含まれるという課題も存在する．現在では，囲碁や将棋等といった複雑なゲームにおいても，コンピュータは人間のトッププレイヤに迫る強さとなっており，単純に人間の強いプレイヤの棋譜を信頼した場合，性能向上に悪影響を及ぼす可能性があると考えられる．

第 2 に，従来手法では，すべての棋譜，局面を均等に扱っており，全体としてプロの指し手との一致率が上がるようにパラメータの調整を行っている．しかし，これは必ずしも強いプレイヤの実現に結び付くとは限らない．棋譜中に含まれる局面には，最善手が指せないと即負けにつながるような重要な局面もあれば，有力な手が多数存在するような局面も存在する．棋譜を教師として，一致率を向上させるようにパラメータを学習する手法では，序盤の駒の位置関係が重視されがちになるが，中盤や終盤の，より戦術的な，勝敗に直結する局面において良い手が指せるかを重視したほうが強いプレイヤとなる可能性があると考えられる．

これらの問題は本質的には「強い」パラメータを得ることが目的であるのに対して，それを「人間の指し手との一致率」といった基準で近似していることに起因しているといえる．強化学習等人間の棋譜によらない手法が成功しているゲームもあるものの [19]，現在までのところ，将棋等の複雑なゲームでは人間の棋譜を教師とした学習を明らかに上回る成果は得られていない．本論文では，人間の棋譜を教師とした学習において教師データに重要度を導入し，対局による重要度の学習と重要度に基づく重み付き学習を組み合わせた学習手法を提案する．

3.2 重要度を導入した学習

前節で述べたとおり，従来の棋譜を教師とした機械学習では，教師データとの指し手の一致率を向上させることを目的としており，プログラムの強さとの関係が不明である．提案手法では「強い」パラメータを得るため，教師データに重要度を導入し，強さに寄与する教師データを重視した学習を行うことを目的とする．

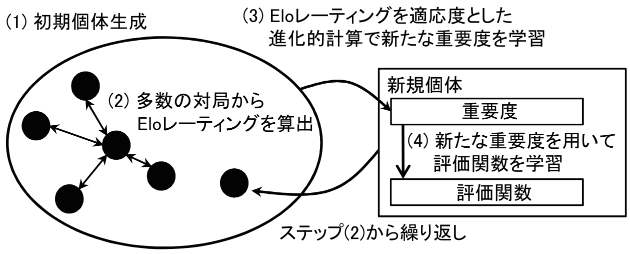

図 3 提案手法の処理（評価関数を学習する場合）
Fig. 3 Procedure of proposed method.

3.2.1 提案手法

提案手法の具体的な処理を図 **3** に示す．図 3 では，評価関数を学習対象としているが，実現確率，playout の方策も同様に学習できる．提案手法では図 3 に示すように，重要度の学習と，棋譜を教師としたパラメータ学習を繰り返し行うことで，「強い」個体を学習する．ここで，それぞれの個体は (1) 重要度，(2) Elo レーティング（適応度），(3) 重要度に従って学習された評価関数，実現確率等のパラメータを持つ．重要度は，評価関数，実現確率等のパラメータの学習のために用いられ，学習データがどの程度強いプログラムの学習に寄与したかを示す．Elo レーティングは，学習された，評価関数，実現確率等を用いた個体間の対局から算出され，重要度を学習する進化的計算の中では適応度の役割を果たす．

図 3 のステップ (1) では，重要度を正規乱数 $\mathcal{N}(1.0, \sigma^2)$ に従って生成し，その重要度により教師データの各局面に重み付けしたパラメータ学習を行う．この操作を繰り返し，N 個の初期個体を生成する．本論文では実験的に，$\sigma^2 = 0.1$，$N = 50$ とした．重要度は局面単位に持たせても良いし，棋譜単位等意味を持ったまとまりごとに持たせても良い．局面単位に重要度を持たせる場合，個別の学習局面それぞれの重要度を算出できるため，悪手を 1 手だけ省くといったような詳細な重要度の割当てが可能となる反面，求めるパラメータの数が膨大となり，多くの計算資源，計算時間を要するという欠点が存在する．そのため，本論文では，重要度を進行度に応じて割り当てる手法，および棋譜ごとに割り当てる手法の 2 種類を検討する．ここで，進行度はゲームの進み具合を意味し，その局面がどの程度終局に近いかを表す．

ステップ (2) では，ステップ (1) で生成された個体どうしで対局を行い，その対局結果から，個体の相対的な強さを Elo レーティングとして算出する．

ステップ (3) では，ステップ (2) で算出された Elo レーティングを適応度とした進化的計算によって，新たな重要度を算出する．具体的な重要度の学習方法については後述する．提案手法では，Elo レーティングを適応度とするこ

とによって，「強い」プログラムを学習するためには，どのような教師データを重視して学習すればよいかを，重要度として自動的に獲得する．

ステップ (4) では，ステップ (3) で学習された重要度に従って，評価関数，実現確率，playout の方策の学習を行う．

以上のステップ (2) からステップ (4) の処理を繰り返すことによって，棋譜を教師とした学習において，「強い」パラメータの学習に寄与する教師データを重視した学習を実現する．

3.2.2 重要度の更新方法

提案手法では，図 3 のステップ (3) の重要度の学習手法として，分布推定アルゴリズム（EDA: Estimation of Distribution Algorithm）を用いる．通常の遺伝的アルゴリズムが直接個体を進化させるのに対し，分布推定アルゴリズムでは，個体の生成確率 h^t を進化させ，h^t に従って，次世代の複数の個体 X_i^t（t は世代数，i は個体番号）を生成する点が特徴である．本論文の手法では，生成確率の更新は，PBILc [20] と同様の方式に従って行う．PBILc では，個体の生成確率 h^t として，正規乱数 $\mathcal{N}(X, \sigma^2)$ を用いる．X，σ は以下の式により更新する．

$$X^{t+1} = (1-\alpha)X^t + \alpha(X^t_{best,1} + X^t_{best,2} - X^t_{worst}) \quad (5)$$

$$\sigma^{t+1} = (1-\alpha)\sigma^t + \alpha\sqrt{\frac{\sum_{j=1}^{K}(X^j - \overline{X})^2}{K}} \quad (6)$$

ここで，$X^t_{best,1}$，$X^t_{best,2}$，X^t_{worst} は t 世代目における，最良の個体，2 番目に良い個体，および最も悪い個体のパラメータ（重要度のベクトル）を意味する．個体の生成確率の平均値 X^t を，式 (5) によって更新することで，徐々に良さそうなパラメータ付近を探索することになる．また，生成確率の σ の値を，上位 K 個の最良個体のパラメータの値の標準偏差を基に更新する．このように更新される生成確率 h^t を用いて次世代の個体を生成することで，よさそうなパタメータの値の付近かつ，分散が大きい（不確定な）部分を探索する手法となっている．本論文における学習では，文献 [20] を参考に，$K = 10$，$\alpha = 0.01$ とした．

なお，文献 [20] では，生成確率 $\mathcal{N}(X, \sigma^2)$ のみを保持し，個体は各世代で作成しているが，本手法における学習では，個体を最も悪いものから 1 つずつ交換する方法をとっている．これは，重要度に従った評価関数等の学習に多くの時間を要すること，また，一度に多くの個体を入れ替えると個体間の相対的な強さを意味する Elo レーティングの値が不安定になりやすいためである．

3.2.3 並列化

提案手法において，時間を要する処理は，ステップ (2) の個体間の対局と，ステップ (4) の評価関数，実現確率等のパラメータを学習する部分になる．提案手法のように個体を 1 つずつ入れ替える場合，ステップ (2)，(3) とステップ (4) を並列に行うことで処理を効率化できる．その場合，重要度を学習する計算機ではつねに対局を行い，異なる計算機で行っている個体の生成（重要度に従った評価関数等のパラメータ学習）が終了したときに，最下位の個体と新規個体を入れ替え，新たな重要度を個体生成側の計算機に渡すことになる．

4. 実験

提案手法を，現在ゲームの分野で行われている主要な棋譜を教師とした学習に対して適用し，その効果を検証した．本論文では，実験対象のゲームとして，将棋を題材とした．具体的には，以下の 3 つの学習における提案手法の有効性を検証した．

(1) 評価関数の学習
(2) 実現確率の学習
(3) モンテカルロ木探索における playout の方策の学習

前章で述べたとおり，本論文では重要度の割当て単位として，棋譜単位，進行度単位の 2 通りについて実験を行う．本実験における進行度は，その時点の手数を終局までの手数で割った 32 段階の値とした．

評価関数のパラメータ学習は文献 [1] の手法を用い，学習時の探索は静止探索のみとした．実現確率のパラメータ学習には，LIBLINEAR [21] を各学習局面に重み付けして学習できるよう改変したものを用いた（オプションは L1 正則化ロジスティック回帰「-s 6」とした）．また，playout の方策の学習には文献 [12] の手法を用い，モンテカルロ木探索の将棋プログラムとしては，文献 [22] の手法を適用したものを用いた．

4.1 学習条件

実験環境を**表 1** に示す．個体どうしの対局，および重要度の学習は 15 台のマシンを用い，探索パラメータ（評価関数，実現確率，playout の方策）の学習には 18 台のマシンを用いた．Elo レーティング算出のための対局数は，各個体あたり 2,000 局（初期個体どうしの対局のみ 4,000 局）とした．ただし，対局数が 1,000 局以上で，レーティング算出対象の新規個体の順位が最下位の場合，学習時間を短縮するため，その時点で対局を打ち切っている．また，学習のステップ数（繰返し回数）は，400 回とした．学習に要した時間は，評価関数の学習では約 12 日，実現確率の

表 1 実験環境
Table 1 Experimental environment.

用途	CPU	メモリ	台数
対局（重要度学習）	Core2 Quad Q9650	8 GB	15
パラメータ学習	Core i7-3930K	16 GB	7
	Core i7-990X	12 GB	1
	Xeon E5506 × 2	24 GB	6
	Opteron 6134 × 2	16 GB	4

© 2014 Information Processing Society of Japan

学習，playout の方策の学習では約 3 日となっている．

なお，重要度の学習，および対局実験における探索ノード数は 1 手 10 万ノードとした．学習では，評価関数，指し手の特徴として以下に示す項目を用いた．本論文の実験では，学習に用いる特徴自体には特別な工夫は行わず，将棋プログラムにおいて一般的に用いられる特徴を用いた．

(1) 駒割り
(2) 自玉，敵玉との位置関係
(3) 利きの関係に基づく駒の位置関係のパターン [23]
(4) 王手
(5) 駒を取る手，リキャプチャ
(6) 成る手
(7) あたりをかける手
(8) 逃げる手
(9) 持ち駒を打つ手
(10) 玉の移動

このうち，(4)～(10) は指し手の特徴となるため，評価関数の学習では (1)～(3) の特徴のみを用いている．

(2) 自玉，敵玉との位置関係は，玉の位置に応じて，駒の種類ごとに位置による点数をテーブル化したものである．この，玉と駒の種類による位置関係を，自玉，敵玉についてそれぞれ特徴としている．

(3) 利きの関係に基づく駒の位置関係のパターンは，利きの重なりがある 2 駒以上の駒の位置関係のパターンを意味する．具体的には，あるマスに着目したとき，そのマスに利きのあるすべての駒の位置関係を特徴として利用する．利きの関係に基づく駒の位置関係の例を**図 4** で示す．

図 4 において，たとえば 2 四の地点に着目すれば（▲ 2 五歩，▲ 1 五銀，▲ 6 八角，▲ 2 八飛，△ 2 三歩，△ 3 三銀，△ 4 二角）というパターンが，7 七の地点に着目すれば（▲ 8 九桂，▲ 7 七銀，▲ 6 七金，▲ 7 八金，▲ 6 八角，▲ 8 八玉）というパターンが抽出される．将棋の学習に用いられる特徴は，一般的に 2 駒間，3 駒間の位置関係が用いられるが，この特徴では，多くの数の駒の位置関係を表現できる．また，利きの情報に基づいているため，意味のあるパターンを生成しやすく，利き情報を持つデータ構造を使用しているプログラムでは，高速にパターンを生成することも可能である．

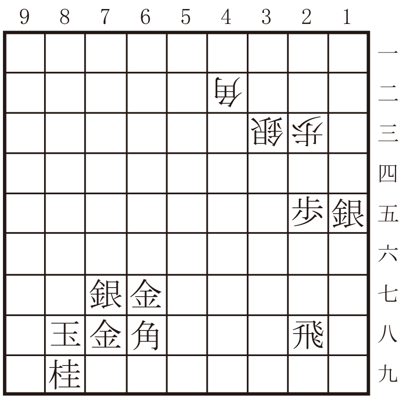

図 4　利きの関係に基づく位置関係のパターンの例
Fig. 4　Example of patterns based on effects of pieces.

(7) あたりをかける手は，歩以外の相手の駒にあたりをかける手，(8) 逃げる手は，歩以外の駒にあたりがかかっているときに，駒損しない位置に移動する手を示す．

なお，学習にはプロやアマチュア高段者の棋譜を用いた．学習棋譜数は，評価関数は 10,000 局，実現確率は 5,000 局とした．

4.2　対局のマッチメイク方法

提案手法では，複数の個体どうしの対局結果を基に算出した Elo レーティングを強さの指標として用いる．なるべく少ない対局数で正確な個体の強さを算出するために，マッチメイクの方法は重要である．本実験では，囲碁や将棋のオンライン対局サーバ [24], [25] で良く用いられている手法を参考に，どのようなマッチメイク方法が適しているかを検討した．

図 5，**図 6** は，強さが既知の 50 プレイヤで繰り返し対局を行ったときの，理想的な順位との誤差の平均を示したものである．本実験では，1,000 回のマッチメイクを行う実験を 10 回行い，各プレイヤの理想的な順位との誤差の平均値を求めている．なお，本実験では，1 回のマッチメイクにつき，同一局面から先後を入れ替えた 2 局の対局をセットで行い，計 2,000 局の対局を行っている．

図 5 は，同一プログラムでノード数を変更したプレイヤ間での実験，図 6 は，評価関数学習時におけるすべての初期個体どうしで対局を行った実験の結果である．

図 5 の実験では，プレイヤ $n (0 \leq n \leq 49)$ の探索ノード数を $1{,}000 + 200 \times n$ とした．この条件では，同一の思考を持つプログラムどうしの対局において，探索ノード数が多いほど強いことは自明であるため，相性が存在しない理想的な順位を持つプログラム間で対局を行う環境での実験となっている．図 6 の実験では，すべての個体どうしで事前に 1,000 局ずつ総当りの対局を行い，その結果を理想的な順位としている．総当りの対局は非常に時間がかかるため，本実験では探索ノード数は 10,000 とした．図 6 は相性や同程度の強さのプログラムが複数存在する，より実際の状況に近い環境でのマッチメイクの妥当性を検証することを目的とした実験となっている．

実験では，次の 4 通りのマッチメイクの方法を実験した．

- 完全にランダムなマッチメイク（random）
- 正規乱数により順位の近いプレイヤ同士を優先的に対局させる方策（ranking）
- 勝ったプレイヤ同士，負けたプレイヤ同士をランダムに対局させる方策（result）
- プレイヤ同士，負けたプレイヤ同士で順位の近いものを優先して対局させ，さらに 10 回に 1 回完全にラン

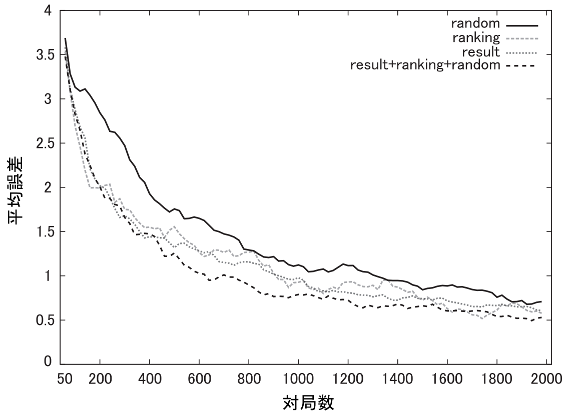

図 5　マッチメイク手法による理想的な順位との誤差（同一プログラムで探索ノード数を変更したプログラム間の対局）
Fig. 5　Errors of rankings (matches among the programs with different search nodes).

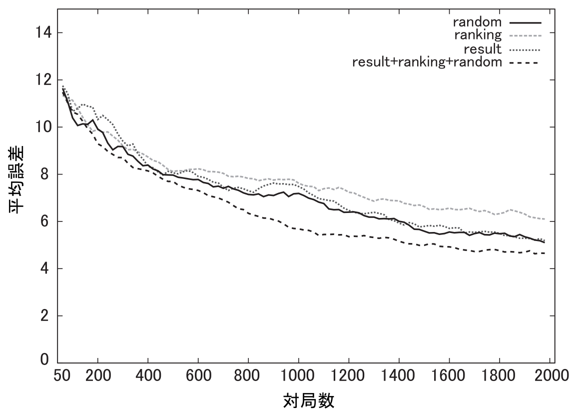

図 6　マッチメイク手法による理想的な順位との誤差（異なるパラメータの評価関数を持つプログラム間の対局）
Fig. 6　Errors of rankings (matches among the programs with different evaluation functions).

ダムなマッチメイクを混ぜた方策（result + ranking + random）

4 つ目の方策で，ランダムなマッチメイクを混ぜている意図は，順位の近いプレイヤで勝ちどうし，負けどうしで対局を組むと，同じプレイヤでの対局のループが生じることがあるため，これを防ぐためである．

実験の結果，図 5 の理想的なプレイヤ同士の対局では，ranking，result，result+ranking+random のいずれの方策もランダムなマッチメイクと比較して良い結果を得た．一方で，図 6 の実験では，result は random とほぼ同等，ranking は random に劣る結果となった．これは，マッチメイクの工夫がプレイヤ間の相性に影響されやすいことを示しているといえる．特に ranking によるマッチメイクでは，順位の近いプレイヤとの対局回数が多くなるため，局所的には正確な順位を求めることができるものの，相性が存在する場合には全体の中での順位に不適当な偏りが生じることが多くなったと考えられる．

result＋ranking＋random によるマッチメイクは，図 5 および図 6 の実験において，ともに安定して最も良い結果を得た．これは，ranking によって順位の近いプレイヤとの優劣を正確に求めつつ，result，random を組み合わせることによって，局所解に陥らないようなマッチメイクが実現できているためであると考えられる．以降の実験では，個体の Elo レーティングを算出するための対局のマッチメイクには，result＋ranking＋random を用いた．

表 2　従来手法に対する勝率（評価関数）
Table 2　Win ratio against conventional methods (learning of evaluation functions).

重要度の学習単位	提案手法	初期個体（best）
進行度	**0.561**	**0.523**
棋譜	**0.581**	**0.552**

表 3　最良の初期個体に対する勝率（評価関数）
Table 3　Win ratio against the best of initial individuals (learning of evaluation functions).

重要度の学習単位	勝率（提案手法）
進行度	**0.544**
棋譜	**0.531**

4.3　対局実験による提案手法の評価

提案手法をコンピュータ将棋における，(1) 評価関数の学習，(2) 実現確率の学習，(3) モンテカルロ木探索中の playout の方策，に適用した際の有効性を対局実験により評価した．比較対象のプログラム（従来手法）は，教師データに重要度による重み付けをせず学習したものである．評価関数等のパラメータの学習では，予備実験においてそれ以上反復回数を増やしても性能向上が認められなくなるまで十分な学習を行ったものを比較対象とした．対局は定跡で 16 手進めた局面から先後を入れ替えて 1,000 セット，2,000 局を行った．

4.3.1　評価関数の学習における提案手法の評価

表 2，**表 3** に評価関数学習時における，提案手法を用いて学習したプログラムの，従来手法，および最良の初期個体との対局結果を示す．なお，以降の実験結果において，表中の太字の数値は有意水準 5%の二項検定で有意な結果を表す．

実験結果から，提案手法が従来手法と比較し，有意に勝ち越していることが分かる．最良の初期個体の重要度を用いて学習を行った場合にも従来手法を上回っているが，提案手法で学習された個体は，より高い勝率を得ることができている．提案手法によって学習された評価関数を用いたプログラムは，最良の初期個体に対しても有意に勝ち越しており，進化的計算によって，より強い個体を学習するための重要度の学習が行われていると考えられる．

4.3.2　実現確率の学習における提案手法の評価

表 4，**表 5** に実現確率学習時における，提案手法を用い

© 2014 Information Processing Society of Japan

表 4 従来手法に対する勝率（実現確率）

Table 4 Win ratio against conventional methods (learning of realization probability).

重要度の学習単位	提案手法	初期個体（best）
進行度	**0.527**	0.506
棋譜	**0.536**	0.486

表 5 最良の初期個体に対する勝率（実現確率）

Table 5 Win ratio against the best of initial individuals (learning of realization probability).

重要度の学習単位	勝率（提案手法）
進行度	0.514
棋譜	**0.542**

表 6 従来手法に対する勝率（モンテカルロ木探索の playout の方策）

Table 6 Win ratio against conventional methods (learning of playout policies).

重要度の学習単位	提案手法	初期個体（best）
進行度	**0.539**	0.505
棋譜	**0.548**	0.514

表 7 最良の初期個体に対する勝率（モンテカルロ木探索の playout の方策）

Table 7 Win ratio against the best of initial individuals (learning of playout policies).

重要度の学習単位	勝率（提案手法）
進行度	**0.540**
棋譜	**0.544**

て学習したプログラムの，従来手法，および最良の初期個体との対局結果を示す．

実験結果から，実現確率探索においても提案手法が従来手法を有意に上回る結果を得た．ただし，評価関数の学習と比較すると，その効果はやや低い結果となった．これは，評価関数の場合には評価値にわずかでも差があれば選択される指し手に直接影響を及ぼすのに対して，実現確率探索の場合には，確率を探索深さに変換して利用することになるが，その際に予測確率の差が小さい指し手間では探索深さとしては差がつきにくいこと等が原因として考えられる．

4.3.3 モンテカルロ木探索の playout の方策の学習における提案手法の評価

表 6，**表 7** にモンテカルロ木探索の playout の方策の学習時における，提案手法を用いて学習したプログラムの，従来手法，および最良の初期個体との対局結果を示す．

実験結果から，モンテカルロ木探索の playout の方策では，実現確率と比較してやや高い効果を得た．これは，モンテカルロ木探索では，終局までの長手数の playout を行うため，通常の探索よりも指し手の予測が結果に与える影響が大きくなったことが考えられる．なお，モンテカルロ木探索の場合，playout の性質がプログラムの強さに与える影響が明らかになっていないという特徴がある．このような，どのようなパラメータが強いプログラムを実現するのか明らかでない課題に対しても，提案手法は効果を発揮すると考えられる．

なお，今回は棋譜単位，進行度単位で重要度を割り当てたが，指し手の予測確率により明確な差が付けられるよう，局面単位で重要度を割り当てることや，正例（棋譜中で指された手）と負例（それ以外の手）で異なる重要度を与えること等により，さらなる性能向上を実現する余地があると考えられる．

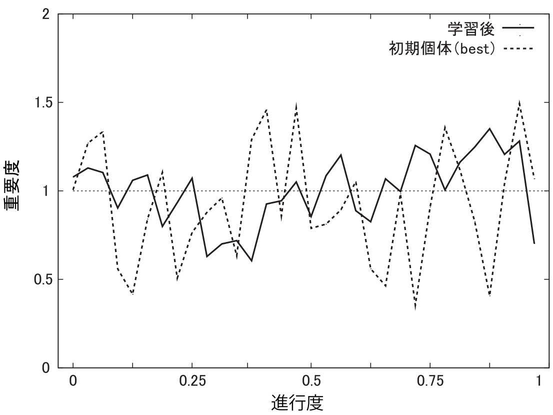

図 7 進行度に応じた教師データの重要度の変化

Fig. 7 Values of weights against progress ratio.

4.4 進行度単位の重要度の分析

図 7 に提案手法を用いて学習した進行度別の重要度（評価関数学習時）を示す．図 7 では，提案手法によって学習された重要度，および参考として初期個体のうち最もレーティングの高かった個体の重要度を示している．なお，重要度の平均はつねに 1.0 になるように正規化している．

図 7 から，提案手法を用いて学習した重要度は，序盤から中盤にかけてやや下がり，終盤で最も高くなっていることが分かる．終盤で重要度が最も高くなった理由としては，終盤では駒の位置関係が直接勝敗に影響しやすく，良い局面，悪い局面がはっきりと分かれるためであると考えられる．文献 [4] では，進行度に応じて終盤ほど正解手とその他の合法手の評価値のマージンが大きくなるように調整しており，終盤の重要度が高くなっている点では，今回の実験結果と傾向が一致しているといえる．

また，今回の実験では，序盤よりも中盤の前半において重要度が低くなる結果となった．これは，序盤は駒組みが明確で評価関数で局面評価がしやすいのに対して，中盤は駒組み（囲い）が崩れることも多く，明確な目指すべき形が定義しにくいことが原因と考えられる．

さらに，最終盤（終局直前の 5〜10 手程度に相当）では，重要度が低くなっていることが分かる．これは最終盤ではすでに勝敗が決しており教師として不適切な局面が存在す

© 2014 Information Processing Society of Japan

表 8　戦術による重要度の違い
Table 8　Differences of weights by tactics.

順位	戦型	戦術		先手側勝率（棋譜数）	重要度の平均
		先手戦術	後手戦術		
1	四間飛車	四間飛車穴熊	銀冠	0.566 (30)	1.121
2	四間飛車	居飛穴模様	藤井システム	0.500 (20)	1.100
3	矢倉	森下システム	△ 9 五歩・8 四歩型	0.500 (96)	1.088
4	矢倉	▲ 4 七銀 3 七桂	金矢倉	0.500 (18)	1.079
5	相掛かり	▲ 2 六飛	△ 5 四歩型	0.542 (24)	1.061
...	...	...	...	...	...
96	三間/四間飛車	居飛車穴熊	本美濃	0.700 (30)	0.951
97	矢倉	引き角	△ 6 二飛戦法	0.474 (19)	0.922
98	三間/四間飛車	左美濃	高美濃	0.611 (18)	0.914
99	中飛車	角交換型	ゴキゲン中飛車	0.519 (27)	0.901
100	角換わり	相腰掛け銀	△ 6 五歩型	0.538 (26)	0.899

ることや，今回の実験で評価関数の学習に用いた探索が静止探索という非常に浅い探索であったことに起因すると考えられる．文献 [1] の学習手法では，探索結果の末端局面を学習に用いるため，学習時に用いる探索深さと局面を正確に評価するために必要な探索深さが大きく異なる局面は，本質的に学習対象に適していないといえる．特に今回の評価関数の学習では，静止探索という非常に浅い探索を用いたため，直接詰みが絡むことが多い最終盤は適切な探索結果を得ることができず，重要度が低くなったと考えられる．

4.5　棋譜単位の重要度の分析

図 7 に，提案手法を用いて棋譜ごとの重要度を学習したときの，戦術による重要度の違い（評価関数学習時）を示す．戦型，戦術の分析には，文献 [26] のソフトを用いた．戦術は，出現数の多いもの上位 100 個について分析を行った．

表 8 から，穴熊や銀冠，矢倉を始めとする，囲いを発展させる戦術の重要度が高くなっていることが分かる．一般的に，穴熊等はコンピュータ将棋でも勝ちやすい戦術といわれており妥当な結果と考えられる．一方で，今回の実験では美濃囲いはやや重要度が低くなる傾向があった．これは美濃囲いで戦うよりも，銀冠，穴熊等の囲いに発展させたほうがより高い勝率を得ることができる可能性が高くなるため，相対的に美濃囲いの重要度を低くする方向に学習が働いたためと考えられる．表 8 から，特に対穴熊では，通常の美濃囲いを選択するよりも，銀冠に発展させる，藤井システムによって穴熊を阻止する等の戦術をとる重要度が高くなっていることが分かる．

表 9 に棋戦による重要度の違いを示す．棋戦は，棋譜数が 10 局以上あるものを分析対象とした．表 9 から，棋戦単位で分析した場合，アマや女流の棋戦の重要度はやや低くなる傾向があったが，プロ棋士の棋戦の間ではほぼ差は見られなかった．実験結果から，今回の実験で用いた評価関数の特徴，および学習方法では，女流やアマ等明らかに実力の差がある棋譜は学習に悪影響を及ぼす可能性があるものの，一定以上の強さが保証されている場合には，その中での棋譜の質の差は重要度に大きな影響を与えてはいないと考えられる．

表 9　棋戦による重要度の違い
Table 9　Differences of weights by tournament.

順位	棋戦	棋譜数	重要度の平均
1	名将戦	41	1.049
2	近将カップ	64	1.043
3	天王戦	85	1.025
4	十段戦	118	1.020
5	達人戦	27	1.020
...	...	...	...
31	全日プロ	400	0.989
32	朝日アマ名人戦	20	0.974
33	早指し戦	299	0.972
34	女流名人戦	62	0.946
35	全日アマ名人戦	10	0.917

その他，対局年，持ち時間，戦型（戦術で分類しない），終局までの手数での分析も行ったが，いずれも明確な傾向は見られなかった．以上の結果から，今回の実験で学習された重要度は，(1) コンピュータが勝ちやすいといわれている戦術が重視されている，(2) 教師としての質（強さ）に大きく差があるものは除外する方向に調整されている，といった傾向が読み取れる結果となった．

4.6　実際の対局時における戦術選択の違い

提案手法では，棋譜単位の重要度を用いた場合，より囲いを発展させる戦術の重要度が高くなる傾向があった．本実験では，この重要度の違いが，実際の対局においてどのように反映されているかを検証した．

表 10 に，4.3.1 項の棋譜単位の重要度を用いた場合の

表 10 実際の対局における戦術選択の違い
Table 10 Differences of tactics selected in games.

戦術	各戦術の選択回数	
	従来手法	提案手法
本美濃	131	77
高美濃	45	54
銀冠	61	175
流れ矢倉	28	15
金矢倉	33	45
居飛車穴熊	210	215
四間飛車穴熊	38	40
三間飛車穴熊	65	39

2,000 局の対局実験（評価関数学習時）において，提案手法，従来手法のプログラムが選択した主要な戦術の違いを示す．対局では 16 手まで定跡で進めているため，居飛車，振り飛車といった大まかな戦型は定跡により決定されているが，最終的な囲いや攻め方の選択は多くの場合，定跡外の通常の思考部で決定することになる．表 10 から，今回の実験結果では，実際の対局時に選択された戦術として，美濃囲い（本美濃）と銀冠において，提案手法と従来手法に特に大きな差が表れていることが分かる．提案手法では，美濃囲いのまま戦うよりも，高美濃，銀冠といった囲いに発展させている傾向が顕著に表れている．その他にも提案手法では，流れ矢倉よりも，より固さを重視した金矢倉が選択される割合が高くなるといった傾向が表れており，学習時の重要度の違いが，実際の対局時の戦術にも影響を与えていることが確認できた．

なお，穴熊については，実際の対局時には，提案手法で選択した割合は従来手法と比較して大きな差は見られなかった．これは，従来手法においても，穴熊は十分に高い価値を獲得しており，提案手法と差が生じにくかったこと等が理由として考えられる．

5. 今後の課題

本論文における実験では，教師データに重要度を導入し，対局による重要度の学習と重要度に基づく重み付き学習を繰り返すことによって，性能向上が実現できることを示した．

今後の課題としては，以下の 2 点があげられる．

(1) 重要度の学習手法，重要度を割り当てる単位の検討
(2) 他のゲームへの適用

(1) について，本提案手法では，重要度の学習には分布推定アルゴリズムを用いた．本論文では，分布推定アルゴリズムの確率分布の更新には，PBILc と同様の更新式を用いたが，その他の手法 [27], [28], [29] を用いることによって，より精度の高い重要度が獲得できる可能性がある．また，本論文では，実験環境の都合上，初期個体 50，学習ステップ 400 という条件で学習を行ったが，これらの条件を増やすことによる性能向上も期待できる．

さらに，重要度を割り当てる単位にも改善の余地があると考えられる．今回の実験では，進行度単位，棋譜単位に重要度を割り当てたが，局面単位，戦術単位，先手と後手で異なる重要度を用いる等の方法が考えられる．局面単位での重要度の学習が可能となれば，提案手法の枠組みによって，棋譜中に存在するミス，悪手といった局面を自動的に除外することも可能になると考えられる．

(2) について，提案手法は「強さ」に基づいた学習であるため，将棋以外の多くのゲームに対して有効であることが期待できる．具体的には，近年研究がさかんな囲碁等への適用については，検討する価値が高いと考えられる．

6. おわりに

本論文では，教師データに重要度を導入し，対局による重要度の学習と重要度に基づいたパラメータの重み付き学習を組み合わせた学習手法を提案した．重要度は，個体の強さを適応度とした進化的計算により学習した．強さを表す指標としては，大規模な計算環境を用いて，個体間で対局を行い，その対局結果から算出した Elo レーティングを用いた．

提案手法をコンピュータ将棋を対象として，(1) 評価関数の学習，(2) 実現確率の学習，(3) モンテカルロ木探索の playout の方策の学習，に適用した結果，従来手法を有意に上回ることに成功し，その有効性を示した．また，学習された重要度を分析した結果，終盤の重要度が高くなる，穴熊等一般的にコンピュータが勝ちやすいといわれている戦術の重要度が高い傾向となる，といった傾向となっており，提案手法を用いることによってゲームの性質にあった教師データの重要度が算出されていることが確認できた．

現在，ゲームプログラミングの分野では，評価関数の学習，探索深さの調整，モンテカルロ木探索の playout の方策等，多く課題において棋譜を教師とした機械学習が取り入れられており，その重要性は今後も増していくと考えられる．一方で，コンピュータの強さは多くのゲームにおいて確実に人間の強さに迫るものとなっており，今後は単純に人間の指し手を教師とした学習では不十分となる場合も考えられる．本論文の提案手法は，教師データに対して重要度という形で意味付けを行うことで，単純な棋譜を教師とした学習を上回ることを示したものであり，今後このような学習手法はさらに重要になると考えられる．

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] Tesauro, G.: Comparison training of chess evaluation functions, *Machines that learn to play games*, pp.117–130, Nova Science Publishers, Inc. (2001).

© 2014 Information Processing Society of Japan

[3] 保木邦仁：「Bonanza 4.1.3」ソースコード，コンピュータ将棋の進歩 6，pp.1–22, 共立出版 (2012).

[4] 鶴岡慶雅：「激指」の最近の改良について—コンピュータ将棋と機械学習，コンピュータ将棋の進歩 6，pp.71–83, 共立出版 (2012).

[5] 金子知適：「GPS 将棋」の評価関数とコンピュータ将棋による棋譜の検討，コンピュータ将棋の進歩 6，pp.25–44, 共立出版 (2012).

[6] Tsuruoka, Y., Yokoyama, D. and Chikayama, T.: Game-Tree Search Algorithm Based On Realization Probability, *ICGA Journal*, Vol.25, No.3, pp.145–152 (2002).

[7] 竹歳正史, 橋本 剛, 梶原羊一郎, 長嶋 淳, 飯田弘之：コンピュータ将棋における実現確率探索の研究，第 7 回ゲーム・プログラミングワークショップ，pp.87–92 (2002).

[8] 橋本 剛, 長嶋 淳, 作田 誠, Uiterwijk, J., 飯田弘之：実現確率探索のゲーム全般への応用 — Lines of Action を題材にして，第 7 回ゲーム・プログラミングワークショップ，pp.81–86 (2002).

[9] 三輪 誠，横山大作，近山 隆：指し手の履歴の抽出に基づくカテゴリの拡張，第 11 回ゲーム・プログラミングワークショップ，pp.64–69 (2006).

[10] 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).

[11] 鶴岡慶雅：最近のコンピュータ将棋の技術背景と激指，情報処理，Vol.49, No.8, pp.982–986 (2008).

[12] Coulom, R.: Computing Elo Ratings of Move Patterns in the Game of Go, *ICGA Journal*, Vol.30, No.4, pp.198–208 (2007).

[13] 金子知適：コンピュータ将棋の評価関数と棋譜を教師とした機械学習（〈レクチャーシリーズ〉コンピュータ将棋の技術〔第 5 回〕），人工知能学会誌，Vol.27, No.1, pp.75–82 (2012).

[14] 川上裕生，浦 晃，三輪 誠，鶴岡慶雅，近山 隆：将棋の評価関数の学習に有用な局面の自動選択，第 18 回ゲーム・プログラミングワークショップ，pp.66–72 (2013).

[15] 竹内 章：コンピュータ将棋「習甦」開発記，コンピュータ将棋協会誌，Vol.22, pp.21–26 (2010).

[16] Beal, D.F. and Smith, M.C.: First Results from Using Temporal Difference Learning in Shogi, *Proc. 1st International Conference on Computers and Games*, pp.113–125 (1999).

[17] 鈴木 彰，柴原一友，但馬康宏，小谷善行：条件付き確率 PIPE による将棋の評価関数の生成，第 10 回ゲーム・プログラミングワークショップ，pp.56–62 (2005).

[18] 矢野友貴, 柴田剛志, 横山大作, 田浦健次朗, 近山 隆：GA と TD(λ) 学習の組み合わせによるゲーム局面評価パラメータの調整，情報処理学会研究報告．GI，［ゲーム情報学］，Vol.2009, No.27, pp.63–70 (2009).

[19] Tesauro, G.: Programming backgammon using self-teaching neural nets, *Artificial Intelligence*, Vol.134, pp.181–199 (2002).

[20] Sebag, M. and Ducoulombier, A.: Extending Population-Based Incremental Learning to Continuous Search Spaces, *Proc. 5th International Conference on Parallel Problem Solving from Nature*, pp.418–427 (1998).

[21] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. and Lin, C.-J.: LIBLINEAR: A Library for Large Linear Classification, *Journal of Machine Learning Research*, Vol.9, pp.1871–1874 (2008).

[22] 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, Vol.50, No.11, pp.2740–2751 (2009).

[23] 佐藤佳州，高橋大介：特徴の生成を組み合わせた機械学習，第 16 回ゲーム・プログラミングワークショップ，pp.135–142 (2011).

[24] available from ⟨http://cgos.boardspace.net/⟩

[25] available from ⟨http://wdoor.c.u-tokyo.ac.jp/shogi/floodgate.html⟩

[26] available from ⟨http://www.geocities.jp/saltedeggplant/⟩

[27] 佐久間淳，小林重信：確率分布推定に基づく実数値 GA の新展開（〈特集〉遺伝的アルゴリズムの発展），人工知能学会誌，Vol.18, No.5, pp.479–486 (2003).

[28] Larranaga, P., Etxeberria, R., Lozano, J.A. and Pena, J.: Optimization by learning and simulation of Bayesian and Gaussian networks, Technical Report, University of the Basque Country (1999). Technical Report EHUKZAA-IK-4/99.

[29] Bosman, P.A. and Thierens, D.: Expanding From Discrete To Continuous Estimation of Distribution Algorithms: The IDEA, *Parallel Problem Solving From Nature – PPSN VI*, Springer, pp.767–776 (2000).

佐藤 佳州 （正会員）

1985 年生．2008 年筑波大学第三学群情報学類卒業．2010 年同大学大学院システム情報工学研究科博士前期課程修了．現在，パナソニック株式会社先端技術研究所．2014 年筑波大学大学院システム情報工学研究科博士後期課程修了．博士（工学）．専門は人工知能，特にゲーム情報学．現在は医療情報処理に関する研究に従事．2013 年度情報処理学会山下記念研究賞受賞．

高橋 大介 （正会員）

1970 年生．1991 年呉工業高等専門学校電気工学科卒業．1993 年豊橋技術科学大学工学部情報工学課程卒業．1995 年同大学大学院工学研究科情報工学専攻修士課程修了．1997 年東京大学大学院理学系研究科情報科学専攻博士課程中退．同年同大学大型計算機センター助手．1999 年同大学情報基盤センター助手．2000 年埼玉大学大学院理工学研究科助手．2001 年筑波大学電子・情報工学系講師．2004 年同大学大学院システム情報工学研究科講師，2006 年同助教授，2007 年同准教授．2011 年同大学システム情報系准教授，2012 年同教授．博士（理学）．並列数値計算アルゴリズムに関する研究に従事．1998 年度情報処理学会山下記念研究賞，1998 年度，2003 年度情報処理学会論文賞各受賞．日本応用数理学会，ACM，IEEE，SIAM 各会員．

© 2014 Information Processing Society of Japan