自己対局による兄弟局面学習における汎用的制御の有効性

築地　毅[†1]　小谷善行[†2]

自己対局による棋譜を使って兄弟局面学習で 5 五将棋の評価関数を学習できることが分かってきた．特に 5 五将棋では，適切な初期値を与えることで学習に成功することができた．しかし，適切な初期値が不明なゲームでは精度が高い着手を選択できないため，有効性は多く示されていない．そこで適切な評価関数の構成が確立していないブロックスデュオを題材とし，ゲームで汎用的に用いられる「進行度」「読み切り制御」「思考時間」の 3 つの制御を加えることによって柿木による手法に有効性があることを示し，学習結果にどのような影響を与えるかについて議論する．実験の結果，学習用棋譜生成時の着手に「進行度」「読み切り制御」を加えた学習では，加えなかった学習に対し 552 勝 393 敗 55 分と有意に勝ち越し，2 つの制御の有効性が示された．また，学習棋譜生成時の思考時間を 10[s] で行った学習は，1[s] で行った学習に対し 545 勝 403 敗 52 分と有意に勝ち越し，思考時間を長くすることで学習の精度を高められることが示された．

An Effectivity of General Control in Learning Evaluation Function by Comparison of Sibling Nodes by Self-play

TSUKIJI Tsuyoshi[†1] and KOTANI Yoshiyuki[†2]

A learning method by comparison of sibling nodes by self-play for 5x5 shogi starts with default parameter setting given by heuristic, but other games whose general evaluation function is unknown have not been experimented yet. We applied the three general controls, "progress value","endgame search" and "thinking time", to the learning method, to show effectivity on the game Blokus duo. The player of applying progress value and endgame search is ' 552 wins 393 losses, 55 draws' against not applying them. The player of 10sec/move learning is '545 wins 403 losses, 52 draws' against 1sec/move learning.

1. はじめに

強いゲームプログラムを作るためには，精度の高い評価関数を設計することが不可欠である．精度の高い評価関数を設計するためには，ゲームの性質をより正確に表現するための特徴を評価関数に数多く持たせ，かつそれぞれの値に適切な重みを与える必要がある．一般的に各評価項目の総和を評価値とすることが多く，その値によって局面の形勢を判断することになる．特徴を増やすほど評価関数の複雑さは増し，プログラマが手調整で適切な重みを与えることは事実上不可能となる．そのため，機械学習による重みの自動的な調整は大きな課題のひとつと言える．

兄弟局面学習はゲームの評価関数学習に大きく貢献した手法であり，金子ら[12] や保木[15] による将棋プログラムはそれぞれ世界コンピュータ選手権で優勝した．しかし，プロの棋譜が存在しないゲームにおける兄弟局面学習の適用は困難であった．そこで，自己対局による棋譜を用いて兄弟局面学習を行う手法が柿木により提案され[10]，5 五将棋プログラム K55 は UEC 杯 5 五将棋大会において優勝を収めた．しかし，5 五将棋は本将棋プログラムで培った有効な初期値が与えられるゲームであり，K55 は人間の手調整を介さない初期値から学習されたわけではなかった．また，本手法の有効性が詳細な実験によって明示されたわけではなかった．

そこで本稿では，自己対局による兄弟局面学習を有効な評価関数の設計が不明であるブロックスデュオで適用し，ゲームにおける汎用的な制御を学習用棋譜生成時の着手に与えるだけで本手法に有効性があることを示し，さらに学習結果にどのような影響をもたらすか議論する．

2. 関連研究

ゲームプログラムの評価関数を学習によって調節する試みは古くから行われてきた．トップレベルの強さ

†1 東京農工大学大学院 工学府 情報工学専攻
Dept. of Computer and Information Sciences, Graduate School of Technology, Tokyo University of Agriculture and Technology
†2 東京農工大学大学院 工学府
Dept. of Computer and Information Sciences, Tokyo University of Agriculture and Technology

を実現した研究としては，Tesauro によるバックギャモンの評価関数を TD 法によって調節した研究[6] や，Buro によるオセロの評価関数を最小二乗法によって調節した研究[3] などが挙げられる．TD 法によって学習した例としては他に，Baxter らによるチェスの評価関数の学習[1] や，Beal らによる将棋の評価関数の学習[2] などが挙げられる．このような TD 法などに代表される，着手する前と着手した後の局面を比較するような手法は，親子局面学習として分類されている．将棋などの複雑なゲームにおいても親子局面学習による評価関数の値の調整は多く挑戦されてきたが，充分な成果をあげた報告はなされていない．

一方，棋譜で指された着手とそれ以外の可能手を比較するような手法は，兄弟局面学習として分類されている．兄弟局面学習は，筆者の調査によると Tesauro によるチェスの評価関数を学習した研究[7] が初出である．Campbell らによるチェスの評価関数を学習した研究[4] なども挙げられるが，これらの研究は完全自動で学習できたわけではなかった．しかし保木は，棋譜に指された着手を実際に探索が指す方向に重みを調節することで，将棋の評価関数を自動的に学習することに成功し，将棋プログラム Bonanza は第 16 回世界コンピュータ将棋選手権において優勝を収めた[15]．保木による成功以降兄弟局面学習は注目を浴びており，本年度の世界コンピュータ将棋選手権で優勝した GPS 将棋も兄弟局面学習によって調整された評価関数を搭載している[12]．

自己対局による兄弟局面学習は，プロがいない環境においても自己対局により棋譜を生成することで兄弟局面学習を行う手法であり，柿木によって発案された[10]．柿木による 5 五将棋プログラム K55 は，第 2 回 UEC 杯 5 五将棋大会において優勝を収めたことから，本手法の有効性は一般的に認められている．他に，小幡らによる探索順序の学習[9] にも用いられており，SEE と比較し効率的なムーブオーダリングを行うことに成功している．また，築地らによるブロックスデュオの評価関数学習[14] にも用いられており，ゲームの特性を棋譜生成時に与えることで本学習が有効であることが示されている．

3. ブロックスデュオ

本研究ではブロックスデュオというゲームを対象として実験を行った．ブロックスデュオとは，株式会社ビバリー[⋆1]が発売している図 1 で示される盤面で行われる二人零和有限確定完全情報ゲームである．情報処理学会プログラミングシンポジウムの分科会である GPCC の主催により，コンピュータ同士の対戦も行われている[13]．GPCC による公式ルール[⋆2]を以下に示す．

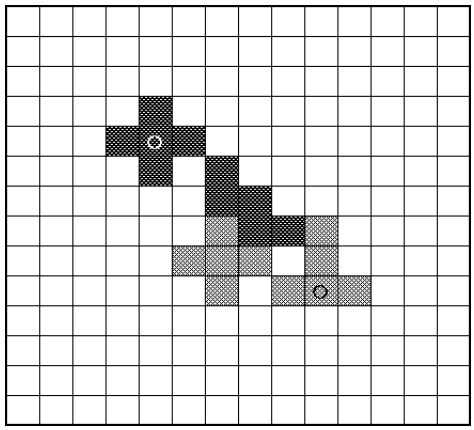

図 1 ブロックスデュオの盤面

二人で行うボードゲームである．ボードは 14 × 14 のマス目でできていて，駒はサイズが 1〜5 のポリオミノ全種類 (一人あたり 21 ピース) である．以下のルールに従って交互にピースを置いていく．置けなくなった人はパスで，二人とも置けなくなったら終了である．ボードを覆うマス目の多い人が勝ちである．

- ボードの一つの対角線の，両端から 5 番目の位置 (2 箇所) に印がついている．それぞれ 1 手目のピースはこのマスを覆うように置かなければならない．
- 2 手目以降のピースは，自分のピースの角同士が接し，辺同士は接しない位置に置かなければならない．相手のピースとはどのように接してもよい．

以降，ボードを覆うマス目のことを”スコア”と呼ぶことにする．

4. 自己対局による兄弟局面学習

ここでは，柿木によって提案された自己対局による兄弟局面学習について述べる．基盤となる理論は保木による手法[15] であり，高段者がいないため学習用棋譜を用意できないゲームにおいて，深い探索による自己対局の棋譜を学習用棋譜として用いた手法である．

4.1 自己対局による兄弟局面学習の理論

自己対局による兄弟局面学習とは，一般的に知られている「深く探索して得られた着手は浅い探索のそれと比較して良い着手である」という仮定の下，深い探

⋆1 株式会社ビバリー, http://www.be-en.co.jp/index.php(2009 年 10 月 4 日アクセス)

⋆2 GPCC, http://hp.vector.co.jp/authors/VA003988/gpcc/gpcc.htm(2009 年 10 月 4 日アクセス)

索で得られる着手を浅い探索でも得られるよう重みを調整する手法である．

まず，学習途中の評価関数によって自己対局を行い学習用棋譜を得る．目的関数を式 (1) のように設計する．

$$J(w) = \sum_{i=0}^{N-1} l(P_{\mathrm{i}}, w) + \gamma M(w) \tag{1}$$

ただし，N をある学習用棋譜における総手数，P_i をある棋譜における i 手目の局面とその兄弟局面すべて，w を学習途中の重みベクトルであるとする．また，$l(P_i, w)$ はこの学習用棋譜中の着手と，他の着手の評価値の違いの度合いを表現する関数であり，式 (2) で表わされる．また，γ はペナルティの強さ，$M(w)$ はペナルティーをかける特徴ベクトル要素の大きさに相当する関数であり，式 (3) で表わされる．

$$l(P, w) = \sum_{m=1}^{M} T\left[f(p_{\mathrm{i,m}}^{\mathrm{leaf}}, w) - f(p_{\mathrm{i,0}}^{\mathrm{leaf}}, w)\right] \tag{2}$$

$$M(w) = \sum_{l} A_l(w)\, w_l^2 \tag{3}$$

$p_{\mathrm{i,m}}$ とは，ある可能手 m によって着された i 手目の着手による局面の特徴ベクトルであり，学習用棋譜による着手を $m = 0$ とする．$p_{\mathrm{i,m}}^{\mathrm{leaf}}$ とは，$p_{\mathrm{i,m}}$ より探索によって得られた局面，M は可能手の数である．$f(p, w)$ とは，p と w の線形和で表わされた評価関数である．T とはシグモイド関数であり，学習用棋譜で実際に採用された着手と，他の可能手による評価値の差を着手一致度に変換する関数である．l とはベクトルの要素である．また，上記の l に対する総和はペナルティをかける特徴に対して取るものとする．$A_l(w)$ は，目的関数の勾配への v_l の寄与の度合いを表わす関数であり，式 (4) で表わされる．

$$A_l(w) = \sum_{i=0}^{N-1} \sum_{m=1}^{M_i - 1} \left| \frac{dT(x)}{dx} \frac{\partial}{\partial w_l} \left[f\left(p_{\mathrm{i,m}}^{\mathrm{leaf}}, w\right) - f\left(p_{\mathrm{i,0}}^{\mathrm{leaf}}, w\right) \right] \right| \tag{4}$$

兄弟局面学習では，式 (1) で表わされた目的関数の最小化を行うことにより，学習用棋譜による着手との一致度を向上させることを考えている．そこで，最急降下法の理論に基づき空間上の坂を繰り返し下っていくことで目的関数の最小化を行う．勾配は式 (5) で求められ，式 (6) により値が更新される．ただし sign は引数の符号を返す関数であり，h は学習率である．

$$\nabla_w J'(w) = \sum_{i=0}^{N-1} \sum_{m=1}^{M_i - 1} \frac{dT(x)}{dx} \nabla_w \left[f\left(p_{\mathrm{i,m}}^{\mathrm{leaf}}, w\right) - f\left(p_{\mathrm{i,0}}^{\mathrm{leaf}}, w\right) \right] + \gamma \nabla_w M(w) \tag{5}$$

$$w_l^{\mathrm{new}} = w_l^{\mathrm{old}} - h\,\mathrm{sign}\left[\frac{\partial J(w)}{\partial w_l}\right] \tag{6}$$

新しく更新された重みベクトルを用いて学習用棋譜を生成することで学習は繰り返し進行される．学習手法の枠組みを図 2 に示す．

1 重みベクトル w を初期化する
2 学習途中の w による評価関数を用いて深い探索による自己対局を行い学習用棋譜を得る
3 得られた学習用棋譜における i-1 手目から指された着手を $p_{\mathrm{i,0}}$，それ以外の可能手を $p_{\mathrm{i,m}}$ とし w を調整する
4 反復回数 N に至っていなければ 2 に戻る

図 2 自己対局による兄弟局面学習の枠組み

柿木による K55 では，得た棋譜は蓄積して繰り返し学習用棋譜として用いている．同一の棋譜を繰り返し学習に用いるため学習が収束しやすいという利点が考えられるが，学習の進んでいない精度の低い棋譜が学習に繰り返し用いられる難点がある．本稿では，学習が進んでいない精度の低い棋譜を学習対象から除く意図より，学習用棋譜は 1 回学習するごとに破棄する手法を取った．また，柿木の手法では $p_{\mathrm{i,m}}^{\mathrm{leaf}}$ を比較対象としているが，本稿においては探索を行わず $p_{\mathrm{i,m}}$ を比較する手法を取った．これはブロックスデュオは可能手が 1000 を超えることもあるほど多いゲームであり全兄弟局面から探索した局面を得ることがプログラム上難しかったことと，ブロックスデュオにおいて静止探索が確立していないため探索による効果が少ないと考えたためである．

兄弟局面学習は，正例以外に負例も与えられるところが親子局面学習と比較して優れている特徴であるとされている．特に，将棋において高段者の棋譜がある場合，兄弟局面学習が親子局面学習に対して優れた結果を残すことが金子らによって示されている[11]．この結果からプロがいない環境でも，正例に加えて負例も与えられる兄弟局面学習を適用できれば，精度の高い学習ができる可能性があると考えられる．

4.2 自己対局による兄弟局面学習の課題

本手法は，深い探索によって得られる着手を教師とするため，着手には一定の精度が求められる．K55 の評価関数の学習では，初期値に本将棋である柿木将棋の開発で培った初期値が用いられており，人間の手調整を介さない初期値から学習されたわけではない．また，学習前の評価関数との比較や，様々な学習条件下における比較もなされていない[10]．築地らによる研究では，「終盤における局面の評価はスコアのみに依存」というブロックスデュオの性質から，学習用棋譜生成時において終盤の評価関数は学習結果とは関係なくスコアだけを与えることで着手を制御するヒューリスティックを用いており，一般的に用いることができる手法ではない[14]．

5. 課題の解決策提案

ここでは，4.2 で述べた自己対局による兄弟局面学習の課題を解決するため，ゲームに依存しない汎用的に用いられている 3 つの制御を棋譜生成時の着手に用いることを提案する．

5.1 読み切り制御による着手制御

将棋では，最終的に王を詰ますことが目的となるため詰探索[5] を搭載しているプログラムが多い．ブロックスデュオでは手数が進めば必ず終局するが，学習途中の不安定な評価関数が選択する着手だけでは必ずしも勝ちを選択できない．例えば終盤においてスコア以外の評価値が大きい場合，スコアを重視しない着手が選択される可能性がある．そこで，探索時に図 3 のように着手を制御して，末端の勝敗情報を教師に加えることにした．以降，図 3 で示した制御を読み切り制御と呼ぶ．

末端局面において
- if(勝局面) 評価値 = +100000+ 評価関数 ()
- if(負局面) 評価値 = −100000+ 評価関数 ()
- if(引分局面) 評価値 = 評価関数 ()

ただし，評価関数に加える値は評価関数の返す値と比較して充分大きい値とする

図 3 読み切り制御

この制御により末端まで読み切れた場合は，勝ち局面に必ず遷移できる着手を教師にできる．そのため，学習途中の精度の悪い評価関数であっても終盤においては正しい着手を得ることができるようになる．なお，将棋では勝ち（詰み）を発見した時点で学習を打ち切る手法が一般的に用いられているようである．

5.2 進行度による評価値制御

多くのゲームプログラムにおいて，進行に応じて評価関数を変化させる手段が有効であると知られており，進行度 s を用いて式 (7)，(8) により評価関数 E を表現した．序盤の評価関数 E_o，終盤の評価関数 E_e，現在手数 t を用意し，最大手数から進行度係数 β を調整した．ただし，$s > 1$ となるときは $s = 1$ とする．

$$s = \beta t \tag{7}$$

$$E = sE_e + (1 - s)\, E_o \tag{8}$$

学習時には，E を重みベクトル w で微分することで，E_o と E_e の重みを同時に調整させている．

この制御により，読み切り制御の導入で終盤の評価関数の精度が学習を行った結果高まれば，式 (8) から序盤においても高い評価関数の精度で着手を得られると期待される．さらに，E を w で微分することで終盤で得られる学習の勾配が序盤に影響することで，序盤の学習の精度を高めることも期待される．

5.3 思考時間を長時間にする

「深く探索して得られた着手は浅い探索のそれと比較して良い着手である」という仮定が正しいならば，思考時間が長いほど深く探索できるために精度が高い学習用棋譜が得られると考えられる．特に終盤において末端まで読み切ることができれば，読み切り制御によって勝ちを確実に選択できる．予備実験として，学習前の評価関数と手調整した評価関数の 2 つのプレイヤを用意した．5.1 による読み切り制御を加えて学習前の評価関数同士，手作業による評価関数同士で，思考時間を 10[s] にしたプレイヤと 1[s] にしたプレイヤが対局を行った．対局の結果，思考時間 10[s] のプレイヤが学習前の評価関数同士で 603 勝 355 敗 42 分，手作業による評価関数同士で 596 勝 349 敗 55 分とともに勝ち越したことから，ブロックスデュオにおいても思考時間を長くするほど精度が高い着手が得られることが分かった．

このことから，学習用棋譜生成時にかける思考時間の長さの違いで学習用棋譜の精度に違いが生じるため，学習で得られる評価関数の精度にも違いが生じることが期待される．

6. 実　　験

6.1 ブロックスデュオの評価関数

ここでは，本稿において用いたブロックスデュオプログラムの評価関数について述べる．評価関数は表 1 に示す特徴により設計されている．

表 1 評価関数の特徴

特徴	重みの数
スコア	1
有効マス	1
死角マス	1
勢力範囲マス 1〜5	5
スコア（盤の重み）	105
有効マス（盤の重み）	105
死角マス（盤の重み）	105
勢力範囲マス 1〜5（盤の重み）	525
計	848

スコアとは盤に置かれたピースのマス目の数，有効マスとは自分の手番に着手することができる自分のピースの角マス数，死角マスとは自分のピースの辺に接している着手できないマス数，勢力範囲とは自分の領域を示すマス数のことを示す．勢力範囲は過去 2 回優勝したプログラム hmmm[⋆1] で用いられている特徴を参考に，有効マスからのマンハッタン距離ごとに分けた大崎らの文献を参考にしたものである[8]．それぞ

⋆1 irori の日記 - ブロックスデュオプログラム対戦結果，http://d.hatena.ne.jp/Irori/20071104/1194151812 (2009 年 10 月 4 日アクセス)

れの特徴に与えられる値は，先手の数−後手の数で表わされる．盤の重みとはそれぞれ盤の座標ごとに価値を与えたものであり，図 4 のように斜めに対称に表現することにした．また，盤の重みの特徴は，先手の特徴があれば+1，後手の特徴があれば-1 で表している．

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
1	1	2	4	7	11	16	22	29	37	46	56	67	79	92
2		3	5	8	12	17	23	30	38	47	57	68	80	93
3			6	9	13	18	24	31	39	48	58	69	81	94
4				10	14	19	25	32	40	49	59	70	82	95
5					15	20	26	33	41	50	60	71	83	96
6						21	27	34	42	51	61	72	84	97
7							28	35	43	52	62	73	85	98
8								36	44	53	63	74	86	99
9									45	54	64	75	87	100
A										55	65	76	88	101
B											66	77	89	102
C												78	90	103
D													91	104
E														105

図 4 ブロックスデュオの盤の重み

6.2 学習実験

ここでは，自己対局による兄弟局面学習によりブロックスデュオの評価関数を学習する．さらに，5 で述べた制御により，得られる結果にどのような違いが生じるか議論する．学習条件は図 5 のとおりである．また，ここでは表 2 で示された条件で学習されたグラフを示すことにする．勝ち以降を学習対象から外す，とは 5.1 において述べた将棋の評価関数の学習で使われている手法を再現したものであり，読み切り制御によって勝ちが発見された場合自己対局を打ち切ることでそれ以降を学習対象から外すことにした．得られた学習曲線を図 6 から図 13 に示す．

- 反復回数 $N = 500$ とした
- 重みの初期値はスコア・有効マス・死角マス・勢力範囲マスの価値を 100，その他の盤の重みを 0 とした
- 進行度を学習時の評価関数に適用し，進行度係数 $\beta = 0.028$ とした
- 学習率 h は一律に 1 とした
- 勾配は可能手数 M で割った値を用いた
- 学習用棋譜は 2 手目までペントミノ (スコアが 5 のピース) のみによるランダム着手，3 手目以降を評価関数による着手によって得た
- 盤の重みには $\gamma = 0.02$ によるペナルティを与えた

図 5 学習条件

読み切り制御を持たせ，かつ勝ち局面以降を学習対象から外さなかった評価関数 B と評価関数 D における終盤のスコアの価値が高くなっていることが分かる．ブロックスデュオでは相手より多くのスコアを稼ぐこ

表 2 学習した評価関数

評価関数名	読切制御有無	思考時間	勝ち以降学習対象から外す
評価関数 A	なし	1[s]	なし
評価関数 B	あり	1[s]	なし
評価関数 C	あり	1[s]	あり
評価関数 D	あり	10[s]	なし

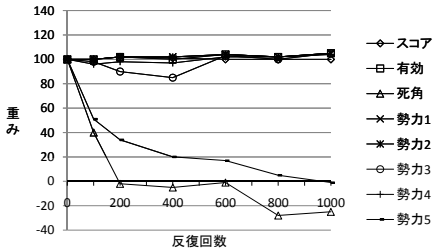

図 6 読切制御なし 1[s](関数 A)：序盤

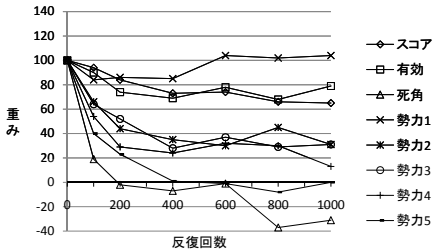

図 7 読切制御なし 1[s](関数 A)：終盤

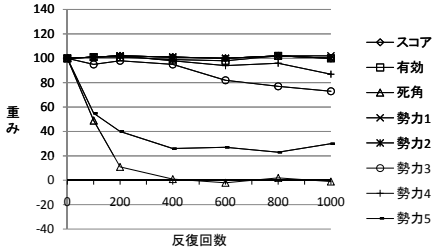

図 8 読切制御あり 1[s](関数 B)：序盤

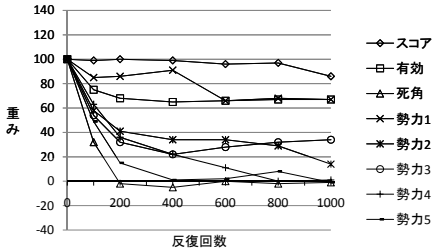

図 9 読切制御あり 1[s](関数 B)：終盤

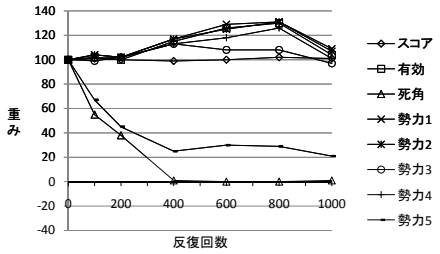

図 10 読切制御あり 1[s] 勝ち以降学習から外す (関数 C)：序盤

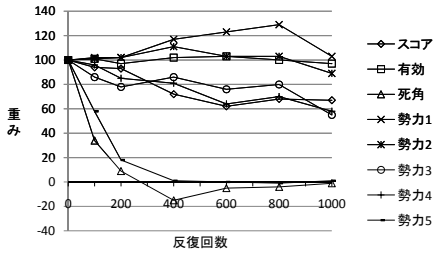

図 11 読切制御あり 1[s] 勝ち以降学習から外す (関数 C)：終盤

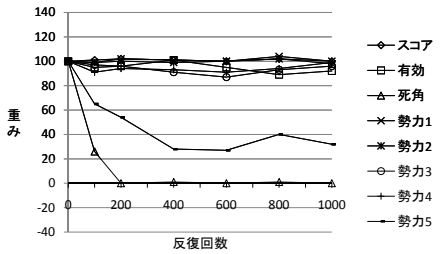

図 12 読切制御あり 10[s](関数 D)：序盤

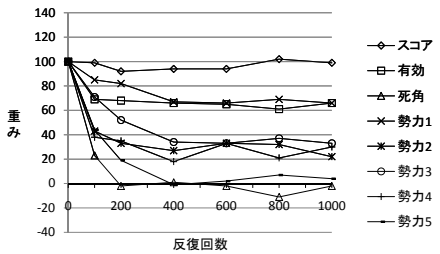

図 13 読切制御あり 10[s](関数 D)：終盤

とが勝利条件であるため，この結果は妥当と言える．

一方，読み切り制御を与えなかった学習による図 7 では，終盤のスコアが高くない結果となった．読み切り制御を与えずに学習用棋譜を得ることで，終盤においてスコアを重視する着手を指されずに学習が進んだことによる結果であると考えられ，妥当な結果が得られたとは考えにくい．また，勝ち局面以降を学習対象からを外した学習による図 11 でも終盤のスコアが高くない結果となった．読み切り制御によりスコア重視の着手を学習する機会が，勝ち局面以降を学習対象から外したことで失われたためと考えられ，これは図 7 の結果と類似したものとなった．

また序盤においては，死角マスと勢力範囲 5 マス以外の価値が学習開始時からほとんど変化しない結果が見られた．これは，着手の一致度を表す関数としてシグモイド関数を用いており，本稿における評価関数の設計で序盤において勢力範囲が多くなる傾向に由来すると考えている．勢力範囲の初期値に大きな値を与えてしまったことによって評価値の差が勾配が求まる幅を超えてしまい，学習が進みにくくなったと考えられる．

6.3 対 局 実 験

ここでは学習によって得られた評価関数を用いて対局実験を行う．対局実験の条件は図 14 のとおりである．

- 先手後手入れ替えて 500 回ずつ行った
- 思考時間 1[s] の反復深化を用い，5.1 による読み切り制御と 5.2 による進行度を着手に加えた
- 2 手目までペントミノ (スコアが 5 のピース) のみによるランダム着手，3 手目以降を評価関数による着手によって得た

図 14 対局実験の条件

6.3.1 制御手法の有効性

ここでは，5 で述べた学習用棋譜生成時における制御が自己対局による兄弟局面学習においてどのような影響を与えるかを，対局実験により調べることにする．対局に用いる評価関数は，6.2 で得られた評価関数を用いる．

表 3 と表 4 では，学習後の評価関数が学習前の評価関数と比較して強化されていることが確認できる．このことから，読み切り制御の有無に関係なく自己対局による兄弟局面学習は有効であることが示された．

表 5 では，読み切り制御を与えた学習用棋譜を用いた学習の方が，読み切り制御を与えなかった学習用棋譜を用いた学習に対して 552 勝 393 敗 55 分と勝ち越していることが分かる．このことから，読み切り制御により制御された着手による学習用棋譜を学習に用いることで，学習の精度を高められることが示された．

これは，終盤においてスコアを重視した評価関数を学習できたことによる成果であると考えられる．

表 6 では，思考時間を 10[s] による学習用棋譜を用いた学習の方が，思考時間を 1[s] による学習用棋譜を用いた学習に対して 545 勝 403 敗 52 分と勝ち越していることが分かる．これは，10[s] で得た学習用棋譜の精度が 1[s] で得たものより精度が高く，精度が高い学習用棋譜を学習に用いることで，より精度の高い学習が行われたためと考えられる．このことから，思考時間を長時間かけた精度の高い学習用棋譜を学習に用いることで，学習の精度を高められることが示された．

勝ち局面以降を学習対象から外した学習においては，表 7 で示されたとおり，外さない学習に対して 346 勝 608 敗 46 分と負け越していることが分かる．これは，勝ち以降を学習できなかったことから終盤において読み切り制御によって得られる適切な着手を学習する機会を失ったことによる結果であると考えられる．図 11 で示したとおり，終盤の学習が適切になされなかった結果，スコアの値が低くなっている．この傾向は読み切り制御を与えなかったため，終盤のスコアの重要さを学習できなかったと考察された図 7 と類似していることは 6.2 で述べたとおりである．このことから本稿において，勝ちを発見した以降の局面を学習から外す手法は終盤の確実に勝ちに遷移できる着手を学習できないため適切でなかった，と結論付けられる．

以上より,「進行度」「読み切り制御」によって着手を制御し，かつ思考時間を長時間与えた学習用棋譜を末端局面まで学習に用いることで，より精度が高く強い評価関数を学習できることを示した．

表 3　学習前の評価関数との比較 1

先手	後手	1[s]	分	学習前
読切制御なし 1[s]	学習前	376	13	111
学習前	読切制御なし 1[s]	233	25	242
計		609	38	353

表 4　学習前の評価関数との比較 2

先手	後手	1[s]	分	学習前
読切制御あり 1[s]	学習前	401	18	81
学習前	読切制御あり 1[s]	296	19	185
計		697	37	266

表 5　読切制御の有無による比較

先手	後手	あり	分	なし
読切制御あり 1[s]	読切制御なし 1[s]	329	26	145
読切制御なし 1[s]	読切制御あり 1[s]	223	29	248
計		552	55	393

表 6　学習時の思考時間による比較

先手	後手	10[s]	分	1[s]
読切制御あり 10[s]	読切制御あり 1[s]	363	26	111
読切制御あり 1[s]	読切制御あり 10[s]	182	26	292
計		545	52	403

6.3.2　反復回数による勝率の推移

ここでは，学習の反復回数によって強さにどのような変化が生じるかについて実験を行う．

表 7　勝ち以降学習から外すことによる比較

先手	後手	外す	分	外さない
読切制御あり 1[s] 外す	読切制御あり 1[s]	248	26	226
読切制御あり 1[s]	読切制御あり 1[s] 外す	98	20	382
計		346	46	608

進行度と読み切り制御を持たせ反復回数が 100 回，200 回，400 回，600 回，800 回，1000 回ごとに重みベクトルを取得し，それぞれ学習前の評価関数と手調整による評価関数とで先手後手それぞれ 500 回ずつ対局を行った．学習前の評価関数に対する勝率の推移は図 15 に，手調整による評価関数に対する勝率の推移を図 16 に示す．なお，手調整による評価関数によるプレイヤは第 2 回大会において 3 位になった筆者のプログラムである「Mate_with_pawn_drop Club ver.22 —僕と私と打ち歩詰めの会—」に同一思考時間でおよそ 60%の勝率を上げる強さであり，学習時と同じ特徴量を用いている．グラフにおける微小時間とは，制限時間が序盤において深さ 1 だけ読み切れる程度の時間まで短く制御されたものである．

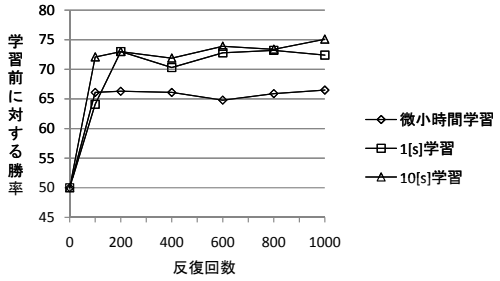

図 15　学習前の評価関数に対する勝率の推移

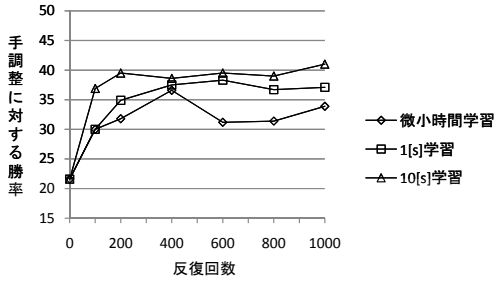

図 16　手調整の評価関数に対する勝率の推移

図 15，16 ともに反復回数 200 回に至るまで勝率は上がり，その後一定の値を推移していることが分かる．また，思考時間が長いほど強い評価関数の学習に成功したことも示されている．しかし，図 16 で示されたように，手調整による評価関数の強さを超えることはできなかった．

末端において読み切り制御を加えたことから，末端における着手は勝ちを選ぶことができる正確な着手であると言える．その着手を教師とする場合，理想的には勝率は反復回数に応じて向上し続けることが期待されるが勝率はほぼ頭打ちとなっており，低下している部分も見受けられる．これは，本学習の枠組みが「自らの着手を教師とする」ものであることから，学習途中の評価関数の着手に特化して学習が進んでしまい，極値に陥っているためであると考えられる．勝ち負けの情報を着手に与える読み切り制御により，評価関数の着手以外による”着手の指針”を与えて着手が特化することを防いでいたが，強い評価関数を学習するためには勝ち負けの情報だけでは不十分であったと考えられる．なお，図 16 から，学習用棋譜生成時における思考時間を長くかけたとしても，思考時間 1[s] と思考時間 10[s] において勝率の差が 5%程度しかないこともあり，手調整による評価関数の強さを超えることは現状の手法において難しいと考えられる．

以上より，勝率が上昇し一定の値を推移する傾向が見られるため強さの向上が確認できたものの，手調整による評価関数を超えることは難しいということが分かった．反復回数を現状より増やすことで勝率の推移を調査することが今後の課題のひとつとして考えられる．

7. おわりに

自己対局による兄弟局面学習を行う際に用いる汎用的手法の有効性について述べた．適切な初期値を与えることができないブロックスデュオにおいて，学習用棋譜を生成する際に「進行度」「読み切り制御」「思考時間」という二人零和有限確定完全情報ゲームにおける汎用的な制御を与えることによる学習結果の相違点について議論した．特に,「進行度」「読み切り制御」を与えることと思考時間を長くすることで学習用棋譜の精度を高め，より強い評価関数の学習に成功することができた．また，反復回数による勝率の推移から本手法によって勝率の向上が見られたものの，現状において手調整による評価関数を超えることは難しいことを示した．

今後の課題として，学習用棋譜の生成時における思考時間をさらに長くした場合に得られる評価関数の精度の推移について調べる必要がある．また，TD 法などの強化学習による精度と自己対局による兄弟局面学習による精度を比較する必要がある．さらに，他のゲームにおいても汎用的な制御を与えるだけで学習が適切に進むのかどうかを実験する必要がある．他に，今回与えた初期値以外で学習が適切になされるかを検証する必要がある．他に，反復回数を現状より増やすことで学習の収束性について調査する必要もある．

参 考 文 献

1) J.Baxter, A.Tridgell, L.Weaver : Experiments in Parameter Learning using Temporal Differences, ICCA Journal, Vol.21, No.2, pp.84–99 (1998).
2) D.F.Beal, M.C.Smith : First Results from Using Temporal Difference Learning in Shogi, LNCS 1558, pp.113–125 (1999).
3) M.Buro : Improving heuristic mini-max search by supervised learning, Artificial Intelligence, Vol.134, No.1–2, pp.85–99 (2002).
4) M.Campbell, A.J.Hoane.Jr, F.Hsu : Deep Blue, Artificial Intelligence, Vol.134, No.1–2, pp.57–83 (2002).
5) A.Nagai, H.Imai : Proof for the Equivalence Between Some Best-First Algorithms and Depth-First Algorithms for AND/OR Trees, IEICE Trans. on Info. and Systems, Vol.E85-D, No.10, pp.1645–1653 (2002).
6) G.Tesauro : Temporal Difference Learning and TD-Gammon, Communications of the ACM, Vol.38, pp.58–68 (1995).
7) G.Tesauro : Comparison training of chess evaluation functions, in: J. Furnkranz, M. Kumbat (Eds.), Machines that Learn to Play Games, Nova Science Publishers, pp.117-130 (2001).
8) 大崎泰寛, 築地毅, 但馬康宏, 小谷善行 : 勝率に近似させた評価関数の性能について, 情報処理学会研究報告, Vol.2009-GI-22, No.5, pp.1–8 (2009).
9) 小幡拓弥, 伊藤毅志 : α β 探索における探索順序の自動学習, 情報処理学会研究報告, Vol.2009-GI-21, pp.49–54 (2009).
10) 柿木義一 : 5 五将棋における評価関数の自動学習, エンターテイメントと認知科学研究ステーション第 5 回招待講演会 (2008).
11) 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の学習, 情報処理学会論文誌, Vol.48, No.11, pp.3438–3445 (2007).
12) 金子知適, 山口和紀 : 将棋の棋譜を利用した，大規模な評価関数の設計, 第 13 回ゲームプログラミングワークショップ, pp.152–159 (2008).
13) 築地毅, 大崎泰寛, 酒井香代子, 藤波順久, 小谷善行 : コンピュータブロックスデュオ大会報告 (2007 年 10 月，2008 年 10 月), 情報処理学会研究報告, Vol.2009-GI-22 No.4, pp.1–8 (2009).
14) 築地毅, 柴原一友, 但馬康宏, 小谷善行 : 先読みを教師とした兄弟局面の比較に基づく評価関数の学習, 情報処理学会研究報告, Vol.2009-GI-21, pp.71–78 (2009).
15) 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).