勝率に基づく評価関数の評価と最適化

竹内聖悟[†] 林芳樹[††] 金子知適[†]
山口和紀[†] 川合慧[†††]

勝率と評価値の関係に基づいた問題点の発見手法を提案し，その有効性を示す．強いプログラムの作成には良い評価関数が不可欠だが，既存の評価関数の問題点の発見や評価値の適切な調整には，対戦などの試行錯誤が必要であり困難であった．本研究ではまず，問題点の発見手法として，評価関数が与える評価値に対する勝率に着目しそのグラフを描くことを提案し，評価関数に欠陥が存在する場合には複数の線として明確に図示されることを示す．さらに，欠陥を解決した評価関数では評価値に対する勝率のグラフが条件によらず一本化されるため，グラフにより評価関数の改善を確認できることを示す．実際に将棋，チェス，オセロについて評価関数の欠陥を図示ができることを示し，将棋においてはその改善がグラフで確認できることを示す．さらに自己対戦から実力が改善されていることを確認した．

Evaluation and Adjustment of Evaluation Functions Based on Relation between Static Values and Win Ratios

Shogo Takeuchi,[†] Yoshiki Hayashi,[††] Tomoyuki Kaneko,[†]
Kazunori Yamaguchi[†] and Satoru Kawai[†††]

Strong game programs need accurate evaluation functions that predict win ratio for a given state. However, it is not easy to construct such functions. We propose to plot evaluation values and win ratio for some sets of states, with an existing evaluation function. If multiple curves appear, it shows that the evaluation function does not work well for states in a certain condition. Improvement is accomplished if new evaluation curves fit into one curve. We applied this method to Shogi, Chess, and Othello, and showed that by plotting values and win ratios we can visualize the faults of evaluation functions. And our experiments with Shogi showed that we can confirm improvement of evaluation function by plotting values and win ratios. Moreover, significant improvement on strength is confirmed by self-plays.

1. はじめに

評価関数は与えられた局面の勝ちやすさを局面の特徴から評価するもので，強いゲームプログラムを作るためには良い評価関数が必要となる．

これまで評価関数の調整は多くが人の手で行われており，囲碁や将棋のように複雑な形勢判断が必要とされるゲームにおいて自動調整が明確に成功を収めた研究は少ない．一方，手作業での調整には，どの特徴をどの程度に評価すべきか，どこに問題点があるかの判断が難しい．また，調整結果を確認するためには，調整前と調整後のプログラムによる問題集の正答数の比較，自己対戦などの時間がかかる手法が通常必要となる．このため，問題点を解決できたかの確認が難しい．

本稿では，評価関数の問題点をグラフで示す手法を提案する．この手法では，評価項目やその重みの適切さの評価が可能であり，新たに追加した評価項目や調整した重みにより問題点が解決したかの確認も容易である．さらにこの手法は自己対戦などよりも計算時間が短いという利点もある．

本手法のアイデアは評価値と勝率の関係を活用することである．評価関数に問題がある場合，局面の勝ちやすさを適切に評価できない．ここで勝ちやすさを勝率で表現できるとすると，同じ評価値でも同じ勝率にならない問題として観測される．つまり，評価関数の評価に問題がある局面では，評価値と勝率の関係が通常と異なっている．

そこで，本稿では評価値と勝率の関係を利用して，

† 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
†† グーグル株式会社
Google Japan Inc.
††† 放送大学
The Univeisity of the Air

勝率と評価値のグラフから評価関数の問題点や評価項目の重みの不適切さを発見することを提案する．さらに棋譜と勝敗をもとに最小二乗法または最尤法を用いて重みを調整することを提案する．

実際に，将棋やチェス，オセロについて，本手法によって評価関数の問題点を発見できることを示した．また，将棋において，玉の危険度差という評価項目に対して自動調整を行い，本手法により問題点の改善を確認した．さらに自己対戦により調整後の評価関数を利用したプログラムが調整前の評価関数を利用するプログラムに対して有意に勝ち越すことを確認した．

以下に，本稿の構成を述べる．2 章では関連研究を，3 章では本手法を具体的に説明し，4 章で評価関数の問題点の可視化を行い，5 章で評価関数の改善とその評価を行い，6 章で結論を述べる．

2. 関 連 研 究

これまでの研究では本稿で提案する手法のような評価関数の性質を図示する例はなかった．ここでは，ゲームプログラミングにおける評価関数の調整の関連研究について述べる．

評価項目の重みを自動的に調整する代表的な研究には Buro のものがある．Buro はオセロにおいて，終局時の石の数の差を理想の評価値として，各局面に対する評価がそれを実現するように重みを調整する手法を提案した[1]．これにより得られた終盤用の評価関数を利用し，中盤と序盤の評価関数を探索結果をもとに調整した．しかし将棋やチェスでは，オセロにおける終局時の石の数のような明確な勝敗の差を表す指標がないため，このような直接的な調整で成功した例はない．

また，評価値の教師例を必要としない手法として TD 法がある．この手法はバックギャモンへと応用され，自己対戦から評価関数の重みの調整を行い，チャンピオンレベルのプログラムを作成するのに成功した[2]．チェスでは，チェスプログラム Knight Cap が，Min-Max 探索向けに TD 法に改良を加えた TD Leaf を用い，人間との対戦から評価関数の重みの学習を行い，マスタクラスの強さとなった[3]．しかし，この手法はトップレベルのプログラムでは使われていない．将棋においても TD 法を用いて駒の価値を学習した例がある[4]．しかし，他の複雑な評価項目の学習に成功した例はなく，トップレベルのプログラムでの利用例もない．

将棋プログラムの評価関数では，序盤は駒の損得，終盤では玉の危険度が重要といわれている．終盤では玉との相対位置を考え，近付くほど評価を高くするという手法は多くのプログラムで用いられている[5)~7)]．また，近年では絶対テーブルを利用する手法も用いられている[8]．これらの評価項目の重みは，多くがプログラマの知識によって決定されている．

最近将棋において成功した手法として，将棋プログラムの Bonanza で用いられた手法があげられる．そのアイデアは，探索の結果が棋譜の指し手と一致するように評価項目の重みを調整すること[9] で，それにより得られた評価関数を用いて Bonanza は 2006 年に世界コンピュータ将棋選手権において優勝した[10]．しかし，棋譜の各局面での探索が必要となるため，学習には 3 カ月かかる☆という難点もある．

3. 勝率に基づく評価関数の問題点の可視化

本章では，勝率と評価値の関係から評価関数の問題点を可視化する手法と，評価項目の重みを棋譜から自動調整する手法について説明する．

提案手法を用いると，以下のようにして評価関数の改善が可能となる：(1) 問題があると予想される条件でグラフを描く．グラフが分かれているなら，評価関数に問題があることが分かる．(2) 条件に対応する新しい評価項目を追加し，グラフを描き直す．グラフが 1 本になっているなら改善したと判断できる．評価項目の重みも本章で説明する自動調整により得られる．

3.1 評価関数の問題点の可視化

理想的な評価関数では局面の勝ちやすさを適切に評価できるので，2 つの局面に同じ評価値を与えるならば，両局面の勝率は同じはずである．しかし，問題のある評価関数では，同じ評価値を与えた 2 つの局面の勝率が異なりうる．

ゲームにおける探索では評価値の高い局面を選択するが，その目的は勝率の高い局面を選択することである．X 軸に先手の評価値，Y 軸に先手の勝率を描くとする．局面に関する条件 Cond が成り立つ局面と成り立たない局面についてプロットし，仮に**図 1** のようにグラフが分離したとすると，勝率の低い局面が選ばれうる．たとえば，探索中で図 1 の点 A，B のどちらかを選択する状況では，A の評価値が B よりも高いため A を選択してしまう．しかし，本来は勝率が高い B を選択するべきである．したがって，この評価関数は条件 Cond に関して局面を適切に評価できておらず，問題のある評価関数だと判断できる．

図 2 のグラフは，5 章で実験対象とする「玉の危険度」に差が生じていることを条件として描いた例であ

☆ 保木（招待講演の質疑，GPW 2006）

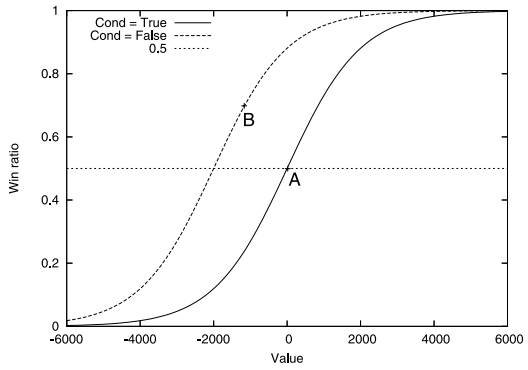


図 1 問題のある評価関数について勝率と評価値をプロットした例
Fig. 1 Example of a bad evaluation function.

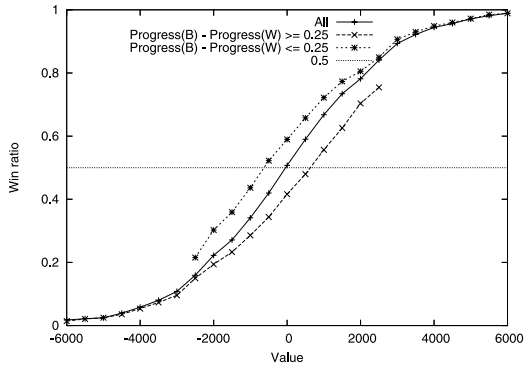


図 2 調整前の GPS 将棋による評価値 – 勝率グラフ（玉の危険度に差がある条件）
Fig. 2 Evaluation curves by GPS Shogi before adjustment (Player's King is in danger).

る．図中の B，W はそれぞれ先手と後手を，Progress は玉の危険度を表している．グラフは複数に分かれており，評価関数には玉の危険度に関して問題があると考えられる．

3.2 自 動 調 整

続いて，評価値と新しい評価項目から勝率を予想するモデルを作り，棋譜から学習を行う方法を説明する．

手法は最小二乗法によるものと最尤法によるロジスティック回帰を用いる．変数 y は局面の勝敗を表し，先手の勝ちなら 1，負けなら 0 と定義した．変数 x_i は局面の評価項目を表し，全体をベクトル X として次のようにまとめた．

$$X = (x_1, ..., x_n)$$

変数 w_i は評価項目 x_i に対する重みを表し，評価項目の種類を n，データサイズを N とする．

以上の変数を利用して，評価関数 $f(X)$ は評価項目の線形モデルとして次式のように表される．

$$f(X) = \sum_{i=1}^{n} w_i x_i$$

3.2.1 最小二乗法（LS）

最も単純な手法として，勝敗を評価関数 $f(X)$ の値から線形に予測し，最小二乗法によりパラメータの調整を行う．目的関数は，勝敗の予想関数 $f(X)$ と実際の勝敗 y との差の二乗和

$$OF_{LS} = \min \sum_{j=1}^{N} (y_j - f(X_j))^2 \qquad (1)$$

として表される．式 (1) で表される目的関数が最小化されたときのパラメータが調整結果である．

3.2.2 最尤法を用いたロジスティック回帰（MLM）

次に，最尤法を用いたロジスティック回帰について説明を行う．勝敗の予想はロジスティック式で表す．

$$g(X) = \frac{1}{(1 + \exp(-f(X)))}$$

この式の定義域は $[-\infty, \infty]$ で値域は $[0, 1]$ となる．次に，上式と実際の勝敗との尤度は，

$$likelihood(X, y) = g(X)^y (1 - g(X))^{(1-y)}$$

となる．勝敗の予想が実際の勝敗を最もうまく説明するためには，全局面での尤度をかけ合わせたものを最大化すればよい．目的関数は

$$OF_{MLM} = \max \prod_{j=1}^{N} g(X_j)^{y_j} (1 - g(X_j))^{(1-y_j)}$$

となる．この目的関数を最大化したときのパラメータを調整結果として得る．対数をとっても大小関係は変化しないので，簡易のために対数をとって計算する．

4. 評価関数の問題点の可視化

前章で提案した評価関数の問題点を可視化する手法の一般性を示すために，将棋，チェス，オセロについて実験を行った．各評価関数について，問題のある条件ではグラフが分かれることを報告する．

4.1 将　　棋

実験で用いた勝率と評価値のデータを棋譜から得た方法を説明する．評価値は棋譜中の各局面における評価値を利用し，勝敗は 10,000 ノードまで詰め将棋探索を行い，詰みを見つけた場合は詰ませる側の勝ち，詰みのない場合は投了した側の負けとして勝敗を決めた．なお，評価値を得る際に静止探索を含めて探索は行わなかった．評価値に対する勝率は，評価値を 500 点ごとに区切り，各区間ごとに勝敗を数え上げて勝率

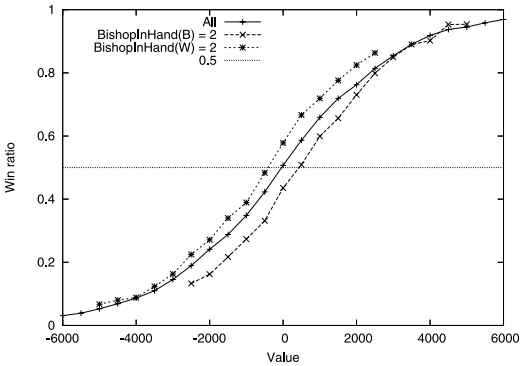


図 3 GPS 将棋による評価値 – 勝率グラフ（持駒に角行を 2 枚）
Fig. 3 Evaluation curves by GPS Shogi (Player has 2 Bishops in Hand).

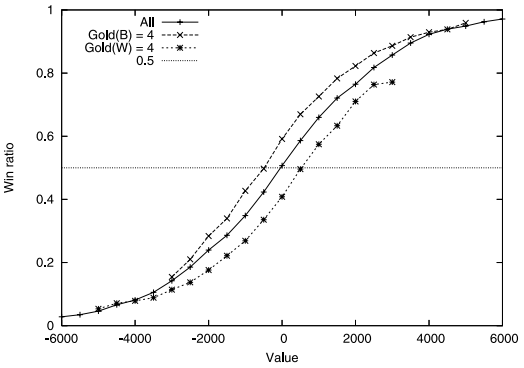


図 4 調整前の GPS 将棋による評価値 – 勝率グラフ（金将を 4 枚）
Fig. 4 Evaluation curves by GPS Shogi before adjustment (Player has 4 Golds).

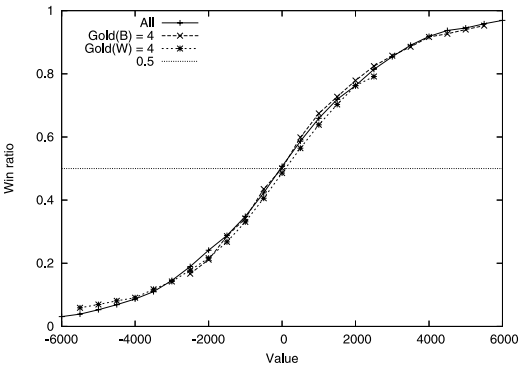


図 5 調整後の GPS 将棋による評価値 – 勝率グラフ（金将を 4 枚）
Fig. 5 Evaluation curves by GPS Shogi after adjustment (Player has 4 Golds).

を得た．なお，グラフでは歩 1 枚が 100 点になるように評価点を換算して表現した．

データの対象は将棋倶楽部 24[11] の棋譜 90,000 局で，区切った評価値区間に 1,000 局面以上あったものを対象とした．

本実験では GPS 将棋☆を使用しており，序盤評価関数は駒の損得と駒の関係[8] からなっている．終盤評価関数については駒と王の位置によって点数を増減するテーブルを利用するなどしている．

将棋では同じ種類の駒を複数枚持駒にしても使いきれないので，2 枚目以降は価値が低いことが経験的に知られている．これに基づき，角行と桂馬について複数枚を持駒にしている場合を調査した．さらに，金は攻めにも守りにも使われる重要な駒であり，金を持たないプレイヤは単なる駒の損得以上に不利といわれる．そこで，金がどちらかのプレイヤに偏っている場合についても調査した．

これらの条件について勝率と評価値のグラフを作成した結果は，**図 3**，**図 4** となり（桂馬についてはグラフを略するが他のグラフと同様な結果が得られた），評価関数がこれらの条件を適切に評価できていないことが確認できた．図 3 での BishopOnHand は各プレイヤが持駒としている角の枚数を表している．図 4 での Gold は各プレイヤが盤上と持駒とで持つ金の枚数を表す．

実際に金の独占に対するボーナスを歩 1.5 枚分としてつけたところ，**図 5** のようにグラフが 1 本となり，評価関数が改善されたと考えられる．

☆ OSL[12]，revision 2602 を利用．

4.2 オ セ ロ

オセロでの実験に利用した Zebra ☆☆ は Gunnar Andersson が開発したトップレベルのオセロプログラムであり，フリーソフトウェアで思考エンジンのソースコードが公開されている．棋譜は GGS（Generic Game Server）☆☆☆ の 100,000 局の棋譜を利用した．

評価関数には，Zebra の評価関数から隅における 5×2 のパターンの評価を除いた評価関数と，オリジナルの評価関数とを利用した．条件は，隅における 5×2 のパターンの評価が 5 より大きいこととした．

上記のような条件でグラフを描いた結果は**図 6**，**図 7** である．図中の Eval（Corner52）は隅の 5×2 のパターンをオリジナルの評価関数で評価した値を表す．

図 6 ではグラフが複数に分かれており，この条件について評価関数に調整の余地があることが分かる．一

☆☆ http://radagast.se/othello
☆☆☆ http://www.cs.ualberta.ca/%7emburo/GGS/game-archive/Othello/

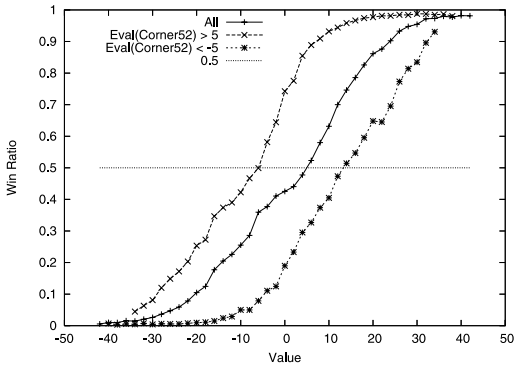


図 6 Corner 5 × 2 を評価しない Zebra による評価値 – 勝率グラフ（Corner 5 × 2）

Fig. 6 Evaluation curves by Zebra without Evaluation of Corner 5 × 2 (Eval(Corner 5 × 2) > 5).

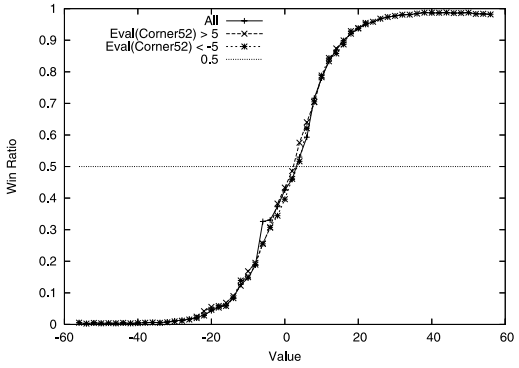


図 7 オリジナルの Zebra による評価値 – 勝率グラフ（Corner 5 × 2）

Fig. 7 Evaluation curves by Zebra with original evaluation function (Eval(Corner 5 × 2) > 5).

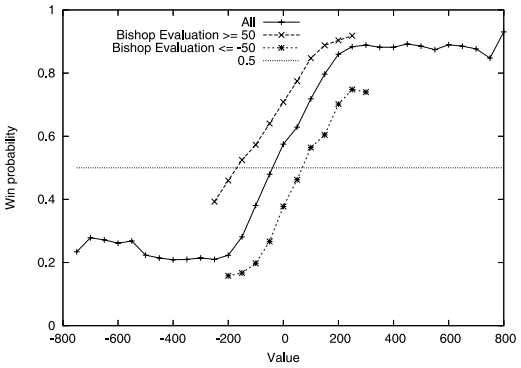


図 8 Crafty にて Bishop Evaluation が 50 以上あるものを分けてプロットした評価値 – 勝率グラフ

Fig. 8 Evaluation curves by Crafty without Bishop Evaluation (Bishop Evaluation ≥ 50).

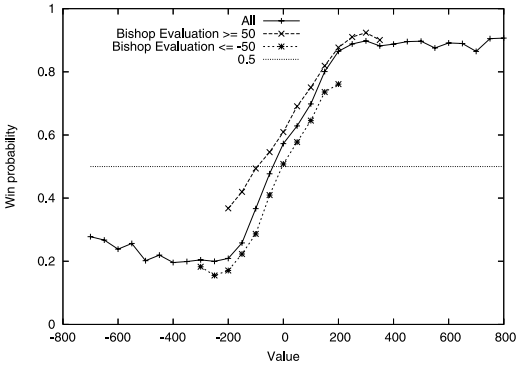


図 9 Crafty にて Bishop Evaluation が 50 以上あるものを分けてプロットした評価値 – 勝率グラフ

Fig. 9 Evaluation curves by Crafty with original evaluation function (Bishop Evaluation ≥ 50).

方，オリジナルの Zebra を利用した場合の結果は図 7 のようにグラフは 1 本となり，この条件での問題はなく，適切な評価が行われていると考えられる．

4.3 チェス

チェスの実験には，Robert Hyatt が開発したチェスプログラム Crafty☆を利用した．実力が高く，ソースコードも公開されている．また，棋譜は ICCF（International Correspondence Chess Federation）☆☆が公開している棋譜 45,955 局分を利用した．

Bishop について評価である Bishop Evaluation を Crafty の評価関数から除いた評価関数とオリジナルの評価関数とを用いた．Bishop Evaluation の評価が 50 以上あるかという条件で，グラフを描いた．

結果は，**図 8**，**図 9** のようになり，Bishop Evaluation を評価しない評価関数ではグラフが複数に分かれ，評価関数を調整する余地があると分かる．オリジナルの評価関数は 1 本に近付いており，適切な評価が行われていることが分かる．図中の Bishop Evaluation はオリジナルの評価関数による先手の Bishop Evaluation の値のことである．

なお静止探索を行わずにグラフを描いた場合，グラフはシグモイドにならなかったが，静止探索を行った結果を利用した場合はシグモイドとなった．このため本実験では評価値を得る際には静止探索を行っている．なお Crafty の評価関数では Pawn の評価値は 100 点となっている．

5. 評価関数の改善と評価

続いて，評価関数の改善が本手法で評価できることを示すために，将棋を題材にさらに深く実験を行った．条件としては，玉の安全度と危険度に着目し，玉の危険度の差を用いた．問題点を改善するため，玉の安全度と危険度の差を評価項目として追加し，自動調整に

☆ ftp://ftp.cis.uab.edu/pub/hyatt
☆☆ http://iccf.com

より評価項目の重みの調整を行った．その結果，グラフから問題点の改善が確認できたとともに，自己対戦でも有意に勝ち越したことを報告する．

5.1 将棋の評価関数

まず実験の対象とした将棋における進行度の評価について説明する．進行度とはゲームの進み具合を表すものである．多くのゲームではゲームの進み具合によって性質が変わるため，それに応じて探索や評価関数を変化させる必要がある．本研究で扱う将棋では，人間にとっては序盤中盤終盤の 3 つに大きく分けられる．序盤では駒得が重視され，終盤では駒得よりも玉の危険度や攻めの速さが重視され，中盤はそれらの中間となっている．多くの将棋プログラムで進行度は用いられており[5),7),13)]，駒の位置や持駒の数，玉の危険度などで表現される．具体的には，

$$\begin{aligned}\text{評価関数} = {} & (1 - \text{進行度}) \times \text{序盤評価値} \\ & + \text{進行度} \times \text{終盤評価値} \end{aligned} \quad (2)$$

などとして，序盤用の駒得重視の評価関数と終盤用の危険度重視の評価関数との配分を決めるという方法が多い．なお，式 (2) では進行度の範囲は 0 から 1 であり，ゲームが進行するにつれて値が大きくなるとした．このように進行度を利用することで，終盤に近付くにつれて終盤用の評価関数の割合が高くなるなど，ゲームの進行に対応した局面の評価が可能となる．

本研究で利用した GPS 将棋の評価関数は式 (2) の形をとっている．本稿では，調整すべきパラメータの少ない枠組みとしてプレイヤ間の進行度差を評価する枠組みを提案し，勝率と評価値の関係に基づいて調整を行う．

GPS 将棋[12)] での全体進行度は，両プレイヤの自玉の 5×3 近傍における敵からの利きの数と持駒の点数の和で計算されている．

5.2 危険度差と安全度差の必要性

次に，新たに導入する措置として，玉の危険度の差を考える．将棋では相手の玉が一方的に危険である場合は勝ちやすく，この指標は評価関数に加えるべき項目と予想できる．プレイヤ別進行度は，自玉の 5×3 近傍における敵からの利きの数と持駒の点数から計算することとした「危険度」と自玉の 5×3 近傍における味方からの利きの数から計算される「安全度」によって構成する．いずれも 0 から 1 の値をとり，1 に近いほど，終盤に近い，自玉が危険，自玉が安全ということを意味する．

玉の危険度を導入する妥当性を調べるために，「先手と後手の玉の危険度に差が 0.25 以上ある」局面について，評価値 500 点ごとにその勝率を調べた結果を図 2 に示す．図中の Progress（B）は先手玉の危険度，Progress（W）は後手玉の危険度を表す．

表 1 訓練例のデータ
Table 1 Data for learning.

データ	局面数	勝ち数	勝率
全体	9,178,020	4,658,800	0.5076
危険度差 ≥ 0.25	747,041	211,193	0.2827
危険度差 ≤ -0.25	778,101	562,826	0.7233

表 2 危険度差と安全度差の調整結果
Table 2 Result of adjustment.

調整	危険度差	安全度差
最小二乗法	−66	94
最尤法	−115	83
手調整	−125	50

グラフを見ると分かるように玉の危険度に差が生じたとき，評価値と勝率の関係が差が生じていないときの関係と異なっており，問題があると考えられる．

5.3 新たな特徴の導入と重みの調整

続いて，玉の危険度差の評価を加えた次のような評価関数を考える．

$$\begin{aligned}\text{評価関数} = {} & (1 - \text{全体進行度}) \times \text{序盤評価値} \\ & + \text{全体進行度} \times (\text{終盤評価値} \\ & + w_1 \times \text{玉の危険度差} \\ & + w_2 \times \text{玉の安全度差}) \end{aligned} \quad (3)$$

この式中の重み（w_1,w_2）を 0 とすれば，式 (2) と一致する．玉の危険度差は主に玉周辺の利きの数からなり，終盤評価値に近いので同じように全体進行度に比例させている．

玉の危険度差の評価は，最小二乗法，最尤法によるロジスティック回帰を行い，手で調整した重みとの比較を行った．なお，これらのデータ処理はオープンソースの統計解析システム R☆を利用して行った．また，訓練例は 4 章で勝率と評価値のグラフを描いたときに利用したデータ，将棋倶楽部 24[11)] の棋譜集 90,000 局の棋譜から得た 9,178,021 局面を対象とした．条件別のデータなどは**表 1** にまとめた．

調整の結果を**表 2** に示す．人手で調整したものと比較すると安全度差の評価が高い傾向があり，最小二乗法は玉の危険度差の評価が他の 2 つに比べずれているが，オーダがずれることはなく，いずれも同じような結果となっている．なお，安全度差の評価が高い理由として，詰め将棋を 10,000 ノード呼んでいることが考えられる．対局した人間が見逃したような詰みを発

☆ The R Project for Statistical Computing, http://www.r-project.org

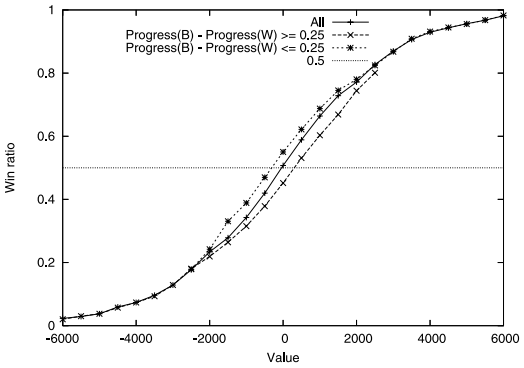


図 10 将棋倶楽部 24 の棋譜による評価値 – 勝率グラフ（最小二乗法による調整後）

Fig. 10 Evaluation curves by GPS Shogi with records in Shogi Club 24 (adjusted by LS).

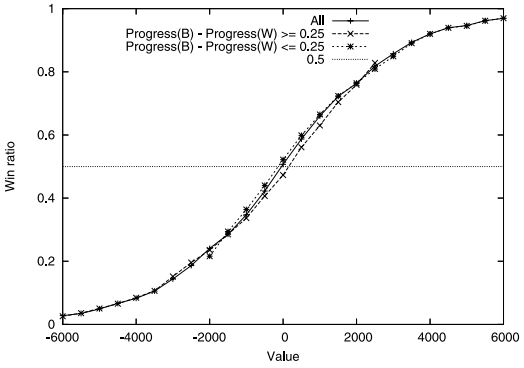


図 11 将棋倶楽部 24 の棋譜による評価値 – 勝率グラフ（最尤法による調整後）

Fig. 11 Evaluation curves by GPS Shogi with records in Shogi Club 24 (adjusted by MLM).

見しても勝ちだと判断するため，守りを重視している可能性がある．

各調整手法ごとの調整後の評価値と勝率のグラフを**図 10**，**図 11**，**図 12** に示す．いずれも調整前の図 2 と比べると，1 本の線に近づいており，評価値と勝率の関係が改善されていることが見て取れる．

5.4 自 己 対 戦

玉の危険度差の評価によるプログラムの棋力向上を確認するため，調整前のプログラムと調整後のプログラムとで 80 局の自己対戦を行った．開始局面は棋譜の 30 手目の局面を利用し，持時間は 25 分とした．結果は**表 3** のようになった．表中の * 印がついた対戦は有意水準 5%の二項検定で有意に勝ち越した対戦である．

調整後のプログラムはすべて，調整前のプログラムに対して有意に勝ち越しており，グラフ上での改善が実際の棋力向上に結び付いていることが確認できた．調整したプログラムどうしの対戦結果を見ても差はほとんどなく，人間の手による調整と同程度の調整を行うことができたといえる．

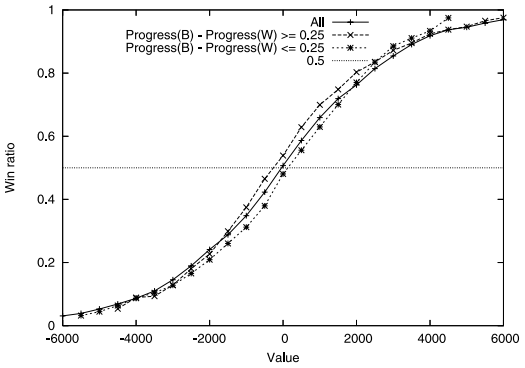


図 12 将棋倶楽部 24 の棋譜による評価値 – 勝率グラフ（手調整後）

Fig. 12 Evaluation curves by GPS Shogi with records in Shogi Club 24 (adjusted by hand).

表 3 自己対戦（勝 - 負 - 引分）

Table 3 Result of self-play.

	調整前	最小二乗法	最尤法
最小二乗法	54 - 24 - 2 *	-	-
最尤法	52 - 28 - 0 *	42 - 36 - 2	-
手調整	59 - 21 - 0 *	38 - 40 - 2	42 - 35 - 3

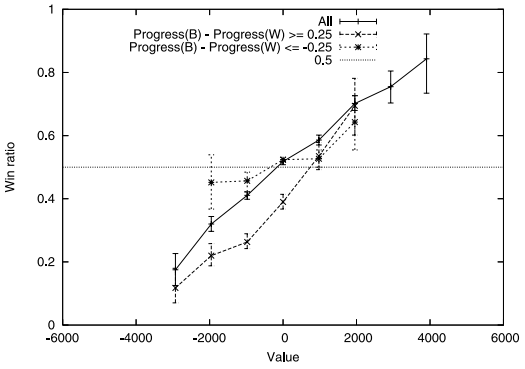


図 13 59 期順位戦の棋譜による評価値 – 勝率グラフ（調整前）

Fig. 13 Evaluation curves by GPS Shogi with records in 59th Junisen (before adjusted).

5.5 プロの棋譜の利用

将棋倶楽部 24 の棋譜はアマチュアによるものだが，玉の危険度差の評価がプロの棋譜でも有効であることを確認するため，プロ棋士の棋譜（59 期順位戦：全 603 局）を対象に 1,000 点ごとに 100 局面以上ある評価値について勝率をプロットした．調整前を**図 13**，表 2 の調整結果を利用したものを**図 14** に示す（グラフの概形はほぼ同じなので最尤法のもののみ示した）．図中のエラーバーは，勝敗の具合が勝率 p の二項分布で表されると仮定したときの 95%信頼区間である．な

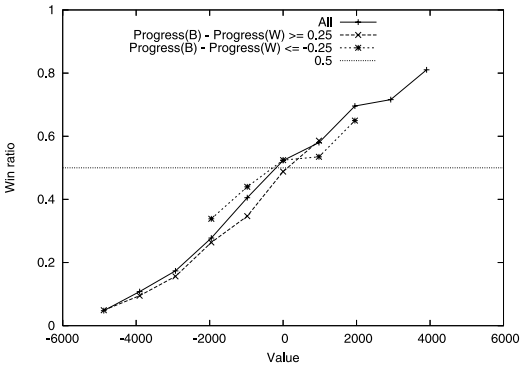


図 14 59 期順位戦の棋譜による評価値 – 勝率グラフ（最尤法による調整結果を利用）

Fig. 14 Evaluation curves by GPS Shogi with records in 59th Junisen (adjusted by MLM).

お，調整後の図では各グラフが接近しエラーバーが重なるため，調整前の図にだけエラーバーをつけた．サンプル数は少ないが，プロの棋譜においてもグラフが 1 本化しており，問題点の改善が確認できる．

6. おわりに

従来，評価関数の問題点を発見するには対戦などの試行錯誤とゲームの知識が必要であり，困難であった．本稿では，問題がある評価関数では，条件によって勝率と評価値の関係が変わってしまうことに着目し，勝率と評価値のグラフを描くことにより評価関数の問題点を簡単に発見する手法を提案した．将棋やチェス，オセロを対象として，本手法を用いて評価関数の問題点を発見できることを示した．

続いて将棋において，評価関数の自動調整を行い，グラフにより問題点の改善を確認した．さらに，調整前後のプログラムによる自己対戦の結果，調整後のプログラムが有意に勝ち越し，総合して本手法の有効性を示すことができた．

今後の課題としては，グラフを描くための条件の発見の自動化があげられる．今回の実験で主な対象とした，玉の危険度に大きな差がある，という条件は将棋に関する人間の知識から得られている．条件の発見を自動化できれば，評価関数の調整がますます容易になり，さらなるゲームプログラムの実力向上が期待できる．

参考文献

1) Buro, M.: Improving heuristic mini-max search by supervised learning, *Artificial Intelligence*, Vol.134, No.1–2, pp.85–99 (2002).
2) Tesauro, G.: Temporal Dfference Learning and TD-Gammon, *Comm. ACM*, Vol.38, No.3, pp.58–68 (1995).
3) Baxter, J., Trigdell, A. and Weaver, L.: KnightCap: A chess program that learns by combining TD(λ) with game-tree search, *Proc. 15th International Conf. on Machine Learning*, pp.28–36, Morgan Kaufmann, San Francisco, CA (1998).
4) Beal, D.F. and Smith, M.C.: First Results from Using Temporal Difference Learning in Shogi., *Computers and Games*, pp.113–125 (1998).
5) 山下 宏：YSS—「コンピュータ将棋の進歩 2」以降の改良，コンピュータ将棋の進歩 5，松原 仁（編），chapter 1, pp.1–32, 共立出版 (2005).
6) 金沢伸一郎：金沢将棋のアルゴリズム，コンピュータ将棋の進歩 3，松原 仁（編），chapter 2, pp.15–26, 共立出版 (2000).
7) 鶴岡慶雅：将棋プログラム「激指」，アマ 4 段を超えるコンピュータ将棋の進歩 4，松原 仁（編），chapter 1, pp.1–17, 共立出版 (2003).
8) 金子知適，田中哲朗，山口和紀，川合 慧：駒の関係を利用した将棋の評価関数，第 8 回ゲームプログラミングワークショップ，pp.14–21 (2003).
9) 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).
10) 瀧沢武信：「全幅探索」と学習による新感覚のコンピュータ将棋の成功とその高速アルゴリズムの及ぼす影響，情報処理，ミニ小特集コンピュータ将棋の新しい動き，Vol.47, No.8, pp.875–881 (2006).
11) 久米 宏：将棋倶楽部 24 万局集，ナイタイ出版 (2002).
12) 田中哲朗，副田俊介，金子知適：高速将棋ライブラリ OpenShogiLib の作成，第 8 回ゲームプログラミングワークショップ (2003).
13) 柿木義一：将棋プログラム K3.0 の思考アルゴリズム，コンピュータ将棋の進歩，松原 仁（編），chapter 1, pp.1–22, 共立出版 (1996).

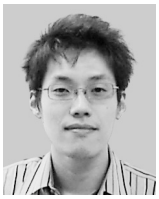
(平成 19 年 1 月 26 日受付)
(平成 19 年 9 月 3 日採録)

竹内 聖悟（学生会員）

2005 年東京大学教養学部卒業．2007 年東京大学大学院総合文化研究科修士課程修了．現在，東京大学大学院総合文化研究科博士課程．人工知能，特に思考ゲームに興味．

林　芳樹

2001年東京大学工学部卒業．2003年東京大学大学院情報理工学系研究科修士課程修了．2005年グーグル株式会社入社．プログラミング言語，思考ゲームに興味を持つ．



金子　知適（正会員）

1997年東京大学教養学部卒業．2002年東京大学大学院総合文化研究科博士課程修了．博士（学術）．2002年東京大学院総合文化研究科助手．思考ゲーム，知識処理に興味を持つ．

山口　和紀（正会員）

1956年生．1979年東京大学理学部数学科卒業．1981年東京大学理学部助手．1985年理学博士（東京大学）．1989年筑波大学電子情報工学系講師．1992年東京大学教養学部助教授．1999年東京大学情報基盤センター教授．2007年東京大学大学院総合文化研究科教授．コンピュータのためのモデリング全般に興味を持つ．ACM会員．

川合　慧（正会員）

昭和42年東京大学理学部物理学科卒業．昭和45年東京大学理学部助手．昭和59年東京大学教育用計算機センター助教授．昭和63年東京大学教養学部教授．平成8年東京大学大学院総合文化研究科教授．平成19年放送大学教授．専門：コンピュータグラフィクス，プログラム言語，情報教育．