

Title	コンピュータ指導碁のための悪手解説
Author(s)	山中, 翠; ビエノ, シモン; 池田, 心
Citation	研究報告ゲーム情報学 (GI), 2016-GI-35(5): 1-8
Issue Date	2016-03-01
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/14080
Rights	<p>社団法人 情報処理学会, 山中翠, ビエノシモン, 池田心, 研究報告ゲーム情報学 (GI), 2016-GI-35(5), 2016, 1-8. ここに掲載した著作物の利用に関する注意: 本著作物の著作権は(社)情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。 Notice for the use of this material: The copyright of this material is retained by the Information Processing Society of Japan (IPSJ). This material is published on this web site with the agreement of the author (s) and the IPSJ. Please be complied with Copyright Law of Japan and the Code of Ethics of the IPSJ if any users wish to reproduce, make derivative work, distribute or make available to the public any part or whole thereof. All Rights Reserved, Copyright © Information Processing Society of Japan.</p>
Description	情報処理学会 第35回ゲーム情報学研究会

コンピュータ指導基のための悪手解説

山中翠^{†1} ビエノシモン^{†1} 池田心^{†1}

概要: 現在、囲碁プログラムは一般の人にとってほぼ十分な強さあり、強さ以外の面に着目した研究も行われ始めている。例えば上達を目的とした「指導基」については、強いプレイヤーなら誰でも適切に指導できるという訳ではなく、専門の指導者にかかれば費用もかかるため、これをプログラムで行えば高い価値がある。指導基プログラムには多くの技術が必要である。これは楽しませることが目的の「接待基」プログラムに必要な技術と共通する点も多いが、悪手を指摘して解説する必要があるという点が大きく異なると我々は考えた。そこで本研究では、指導基プログラムへの第一歩として、指導基とりわけ悪手の解説について考察し、それを実現するのに必要な要素やとりうるアプローチをまとめることを目標とする。その上で、モンテカルロ木探索による囲碁プログラムの機能を援用しやすいものについて実装し、プロトタイプとして示す。

キーワード: 指導基, モンテカルロ木探索, 想定図, 悪手解説

Bad Move Explanation for Teaching Games with a Go Program

MIDORI YAMANAKA^{†1} VIENNOT SIMON^{†1} KOKOLO IKEDA^{†1}

1. はじめに

ゲーム情報学研究では、強いコンピュータゲームプレイヤー (AI)、人を楽しませる AI、人に教える AI などが目指されている。どんな目的にも一定の強さは必要なことからまずは強さが主に求められてきたが、チェスでは 1997 年に DeepBlue が世界チャンピオンを破り、2010 年代前半には将棋でも上位プロと互角以上の強さを持つようになり、さらに 2015 年囲碁においても AlphaGo がプロ棋士を破るなど [1]、これら古典的ボードゲームでは AI の強さは一般プレイヤーには十分なものになりつつある。これにより、楽しませる AI や教える AI に取り組む環境ができつつあるともいえる、そのような研究も増えてきている [2][3][4]。

我々は、囲碁における“教える AI”として、特殊な囲碁対局形式である“指導基”を打てるような AI を作成することを目指す。指導基は通常人間の上級者が下位の者に対して教育目的で行う対局であるが、単に囲碁が強いだけでは適切な指導ができないことを強調したい。指導法の技術的な面のみならず、家族間では感情的になることが多いなど、初級者中級者にとって適切な指導者を身近に見つけることは難しい。専門の指導者にかかり上質な指導を受けることは可能だが、大抵は 1 局数千円程度の費用がかかり、移動の手間も小さくはない。もしアマチュア指導の一部でもコンピュータプログラムが担うことができれば、囲碁普及のためにも価値は高いと考える。

本研究ではまず、指導の専門家が実際どのように指導を行い何に気を付けているのかを聞き取り調査し、そこから

何が囲碁プログラムに求められるのかを見定める。そしてその結果明らかになった“悪手の指摘と解説”の必要性および実装可能性について論じる。人に教える指導基、あるいは人を楽しませる接待基は、非常に大きく遠い目標である。本論文ではそのための第一歩として、悪手解説が「悪手検出」「悪手分類」「結果説明」「想定図作成」の 4 つの段階に分けられること、そのそれぞれに標準的な囲碁プログラムの機能が流用できるであろうことを述べ、いくつかの例をプロトタイプとして示す。

2. 関連研究

チェス・囲碁・将棋といった古典的ボードゲームで強い AI を作る場合、完全情報性から木探索が主に用いられる。さらに上級者棋譜の入手が容易なことから状態 (局面) 評価関数や行動 (着手) 評価関数を機械学習によって調整し木探索に援用することが一般的である。詳しい発展の歴史は省略するが、これら用いられる手法は多くの場合教える AI や解説などにも援用される。

将棋では、コンピュータプレイヤーの進歩が囲碁よりも早かったこともありいくつかの解説に関する取組みがなされている。亀甲らは将棋の対戦局面の解説文を、機械学習と自然言語処理の組み合わせにより自動生成する高度な取り組みを行っている [4]。さらに中盤戦では数手先についての解説が行われていることを踏まえ、それがどの手順を表しているのか候補を列挙し、局面評価値等で選択するという発展を行っている [5]。石脇らは将棋の部分問題である詰将棋について、正解手順だけでなく不正解手順を解説する試みを行っている [6]。解説文はテンプレートにより作成し、また初心者にとっての「一見良い手」を定義することで不正解手順を選択する工夫を行っている。

^{†1} 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
kokolo@jaist.ac.jp

著者らは、人間を楽しませるための“接待碁”を AI に打たせるための試みを行い、接待碁では「相手の好みを推測する」「不自然な手を打たずに形勢を制御して負けてあげる」といったことが必要であると主張している(図1)。その上でそのいくつかについて、モンテカルロ木探索の着手選択部やシミュレーション部末端に簡単な変更を加えることで有望な結果が得られることを示した[3]。また、将棋では「▲2一銀」などと座標のみで着手が表せるのと同じに、囲碁では「D6ノビ」などと形の名前を表す言葉が用いられることを踏まえ、形の名前の決定ルールを機械学習により定める取り組みを行い、上級者レベルの名前付けに成功している[10]。

これらを踏まえ、本研究では指導碁や悪手解説の特性を考慮したうえで、技術の適用可能性を考察していく。

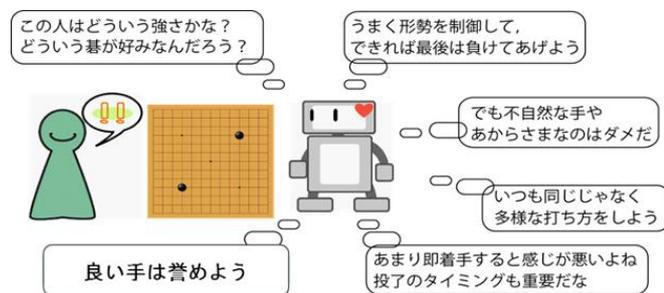


図1： 接待碁の6つの要素

3. 指導碁と悪手指摘

本研究ではまず、日本棋院および神奈川の碁会所に赴き、囲碁教室や指導碁においてどのような指導が行われ、専門の指導者が何に気を付けて指導しているかを調査した。これと合わせ、市販の棋書(囲碁の参考書・問題集)についても調査し、関連性を調べた。

【指導内容】

アマチュア囲碁プレイヤーの棋力には大きな幅があり、適切な指導内容も大きく異なる。入門から初級のプレイヤーへの指導としては教室形式(1対多)での解説が頻繁に行われ、概念や考え方の説明がなされていた。囲碁は目的やその達成法が分かりにくいゲームなので、対戦等による自己訓練の前にこれらは必須のようである。

教室形式の指導では他にも、詰め碁が提示され生徒がそれぞれ取り組んだあと正解不正解の解説がなされるものや、定石の変化や使い方の解説、棋譜が大盤解説されるものも頻繁に行われる。プロの棋譜を解説するときには変化図を示しながら「正しい考え方」のお手本としており、アマや生徒の棋譜を解説するときには(好ましい手は褒めつつ)悪かった手を指摘し、その理由やどうすればよかったかを説明している。詰め碁やプロ棋譜の解説は概ね中上級者向けであった。

囲碁教室でも対戦は頻繁に行われる。指導者と1対1または1対数人の“指導碁”を行うこともあれば、生徒同士が打ちそれを指導者が巡回してコメントすること、あるいは棋譜に後日添削が行われるような場合もあった。多くの場合指導碁は生徒にとって最も期待する指導であり、一般論ではなく自分の打った手に対しての具体的なコメントがもらえる貴重な機会である。

【指導碁】

指導碁もまた生徒の棋力によって形式は異なる。上級者に対しては適切なハンデを与える他は全く通常の対戦のように打つというプロもいる一方で、多くの場合は何らかの手加減や誘導が行われるようである。つまり、生徒が多少悪手を打ったとしても何度か勝ちへのチャンスを与えたり、指導に適した難しすぎず簡単すぎない局面に導く、あるいはいつもと違う状況に導くということである。何度も指導を担当している生徒に対してはその得手不得手を把握できているため、弱点の補強になるような誘導も行われていた。

指導碁で指導者側に心がけられている内容は、接待碁に求められる内容と重複する部分も多かった。これは囲碁を続けてもらおうと思っただけでなく楽しんでもらうことが大事であることを考えれば自然なことである。しかし異なる点も以下のようにいくつか存在した。

- 手加減するにしてもお手本にならないような(真似されては困る)手は控えるべきであること
- そのレベルで理解できるように悪い手を生徒が打った場合には、それを見逃さずに生じるべき結果を生じさせ負の報酬を与えること
- さらにそれを指摘して解説すること

【悪手の指摘】

悪い手を生徒が打った場合の解説も棋力により形式は異なる。上級者では指摘は局後に行われるが、中級者程度までは、着手された時点あるいはせいぜい数手進んだ時点で指摘が行われることが多い。これはある程度強くならなければ途中の局面や自分が着手した理由を覚えていないからである。ゲームの進行に致命的な間違いの場合は初級者ならばその場所から打ち直すなどもしばしば見受けられた。

悪手の指摘では、単に「この手が悪手」「ここに打つべき」という情報だけでなく、「どうしてその手が悪いのか(理由)」「その結果どうなってしまったのか(結果)」「どう考え、どう打つべきだったのか(正解)」「そうすればどういう進行になると予想されるのか(想定図・変化図)」などが示されていた。こういった情報は大変有益であるため、生徒の手がさほど悪くなくとも、有力な対案を示すこともしばしば行われていた。

【棋書】

市販の囲碁の本にも、入門書、戦略指南、定石解説、棋

譜解説、詰碁集、問題集などさまざまなものが存在し、これは前述の指導内容と対応する部分も多い。多くの場合少なくとも一部には問題形式のページが含まれ、そこでは対象レベルに合わせた正解図、変化図、失敗図などとともに理由や結果の説明がなされていた。

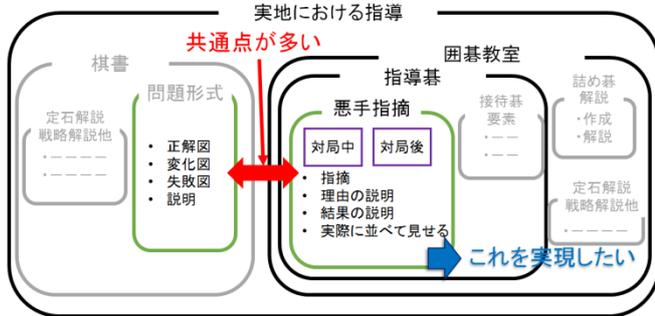


図2：調査結果の概略

【実現イメージ】

以上の調査結果を踏まえ、我々はコンピュータ囲碁プログラムが代替できる部分として指導碁を行うこと、中でも悪手解説を想定図を用いて行うことを中心課題とする(図2)。教室形式の指導は1対多で行えばビデオ等でも代替できるのに比べ指導碁は1人1人1局1局に対応しなければいけないこと、指導碁の要素は接待碁の要素と重なる部分が多いが悪手解説が大きな違いであることがその理由である。将来的には教育的詰め碁の自動生成なども試みたいが、想定図の作成には本研究と同様の技術が流用できると考える。

図3は本研究の実現イメージである。まず人間プレイヤーの着手の中から悪手を検出し(1.)、それを分類して説明し(2.)、どうなってしまったかを図を用いて説明し(3.)、どう打てば良かったかを説明する(4.)。

4. 悪手の定義と種類

一口に悪手と言っても、その定義を明確に行うことは困難である。min-max 原理により定まるはずの最善手集合以外を悪手と呼ぶことも可能だろうが、それは現実的な判別や中級者程度への指摘としては全く適当ではない。人間の指導者は生徒のレベルに合わせ、説明がしやすく理解できるもの、程度のひどいものなどを選んで解説しているはずである。

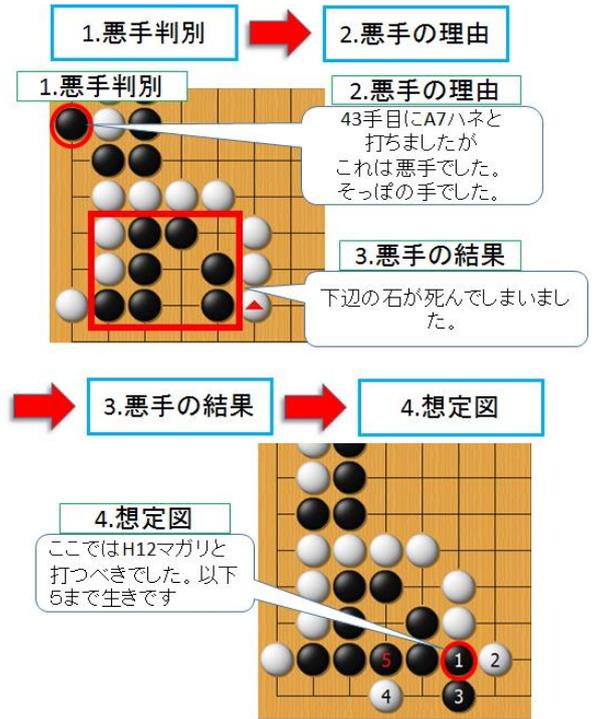


図3：悪手指摘のイメージ

【コンピュータによる悪手の検出】

将棋では、前述の亀甲らの論文[5]や山下の論文[7]において悪手の概念が扱われており、強いAIが計算した局面評価値を参照している。これらの研究は悪手であるかどうか・悪手を指摘すべきかどうかという2値の問題を直接扱っているわけではないが、検出への応用も可能だろう。

比べて囲碁の場合には局面の評価として、勝率と地合いという全く異なる2つの指標があることに注意しなければならない。例えば、ほぼ100%勝つ終盤の局面であっても、12目勝ちを9目勝ちにしてしまうような手は、指導碁では悪手として指摘する可能性がある。あるいは、たとえ影響が微小であっても、「明らかに悪い形であり、近くに良い形の手がある」場合には指摘する可能性が高い。

この検出の問題はいずれは、棋力・勝率の変動・地合いの変動・着手の形の良さ等を入力とし、指摘の有無を出力とするような機械学習によって解決されるべきであろうが、十分な数のデータ収集には大きなコストが必要だろう。

【悪手の種類】

悪手は「最善手と比べある程度悪い」という以外にも、何らかの理由付け・種類分けが可能である。我々のアプローチ(図3)では理由付けと結果は別個に扱うことを想定しているが、結果も含めての分類になってしまうものも多い。以下、検出の難しさや指導の高度さを考慮せずに、聞き取り調査や棋書調査の中で頻繁に出てきた悪手の種類を列挙する。

- 形の悪い手, 筋違いの手. 主に局所的に, 良い手と比べて石の働きの乏しい手.
- 弱い自分の石を助けなかった手, 弱い相手の石を攻撃しなかった手.
- 逆に, 強い自分の石を補強したり, 取れている相手の石に無駄な手入れをした手.
- 味消しの手, コウ材を消した手. これらを検出するのは比較的困難だと想像される.
- 手拍子の手. 相手の着手の近くの一見形の良い手が実際には悪い結果を導く手.
- そっぽの手. 相手の着手に応対すべきだったのにしないで悪い結果を招く手.
- おつきあいの手. そっぽとは逆に, 相手の着手に無駄に対応してしまった手. 関連して, 先手を取られた手.
- 定石間違いの手.
- 方向間違いの手. 部分的にはありうる2つの近い手のうち, 大局的には重視する方向を誤った手.
- 大場を見過ごした手, 急場を見過ごした手.
- 無謀な侵入をした手.
- 正しく読み切れなかった手. 局所的な戦闘や攻め合いで, 先読みを途中で失敗した手.
- 部分的に損をした手. ヨセに多い.

5. 理由説明と結果説明

悪手に対する解説を自然言語で行うことは, もしかしたら亀甲らのもの[4]と類似した方法で可能かもしれない. 局面の特徴量と解説文を用いた機械学習の代わりに, 局面および着手の特徴量と悪手指摘文を用いた機械学習を行うことは原理的には可能である. しかし学習データの入手難さや, 「この石が死んだ」「ここに地模様できた」などの盤面上での説明のため, まずはルールベースで限られた悪手種類・悪手結果を判別し, テンプレートに流し込んで説明文を作成するアプローチを取る.

【有用な囲碁プログラムの機能】

まず, 判別や説明のために有用であろう, 囲碁プログラムの入出力関数を示す. 囲碁プログラムは原則的には「ある盤面に対して一手を返す」機能だけを持っていいが, 実際にはそれは「一見した手の良さを求める[8]」「ある場所が黒地になるか白地になるか予測する」などのより粒度の細かい様々な機能の組み合わせで実装されているはずである. ここでは, 囲碁プログラム Nomitan[9]にもともと備わっていた, あるいは今回のために拡張・追加した要素的な機能を簡単に紹介する. 2010年代の標準的なモンテカルロ木探索プログラムではこれらの多くは容易に追加できると予測する. 以下の関数で Pos は位置クラス, Board は局面クラスを表す.

- Pos getBestMove(Board b): 局面 b における最善手を求める関数. ただし質はプログラムによるため厳密には最善手とは言えない, あくまで疑似的なものである.
- double getDistance(Pos p1, Pos p2): 2つの着点の距離を求める. 石の連なりを考慮するなどさまざまな距離が考えられるが, 今回は単純に Euclid 距離とした.
- double getWinratio(Board b, Pos p): 局面 b で着手 p を打った場合の勝率を求める関数. 実際には, 予測地合い差を返すような関数も必要だろう.
- double[] getSelectProbs(Board b): 局面 b でどの場所が一見打ちたいのか, 確率分布を求める関数. 機械学習で求めて用いるが, AlphaGo のように高度なモデルを用いた場合[1]もはや“一見”ではないような手に高い確率が振られる可能性があることは注意したい. 図4は論文[9]にある例である.
- double[] getOwnership(Board b): 局面 b からモンテカルロシミュレーションまたはモンテカルロ木探索を行ったときに, 各地点が黒のものになる確率. 石の強弱や模様の有無を判別できる. これも Zen などの高度なシミュレーションを用いる際には中級者レベルの直感的なものとの違いを考慮する必要がある.
- double[] getCriticality(Board b): 局面 b で, 各地点を占めることが勝敗にどの程度直結するか, 地点の重要度を求める関数[11]. 通常 Ownership と同時に求める. 図5に, Ownership と Criticality の例を示す.
- string getShapeName(Board b, Pos p): 局面 b における着手 p を「ノビ」「アタリ」などと日本語表記する関数. これは穴戸らの研究[10]に基づく.
- Pos doAreaSearch(Board b, Pos[] searchArea, Pos[] weightArea, double weight): 想定図作成に用いる. 局面 b で次の一手を求めるが, 探索範囲を searchArea に限定する. またシミュレーションの最後に勝敗を決める際, weightArea 内の重みを weight 倍する. このような「場所による重みづけ」は地合派・実利派の演出に用いたもの[3]と同じ技術である.

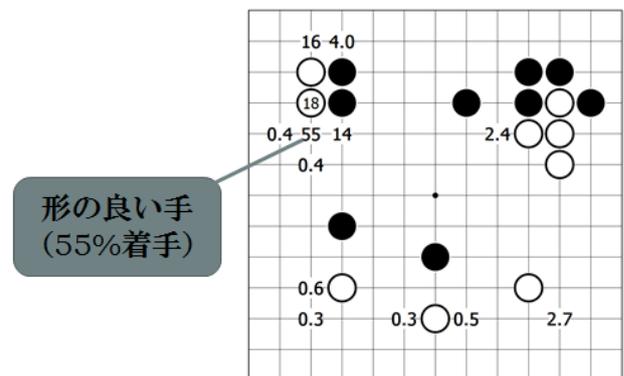


図4: 白18に対する黒の着手選択確率

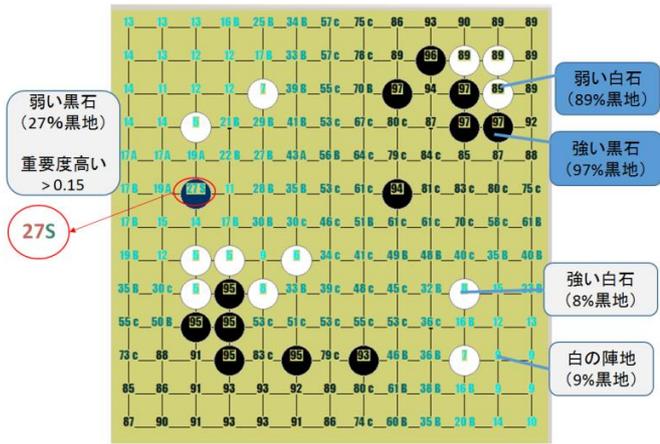


図5 : ownership, criticality の例

【理由説明】

4章最後の悪手分類のそれぞれについて、上に示した関数を用いて判別を行い、「A7 ハネはそっぽの悪手でした」などと説明することを考える。「ハネ」を言う部分が getShapeName 関数である。

ここで簡単のため、局面と着手に番号を振る。B1 を検出対象の黒の悪手、S1 をそれが打たれたときの(直前の)局面、W0 を直前の白の着手などとし、局面 S0→着手 W0→局面 S1→着手 B1→S2→W2→S3 と表記することにする。

まずはプロトタイプとして、関連性のある3つの悪手「そっぽ」「手拍子」「お付き合い」を以下のように分類することにした。

1. getWinratio(S0,W0) と getWinratio(S2,W2)を比べ、白にとっての勝率が5%以上向上している場合、B1 は悪手とした。白をプログラムに打たせている場合、この判別はコストなしに行える。
2. getSelectProbs を使い、B1 の形の良さを調べる。これが一定値(例えば0.05)以下の場合、一見良い手ではないので、この3つには分類しない。
3. getBestMove(S1)により、最善手 B*1 を求める。
4. getDistance を使い、W0, B1, B*1, W2 の位置関係を調べる。

(ア) W0, B*1, W2 が近く、B1 が遠い場合、B1 を“そっぽの手”と判断する。つまり、打つべき手 B*1 は直前手 W0 の近くにあったのに遠くに打ってしまい、相手に W0 の近くの手 W2 で咎められたということである。

(イ) W0 と B1 および B*1 が近い場合、B1 を“手拍子の手”と判断する。局所的な戦闘で一見良い手を打ったが別の近い手の方が良かったということである。なお同じような局所戦闘で単なる「筋違いの手」「形が悪い手」もこのような距離の条件を満たすが、形が良い手だけを2で選んでいるため、手拍子の手とは分類されない。

(ウ) W0 と B1 が近く、W0 と B*1 や W2 が遠い場合、B1 を“お付き合いの手”と判断する。つまり、打つべき手は直前手とは遠くにあったのに近くに打ってしまって損をした(後手を引いた)ということである。

実際に検出できる、そっぽ・手拍子・お付き合いの人工的な局面例を図6, 7, 8に示す。

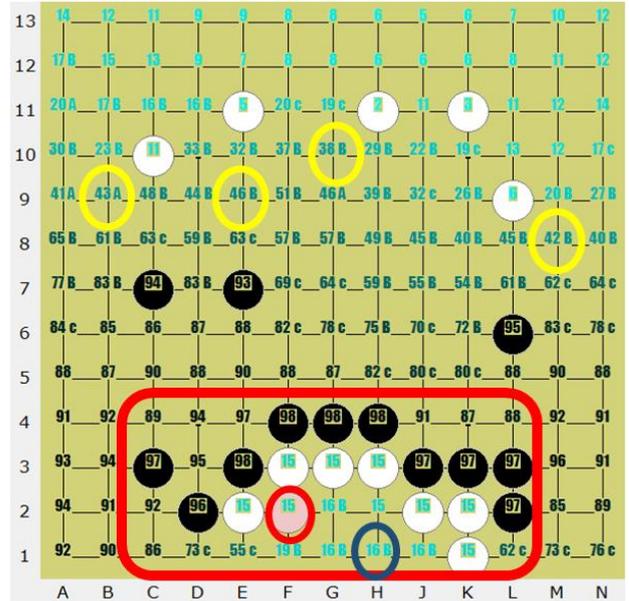


図6 : そっぽの例. 白F2 ツギ(赤丸)に対し、黒H1 オキ(青丸)ならば下辺白を殺せる。黄丸の部分に打つような手は部分的には悪くないが、すべてそっぽである。

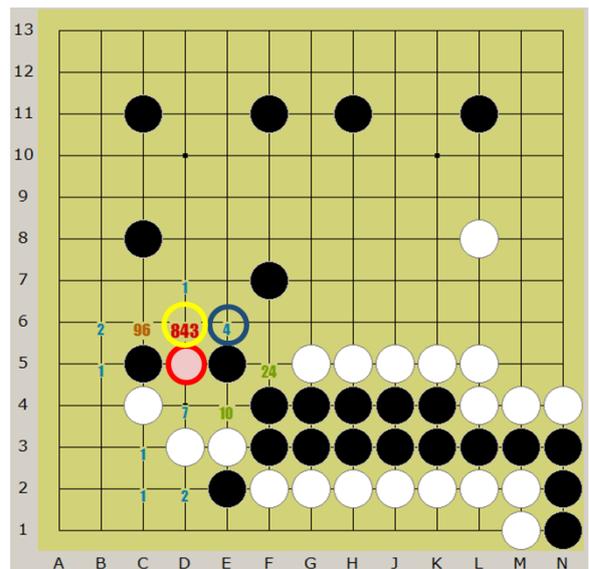


図7 : 手拍子の例. 白D5 ハネコミ(赤丸)に対し、一見黒D6 オサエが自然に見える(黄丸、選択確率84.3%)が、実際には白F5 アテからウツェガエシで黒大石を取られる。黒E6 ヒキ(青丸)などと打つべきである。

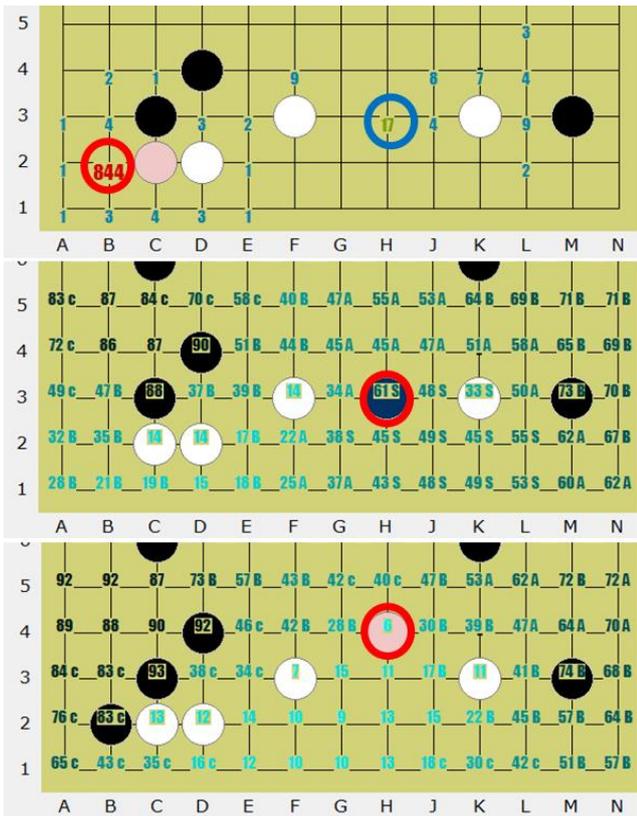


図 8 : お付き合いの例. 上段, 白 C2 ハイに対し, 黒 B2 オサエは自然に見える (赤丸, 84.4%) が, 黒 H3 ウチコミに打ちたい (青丸, 1.7%). 中段は打ちこんだ図, 下段はお付き合いで守られた図. 下辺白の ownership に違いがあるのが分かる.

【結果説明】

悪手の理由 (分類) と, 悪手の結果は関連する場合も多いが独立の場合も多い. 例えば {そっぽ・お付き合い・手拍子} のそれぞれに対し, {黒が死んだ, 黒が攻められた, 白が生きた, 白が強くなった, 急場に先行された, 地を損じた, 模様を張られた, 模様に入らされた} それぞれは, 一部の例外を除いて組み合わせとして成立する. 組み合わせの数を考えれば, 理由と結果をそれぞれ独立に判別して解説文を作り, 両方を並べて見せるのが容易であると考えられる.

プロトタイプでは, B1 着手前の局面 S1 と, W2 着手後の局面 S3 に対して getOwnership 関数を用いて, 30%以上の低下がみられ, かつ S3 で 20%以下になっている黒石があった場合, 「黒石が取られてしまった」と結果を説明する. それが白石だった場合は, 「白石を助けられてしまった」と結果を説明する. 著者の一人と Nomitan との実戦棋譜による検出例を図 9 に示す.

この方法ではいわゆるカス石を取られた・助けられた場合にも説明してしまうことになるので, より良い別の説明がありうる場合を考えると, 石数や criticality で閾値を設け

るべきかもしれない. ただし勝率 5%低下の条件があるため, 単にカス石を取られただけではこういった説明は行われない.



図 9 : 対局しながらの悪手指摘の様子, 実戦例. 表示のしかたなどには改良の余地は大きい.

6. 想定図作成

黒の着手 B1 が悪く, ある結果を導いたのであれば, 通常は正しい着手 B*1 はその逆の結果を導くはずである. 正解手順の想定図作成のためには, 単に正しい手とその principal variation を表示するのではなく, 「逆の結果」の説明ができるものであることが望ましい. つまり, 本稿のプロトタイプで実装した「B1 によって黒石が死んだ」という説明に対しては, 「B*1 によって黒石が生きた」という結論を説明できるような図を得たいのである.

【想定図に載せる手数】

想定図は見やすく分かりやすくあるべきである. そのためには, “必要どころまで” 手順を進める必要がある. 例えば図 10 (イ) では中級者はまだ黒が死んだことを納得できないかもしれず, 逆に (ハ) では冗長すぎるかもしれない. これらは, レベルによっても感じ方が異なるものである.

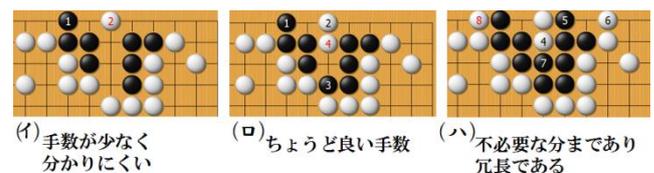


図 10 : さまざまな手数の想定図.

死活に関しては、ownership が一定以上・以下になった場合は「明らかな死」「明らかな生き」と判断できるかもしれない。しかし、高度なヒューリスティックを組み込んだプログラムの場合、例えば「隅の六目」と呼ばれるような頻出の死形を確実に死ぬように (ownership がほぼ 0 になるように) 調整している可能性があり、これを「明らかな死」とされると初中級者にとっては困る可能性がある。このような場合には一定割合でランダムなシミュレーションを行うなどの方法がありうるかもしれない。

布石や中盤の、石の死活に関係ない想定図に関してはさらに判断が困難である。getSelectProbs の高い場所が近くになくなったことを手どまりと見なしたり、それと手数を組み合わせる機械学習を行うなどが有望だろう。

【想定図に載せる手の作成】

プロトタイプで用いたような「B1 は悪手で、黒石が殺された (ことによって勝率が 5% 以上下がった)」場合、普通に getBestMove (S1) を用いれば黒石を生かすような手が B*1 として選ばれるはずである。しかしながら、同様の手順で「その黒を脅かすような白の手」を打たせることは通常期待できない。なぜなら、もし B*1 で黒石が本当に生きたなら、それを攻撃してよりはっきりと生きられてしまうことはコウ材の減少にも繋がり大抵は悪手だからである。内部から眼形を脅かすような手は外部からのキカシに比べ地の損になることも原因の一つである。

そこで、5 章前半で提示した doAreaSearch を用いて、「着手範囲を限定し」「黒石を殺すことにボーナスを与えた」探索を行うことが有効になる。この二つの改変は両方必要である。ボーナスを与えるだけでも良さそうに思えるかもしれないが、探索の水平線効果により、「今はその付近に触れないほうが殺せる見込みがある」とプログラムが判断してしまうことが多いためである。

【重み (ボーナス) の影響】

図 1 1 (a) は、黒の手入れが必要な局面である。これを手抜きして白 F2 オサエにより殺された場合に、どう打てば良かったかの想定図を出すことを考える。

殺された石の周囲に着手を限定し、doAreaSearch を用いると、1 手目と 2 手目はどのような重みでも黒 F2 マガリ、白 G2 オサエとなる。しかしボーナスを 100 目とした場合、3 手目黒は図 1 1 (b) のように眼を持つ箇所を打ってしまった。これは、生きを確実にしようとするあまり、全体の地合いを全く考えなくなってしまったためである。ボーナスを 10 目とした場合、図 1 1 (c) のように 3 目程度(b)よりも得な手を打ってくれる。

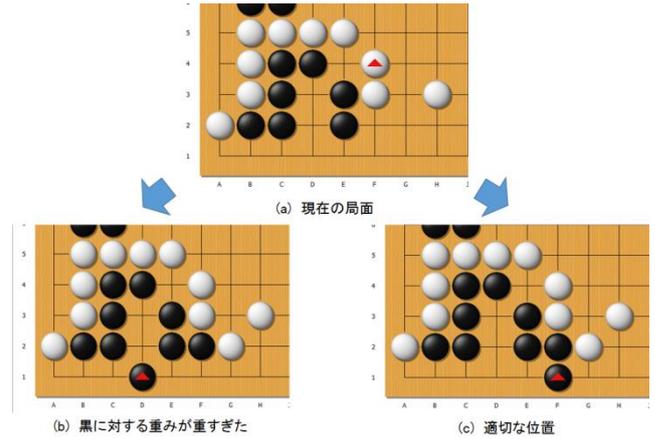


図 1 1 : ボーナスの影響. 損な生き方(b)と得な生き方(c).

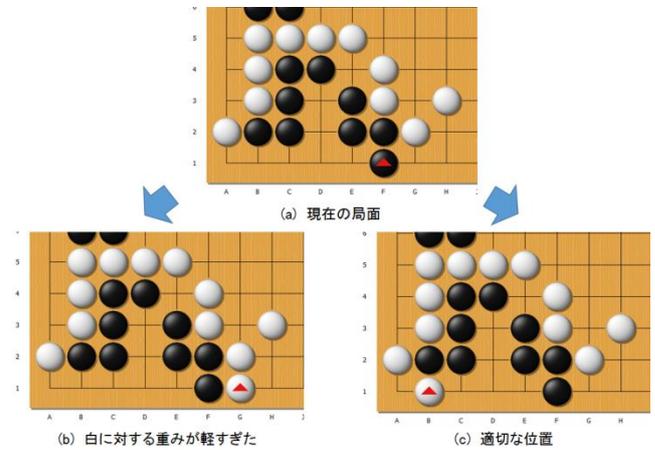


図 1 2 : ボーナスの影響. 消極的な手(b)と攻撃的な手(c).

図 1 2 (a) は図 1 1 (c) と同じ局面である。ここで今度は白にボーナス 10 目の doAreaSearch を行うと、図 1 2 (b) のように、ヨセのような手を打ってしまう。これはある意味“正しい手”であるが、想定図では何らかの攻撃を仕掛けてくれないとつまらないため、望ましくない。ボーナスを 100 目にする、地を気にせず少しでも殺せる可能性が高くなるように、図 1 2 (c) の B1 ハネ、あるいは D1 オキに打ってくれる。

【想定図作成手順】

以上をまとめると、黒石を殺された・白石に生きられた場合の想定図作成手順は以下の通りになる。

1. B1 が悪手の場合、局面を S1 まで戻す。
2. 殺された・生きられた石、およびその周囲の範囲を保存する。
3. 最善手 B*1 を getBestMove で求め、着手する。
4. getOwnership で対象の石の ownership を調べ、90% 以上であれば「生ききった」「殺しきった」と判定して想定図作成を終了する。

5. ボーナスの大きい doAreaSearch により, 白の着手を決定する.
6. ボーナスの小さい doAreaSearch により, 黒の着手を決定し, 4.に戻る.

図 1 3 は実戦例でのプロトタイプの想定図表示例である. 左下に手を入れず, M9 のそばに打って殺された例であり, 正しくは F2 マガリから D2 ツギまでと打つべきであったことが示されている.

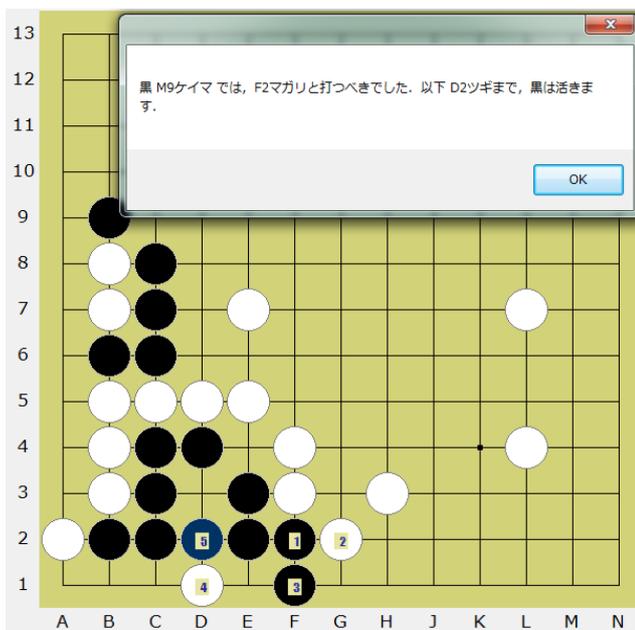


図 1 3 : 対局しながらの想定図表示の様子, 実戦例.

7. おわりに

本研究では, 教える囲碁 AI という遠く高い目標を目指し, 実際の指導の場での聞き取り調査を踏まえ, まず指導碁, なかでも悪手の指摘と解説に取り組むことに価値があることを主張した. そのうえで, 悪手指摘は「検出」「分類」「結果説明」「想定図作成」の4つのステップに分けることができる場合が多いことを述べた. さらに, それぞれについて非常に単純な方法を用いてかつ限られた種類に対してにすぎないものの, 実際に既存の囲碁プログラムの機能を流用拡張することが有望であることを示した.

論文中に示した例はうまくいった例に過ぎず, 実際に悪手の分類や結果の対象を拡げていった場合にどこまでルールベースで対応できるのかには疑問が残る. 全てを機械学習に行わせることは今のところ難しいと考えているが, 少なくとも数百程度の悪手解説例を準備し, それに合う条件設定を学習あるいは最適化することは行いたい. さらに, 本稿は工学的評価が行われていないという意味で未完成の研究であり, さまざまな指標がありうる評価法の確立からはじめ, 実験を行っていきたいと考えている.

謝辞 本研究は JSPS 科研費 26330417 の助成を受けたものである.

参考文献

- [1] David Silver, Aja Huang ら, Mastering the game of Go with deep neural networks and tree search, Nature 529, pp.484-489, 2016
- [2] 藤井叙人ら, 生物の基本原則の導入によるビデオゲーム COM プレイヤーの『人間らしい』振る舞いの自動獲得, IPSJ EC 研究会, Vol. 2013-EC-27 No.16, 2013
- [3] 池田心, Simon Viennot, モンテカルロ碁における多様な戦略の演出と形勢の制御~接待碁 AI に向けて, IPSJ GPW2012, pp. 47-54, 2012
- [4] 亀甲博貴, 三輪誠, 鶴岡慶雅, 森信介, 近山隆, ロジスティック回帰による言語モデルを用いた将棋解説文の自動生成, 言語処理学会第 20 回年次大会, pp. 8-18, 2014
- [5] 亀甲博貴, 森信介, 鶴岡慶雅, 将棋解説文のグラウンディングのための指し手表現と局面状態の対応付け, IPSJ GPW2014, pp. 202-209, 2014
- [6] 石脇滉己, 荒川達也, 「一見良い手」を含めた初心者向け詰将棋解説文生成の提案, IPSJ GI 研究会, Vol.2015-GI-34 No.5, pp.1-7, 2015
- [7] 山下宏, 将棋名人のレーティングと棋譜分析, IPSJ GPW2014, pp. 9-16, 2014
- [8] Remi Coulom, Computing Elo Ratings of Move Patterns in the Game of Go, ICGA Workshop, 2007
- [9] 池田心, ビエノシモン, 行動評価関数を用いたモンテカルロ木探索の重点化と見落としの抑制, 情報処理学会論文誌, Vol.55 No.11, pp.2377-2388, 2014
- [10] 穴戸崇音, 池田心, ビエノシモン, 機械学習を用いた囲碁の着手の日本語表現, IPSJ GI 研究会, 2015-GI-33(4), pp.1-7, 2015
- [11] Remi Coulom, Criticality:A Monte-Carlo Heuristic for Go Programs, Invited talk at the University of Electro-Communications, Tokyo, Japan, 2009