

探索結果を利用した実現確率探索

佐藤佳州^{†1} 高橋大介^{†1}

本論文では、探索結果に基づく実現確率探索を提案する。実現確率による探索打ち切りアルゴリズム（実現確率探索）は、コンピュータ将棋において注目を集めている探索法の一つであり、多くのトップレベルのプログラムがこのアルゴリズムをベースとした探索法を用いている。実現確率探索は探索深さの決定にプロの棋譜から求めた指し手の確率を用いることで、あり得そうな展開を深く探索するという特徴を持つアルゴリズムである。この手法は、よい結果を収めているものの、探索の残り深さや評価関数の性質などによらず確率が常に一定であるという点に改善の余地があると考えられる。また、ある程度探索を行いよさそうな手が分かってきた場合にも、プロの棋譜から求めた確率に基づき確率を決定してしまう点も問題点と考えられる。

本論文ではこれらの問題点について、プログラムの探索結果を用い、深さに応じた確率を利用する、探索中に得られる情報を特徴として利用する、といった改良を行うことにより改善を試みた。実験の結果、プロの棋譜を基にした従来の実現確率探索と比較し、6割程度の勝率を得ることに成功し、提案手法の有効性を示した。

The Realization Probability Search Based on Search Results

YOSHIKUNI SATO^{†1} and DAISUKE TAKAHASHI^{†1}

In this paper, we propose the realization probability search algorithm based on search results. The realization probability search is one of the search algorithms attracting much attention in the Computer-Shogi area, and many top-level programs use this algorithm. This method decides search depths according to probabilities of moves obtained from game records by professional players, and searches deeper for more probable moves.

The realization probability search is an efficient algorithm, but it is considered that it still has room for improvement because probabilities of moves are fixed at any time.

We improved the realization probability search by using the search results. In our method, probabilities change according to the rest of search depths and heuristics obtained while searching. In the result of our experiments, our program based on the proposed method is superior to existing methods.

1. はじめに

ゲームプログラミングにおいて、探索は重要な要素であり、多くの研究が行われてきた。探索の手法としては、大きく分けて全幅探索と選択探索が存在する。将棋では合法手の多さから、以前は選択探索が主流であったが、Bonanza¹⁾ 登場以降全幅探索も見直されつつあり、どちらが優れているかという結論は今のところ得られていない。

選択探索のうち、注目を集めている手法の一つとして実現確率による探索打ち切りアルゴリズム²⁾がある。

この手法は、プロの棋譜を基にした学習により指し手の探索深さを決定するというもので、あり得そうな展開をより深く探索できるという特徴をもつ。実現確率による探索打ち切りアルゴリズムは世界コンピュータ将棋選手権でも多くのプログラムで採用され、成功を収めている探索法の一つであるといえる。

実現確率による探索打ち切りアルゴリズムは非常に優れた探索法であるが、プロの棋譜から求めた指し手の選択される確率を利用するため、探索の残り深さや評価関数の性質などによらず、どのような状況でも確率が一定になってしまうといった点について、改善の余地があると考えられる。また、実際に探索を行っていくと、表面的な特徴だけでは分からないような指し

^{†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

手の善悪が分かってくる．このような場合，指し手の表面的な特徴に基づいた確率よりも，探索中に得られる情報（ある手が最善手であった割合など）を重視した方がよい性能を得ることができる可能性がある．

本論文では，このような問題点をプログラムの探索結果を利用した確率を用いることにより改善した実現確率探索を提案する．

2. 関連研究

2.1 実現確率による探索打ち切りアルゴリズム

実現確率による探索打ち切りアルゴリズム²⁾（以下，実現確率探索）は鶴岡によって提案された手法である．この探索法は多くの将棋プログラムで採用されており，盛んに研究が行われている³⁾⁻⁶⁾．

一般的な探索法では，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．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．

（局面の実現確率）

$$= (\text{親の局面の実現確率}) \times (\text{遷移確率})$$

ルート局面の確率は 1.0 とし，遷移確率には指し手の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．

指し手の選択確率については，以前は指し手の表面的な特徴を 1 つだけ利用して算出するのが一般的であったが，現在の「激指」⁷⁾ ではロジスティック回帰を利用し，指し手についての様々な情報を総合的に使って確率を算出するといった手法を用いている．本論文でも指し手の選択確率の算出にはロジスティック回帰を利用した手法を用いる．

2.2 自動実現確率探索 (ARPS)

自動実現確率探索 (Automatic Realization-Probability Search, ARPS)⁸⁾ は橋本らによって提案された手法である．実現確率探索には指し手の確率を求めるために優れた棋譜を大量に要するといった問題点がある．ARPS ではこの問題を，コンピュータ同士の自動対局によって得られた棋譜を利用することにより解決している．手本となるべき棋譜のほとんどないゲームである「Lines of Actions」に適應し，通常の反復深化法に勝ち越すといった結果を得ている．

本論文の提案手法もプログラムの探索結果を用いており，ARPS を発展させた形のアルゴリズムとなっている．

3. 提案手法

3.1 実現確率探索の問題点

実現確率探索の基本的な考え方は，指し手の特徴に応じ，プロがその特徴をもつ手を選択した割合によって，探索深さを決定するというものである．この手法は多くの場合において，ある程度よい結果を得ているものの，必ずしも最善とはいえない．

プロの棋譜を用いた実現確率探索の問題点として，探索の残り深さやプログラムの評価関数の性質によらず，確率が常に固定であるといった点があげられる．例えば探索の末端に近いノードで，10 手先によくある可能性のある手を深く探索しても無駄になる可能性が高い．探索の残り深さが少ない局面では，単純に「駒を取る手」など，直接的に得をする手の確率を高くした方がよいと考えられる．また，評価関数と確率の相性が悪い場合には，（評価関数の性質上）よいと評価できない指し手を無駄に深く探索する，あるいは評価値がよい手に低い確率を割り当て再探索が頻繁に起こる，といった問題が生じる可能性がある．このような問題を回避するため，確率の値は固定ではなく，評価関数や探索の残り深さに応じたものにすることが望ましいと考えられる．

その他の問題点としては，ある程度探索を行い，よさそうな手が分かってきた場合にも，そのような情報は確率には全く反映されないといった点があげられる．また，実現確率探索の性能はどのような指し手の特徴を用いるかに大きく依存するが，指し手を詳細な特徴に分類することは容易ではないといった点も大きな問題点の一つといえる．

3.2 探索結果を利用した実現確率探索

本論文では，プログラムの探索結果を学習データとして利用した実現確率探索を提案する．プロの棋譜の代わりに，プログラムの探索結果を利用するという考え方は，ARPS (自動実現確率探索) でも用いられている．しかし，ARPS では，プロの棋譜を入手できないようなゲームに対する適用を目的としており，単純に適用しただけでは，従来の実現確率探索を上回ることは難しいと考えられる．提案手法では，探索結果を利用し評価関数との相性を考慮するほか，

- 探索中に得られる情報を特徴として利用する
- 探索の残り深さに応じた指し手の確率を用いる

- 学習データとして探索結果とプロの棋譜を併用する

といった手法により、従来の実現確率探索を上回る性能を得ることを目的とする。具体的な手順は次に示す通りである。

- (1) 事前に、適当な局面においてプログラムで深さ $1 \sim n$ の探索を行い、最善手を求める（今回は実験的に $n = 7$ とした）。この操作を複数の局面について行い、確率を求めるための学習データを作成する。
- (2) 学習データとして、深さ $1 \sim n$ の探索結果を用いた場合について、指し手が最善手として選択される確率をそれぞれ算出する。
- (3) 探索を行う際には、残り深さが d 以上 $d+1$ 未満のときには、深さ d の探索結果を学習データとした確率を用いる^{*1}。 $d > n$ の場合には、常に深さ n の探索結果を用いる（以下、提案手法 1）、プロの棋譜から求めた確率を用いる（以下、提案手法 2）の 2 通りの手法を検討する。

以下で提案手法の特徴について具体的に説明していく。3.2.1 節、3.2.2 節で述べる探索中に得られる情報の利用、探索の残り深さに応じた確率の利用は提案手法 1、提案手法 2 で共通して行う手法である。3.2.3 節で述べるプロの棋譜の併用は提案手法 2 のみで行う。なお、以降本論文中で単に提案手法と表記した場合、提案手法 1、提案手法 2 の両方に共通する特徴であることを示す。

3.2.1 探索中に得られる情報の利用

提案手法の特徴の一つとして、探索中に得られる情報としてキラームーブ、ヒストリーヒューリスティックを指し手の特徴として用いる点があげられる。キラームーブは兄弟ノードの最善手、ヒストリーヒューリスティックはある手が最善手であった割合を示す。これらの値はコンピュータ将棋やコンピュータチェスでは、探索中の指し手の並べ替え（ムーブオーダーリング）などによく用いられている。キラームーブ、ヒストリーヒューリスティックはともに探索中に変化する値であり、実際に探索を行わないプロの棋譜を学習データとした通常の実現確率探索では利用できない。これらの特徴を用いることにより、表面的な特徴だけでは分からない指し手の善悪を確率に反映できると期待できる。

3.2.2 探索の残り深さに応じた確率の利用

提案手法では、探索の残り深さに応じた確率を用い

ている。探索の残り深さが少ない場面では浅い探索結果を基にした確率を利用し、逆に探索の残り深さが深い場面では深い探索結果を基にした確率を利用する。このようにすることで、残り深さが少ない局面では、浅い探索でよい結果が得られそうな指し手の確率が高くなる、といったような残り深さに応じて適切な確率を用いることができると考えられる。

3.2.3 プロの棋譜の併用

提案手法では指し手の確率を求めるために、プログラムの探索結果を作成しなければならない。多くの局面について探索を行う必要があるため、非常に時間のかかる処理になり、探索深さも限られたものになるという問題がある。この問題を改善するため、探索の残り深さが多い場合、学習用のデータとしてプロの棋譜を併用することを検討する。この手法は、探索深さを十分に増やした場合には、理想的にはプロの指し手とプログラムの指し手がほぼ一致する、という考えに基づくものである。

提案手法 1（プロの棋譜を併用しない）の利点としては、ARPS と同様、人間の知識がないゲームにも適用できることがあげられるのに対し、提案手法 2（プロの棋譜を併用する）の利点としては、探索の残り深さが十分な場合に精度のよい確率を得ることができると期待できる点があげられる。

4. 実装

4.1 実装、実験に用いたプログラム

提案手法の実装、実験には著者が開発中のプログラム「棋理」を用いた。「棋理」は探索法として、全幅探索を採用しており、強さはアマチュア三段程度^{*2}である。本研究では「棋理」の探索部分に通常の実現確率探索、提案手法をそれぞれ実装し、全幅探索も含めた性能の比較を行う。

なお、「棋理」の探索中では以下の枝刈りを行っている。

- null move pruning
- futility pruning
- history pruning

このうち history pruning については、全幅探索のみで行っている。history pruning は単純に実現確率探索と組み合わせてもよい結果を得ることができなかったためである。また、探索の末端では静止探索を行っている。

*1 今回実験を行うプログラムでは、実装面や速度面の都合から、残り深さ 1 未満では実現確率探索を用いていないため、 $d = 0$ の場合は考えない。

*2 世界コンピュータ将棋選手権や floodgate⁹⁾ での成績、次の一手問題の正答数から推定。

4.2 用いた特徴, 学習データ

実現確率探索の指し手の分類に用いた特徴は以下の通りである.

- 駒の損得
- 駒を取る手, 直前に動いた駒を取る手
- 成る手
- 逃げる手
- 当たりをかける手
- 王手
- 王手を防ぐ手 (玉の移動, 合駒)
- 移動による玉の危険度の変化
- 玉との相対位置テーブルの値の増減
- キラームーブ
- ヒストリーヒューリスティック

さらにそれぞれの項目について, 駒の種類, 移動する手が打つ手かといった項目を考慮し, 50種類程度の特徴を用いている「激指」では移動先, 移動元の 3×3 の盤面パターンなど 10万程度の特徴を用いていることを考えると, 本研究で用いた特徴の数は少ないといえる.

指し手の確率の算出には以下の学習データを用いた.

- プロの棋譜 300局 (通常の実現確率の算出に用いる)
- 上記の棋譜に現れるすべての局面について, 深さ 1~7 の全幅探索を行った結果

ロジスティック回帰の計算部分には LIBLINEAR¹⁰⁾ を利用している. LIBLINEAR とは SVM やロジスティック回帰といったデータの分類に関する計算を高速に行うライブラリである.

なお, 指し手はすべて生成した上で, 特徴に分類する. 確率が低い手についても知識による前向き枝刈りは行わないため全幅探索に近い形の実装になっているといえる. また, 1手で消費する深さは確率に応じて深さ 3/4 から 3 相当に制限している. 消費する深さの下限が大きめの値に設定してあるのは, 予備実験では延長をあまり行わない方がよい結果を得たためである. この範囲は実験的に決定したため, さらに検討の余地はあると考えられる.

5. 実験結果

5.1 実験環境

本研究の実験環境を表 1 に示す. 実験結果はすべて 1CPU で行ったものである.

自己対局は, 定跡で 16 手目まで進めた局面 (同一局面や, 極端に差がついた局面は除く) から開始し, 先後を入れ替えて 250 セット, 計 500 局の対局を行っ

表 1 実験環境

CPU	Xeon X5355 2.66GHz
メモリ	2GB

た. なお, 引き分けが生じた場合にはその対局は無効とし, 開始局面を変更した上で勝敗がつくまで再対局を行っている. 思考時間は 1 手 4 秒とした.

5.2 探索速度

表 2 に探索法ごとに 1 秒間に探索できるノード数を計測した結果を示す. ここで示す nps の値は静止探索中のノードもカウントしたものである.

表 2 1 秒間に探索できるノード数

手法	1 秒間に探索できるノード数 (nps)
全幅探索	342,093
提案手法 1	287,102
提案手法 2	288,985

提案手法の探索速度は, 全幅探索と比較し, 約 16% の速度低下となっている. ただし, 指し手の特徴の分類やロジスティック回帰など時間がかかる処理が増えていることを考えると, 速度低下としては小さいといえる. これは, 残り深さ 1 では実現確率探索を用いていないことや, 特徴の数がそれほど多くないことが理由だと考えられる.

5.3 探索中に得られる情報を特徴として利用した場合の効果

探索中に得られる情報 (キラームーブ, ヒストリーヒューリスティック) を利用した場合の効果を検証した. 表 3 は探索中に得られる情報を利用した場合と利用しない場合で対局を行った結果である. 学習データとしては深さ 7 の探索結果を用いている.

表 3 探索中に得られる情報を特徴として利用した場合の効果

手法	勝敗	勝率
探索中の情報を利用した場合	296 勝 204 敗	0.592

実験の結果, 探索中の情報を利用したプログラムが勝率 0.592 で勝ち越した (有意水準 5% の二項検定で有意). この結果から, 探索中に得られる情報を利用することは有効であると考えられる.

以下に探索中の情報を特徴として利用した場合の例を示す. 表 4 は学習データとして深さ 7 の探索結果を利用した場合の図 1 における指し手の確率 (上位 10 手) である. 比較として表 5 にプロの棋譜に基づく通常の実現確率探索を用いた場合の確率を示す.

【図は 2 手目△3四歩まで】

	9	8	7	6	5	4	3	2	1	
	香	桂	銀	金	玉	金	銀	桂	香	▲先手なし
		飛						馬		一
	歩	歩	歩	歩	歩	歩		歩	歩	二
										三
										四
			歩							五
										六
▽読み主	歩	歩	歩	歩	歩	歩	歩	歩	歩	七
		角						飛		八
	香	桂	銀	金	玉	金	銀	桂	香	九

図 1 探索中に得られる情報を特徴として利用した場合の効果 (例)

表 4 図 1 における指し手の確率 (深さ 7 の探索結果を学習データとして利用した場合)

オーダー	探索開始時		探索中	
	指し手	確率	指し手	確率
1	2 二角成	0.1484	2 六歩	0.1110
2	4 八銀	0.0453	2 二角成	0.0989
3	7 七角	0.0447	4 八銀	0.0362
4	2 六歩	0.0428	6 六歩	0.0312
5	6 八飛	0.0422	7 七角	0.0310
6	7 八飛	0.0404	5 八金右	0.0208
7	3 八銀	0.0393	5 六歩	0.0205
8	5 六歩	0.0391	9 六歩	0.0194
9	6 六歩	0.0390	7 八金	0.0189
10	5 八飛	0.0388	5 八金左	0.0189

表 5 図 1 における指し手の確率 (プロの棋譜を学習データとした場合)

オーダー	指し手	確率
1	2 二角成	0.1050
2	4 八銀	0.0411
3	6 八飛	0.0391
4	2 六歩	0.0305
5	5 六歩	0.0305
6	3 八銀	0.0297
7	6 六歩	0.0294
8	7 八飛	0.0275
9	7 八金	0.0275
10	5 八金右	0.0248

表 4 の探索開始時は初めてノードを訪問したときの確率, 探索中は探索を開始して 4 秒後の確率を表している. 表中の確率が高い指し手ほど, より深く探索されることになる.

探索前の指し手の確率はプロの棋譜を基にした通常の実現確率と比較し, 概ね高い確率が割り振られている. これは, history heuristic (ある指し手が最善手

であった割合)の初期値をすべて 1.0 に設定しているためだと考えられる. 一方, 探索中 (探索開始 4 秒後) の確率は探索開始時と比べ, 上位数手のみに高い確率が割り当てられ, その他の手の確率は通常の実現確率探索の指し手の確率よりも低くなっていることが分かる.

この結果から, キラームーブ, ヒストリーヒューリスティックといった探索中に得られる情報を特徴として利用することで, 初めは通常の実現確率探索と同様, 指し手の表面的な特徴によりよさそうな手を深く探索し, 徐々に探索中によりと判断された手に絞って深く探索していくという探索が実現されていると考えることができる.

5.4 探索の残り深さに応じた確率を用いる効果

探索の残り深さに応じた確率を用いることによる効果を検証した. 表 6 は, 深さ 1~7 の探索結果を利用し, 探索の残り深さに応じて確率を使い分けた場合 (提案手法 1) と残り深さによらず常に深さ 7 の探索結果を学習データとした確率を用いた場合の対局結果である.

表 6 深さ 7 の探索結果のみを学習データとした確率を用いた場合との対局結果

手法	勝敗	勝率
提案手法 1	273 勝 227 敗	0.546

実験の結果, 探索の残り深さに応じた確率を用いた場合が勝率 0.546 で勝ち越した (有意水準 5% の二項検定で有意). この結果から, 探索の残り深さに応じた確率を用いることは有効であると考えられる.

以下に探索の残り深さに応じた確率を用いた例を示す. 表 7 は提案手法を用いた場合の図 2 の局面における指し手の確率である. 深さ 1, および深さ 7 の探索結果を学習データとした場合の指し手の確率をそれぞれ上位 5 手を示している.

深さ 1 の場合には, 深さ 7 の場合と比較し, 単に駒を取る手など直接的に得をする手の確率が高くなっている. また, 深さ 7 の場合には大駒を切る手などにも比較的高い確率が割り振られているのに対し, 深さ 1 の場合にはそのような駒損をする手は上位には挙がっていない.

残り深さが少ない場合には, 読みを入れなければよいと判断できないような指し手の確率は低くなっており, 枝刈りのような効果を果たしているといえる. 通常の実現確率探索では, 残り確率から考えて意味のな



図 2 深さに応じた学習データを用いる効果 (例)

表 7 提案手法を利用した場合の図 2 における指し手の確率

オーダー	深さ 1		深さ 7	
	指し手	確率	指し手	確率
1	6 八飛	0.1241	1 三桂成	0.1325
2	8 六馬	0.1023	6 八飛成	0.1084
3	6 八飛成	0.0953	6 八飛	0.0912
4	1 三桂成	0.0835	3 一飛成	0.0369
5	4 六香	0.0481	3 二馬	0.0348

い手は生成しないといったようなことを手動で行っている²⁾。例えば送りの手筋などは、駒を一旦損して、取り返すところまで読めなければならぬため、残りの確率の少ないときには生成されない。提案手法を用いれば、このような残り深さに応じた確率の算出も自動でできる可能性があると考えられる。

5.5 学習に用いる探索結果の探索深さを増やすことによる性能の変化

図 3 に、指し手の確率の学習に用いる探索結果の深さを増やすことによる性能の変化を示す。勝率は、提案手法 1 において、深さ 1 のみの探索結果を用いた場合との対局結果を示している。

実験結果から、深い部分までの探索結果を用いることで性能が向上していることが分かる。本実験では、深さ 7 までの探索結果しか用いていないが、さらに深い部分までの探索結果を利用することで性能を改善できる可能性もあると考えられる。

5.6 プロの棋譜を併用する効果

探索の残り深さが多い場合に、プロの棋譜を併用した確率を用いる効果を検証した。表 8 に提案手法 1 (プロの棋譜を併用しない場合) との対局結果を示す。

プロの棋譜を併用した場合が勝率 0.530 と勝ち越し

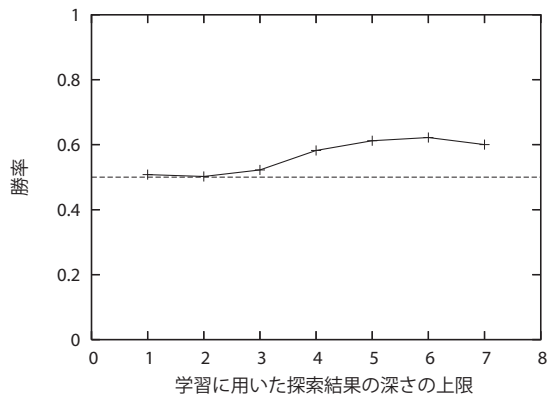


図 3 学習に用いる探索結果の深さを増やすことによる性能の変化

表 8 提案手法 1 (プロの棋譜を併用しない) との対局結果

手法	勝敗	勝率
提案手法 2	265 勝 235 敗	0.530

ているものの、有意な結果を得ることはできなかった。ただし、プロの棋譜を併用した確率に切り替える深さを調整することなどにより、性能をさらに改善できる余地は残されていると考えられる。また、本実験の思考時間は 1 手 4 秒と比較的短い設定となっているため、残り深さが多い部分でプロの棋譜を併用した効果が現れにくかった可能性もあると考えられる。

5.7 自己対局による性能評価

表 9、表 10 は、提案手法と全幅探索および通常の実現確率探索の性能を比較した結果である。比較に用いた全幅探索のアルゴリズムは以下の特徴をもつ、一般的なものである。

- PVS (Principal Variation Search) + 静止探索
- 王手, リキャプチャ, one-reply 延長
- 枝刈り (null move pruning, futility pruning, history pruning)

表 9 全幅探索との対局結果

手法	勝敗	勝率
通常の実現確率探索	311 勝 189 敗	0.622
提案手法 1	349 勝 151 敗	0.698
提案手法 2	356 勝 144 敗	0.712

表 10 通常の実現確率探索との対局結果

手法	勝敗	勝率
提案手法 1	291 勝 209 敗	0.582
提案手法 2	294 勝 206 敗	0.588

全幅探索，通常の実現確率探索との対局において，提案手法 1，提案手法 2 がともに勝ち越していることが分かる（すべて有意水準 5% の二項検定で有意）．この実験では，プロの棋譜を学習データとして併用した提案手法 2 が提案手法 1 よりもわずかによい結果を得た．

探索結果のみを学習データとした提案手法 1 についても，従来の実現確率探索を上回る結果を得た．今回の実験では，学習データとして，深さ 7 までの探索結果しか用いていないため，より深い部分までの探索結果を利用することで，さらに性能を改善できる可能性がある．また，この結果からプロの棋譜等の優れた学習データを得ることが難しいゲームにおいても，従来の実現確率探索と同等以上の性能を得ることができると考えられる．

5.8 探索ノードの深さ

図 4 に探索法による探索ノードの深さの違いを示す．実際の探索中にある深さに対応するノードが全探索ノードのうちどれくらいの割合を占めているかを示している．静止探索中のノードはカウントしていない．

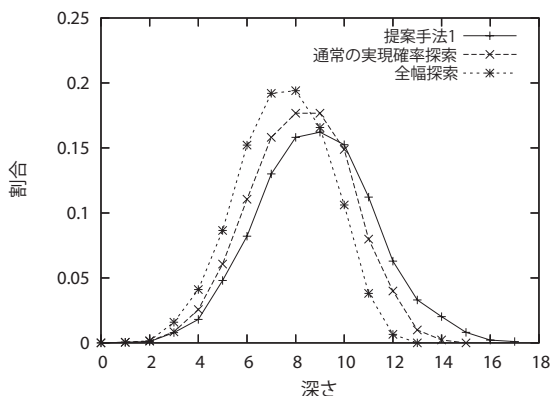


図 4 探索ノードの深さ（思考時間 1 手 4 秒）

実験結果から，全幅探索と通常の実現確率探索では，実現確率探索の方が深いノードを多く探索していることが確認できる．ただし，文献 2) の結果と比較すると，その差はやや小さいといえる．これは，本実験で用いた全幅探索では，history pruning や null move pruning, futility pruning といった枝刈り手法を用いることや，王手や駒の取り返しなどの戦術的な手を延長することにより，単純な手法よりも深くまで探索できているためだと考えられる．また，実現確率探索において 1 手当たりの消費する深さに制限を設けていることも理由の一つであると考えられる．

提案手法は，通常の実現確率探索と比較し，より深いノードを多く探索していることが分かる．この理由としては，探索中に得られる情報を特徴として利用している効果が大きいと考えられる．通常の実現確率探索では指し手の確率が終始一定なのに対して，提案手法の場合には，実際に探索してみてもよくなかった手の確率は動的に下がって行くという特徴がある．この結果，通常の実現確率探索よりも深いノードまで探索できていると考えられる．

なお，グラフ中には提案手法 2（学習データとしてプロの棋譜を併用した場合）の結果は載せていないが，提案手法 1 とほぼ同じ結果となっている．

5.9 floodgate でのレーティング

floodgate⁹⁾ においてレーティングの測定を行った．floodgate とはサーバー上で他の将棋プログラムと連続対局を行い，プログラムの相対的な強さをレーティングとして求めることのできるシステムである．参加しているプログラムの数や強さが常に一定ではないといった問題点が存在するが，ある程度長い思考時間でさまざまなプログラムと対局できることからコンピュータ将棋の性能の有力な評価指標の一つになり得ると考えられる．

表 11 は探索法としてそれぞれ，全幅探索，通常の実現確率探索，提案手法を用いた場合の，100 局対局を行った時点でのレーティングである．

表 11 floodgate のレーティング（100 対局時）

探索手法	レーティング
提案手法 2	2080
提案手法 1	2067
通常の実現確率探索	1933
全幅探索	1901

探索法として全幅探索および通常の実現確率探索を用いたプログラムと比較して，提案手法を用いたプログラムのレーティングが高くなっていることが分かる．この実験結果から，提案手法がある程度長い思考時間でも有効であること，他のプログラムとの対局においても有効であることが示しているといえる．

6. 今後の課題

6.1 特徴の数や学習に用いる探索結果の深さを増やした場合の性能の比較

50 程度という比較的少ない特徴を用いた今回の実験では，提案手法は従来の実現確率探索を上回る性能を得た．

ただし、実現確率探索の性能は指し手の分類に用いる特徴の数や性質に大きく依存する。指し手の詳細な分類は容易な作業ではないため、特徴の数が少なくてもよい結果が得られるという点は提案手法の利点といえるが、特徴の数を増やし、詳細な指し手の分類を行った際にも同様の結果が得られるかさらに検討する必要があると考えられる。

また、実験結果から指し手の学習に用いる探索結果の深さを増やすことは性能の向上に有望であると考えられる。本論文では深さ7までの探索結果しか用いていないが、さらに深さを増やした時の性能の変化についても実験を行いたい。同時に、今回の実験では有意な結果を得ることのできなかつた、探索の残り深さが十分に深い部分においてプロの棋譜から求めた確率を併用する手法についても、さらに検討の余地があると考えている。

6.2 評価関数の性質に応じた性能の比較

提案手法では、プログラムの探索結果を用いて確率を算出するため、評価関数の性質によって、性能が変化することが考えられる。強いプログラムで求めた探索結果を用いた方がよい結果が得られるのか、あるいは弱くても自分のプログラムの探索結果を用いた方がよい結果が得られるのかといった点について、異なる評価関数を用いた実験などにより、検討していきたいと考えている。

6.3 優れた棋譜を得ることが難しいゲームにおける有効性の評価

提案手法1は、プロの棋譜を用いる必要がないため、ARPSと同様、優れた棋譜が得にくいゲームへの適用も可能である。本論文ではゲームの対象としてコンピュータ将棋を用いたが、それ以外の特にプロの棋譜などの優れた学習データが入手しにくいようなゲームにおける提案手法の有効性についても検証していきたいと考えている。

ARPSでは、単純な反復深化法との比較ではよい結果を得ているが、コンピュータ将棋の全幅探索はここ数年で大きく進歩しており、現在実験を行うと異なる結果が得られることも考えられる。そういったことも踏まえ、提案手法、ARPS、全幅探索の比較を行い各手法の有効性について検証したい。

7. おわりに

本稿では、プログラムの探索結果を利用した実現確率探索を提案した。実験結果から、探索中に得られる情報を特徴として利用することや深さに応じた学習

データから求めた確率を利用するといった改良により実現確率探索の性能を向上させることができることを示した。自己対局や floodgate におけるレーティングの結果では、提案手法は全幅探索や従来の実現確率探索を上回る結果を得ることに成功し、有効な探索手法となり得ることを示した。

探索は評価関数と並びプログラムの強さを決定する重要な要素である。評価関数が機械学習により成功を収めつつあるのに対し、探索はまだ比較的手動の部分が多いといえる。現在、計算機のハードウェアの性能は非常に高性能になっており、コンピュータ将棋の分野でも行うことのできる手法の幅は大きく広がっている。このような背景を活かし、提案手法のような高いマシンパワーを活かしたチューニングを行うことで、探索の性能も大きく改善できる可能性があると考えている。

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) 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, pp.1-17 (2003).
- 3) 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).
- 4) 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).
- 5) 棚瀬寧: 棚瀬将棋の技術背景, 情報処理, Vol.50, No.8, pp.987-992 (2008).
- 6) 金子知適: 最近のコンピュータ将棋の技術背景とGPS将棋, 情報処理, Vol.50, No.8, pp.878-886 (2009).
- 7) 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).
- 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) floodgate, <http://wdoor.c.u-tokyo.ac.jp/shogi/floodgate.html>
- 10) Fan, Rong E., Chang, Kai W., Hsieh, Cho J., Wang, Xiang R. and Lin, Chih J.: LIBLINEAR: A Library for Large Linear Classification, <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>