Vol.2016-GI-35 No.8
2016/3/8

学習の教師例となる指手の選択を目的とした，複数プログラムを用いる探索についての考察

竹内 聖悟[1,a)]

概要：強いプレイヤを作成するために，将棋を始めとした様々なゲームで棋譜からの学習が行われており，学習のためのデータ生成も研究課題となっている．例えば将棋の学習への利用を目的とした，局面と指手のペアの生成の研究がある．この研究の中で，教師例となる指手の質が重要であることが述べられている．従来は，深い探索によって得られる指手を教師として利用しているが，1 つのプログラム，特にそれが学習するプログラムと同じ場合，局所解に陥り，適当でないことがある．ところで，教師例となる指手の生成は対局後に行うため，対局中とは異なった様々なリソース，時間や他のプログラムなど，を使うことが可能である．本研究では，学習の教師となる質の高い指手の選択を目的として，複数プログラムを用いる探索を提案する．対象の局面を各プログラムで探索し，次に各プログラムの最善手を指した局面から探索を行う．この探索結果から各自の最善手を決定する手法である．複数プログラムの利用により，読み抜けを防ぎ，質を高めることができると考えられる．また，再帰的に上記の探索を行うことも可能であり，さらに質を高められると考えている．
将棋を対象として提案手法の性能評価を行った．正答数の比較の結果から，単一プログラムの深い探索や異種合議と同程度の性能を確認した．また，複数プログラムを用いることで，単一プログラムだけでの探索よりも幅広く質の高い指手を探索していることを確認できた．

Search Method using a few Programs to Generate Moves for Training

SHOGO TAKEUCHI[1,a)]

Abstract: Training parameters plays an important role in developing strong game programs. We required a large number of training positions and moves for the training. Thus, generating training data is an issue. Ura et al. reported that self-generated data may have a bad effect on the training. Therefore, we proposed a search method with two or more programs in order to avoid self-generation. Experimental results showed that performance of the proposed method is almost the same with the normal search and majority voting.

1. はじめに

囲碁や将棋などのゲームにおいて，強いプレイヤの作成のために棋譜を用いたパラメータの調整が行われている．例えば，将棋では人間プレイヤの棋譜を使った評価関数のパラメータ調整 [3]，囲碁では人間プレイヤの棋譜を使ったモンテカルロ木探索の方策の学習 [5] などが行われ，成功している．学習のためにはデータが必要であるが，強い人間プレイヤの棋譜数には限りがある他，プレイヤの数が少ないゲームや新しいゲームでは棋譜を得ることが難しい．そのため，学習データの自動生成についての研究されている．

将棋の評価関数の学習を目的とした学習データの生成には，中澤らの研究 [11] や Ura らの研究 [6] がある．中澤らは合議プレイヤを用いて学習データを生成し学習を行った．合議を行うことで勝率が向上するため指手の質が向上すること，学習プログラム単体では選ばない指手を合議が選ぶことによる学習の性能向上が期待される．しかし，プロ棋士の棋譜を用いるよりも性能が悪く，合議なしの棋譜を用いた場合と比べても有意な結果は得られず，原因を探

[1] 北海道大学大学院 情報科学研究科
[a)] takeuchi@erato.ist.hokudai.ac.jp

索深さと述べている [11]．Ura らは自己対戦の棋譜を元に局面を生成し，教師手は深い探索により生成した [6]．得られた棋譜データを用いて学習した評価関数は，プロ棋士の棋譜を用いたものよりも性能は悪かった．更にプロ棋士の棋譜に対し，プロの指手ではなく，深い探索結果を教師とすると勝率が下がることが確認され，教師手を学習プログラムが生成していることが性能の悪さの原因と考察している．将棋プログラム同士の棋譜においてもプロ棋士の棋譜と同様の実験結果を得ており，考察を補強している．また，チェスにおいて強化学習を行った研究では，自己対戦よりも対局サーバーにおいて多様な相手と対戦する方が学習の効率が良くなることが述べられており [1], [7]，学習プログラムによる教師手生成が性能の悪さの原因という説の傍証になっていると言える．

ところで，学習のための局面や教師を用意することを考えると，対局中に比べて様々なリソースを利用することが可能である．例えば，教師の指手として深い探索を行うというのは，対局中よりも多くの時間を利用しており，合議による指手選択も複数の評価関数というリソースを使っていると言える．質の高い教師手を用意するために，時間以外のリソースを用いるのは自然である．

本稿では，教師例となる指手の選択を目的として，局所解を回避するような探索を考える．単一プログラムの場合，特にそれが学習するプログラムと同一である場合，探索が局所解に陥る可能性がある．そこで局所解を回避するために，複数のプログラムを利用した指手の探索手法を提案する．複数プログラムの利用には合議がある．合議は複数のプログラムで独立に探索を行い多数決を取る手法であるが，提案手法は，それぞれの探索結果に対してさらに探索を行う点で異なる．異なるプレイヤの最善手や上位手を探索することにより，1 プログラムでの見落としや過剰な枝刈を避けることが出来ると考えられる．提案手法の有効性を示すために将棋を対象として実験を行う，問題集の正答数や候補手の比較などから評価を行う．

2. 関連研究

学習データの生成についての関連研究と，複数プレイヤの利用についての関連研究を述べる．

2.1 学習データの生成

Ura らは，将棋プログラムにおける学習のための局面生成を目的とした研究を行った [6]．自己対戦の棋譜を元に，棋譜中の局面，そこから探索を行った PV 末端局面，およびランダムに 2 手進めた局面を生成し，学習を行い，生成された局面の質の評価を行ったが，プロ棋士の棋譜を利用するほどの性能は得られなかった．教師手として学習プログラムによる深い探索の結果を用いたが，この教師手が性能の悪さの原因があると考察している．プロ棋士の棋譜を対象として，プロ棋士の指手を教師とした場合と深い探索の結果を教師とした場合とを比較した結果，後者は勝率が下がっており，教師として十分ではない可能性を示唆している．また，プログラム同士の棋譜でも同様の結果を得ており，学習プログラムによる教師の生成に課題があるという考察が補強されている．

また，中澤らは将棋を題材として合議プレイヤの棋譜からの学習を研究した [11]．一般に合議によって棋力の向上が得られるため，教師手の質の向上が期待されるが，学習結果としては合議を用いないものと有意な差は得られなかった．合議に用いたプレイヤは学習するプログラムがベースとなっていることから，こちらでも教師手の問題があることが考えられる．

2.2 複数プレイヤの利用

複数プレイヤの利用については，合議 [4] や相手モデル探索が関連研究として挙げられる．多数決合議とは，複数のプレイヤにそれぞれ独立に探索を行わせ，その探索結果の多数決を取る手法である．これにより勝率が向上することが示されている．複数のプレイヤとして，評価関数に乱数を加える手法の他，異なるプログラムを用いる異種合議と呼ばれる手法がある．合議は各プレイヤの指手から最終的な最善手を選ぶ手法であり，それぞれの探索は完全に独立して行われる．提案手法では互いの探索結果を探索する点が異なる．

相手モデル探索は，相手モデルを推定し，探索中に推定した相手の評価関数を利用するものである．提案手法は探索結果を利用しており，相手プレイヤの利用方法が異なる．また，Donkers らは，用いる評価関数対が Admissible の時のみ安全な探索が行えること，現実的にはその許容性を達成することが難しいことを述べている [2]．

3. 提案手法

学習の教師となる指手を生成する手法について説明する．これまで述べたように，単一のプログラムによる探索では学習データとして不向きであると考えられる．それを回避するために，複数プログラムを用いることが着眼点である．更に，複数プログラムの探索で得られた指手について探索を行うことで，質を高めることを試みる．

提案手法は 2 ステージの手法で，まず複数プログラムで独立に探索を行い，最善手を全プログラムで共有する．次に，それらの指手について探索を行い，その局面での最善手をまた共有し，以下再帰的に探索を繰り返し，予め定めた深さまで繰り返す．上記のように作成した探索木について，末端の評価値を用いて minmax 探索的にルート局面の最善手を得る．

複数プログラムを用いた探索としては異種合議がある．異種合議と提案手法との違いとしては，各プログラムの探

索結果を共有して探索を行うことがある．他に，2 プログラムからでも可能であること，2 回目以降の探索によって得られる局面が学習データの候補となることが挙げられる．

通常，探索結果としては最善応手手順（Principal Variation, 以下 PV と略す）が得られる．提案手法を用いると，PV 中の読み抜けなどの指摘が可能となる．山下はプレイヤのレーティングの計算に，棋譜の悪手や好手を用いている [8] が，この悪手や好手の定義として，その手を指した後の評価値の変動を用いている．1 手進めるだけで評価値が変動することは多く，提案手法のように手を進めて探索を行うことで質の高い指手を得ることが期待される．

また，ルート局面において Multi-PV 探索を利用することで，候補となる指手を増やし，精度を高めることも考えられる．Multi-PV 探索とは，探索結果として上位複数の指手，及びその応手手順を返すような探索である．通常は正確な評価値は最善手だけにあれば良いが，Multi-PV 探索では複数の指手に正確な評価値が必要となるため，通常探索よりも探索速度が遅くなる傾向にある．

4. 実験

提案手法の有効性を評価するため，問題集を用いて実験する．正答数の他，探索の途中で現れる最善手候補についても比較を行い，評価を行う．

比較のため，通常探索の他に，ルート局面で Multi-PV 探索を行い，各指手を進めた局面から再度探索を行い，その評価値から最善手を選ぶ手法を用いる（Multi-PV と表記する）．評価実験には，ラクラク次の一手 基本手筋集 [9]（全 216 問），ラクラク次の一手 2 基本手筋集 [10]（全 216 問）の計 432 問を利用し，計算機環境として，Intel(R) Xeon(R) CPU E5-2650 v3 (10 core 2.30GHz) 1CPU を用いた．また，探索プログラムとして GPSFish[*1] と Apery[*2] を利用した．

この実験での提案手法は，複数プログラムとして Apery と GPSFish の 2 プログラムを用い，ルート局面では Multi-PV 探索を行わず，再帰的な探索も 1 度しか行わない．

正答数の比較を行った結果を示す．通常探索，Multi-PV 後にその指手を探索する手法，提案手法の結果を表 1 に示す．提案手法では探索を複数回行うが，探索 1 回あたりの時間を 10 秒として，全体でかかった時間も併記している．また，提案手法では全体として 40 秒の時間がかかるので比較として，40 秒の通常探索の結果も示した．表中の差分下にある” 通常” は通常探索 10 秒との比較結果である．”1st” は，最初にルート局面で行った探索結果の正答数との比較を示す．統計的に有意な差は得られなかったが，提案手法により正答数が改善する傾向が見られた．

[*1] http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi/index.php?GPSFish
[*2] https://github.com/HiraokaTakuya/apery

表 1　1 度あたりの探索時間を 10 秒に揃えた時の正答数（全 432 問）

探索		Apery			GPSFish		
手法	時間（秒）	正答	差分 通常	差分 1st	#Ans.	差分 通常	差分 1st
通常探索	10	345	-	-	373	-	-
通常探索	40	353	-	-	374	-	-
Multi-PV(2)	30	350	+5	+1	339	-34	-8
Multi-PV(3)	40	346	+1	+1	346	-27	+11
提案手法	40	362	+17	+21	379	+6	+5

表 2　Multi-PV 探索を利用した時の正答数（全 432 問）

探索		Apery			GPSFish		
#PV	時間（秒）	正答	差分 通常	差分 1st	#Ans.	差分 通常	差分 1st
1	40	362	+17	+21	379	+6	+5
2	60	356	+11	+12	362	-11	+17
3	80	354	+9	+10	356	-17	+14

表 3　探索深さを増した時の正答数（全 432 問）

探索		Apery			GPSFish		
深さ	時間（秒）	正答	差分 通常	差分 1st	#Ans.	差分 通常	差分 1st
1	40	362	+17	+21	379	+6	+5
2	100	367	+22	+20	377	+4	+9
3	220	361	+16	+14	378	+5	+14

4.1 Multi-PV 探索の利用

提案手法のルート局面において Multi-PV 探索を利用し，探索候補を拡げることで，多様性や精度が改善することが期待される．この時の問題集の正答数を表 2 にまとめた．#PV が 1 となっているのは提案手法で Multi-PV を行わない場合の結果である．

Multi-PV を使う手法はいずれも正答数が下がった．複数の候補手の評価値を正確に探索するためのコストが大きく，ルート局面での探索で良い指し手候補を選ぶことができず，その後の探索の効果が出ていないのではないかと考えられる．ルート局面で良い指し手候補を選ぶことができないという仮説を補強する実験結果として，表 1 の GPSFish の Multi-PV 時の正答数の減少がある．GPSFish は Multi-PV 探索 (#PV=2,3) により，347, 335 問しか正答しておらず，通常探索よりも大きく正答数が減少していることがわかり，ルート局面での探索候補の質が低いことが分かる．

4.2 Multi-PV 探索の利用

提案手法では，ルート局面での探索結果を全プログラムで共有し，探索を行う．これを再帰的に行うことも提案手法の 1 つであり，探索を深めることで性能が向上することが期待できる．この時の問題集の正答数を表 2 にまとめた．

実験結果からは深さによる性能への影響は確認できない．理由は不明であるが，サンプル数の少なさが原因の一

表 4　候補手中の正答数の比較

	Apery	GPSFish
通常探索 (m=1)	345	373
通常探索 (m=2)	372	370
通常探索 (m=3)	381	379
提案手法 (m=1)	393	393
提案手法 (m=2)	399	399
提案手法 (m=3)	392	392

表 5　異種合議との比較，探索 1 回 10 秒（全 432 問）

	Apery	GPSFish	Blunder
合議	373	372	370
提案手法	359	366	359
通常探索	347	364	354

つとして考えられる．

4.3 新しい指し手の発見について

提案手法によって得られる指手や探索候補の多様性とその質について評価を行う．ルート局面での最初の探索結果として得られる指手の中に，どれだけ正答が含まれているかを示す．例えば，提案手法で Multi-PV を行なわない場合はルート局面で各プログラムが探索した最善手の 2 手の内に正答が含まれるかどうかである．

結果を表 4 に示す．手法の後の”(m=1)” は Multi-PV 探索の PV 数を表している．通常探索 (m=2) と提案手法 (m=1) とでは，候補となる指手の数は同一であるが，提案手法の方が多く正答数を含んでいる．提案手法のように複数プログラムを利用することで，多様性と質の高さが得られていることが確認できる．

4.4 異種合議との比較

提案手法と異種合議との性能比較を行う．異種合議では多数決を取るために，3 つ以上のプログラムが必要となる，この実験では 3 つ目のプログラムとして，Blunder (BlunderXX-20130508)[*3] を利用した．

この実験での合議は，3 つのプログラムで 10 秒探索を行い，その結果の多数決を取る．同票の場合は予め定めたリーダープログラムが選んだ指手を選ぶ．結果を表 5 にまとめた．合議でリーダープログラムを変えた場合の結果があるため，3 つの欄が存在する．提案手法，各プログラムが 10 秒で探索を行った結果を進めて探索を行い，その評価値から最善手を選ぶ手法，最初に行った各プログラムの通常探索の結果も併記した．有意な差ではないが，合議が最も正答数が高くなった．一方で，Apery と GPSFish の 2 プログラムを利用した提案手法の結果と比較すると，Apery では改善しているが，GPSFish では明確な差は得られていない．

[*3] http://ak110.github.io/

5. 終わりに

学習のためのデータ生成として教師となる指手の生成についての研究を行った．学習プログラムによって教師を生成する手法では局所解に陥るためか，性能が悪くなるという知見がある．本研究では，学習の教師となる質の高い指手の選択を目的として，複数プログラムを用いる探索を提案した．学習データの生成は対局中と異なり，時間や他のプログラムなどの様々なリソースを使うことが可能であるので，質の高いデータの生成のために複数プログラムや多くの時間を利用することは自然である．提案手法では，対象の局面を各プログラムで探索し，次に各プログラムの最善手を指した局面から探索を行い，これを再帰的に繰り返して探索木を構築し，その探索木から各自の最善手を決定する手法である．複数プログラムの利用により，読み抜けを防ぎ，質を高めることができると考えられる．

問題集による性能評価を行い，提案手法によって得られる指手について深い探索や異種合議と同等の性能であることを確認した．また，複数プログラムを用いることで，探索候補の多様性が増すことも確認した．このことから，学習プログラムによる深い探索で局所解に至ることを，提案手法で回避することと指手の質を高めることが期待できる．一方で，Multi-PV や探索深さを深めることで質の向上が期待されたが，改善が確認できない結果であった．これらの結果について考察し，原因を取り除くことで質の向上を行いたい．また，多様性や質の高い指手が得られることが期待できるが，問題集での評価であるため，実際に学習に用いた時の性能については不明である．今後は実際に得られた指手を教師として学習を行うことが課題である．

謝辞　本研究は JSPS 科研費 26730181 の助成を受けたものです．

参考文献

[1] Jonathan Baxter, Andrew Tridgell, and Lex Weaver. Learning to play chess using temporal differences. *Machine Learning*, Vol. 40, No. 3, pp. 243–263, 2000.

[2] H.H.L.M Donkers, J.W.H.M Uiterwijk, and H.J van den Herik. Admissibility in opponent-model search. *Information Sciences*, Vol. 154, No. 3–4, pp. 119–140, 2003. Heuristic Search and Computer Game Playing.

[3] Kunihito Hoki and Tomoyuki Kaneko. Large-scale optimization for evaluation functions with minimax search. *Journal of Artificial Intelligence Research*, Vol. 49, No. 1, pp. 527–568, January 2014.

[4] Takuya Obata, Takuya Sugiyama, Kunihito Hoki, and Takeshi Ito. Consultation algorithm for computer shogi: Move decisions by majority. In H.Jaap Herik, Hiroyuki Iida, and Aske Plaat, editors, *Computers and Games*, Vol. 6515 of *Lecture Notes in Computer Science*, pp. 156–165. Springer Berlin Heidelberg, 2011.

[5] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrit-

twieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, Vol. 529, No. 7587, pp. 484–489, 01 2016.

[6] Akira Ura, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. Comparison training of shogi evaluation functions with self-generated training positions and moves. In H. Jaap van den Herik, Hiroyuki Iida, and Aske Plaat, editors, *Computers and Games*, Lecture Notes in Computer Science, pp. 208–220. Springer International Publishing, 2014.

[7] Joel Veness, David Silver, William Uther, and Alan Blair. Bootstrapping from game tree search. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pp. 1937–1945. 2009.

[8] 山下宏. 将棋名人のレーティングと棋譜分析. 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. 9–16, oct 2014.

[9] 日本将棋連盟書籍（編）. ラクラク次の一手 基本手筋集. 日本将棋連盟, 2003.

[10] 日本将棋連盟書籍（編）. ラクラク次の一手 2 基本手筋集. 日本将棋連盟, 2003.

[11] 中澤隆久, 林伸也, 鶴岡慶雅, 田浦健次朗, 近山隆. 合議アルゴリズムプレイヤの棋譜を用いた将棋の評価関数の学習. 情報処理学会研究会報告, 2012-GI-27, pp. 1–4, February 2012.