

記憶に基づく推論を使った囲碁プログラム「勝也」の試作

清 慎一

川嶋俊明

ssei@ssl.fujitsu.co.jp

kawkaw@ssl.fujitsu.co.jp

(株) 富士通ソーシアルサイエンスラボラトリ

我々は、記憶に基づく推論を使った囲碁プログラム「勝也」を開発した。「勝也」は、実際の対局で有段者の打った部分的な形を事例として記憶しており、記憶に基づく推論を使って次の一手を求める。この方法により、人間の思考方法を解析してプログラム化しなくとも、良い手を打つことができるようになった。本論文では、囲碁プログラム「勝也」の方式について説明する。

The Experiment of the Go-program “KATSUNARI” using Memory-Based Reasoning

Shinichi Sei

Toshiaki Kawashima

Fujitsu Social Science Laboratory Ltd.

We have implemented the Go-program, KATSUNARI, which uses memory-based reasoning. In KATSUNARI, we adopted the pairs of pattern and next move which higher grade player actually put as the cases. In order to using memory-based reasoning, KATSUNARI created good moves without analyzing and representing human's skill. This report describes the algorithm of KATSUNARI.

1 はじめに

囲碁プログラムが誕生してから約30年経つが、一番強いプログラムでもアマチュア5級程度である¹。強い囲碁プログラムを作れない主な理由として、(1) 盤面が広過ぎて、コンピュータチェスで成功した手法である探索が使えない、(2) 精度の高い形勢判断を行なう評価関数の作り方が不明である、(3) 人間の思考方法が解明できず、良い候補手を生成することが難しい、などが挙げられる。

我々は(3)の問題を解決する方式として、高段者の実際の対局に現れた石の配置(囲碁用語で形という)を直接利用して候補手を求める方式を考案した[8, 9]。その方式では、プログラムはあらかじめ高段者が実際の対局で打った手と、その時の形を「形データ」として多数記憶しておく。そして、ある盤面が与えられた時に、(1) その盤面とプログラムが持っている最も類似した「形データ」を探し、(2) その形ではどこに打ったかを求め、(3) その位置を候補手とする。このように、プログラムが持つ沢山の事例と、与えられた問題の状況とを比較して、最も類似した事例の解を回答とする方式は、「記憶に基づく推論」[5]と言われ、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いる。記憶に基づく推論を囲碁プログラムに利用することによって、設計者が高段者の思考方法を解析してプログラム化しなくとも、良い手を求めることができるようになる。

しかし、形として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。そのために我々のプログラムでは、盤面の一部を切り出して記憶している。しかしながら、記憶する「形データ」の一つ一つのサイズが小さいため、ある盤面が与えられたとき、複数の箇所が候補手として提案されてしまう。また、この方式だけでは、石の生死について正しく判断することができないという欠点もある。我々は、以上の欠点を補いつつ、実際の対局に現れた形を直接利用して候補手を求める囲碁プログラム「勝也」を試作した²。

2 記憶に基づく推論による候補手

既存の囲碁プログラムは、殺す手、生きる手、つなぐ手、切る手、模様を広げる手、模様を狭める手、地を囲う手、…など、何種類もの候補手を生成している。しかし、それらを全てをプログラム化することは大変手間のかかる作業である。そのうえ、上級者の思考方法を分析しアルゴリズム化できなくては、プログラムはそれぞれの手の意図を正確に反映する候補手を生成できない。また、複数の目的を達成する候補手の最適な位置を求めることは、さらに難しい。序盤の定石は囲碁の本に書いてある通り入力すれば済む³が、それ以外の手はそのままプログラムできるほど、詳しい分析結果が書かれた本はない。強い囲碁プログラムが存在しない理由の1つに、このような、人間の思考方法の解析の困難さが挙げられる。

「勝也」はこの問題を解決するため、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いる、「記憶に基づく推論」方式を使って候補手を求める。類似度の精度が高ければ、有効な事例を多数記憶すればするほど良い候補手を生成できるため、人間の思考方法の解析やそのプログラム化などの作業をせずに強い囲碁プログラムが生成できる。

¹1995年 FOST 杯囲碁ソフト選手権で優勝した Handtalk に、日本棋院からアマ5級の認定証が贈られた。

²強さを評価するために96年 FOST 杯囲碁ソフト選手権に出場するが、選手権は本ワークショップの直前に開催されるので、論文執筆時には結果はわからない。

³手順はそれで良いが、複数ある枝のうちどの手を選択すべきかは周囲の状況によるので、実は簡単にはプログラム化できない。

2.1 データベース

「勝也」は、高段者が実際の対局で打った手と、その時の局所的な石の配置を「形データ」として多数記憶している。

記憶している形データの例を図1に示す。形データは、{ 盤面の局所的な石の配置 (候補手位置を中心としたマンハッタン距離4以下の菱形)、その配置における候補手位置、形の緊急度 } の3つの組から構成されている。形データは、プロ棋士の実戦譜と、アマ高段者と初級者の対戦から収集した。プロ棋士の実戦譜からは、全ての手を収集している。プロ棋士の打った手は、全て良い手と考えられるからである⁴。また、アマ高段者と初級者の対戦からも、アマ高段者の打った手を収集している。プロ棋士の実戦譜だけでなく、アマ高段者と初級者の対戦からも収集するのは、「相手の悪い手をとがめる手」が、プロ棋士同士の実戦譜からは集められないからである。現在、「勝也」は約 50,000 の形データを持っている。形の緊急度とは、「相手の着手にすぐに対応する手であるかどうか (相手が着手した直後に、その近傍に打ったかどうか)」を基に算出した値であり、候補手の評価値の算出に利用される。

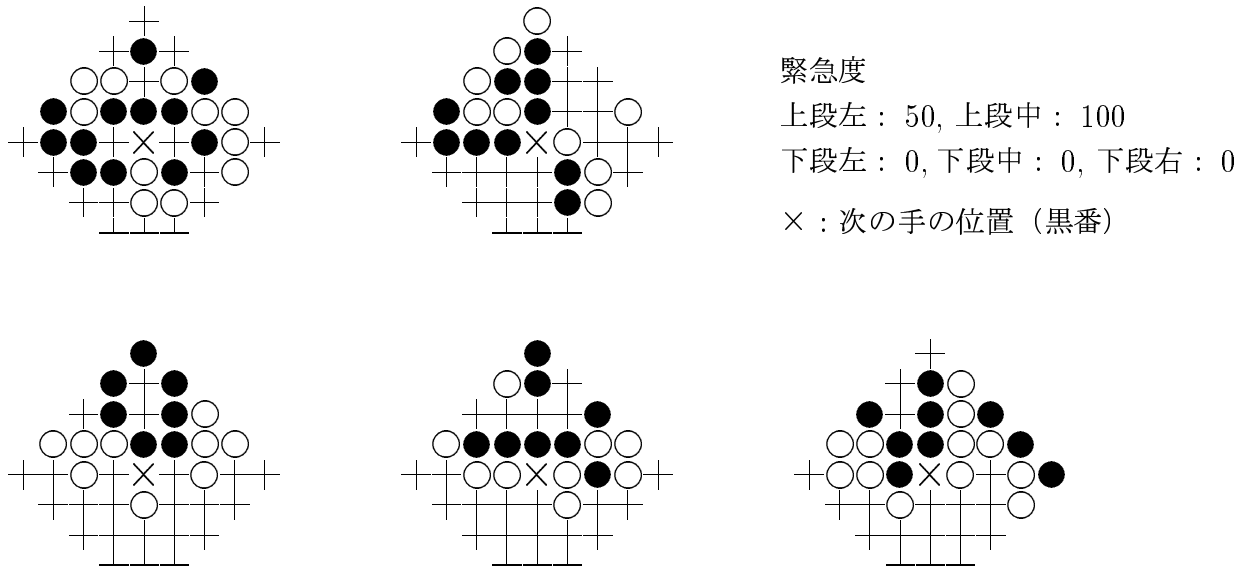


図 1: Pattern Data

2.2 候補手生成手順

候補手生成の手順は次の通りである。

- [1]** 候補手として考慮するための、類似度の最低点 (閾値) を決める。
- [2]** 盤面上の全ての点に対し、その周囲の形と、形データとを比較し、最も類似している形データとその類似度を求める。
- [3]** 最初に設定した類似度の最低点を越えるデータがある座標を、候補手とする。

類似度は、「石の違い度 (白/黒/空点/盤外)」「候補手位置と、その周囲に存在する石までの距離」「候補手位置と、その位置と最も近い石までの距離」を因子として算出する。

⁴少なくとも我々アマチュアには、プロ棋士の打った手の善悪はわからない。

$$\text{類似度} = F(\text{距離係数, 最も近い石までの距離, 石の違い度})$$

3 囲碁プログラム「勝也」

3.1 システム概要

「勝也」のシステム構成を図2に示す。対戦相手が打った着手位置を入力すると、「盤面認識」にて現在の状態を把握する。次に「候補手生成」にて各種の候補手を生成し、「着手決定」にて次の一手を求める。

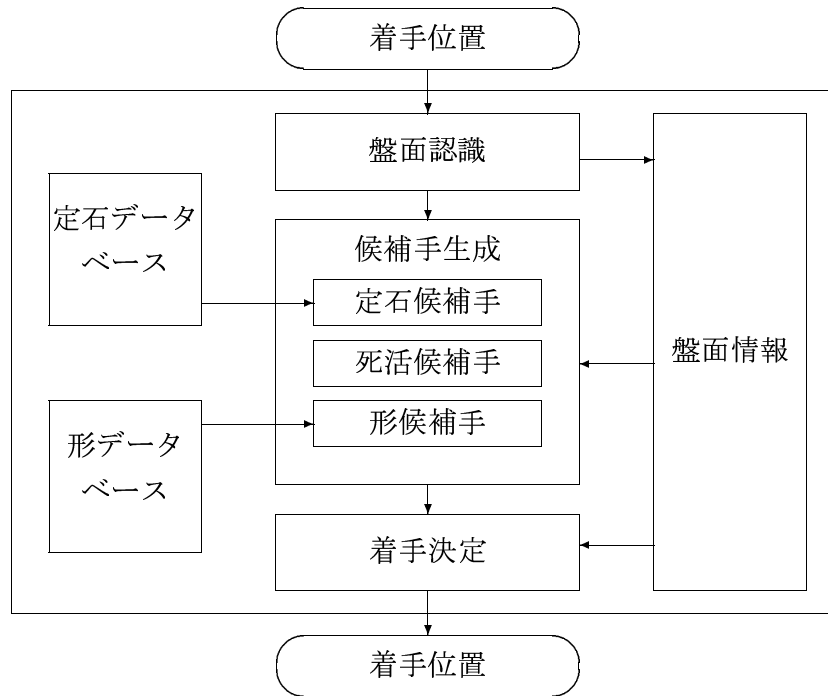


図 2: System Configuration

3.2 盤面認識

人間が盤面を認識する際、各石を個別にではなく、戦略的に意味のある石の集団として捉えていると考えられており、既存の多くの囲碁プログラムは、その考えに基づいて盤面認識の処理を行なっている。その戦略的に意味のある集団として、同色のつながった石や、そのつながった石が連結してできた集団をなどが提案されている [1, 2, 3, 4, 6, 7, 10]。そして、それらの集団のデータの属性（生死の状態や、その集団が及ぼす影響力の範囲の大きさなど）を基に、次の手の候補手や、その候補手の評価値を算出する材料としている。「勝也」も同様の考えに基づいて盤面認識の処理を行なっている。

「勝也」は、盤面そのものと、戦略的に意味のある集団として認識した結果である連と群の2つのデータを盤面情報として蓄える(表1)。この認識結果は、候補手生成及び着手決定時の資料として使われる。例えば、連の生死が「中立(守り番なら生き、攻め番なら死にの状態)」であった場合、攻め側からはその連を捕獲する手を、守り側からはその連が脱出する手を「死活候補手」としている。群の緊急度は、着手決定時において、盤面のどの部分に着目するか、どの部分の手を優先的に打たせるかの材料となっている。

表 1: Data Structure

| データ名 | 意味 | 主な属性 |
|------|----------------------|------------|
| 連 | 縦横につながった同色の石、生死の最小単位 | 生死、ダメ数 |
| 群 | 戦術上の同色の石の集合 | 群サイズ、地、緊急度 |

3.3 候補手生成

「勝也」は、形候補手、定石候補手、死活候補手の3種類の候補手を生成する(表2)。3種類の候補手について、それぞれ有望な候補手を複数作成する。また、それぞれの候補手がどれくらい有望な手であるかを示す基礎評価値も算出する。

表 2: Candidate

| 候補手名 | 意味 | 生成方法 |
|-------|-----------|-----------|
| 形候補手 | 良い形を形成する手 | 記憶に基づく推論 |
| 定石候補手 | 序盤の隅の定石の手 | パターンマッチング |
| 死活候補手 | 連の捕獲/脱出の手 | 局所的な探索 |

3.3.1 形候補手

形候補手は、記憶に基づく推論を使って求める。「勝也」では、記憶する事例を形(石の配置)としているため、記憶に基づく推論で求めた候補手を「形候補手」と名付けている。

形候補手の基礎評価値は、形の緊急度と、類似検索によって求めた類似度、候補手位置とその周囲の最も近い石までの距離の3つの要素を基に算出する。

$$\text{評価値} = F(\text{形の緊急度}, \text{類似度}, \text{最も近い石までの距離})$$

3.3.2 定石候補手

序盤の、特に隅における候補手は「定石」と呼ばれ、最善の候補手の手順が研究されている。形データベースには、定石に従って打たれた手も含まれるが、それ以外の手も多く、候補手の類似検索に要する時間は長い。また、定石一つのサイズは、形データのサイズよりも大きいので、正確に定石を打つことが難しい。「勝也」は、それらの問題を解決するために、定石データベースを形データベースとは別に用意し、完全に一致するものを検索することにより定石候補手を生成する。

定石候補手の基礎評価値は、定石データベースに蓄えられている。定石の各手には、(1)この形になったらすぐに打つべき手と、(2)すぐにはないが機会があったら打つべき手、の2値からなる評価値が登録されている。

3.3.3 死活候補手

囲碁において、生死の最小単位となるのは、縦横につながった同色の石の集まりである。「勝也」ではこの石の集まりを「連」と呼び、盤面認識のフェーズで連に関する情報を作成している。

この連を「殺す／生きる」手を、「死活候補手」と呼ぶ。

形候補手は、「良い形」になる候補手位置を求めることはできるが、「形は悪くとも、連を{殺す／生きる}手」を求めることはできない。それは、連のサイズ及びそれを囲む石の範囲は、形データのサイズよりも一般に大きいからである。そこで、死活候補手は局所的な探索によって求め、その結果を死活候補手として利用している。

死活候補手の基礎評価値は、死活の対象となる石の属性を因子とする関数によって算出する。

3.4 着手決定

「候補手生成」で作成された複数の候補手と、盤面の状態を考慮して、その中から最も良い手を「次の一手」として求める。

着手決定におけるアルゴリズムは次の通りである⁵。

- 【1】 生成した全ての候補手を集める。
- 【2】 盤面上で緊急度の高い場所（忙しい場所）を求める。
- 【3】 盤面上の緊急度と各候補手の基礎評価値から、盤面上の全ての座標について総合評価値を求める。
- 【4】 最も総合評価値の高い座標を、最も良い手とする。

最初に、生成した全ての候補手を集める。次に、盤面上で緊急度の高い場所（忙しい場所）を求める。盤面上の緊急度は、「盤面認識」で求めた緊急度の高い群とその群の周囲としている。緊急度の高い群に対しては、すぐに何らかの行動を起こす（逃げる、殺すなど）べきだからである。

次に、盤面上の緊急度と各候補手の基礎評価値から、盤面上の全ての座標について総合評価値を求める。

$$\text{総合評価値} = F(\text{盤面上の緊急度}, \text{各候補手の基礎評価値})$$

最後に、総合評価値が最も高い座標を、最も良い手の座標とみなし、その座標を「次の一手」として出力する。

既存の囲碁プログラムでは、候補手の評価値は、地や模様をどれだけ作ったかを評価値の材料としているが、「勝也」はその方式を採用していない。緊急度により着目すべき範囲を求め、その中で最も良い形になるような手を候補手としている。これは、地や模様をどれだけ作ったかを正確に計算することが難しいためである。群の生死が絡む場合の手、複数の意図（攻めながら地を囲うなど）を達成する手、打った直後でなく後々効いてくるような手の評価は特に難しい。また、現在のコンピュータ囲碁のような初級レベルでは、「序盤はどこに打っても大差ない」という経験則からも、このような方針を立てた。

⁵FOST 杯に参加する際は、処理時間の節約などのため、処理の順番など説明と若干異なる。

4 実行例

実行例として、「勝也」とサンソフト社「ゲームの鉄人」の囲碁との対局の一部を図3に示す。黒番が「勝也」である。この勝負は後半に大石を取られたため「勝也」が負けたが、中盤までは「勝也」が優勢であった。

図に示した30手目以降では定石候補手は生成されていない。また、特に捕獲可能な連も存在しないため、死活候補手も生成されていない。この図の「勝也」の候補手は全て、「記憶に基づく推論」による形候補手の手だけである。

「勝也」は、相手の模様を狭める手という知識を持っていないが、31手目(左図K-5)のように、相手の模様を狭める手を打っていることがわかる。また、33手目(左図H-5)のように、攻めながら相手の模様を狭めるという2つの目標を同時に達成する手も打っている。右図では、打ち込みの51手目(F-11)、57手目(C-9)、逃げる手の53手目(F-14)、守りの手の61手目(O-13)なども打っている。この対局を見る限り、形候補手だけでも、初級者レベル以上の手を打ち続けることができるように見える。

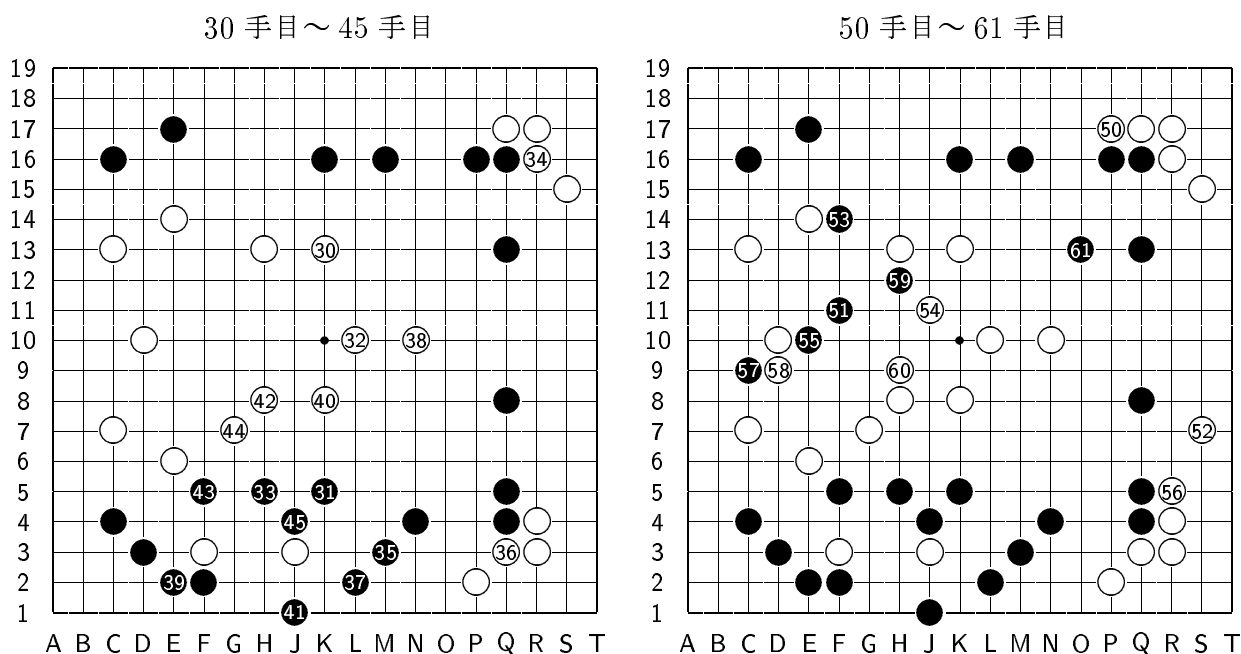


図 3: Example

5 おわりに

我々は、記憶に基づく推論を使った囲碁プログラム「勝也」を試作した。実験では、人間の思考方法を解析してプログラミングしなくとも、良い手を生成することができることがわかった。

しかし、我々は「勝也」の評価を始めたばかりなので、この方式でどの程度強いプログラムが実現できるかはまだ不明である。今後は、他の囲碁プログラムと数多く対戦させて強さを評価すること、記憶するデータ数による強さの違いの評価などを行ないたい。

また、「勝也」は部分的に非常に良い手を生成することができるので、局所的な探索における候補手の生成に利用すれば、人間のような深く狭い(展開する枝数が少なく、深く読む)先読みが実現に近付くのではないかと考えられる。今後の課題として、記憶に基づく推論を利用した探索について考えていきたい。

謝辞

本研究の機会をいただいた守屋事業部長代理、田中プロジェクト担当部長に感謝します。「勝也」のプログラミングに協力してくれた岡野幸世に感謝します。

参考文献

- [1] Chen, K : The Move Decision Process of Go Intellect, Computer Go, No.14, pp.9-17, 1990
- [2] Fotland, D. : Knowledge Representation in The Many Faces of Go, available from Go Archive as Go/comp/mfg.Z, 1993
- [3] Hsu Shun-chin, Liu Dong-Yeh : The Design and Construction of the Computer Go Program Dragon II, Computer Go, No.10, pp.10-20, 1989
- [4] Kierulf, A., Chen, K., Nievergelt, J. : Smart Game Board and Go Explorer : A study in software and knowledge engineering, CACM, Vol.33, No2, pp.152-166, 1990
- [5] 松原 仁 : 推論技術の観点からみた事例に基づく推論, 人工知能学会誌, Vol.7 No.4, pp.567-575, July 1992
- [6] Reitman, W., Wilcox, B. : The Structure and Performance of The INTERIM.2 Go Program, Proc.6th IJCAI, pp.711-719, 1979
- [7] 実近 憲昭, 他 : 囲碁システム「碁世代の方法」, ICOT 研究速報 TM-618, 1988
- [8] 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
- [9] 清 慎一, 川嶋俊明 : 事例ベース推論により囲碁における次の一手を求めるプログラムの試作, 情報処理学会第50回全国大会, Vol.2 pp.143-144, 1995
- [10] Wilcox, B. : Reflections on Building Two Go Programs, SIGART Newsletter, Vol.10, Number 94, pp.29-43, 1985