

モンテカルロ碁で用いるパターンの大きさに関する考察

野口 陽来, 松井 利樹, 橋本 隼一, 橋本 剛

北陸先端科学技術大学院大学

概要

現在, コンピュータ碁碁ではモンテカルロ法を用いた探索法が主流となっている. その中でより強いプログラムを作るためにプレイアウトの精度を上げる研究が盛んに行われている. その中でパターンを用いてプレイアウトの精度を高める方法が, その方法を使ったプログラムが大きな大会で優勝したため注目されている. 本研究では大きさの異なる二つのパターンを棋譜より抽出し, それらのパターンを使用したプレイアウトの性能を比較した. その結果大きいパターンを使用したプレイアウトの方がパターンを使用しないそれより高い性能を示すことがわかった. またパターンはそれに応じた使い方をしなければ効果が出ないことが予想された.

Studies of Pattern Size for Monte Carlo Playout

Haruki Noguchi, Toshiki Matsui, Junichi Hashimoto, Tsuyoshi Hashimoto
Japan Advanced Institute of Science and Technology

Abstract

Today, Monte-Carlo method is mainly used in the domain of computer go. There are many studies that improve the accuracy of the Monte-Carlo playout. In this paper, we extract two set of patterns of different size from game records and compare the performances of these patterns. The experimental results show that the bigger patterns have performance better in the playout. Moreover the performance depend on how the patterns are used.

1. はじめに

現在, コンピュータ碁碁ではほとんどのプログラムがモンテカルロ法 (MC 法) を採用している. これは MC 法を用いることで従来の知識駆動型のプログラムを超えることができる可能性が示唆され[1], また MC 法を用いたプログラム Crazy Stone[2]が Computer Olympiad の碁碁部門 (9 路) で優勝したためである.

MC 法には従来の方法で困難とされている評価関数の作成を行わずにプログラムを作成することができるという特徴がある. MC 法では評価関数の代わりにプレイアウトと呼ばれるランダムなゲームを数多くプレイして局面の統計情報を収集し, 局面を評価する. また MC 法で評価するノードを選択するアルゴリズムとして Upper Confidence Bound applied to Trees (UCT) が使われている. これにより探索と冒険のバランスを調節することができる.

MC 法の改良方法として速度の向上と精度の向上が考えられる. 速度の向上法としては並列化が研究されている[4]. 精度の向上としてパターンを用いる方法[5][6]等が提案されている.

速度の向上による性能の向上はいずれ頭打ちになり, それ以上速度を上げても性能が上がりなくなると考えられる. そこで本研究ではプレイアウトの精度を向上することに焦点を当てる. プレイアウトの精度を向上させることで少ないプレイアウトの試行回数で精度の高い探索が行えることになり, プログラムがより強くなると考えられる.

精度の向上のためにパターンを用いる研究が行われている. パターンを適切に使用することでより精度の高いプレイアウトが得られると考えられている. 本研究ではパターンの大きさを変えて強さに対する影響を調査する.

2. 先行研究

ここではパターンとは限定された狭い領域での特定の石の配置のことをいう. パターンをプレイアウトに使用することでプレイアウトの精度を上げる研究がされている.

パターンの使い方として二通りの方法が研究されている。ローカルな応手を得るために、直前の手の近傍にパターンと一致する石の配置が存在するとき、そこに優先的に着手する方法[5]、グローバルでよい手を見つけるために各パターンに優劣をつけて盤面全体から優れたパターンを探しそこに着手する方法[6]である。前者の研究では3×3のハンドチューニングによって作られたパターンを利用している。後者では1×1～3×3までの大きさのパターンに対して強化学習を行いパターンに優劣をつけている。

3. パターンの大きさ

パターンを使用することでよい着手を選択することができるため、プレイアウトの精度が向上すると考えられる。そこで大きなパターンを使用すればさらにプレイアウトの精度が向上するのではないだろうかと考えた。

使用するパターンとして3×3の8近傍パターンと、5×5の正方形から端の四点を除いた20近傍パターンを使用した。端の点を除いた理由はノビやハネ、一間トビ等と比べてその形の重要度が低いと考えたからである。

4. プログラム

実験には自作の囲碁プログラムを用いた。プレイアウトでは基本的な眼をつぶさないというルールと、UCTではUCB1-TUNED[2]を実装した。UCTにおける反復の速度は約8000回/秒である。パターン等を用いないプログラムをCGOS[7]で走らせて性能を測定したところ、レーティングが1500程度であった。

パターンの使用方法は盤面全体からマッチするものを探し、マッチしたところに打つ確率を高くする方法とした。打たれる確率として棋譜で打たれた頻度をそのまま利用した。

5. 使用したパターン

使用したパターンは棋譜から取得した。棋譜はKGS[8]サーバの19路盤の棋譜をダウンロードして使用した。棋譜の中で直前の手の近傍に打たれた手の周りの石の配置をパターンとして保持しておき、30回以上現れたもののみを使用した。直前の手の近傍のみに限定したのは、相手の手に対して近傍に打つ手は重要度が高い手が多くまた愚形になっている可能性も低いと考えたからである。

6. 性能評価

① パターン

得られたパターンのいくつかを表1、表2に示した。「□」はマッチする空点、「+」は空点、「×」が盤外、「●」が自分の石、「○」が相手の石を表す。ハネやノビ、あたり逃れ等がパターンとして抽出されていることがわかる。

表1：小さいパターン

+++●	+++	+●○	+++	+++
+□○	+□●	+□○	+□●	+□○
+++	+○	+++	+○+	+++
○+++	+++	+++	++●	○●○
●□+	○□+	+□+	+□○	+□●
○+++	●○+	+○	++●	+++
●+○	++●	×××	+++	○●●
○□+	+□○	+□○	●□●	+□○
+++	+●+	++●	+○+	+++

表2：大きいパターン

+++	+++	+++	+++	+++
+++++	++○●+	+++++	+++○○	+++++
++□+○	++□++	++□++	++□●●	++□●+
+++++	+++++	++++○	+++++	++○++
+++	+++	+++	+++	+++
+++	+++	+++	●++	+++
+++++	+++++	+++++	○●+++	×++●+
++□○+	●+□++	++□++	+○□++	×+□○+
+++++	+++++	+○●++	+++++	×++++
+++	○++	+++	+++	+++
+++	+++	+++	+++	+++
+++++	+++++	+++++	+++○●	+●+++
++□○○	++□++	++□++	++□●○	++□++
+++●●	++●○+	●+○++	+++○+	+○●++
+++	++○	+++	+++	+○+

② プレイアウト

各プレイアウトの様子を図1-3に示した。小さいパターンにマッチしたときの確率の増分が小さいので利用したプレイアウトにはあまり影響が見られなかった。大きいパターンではケイマ(5) やツケ(17) 等パターンが利用された痕跡が見られた。

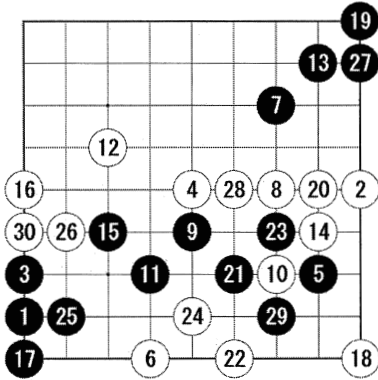


図1：パターンを使用しない

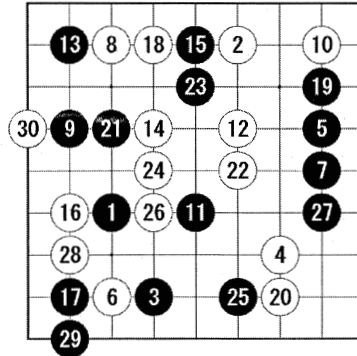


図2：小さいパターン

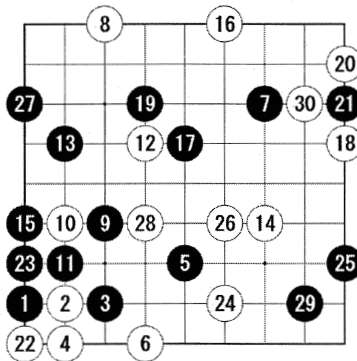


図3：大きいパターン

③ パターンを用いたプレイアウトの勝率

パターンを用いないプログラムとパターンを用いるプログラムを一手あたりの思考時間を等しくして戦わせたときの結果を表3に示す。パターンを使ったほうはパターンの大小問わず負け越している。これはパターンの効果以上に処理の速度が減少しているのが原因と考えられるので、処理速度を同じにしたと仮定して比較した。

図4に速度と勝率の関係を示した。これを用いると速度が14.5%の時の勝率は3.24%、17.4%の時の勝率は5.23%となった。直線から得られた値とサンプルの勝率を比較した。有意確率95%で大きいパターンを使ったほうは強くなっているといえ、小さいパターンを使ったほうは強さが異なるとはいえなかった。

表3：パターンの効果

パターンサイズ	大	小
処理速度[%]	14.5	17.4
ゲーム数	600	233
勝数	74	7

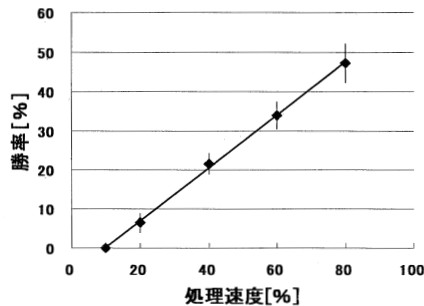


図4：速度と勝率の関係

7. 考察

二種類のパターンの精度の比較を行った結果、大きいパターンでは性能の向上が見られ、小さいパターンでは性能の向上は見られなかった。パターンは同一の棋譜から生成したので似たようなパターンが得られたと考えられる。例えば小さなパターンでノビのパターンが生成されていれば、大きなパターンでも同様のものが得られている。8近傍のパターンではあまりに近くの情報しかないため最終手の近傍で使うわないとあまりにいい加減な手を打ってしまい精度が下がったのかもしれない。

8. まとめと今後の課題

大小二つのパターンを比較した結果、大きいパターンを用いた方がプレイアウトの精度が高くなることがわかった。しかし勝率が高いことを精度が高いと定義したが、なにが原因となり精度が向上したかわかっていない。今後はパターンの何が性能の向上に必要な要素なのかを検討したい。

参考文献

- [1] B. Bouzy and B. Helmstetter, Monte Carlo Go Developments, In E. A. Heinz, H. J. van den Herik, H. Iida, editor, 10th Advances in Computer Games, pp. 159–174, Graz, 2003, Kluwer Academic Publishers.
- [2] R. Coulom, Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search, In P. Ciancarini and H. J. van den Herik, editors, Proceedings of the 5th International Conference on Computers and Games, Turin, Italy, 2006.
- [3] L. Kocsis and Cs. Szepesvari, Bandit Based Monte-Carlo Planning, In 15th European Conference on Machine Learning (ECML), pages 282–293, 2006.
- [4] T. Cazenave and N. Jouandeau, On the Parallelization of UCT, CGW 2007, pp. 93-101, June 2007.
- [5] S. Gelly, Y. Wang, R. Munos, and O. Teytaud, Modification of UCT with Patterns in Monte-Carlo Go, Technical Report 6062, INRIA.
- [6] S. Gelly and D. Silver, Combining Online and Offline Knowledge in UCT, In Zoubin Ghahramani, editor, Proceedings of the International Conference of Machine Learning, ICML 2007, pp. 273-280, 2007.
- [7] Computer Go Server, <http://cgos.boardspace.net/>.
- [8] KGS Go Server, <http://www.gokgs.com/index.shtml>.