

プロの手を真似する囲碁プログラムは強くなるか？

Evaluation of the Method to Create Candidates using Pattern Knowledge Extracted from Professional Player's Scores in Go-Game

清 慎一

SEI, Shinichi

ssei@ssl.fujitsu.co.jp

川嶋俊明

KAWASHIMA, Toshiaki

kawkaw@ssl.fujitsu.co.jp

We had proposed the method to create candidates using pattern knowledge extracted from professional player's scores in Go-game. And then, we implemented the method in Go-program "KATSUNARI". KATSUNARI could create many good candidates, however the result in computer Go championship was not good. The reason was that KATSUNARI created not only good candidates but also undesirable candidates and it sometimes selected undesirable one as next move. We found that to choose good one from many candidates is very important to make strong program using this method. In this paper, we report about evaluation of the method.

1 はじめに

強い囲碁プログラムを作ることができない理由の一つに、「人間の思考方法が解明できないので良い候補手を生成することが難しい」という問題がある。この問題を解決する方法として、プロ棋士の実際の対局に現れた石の配置を大量に記憶し、その記憶したデータを直接利用して候補手を求める方法を考案し [3]、囲碁プログラム「勝也」を試作した [4]。この方式を用いることで、人間がどのように候補手を生成するのかを解析して、それを実現するためのアルゴリズムを考えプログラム化するという、大変手間のかかる作業をすることなく、良い候補手を生成することができた。この方式の効果をさらに調査するために、囲碁プログラムの大会である「囲碁ソフト選手権 (FOST 杯)」に出場したが、良い成績を残すことはできなかった。そこで、なぜ良い候補手を生成したのに成績が良くなかったのかを調査したので、ここに報告する。

2 大量の形データからの候補手生成

囲碁において、着手を決定する際には、好形になる手を選ぶことが大切であり、多くの場合局所的な石の配置から次の一手を決めている。上級者が詰碁において、局所的な石の配置から候補手を生成していたという報告もある [6]。そこで、プロ棋士が実際に着手した手とその周辺の石の配置を大量に記憶しておき、「次の一手」を求める際にその記憶したデータと最も似ている手を選べば、強くなるのではないかという発想のもと、囲碁プログラム「勝也」を開発した。

2.1 記憶に基づく推論

記憶したデータと最も似ている手を選び候補手とする方式として、記憶に基づく推論 (Memory-Based Reasoning)[5] を利用した。

記憶に基づく推論のプロセスは以下の通りである。

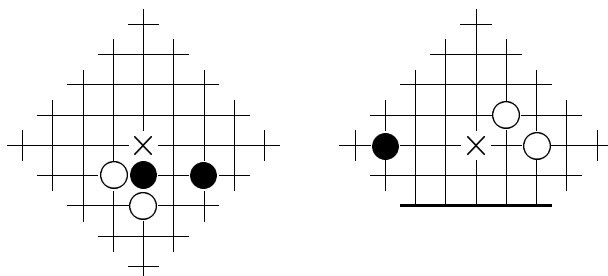
1. 過去に起きた事例を収集しデータベースに保存する
2. データベースから、与えられた問題と最も似ている事例を求める
3. その事例に書かれている解を、問題の回答として出力する

この方式は、問題の解決方法が確立されていない分野について有効な解決方式として知られており、単語の発音（与えられた文字列をどう発音するか）[5] や機械翻訳などへの応用が研究されている [1][2]。

2.2 データベース

プロ棋士が実際の対局で打った手とその周囲の石の配置を事例として、データベースに記憶した。データは主に、トッププロ同士の対局による棋譜から収集した¹。本来は、盤面全体を一つのデータとして記憶すべきなのだが、盤面全体を記憶するには碁盤は広過ぎて、記憶量・検索時間から考えて、現在のコンピュータには適当でない。また、人間の上級者が持っている知識も盤面全体ではないと考えられるので、盤面の一部分を切り出して一つのデータとした。

記憶する一つ一つのデータの大きさや形状は図のように、着手位置を中心とするマンハッタン距離4の菱形とした(図1)。囲碁用語における大ゲイマや三間にある石の距離が、マンハッタン距離で4になるところから、妥当な範囲ではないかと考えたからである。



× : 黒番の次の一手の位置

図 1: 記憶したデータの形状

2.3 候補手生成

ある局面において、勝也が記憶に基づく推論を使って候補手を生成するプロセスは次の通りである。

¹最初は、上級者が記憶している好形を全部列挙してもらおうとしたが、それは大変な作業な上、人によって覚えている形や数に違いがあり、どの形を好形とすべきか判断が難しいので、この方法は採用しなかった。

1. 対象となる局面におけるすべての打着可能点について、その点を中心とする菱形を取り出す
2. その菱形と、データベース内にあるすべてのデータとの類似度を求める
3. ある閾値以上の類似度を持つデータが存在したら、盤上のその座標（抽出した菱形の中心の座標）を、この局面における候補手とする

ここで、どのような石の配置を「似ている」と定義するかという問題がある。我々は、菱形内の各点について、その点の状態（白か黒か空点か盤外か）の類似度と、その点が菱形の中心からどの位離れているかを基に算出した。点の状態の類似度は、同色の石（または空点同士、盤外同士）なら同じに、黒と白なら大きく異なり、黒（または白）と空点なら少し異なり、盤外と盤上の点（黒／白／空点）なら非常に大きく異なる、という具合に定義した。また、その点が菱形の中心からどの位離れているかは、単純に中心からの距離ではなく、その菱形の中心に最も近い石までの距離も考慮に入れて数値を調整している。中心に近いところに石がある場合は、遠い石の存在をあまり考慮する必要はないと考えられるので、それを反映させるためである。

2.4 候補手例

勝也によって求まった候補手の例を図2に示す。使ったデータベースの規模は、プロの棋譜200局から収集した約30,000データである。類似度の閾値は、菱形の中心から距離1に石がある場合は、中心から距離2にある石まで完全に同じで、距離3にある点は2箇所の配置が異なる程度に設定した。菱形の中心から距離1に石がない場合は、菱形内の石の配置がほぼ同じになるように設定した。また、菱形内に石が全く存在しない場合は候補手から除いた。

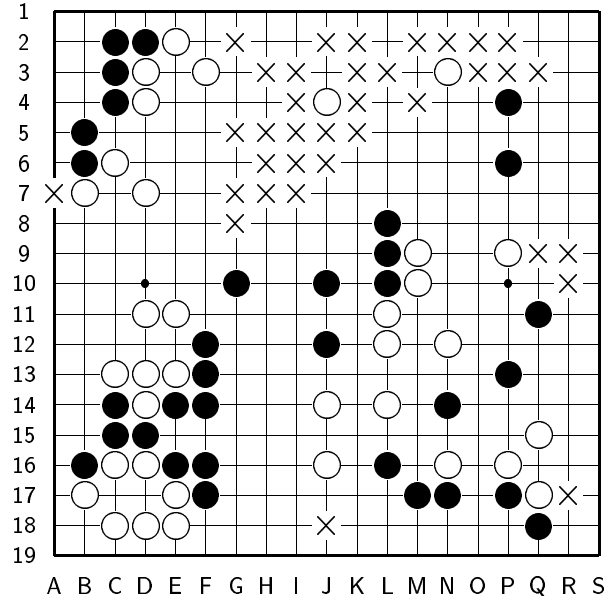


図2: プログラムが求めた候補手の例 (黒番)

筆者（アマチュア四段）の主観ではあるが、図2を見たところ、上辺への模様を消す手、敵の石を攻める手、自分の地を増やす手など、良い手がたくさん候補手として挙がっているように見える。この対局でプロが実際に着手した手 (R,17) も、候補手として挙げられた。

3 評価

本方式で着手を生成した囲碁プログラム「勝也」の囲碁ソフト選手権での成績は、決して良いとは言えない。初めて参加した1996年は4勝5敗で参加19チーム中13位、プログラムに改良を加えて参加した1997年は5勝5敗で参加38チーム中20位であった。良くない成績の理由がどこにあるのかを本章で検討する。

3.1 良い手が網羅されているか

図2において、すぐに打つべき手ではないが、両ノゾキ(K,15)や、白石の連絡を妨げる手(M,15)のように、重要な手が幾つか候補手として挙がっていないことがわかる。この理由として、収集したデータが少ないのではないかと考えられる。まずは、データ数により求まる候補手数の違いについて調査した。

データベースの作成に使用しなかったプロの棋譜から、最後まで打ち続けた(投了しなかった)ものを選び、その手数を六等分した等分点の局面で、候補手生成を行なった。求めた候補手数の平均を表1に示す。またデータベースが大きくなると、候補手数がどのくらい増えるかを示したのが表2である。ちなみに、総手数を六等分した平均値は約39手であった。

表 1: データベースの規模と候補手数

局面の進行度	1/6	2/6	3/6	4/6	5/6
約 40,000 データ	105	63	32	19	11
約 20,000 データ	86	47	23	13	6
約 10,000 データ	66	33	17	9	3

表 2: データベースの規模と候補手数の増加率

局面の進行度	1/6	2/6	3/6	4/6	5/6
40,000 データ / 20,000 データ	1.23	1.36	1.36	1.44	1.69
20,000 データ / 10,000 データ	1.30	1.42	1.36	1.55	1.88

表1から、データ数が40,000もあると、序盤では十分過ぎる(100以上!)ほどの候補手が生成されていることがわかった。しかしながら、対局が進むにつれ候補手数は激減し、終盤では候補手がほとんど出なくなることがわかった。

表2において、データ数が10,000から20,000になる場合と、20,000から40,000になる場合の候補手数の増加率にあまり差がないことから、収集するデータ数を増やすと、さらに候補手数は増え続けることが予想される。これは、40,000データでは、まだプロが打った手は網羅されていないことを示している。

プロが実際に打った手が候補手数の中に存在した割合を表3に示す。この表からも、収集したデータ数が40,000では、不十分ではないかと考えられる。

表 3: プロが実際に打った手が全候補手の中に存在した割合

局面の進行度	1/6	2/6	3/6	4/6	5/6
約 40,000 データ	70 %	60 %	40 %	30 %	10 %
約 20,000 データ	70 %	40 %	40 %	30 %	10 %
約 10,000 データ	70 %	30 %	30 %	30 %	10 %

3.2 どのような手が候補手として求まりにくいのか

次に、プロのどの着手が、候補手として挙げられなかったのかを具体的に調査した。プロが打った手と、プログラムが求めた候補手位置を図 3 に示す。プログラムが求めた候補手には×を、プロ棋士が打った手の位置を P で示す。実験に使ったデータ数は約 40,000 である。

- 石の捕獲／逃亡の手 (図 3左)

調査した中で、プロの着手位置と一致しなかった手のうち、最も多かったのはこの手である。石が少ない場合は形良く捕獲（逃亡）することは重要である。しかしこの手は、ダメが詰まってくるに従い、単にダメを減らす（増やす）手となる。そのため、好形を作る手とは無関係になり、本方式では求めにくい手と考えられる。

- 先手で効かす手 (図 3右)

形の急所であるので、本方式で求まっても良さそうな手である。実験では、このような先手で効かす手も候補手になることもあったが、石が混み合ってくると候補手にならなかった。石が多いと、データベースに存在しない確率が高いためと考えられる。

- 終盤のヨセの手 (図 4左)

終盤は石が混み合っているため、これもデータベースに存在しないものが多かったと考えられる。また、この手は一線のハネツギのような手筋と異なり、好形とは無縁の手であるため、本方式では求めにくいと考えられる。

- 切りの手 (図 4右)

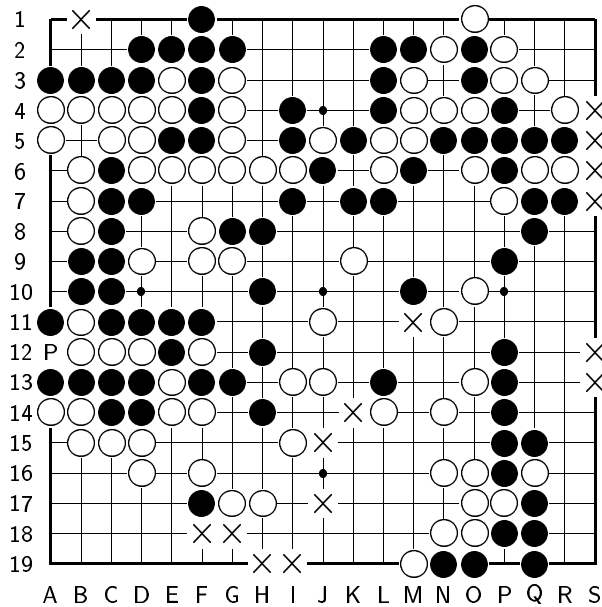
切りやツナギの手は重要であるが、キリやツナギの手が存在するのは、白黒両者の石が集中している場所である。これも、石が混み合っているため、データベースに存在しなかったと考えられる。

3.3 悪い手が出ていないか

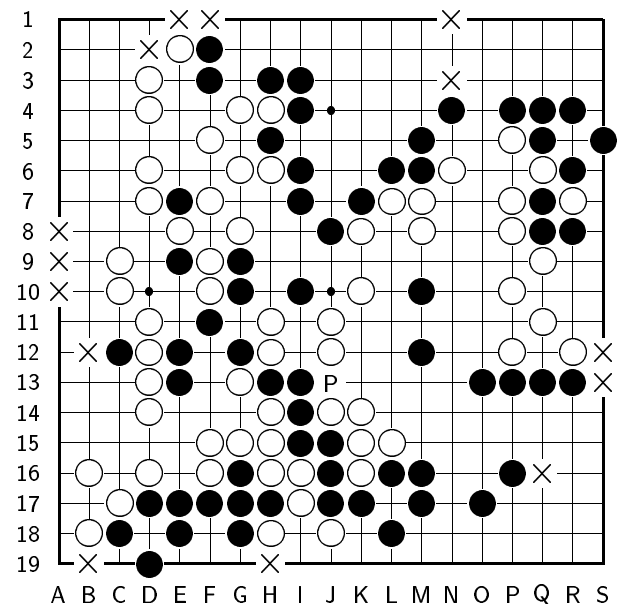
良い候補手がたくさん生成されても、悪い候補手が生成され、その悪い手が次の一手に選ばれてしまっても意味がない。そこで、悪い手が候補手として挙げられていないかを調査した。図 5、典型的な悪手の例を示す。これはいずれも、1996 年の FOST 杯での勝也の実戦譜である。

- 価値の低い手 (図 5左)

黒 55 は二線のハサミツケという典型的な手筋であるが、この局面では価値が低い。そのうえ後手である。



石の捕獲／逃亡の手 (黒番)



先手で効かず手 (黒番)

図 3: プロが打った手が出なかった例 1

- 石の生死を考慮していない手 (図 5右)

白 42 のノビは形の良い手であるが、白が全滅しかねない危険な手である（実戦では白が死んだ）。生死を考慮した上での良い形でなければ、意味がない。

- 手筋の途中で方向転換 (図 6左)

白 36 は三線の二間開きの内側にツケという、よく見かける手筋である。しかし、黒が二線にハネた後に (R,11) と切るべきところを (P,12) とノビてしまった。候補手には、(R,11) も出ていたのだが、(P,12) の価値より低いと判断された。手筋の途中で方向転換するぐらいなら、始めから白 36 とツケない方がよい。

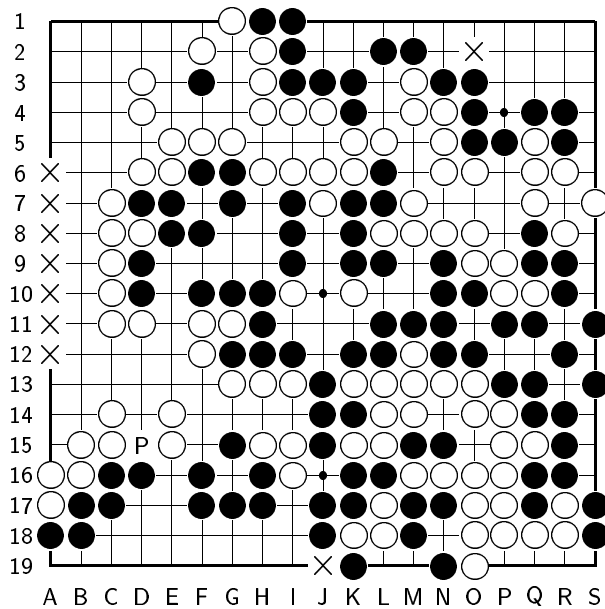
- 相場がわからず欲張りな手 (図 6右)

黒 35 からの一連の手は、白 32 を捕獲できれば大成果だが、それは不可能である。上級者なら、右辺を割ったことで満足すべきところである。局所的に見ると、黒 31 のツケ、黒 33 のオサエ、黒 35 の切り、黒 37 と黒 39 のオサエ、すべて良い手なのだが、全部併せると共倒れになってしまう手である。

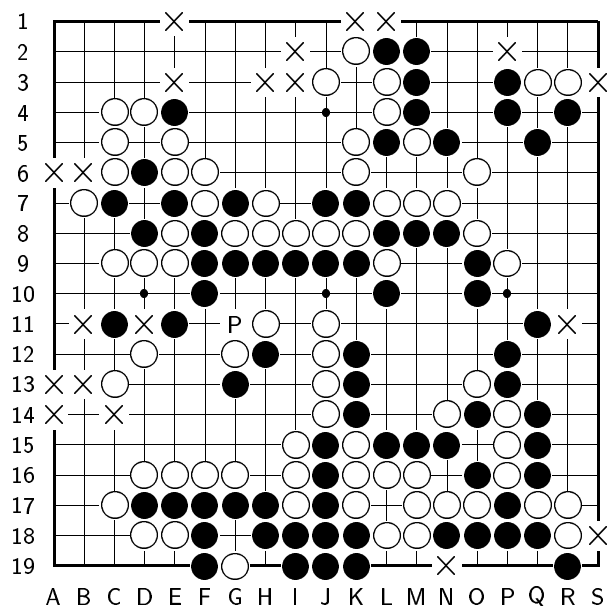
- 中手 (図 3左)

図 3 は実戦譜ではないが、(B,1) に中手のような手が出ている。「2 の 1」は死活の急所だが、この黒は右辺につながっているので、中手にならない。この手は全く無駄な一着である。

- ヨセ (図 3右)



終盤のヨセの手 (黒番)



切りの手 (黒番)

図 4: プロが打った手が出なかった例 2

左辺の一線に候補手が出ている。これは、ヨセにおいて一線へ飛び込む手がデータが存在したためである。しかし、この局面では近くに味方が存在せず、ヨセになっていない。

悪い手とは言っても、どれも部分的には意味のある形の手であった。しかし、その手を選んだタイミングが悪かったり、他の手と効果を打ち消し合ったりしたために、悪手になっている。局所的な石の配置を調べただけでは、必ずしも良い手ばかりが挙げられるわけではないようである。

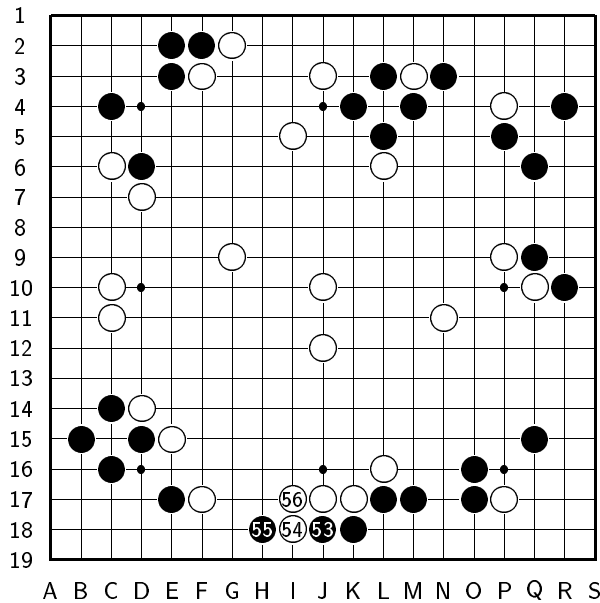
また、いずれの図でも、もっと良い手も候補手として挙がっていた。悪い手を出さないことも問題だが、良い手を選ばなかったことも問題であることがわかった。

3.4 相手のおかしな手をうまく咎める手は出ているか

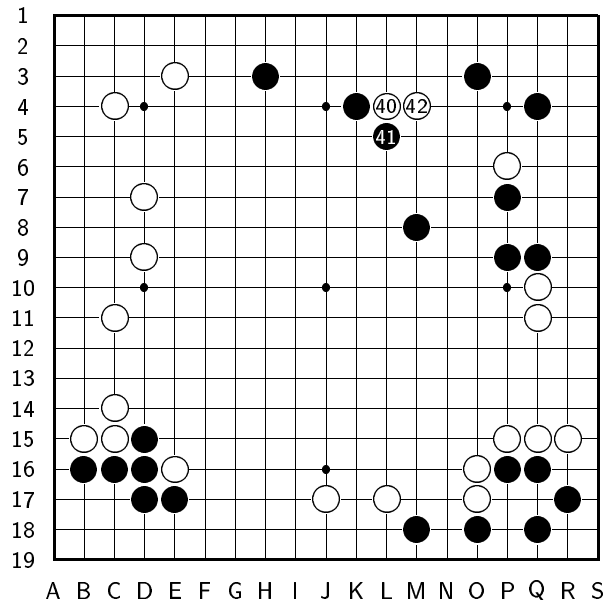
FOST 杯に参加したプログラムは、最も強いものでも約 5 級、弱いものは初心者程度の強さしかない。勝也はプロの対局からデータを抽出しているので、弱い者が打つおかしな手をうまく咎める手が出ていないために、FOST 杯での成績が悪かったことも考えられる。なお、ここで言う「おかしな手」とは形の悪い手のことであり、他により良い手があるために価値が低いという意味ではない。

図 7 は FOST 杯で勝也の実戦から取り出した、相手のおかしな手に対する勝也の候補手の位置である。左図の黒 23 に対して、勝也は (E,15) と正しく咎める手が生成された。右図の黒 19 に対しても、(O,16) と正しく咎める手が生成された。

なお、これらの手と同形のデータは、データベースには存在しなかった。類似した形がデータベースに存在したために、生成された手であった。類似した形がデータベースに存在した理由は、「たまたま、プロがおかしな手を打って、その手を咎める手が収集された」のではなく、「プロは



価値の低い手 (黒番)



石の生死を考慮していない手 (白番)

図 5: 悪い手が出た例 1

相手の石が無意味になるように打つので、その結果、相手の石がおかしな配置になった図が収集された」と考えられる。プロの対局から集めたデータでも、おかしな手を正しく咎めることができる場合があることがわかった。

しかしながら、実戦で勝也はどちらの手も打たなかった（勝也の打った手の位置は K）。咎める手がないのではなく、咎める手を選ばなかったのが、悪い成績の原因の一つのようである。

4 まとめ

プロ棋士の実際の対局に現れた石の配置を大量に記憶し、その記憶したデータを直接利用して候補手を求める方式を採用したプログラムが、なぜ強くないのかを知るために候補手の性質を調べた。その結果をまとめると以下ようになる。

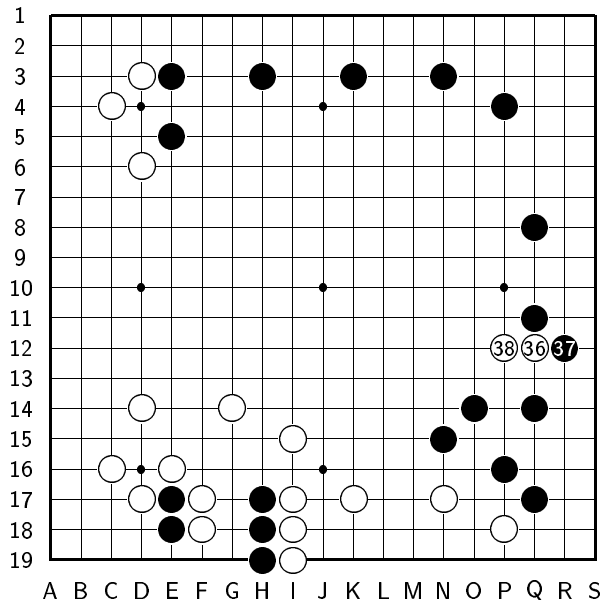
- 良い候補手が出ていないことがある

40,000 データでは量が少ないため、良い手が網羅されていない。特に石がたくさんあるデータでは、良い手のデータが存在する確率が低くなる。

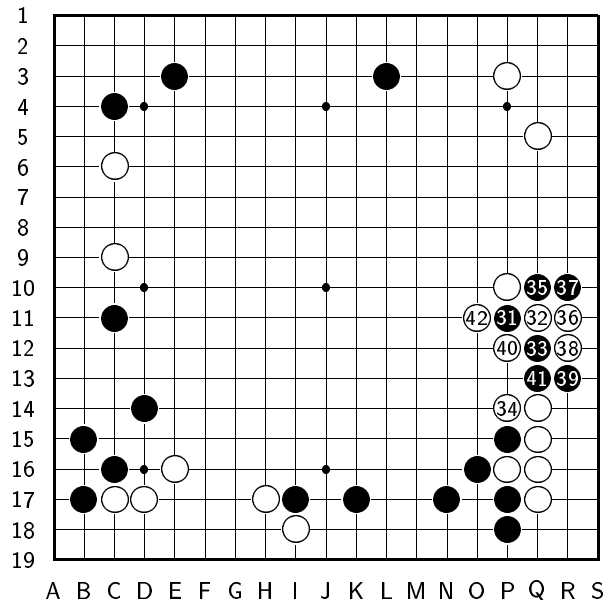
- 悪い候補手が出ていることがある

部分的には良いが、全局的に見ると良くない手というのが存在した。その上、良い候補手が挙がっていても、価値の低い手を選ばれることがあった。

本方式には「良い候補手を生成できる」という利点はあるが、すべての良い候補手を生成するためには、まだ問題があることがわかった。今後は、データ数を増やして実験を行ったり、収集



手筋の途中で方向転換 (白番)



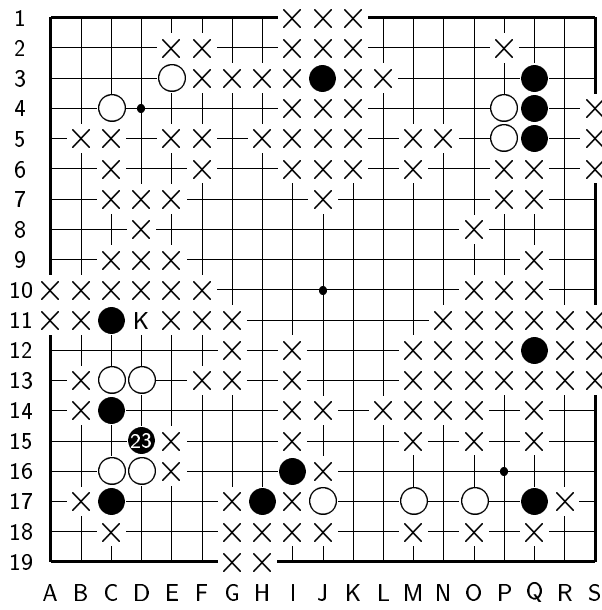
相場がわからず欲張りな手 (黒番)

図 6: 悪い手が出た例 2

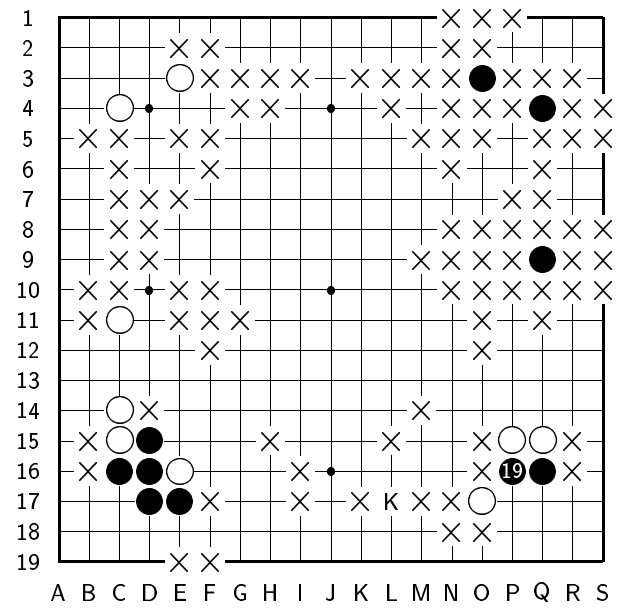
するデータの形状やサイズ (マンハッタン距離 4 の菱形) についての検討も行なってみたい。また、悪い候補手も生成されてしまうこともわかったので、候補手の中から、どのようにして良い手を選択するかという問題も解決しなければならない。

参考文献

- [1] Kitano, H. : A Comprehensive and Practical Model of Memory-Based Machine Translation. Proc. of 13rd International Joint Conference on Artificial Intelligence, pp.1276-1282, 1993
- [2] Sato, S. : Example-Based Translation of Technical Terms. 5th Int. Conf. on Theoretical and Methodological Issues in Machine Translation, 1993
- [3] 清 慎一, 川嶋俊明 : 「局所パターン」知識主導型の囲碁プログラムの試み, 第 1 回ゲームプログラミングワークショップ, pp.97-104, 1994
- [4] 清 慎一, 川嶋俊明 : 記憶に基づく推論を使った囲碁プログラム「勝也」の試作, 第 3 回ゲームプログラミングワークショップ, pp.115-122, 1996
- [5] Stanfill, C and Waltz, D. : Toward Memory-Based Reasoning., Communications of the ACM, Vol.29, 1986
- [6] 吉川 厚, 斉藤康己 : 先読みをしないと詰め碁は解けない?, 第 3 回ゲームプログラミングワークショップ, pp.76-83, 1996



Gogol(黒) vs 勝也 (白)



Go Intellect(黒) vs 勝也 (白)

図 7: 相手の悪手に対する候補手