

TM-0618

囲碁システム「碁世代」の方法

実近憲昭(ETL), 沖廣明(未来工)

吉川貞行(NTT)

November, 1988

©1988, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

囲碁システム「碁世代」の方法

実近 憲昭*

沖 廣明

工業技術院 電子技術総合研究所

(株) 未来技術研究所

吉川 貞行[†] 吉岡 勉 内田 俊一

(財) 新世代コンピュータ技術開発機構

11 月 10 日, 1988

概要

本論文は、第五世代コンピュータプロジェクトにおける囲碁システム「碁世代」の研究開発について紹介したものである。本プロジェクトでは、囲碁プログラムの研究開発を通じて、探索問題、曖昧性の処理、例外処理、協調問題解決などの人工知能(AI)の基本的な問題を提起し、問題解決の新しいアプローチを探ることを目的としている。囲碁システム「碁世代」は、第五世代コンピュータプロジェクトの中期計画の一環として開始され、この分野の研究者を集めたワーキンググループを組織すると共に、この分野での実績のある電子技術総合研究所との共同研究を行い、この研究を推進することとした。中期の研究開発では、複数の人工知能的テーマに視点をおいて、人間プレイヤーのシミュレーションを試みた。昭和61年9月に概念設計書が完成し、昭和62年2月には簡易版ながらも序盤から終盤まで打ち切ることができるシステムが完成した。昭和62年度からは、システムの見直し、改良を加え、更に、ハードウェアもPSI-IからPSI-IIに移行したことにより、速度的にも対局に耐え得るものとなった。本論文は、囲碁システム「碁世代」の研究開発の中間報告として、研究の概要について述べる。

1 はじめに

囲碁は人工知能(AI)の核心に迫るテーマを比較的純粋な形でいくつか内蔵している。囲碁プログラムを研究開発することで、新しいプログラミング言語や計算機システムが、どの程度AIの問題に適しているかを評価することが可能であり、また、人間の行っている思考過程を計算機システム上でモデル化することを試みることで、AI問題解決の新しい方法が見出せるかもしれない。囲碁は、チェスなどとくらべ、はるかに大きな探索空間を持ち、その探索一つとってみても単純な手法は通用せず、知識を有効に用いた知的な探索が必要となる。また、いくつかの抽象概念に基づく曖昧な思考は、計算機システムの上で扱うためには、新しい方法を必要とする。このようなAIの基本的な問題を提起することが、正に、第五世代コンピュータプロジェクトにおいて囲碁の問題を取り上げる理由であるといえよう。

囲碁において扱われる、AIの核心に迫る研究テーマをあげると次のようになる。

*現 (株)AI 言語研究所

[†]現 NTT データ通信株式会社

1. 探索：探索パラダイムの限界の打破
2. 曖昧：曖昧概念の処理、曖昧思考の導入
3. 例外：未知の状況への対処（例外処理能力）
4. 協調：協調システムの有機的統合処理
5. 学習：経験の蓄積と法則化

基世代においては、5の学習を除いた残りのテーマについて取り上げた。ここで、簡単に各テーマの意義と概要を述べておこう。

(1) 探索

探索は万能のアルゴリズムと言われており、AIにおいても、基本的な手法の1つとして定式化されている。そのメカニズムの中心は、まず可能な候補手を列挙し、次に各候補手を適用してその結果を調べることである。これはいわゆる generate and test と呼ばれている。そしてこのメカニズムそのものが組合せ爆発の根源である。この爆発を制御するために、検査すべき候補手を出来る限りしぼる努力がなされる。しぼるためにはなんらかの検査が必要となる。これは一見堂々巡りのように見えるがそうではない。このことはしぼる過程を複数段階に分けて行うことにより、比較的無理なく達成される。

即ち、最初は粗い検査（コスト小）により、篩にかけ、候補手数が少なくなるにつれて検査の精度を上げて行けばよい。この原理は、日常生活でもよく使われている。例えば入学試験はよく二度に分けて行なわれる。この原理は認識の段階的深化としてより一般的な立場から捉えることも出来る。

(2) 曖昧

曖昧概念とこれに基づく曖昧思考はファジィ理論の中心テーマでもあるが、これは曖昧さが人間にとって本質的であるという観点に立っている。より狭い見方をすれば、定量的、数値的処理よりも定性的、記号的処理を目指しているともいえる。従来、絶対的ともいえる信頼を得てきた科学的方法は分析的解析的な性格が強く、細かく分析して行けばいつかは真理に到達するという考えが根底にあるようである。また、全てのデータが努力すればやがては手にはいるという考えも一方にある。しかし、人間の扱う概念の中には分析すればかえって実体を失ったり、あるいは無意味になったりするものも多い。また時間とメモリの制約から不完全な知識のまま行動したり、判断を下すことは日常的なことである。

基における抽象概念のいくつかは「自然界」に存在しないものであり（例、厚み）、これに基づく思考は必然的に曖昧思考になる。

(3) 例外

人間は、自分の知識外の状況に置かれたとき、非常事態の知識体系に切り替え問題解決を試みる。このメカニズムを模擬するためには、まず定常状態と非常事態を区別出来ることが必要であり、次に非常事態の意味を知る必要がある。非常事態は不利な場合もあれば有利な場合もある。例えば不利の場合には、不利と判断する原因をつきとめる。その原因を排除または回避する手段を、原因の性質に応じていくつか用意す

る。非常事態における価値基準は定常状態のときと異なる。非常事態である目的を達成するために提示される手段は、ある定常状態で同じ目的を満たす手段と比べて、一般に条件が緩められる。例えば部屋から出るとき通常はドアから出るが、鍵が壊れてドアが開かないときは窓から出ることも考える。ドアも窓も外に通じる通路という点で共通した手段だからである。一般に非常事態での処理は非能率で不正確である。なるべく定常状態でカバーした方がよい。即ち、なるべく多くの場合を想定し定常状態として処理する。非常事態も同じ事を何度か経験すれば新しい定常状態の1つとして分類されるようになる。従って、このテーマは学習と密接な関係にある。

(4) 協調

特に定まった解法の無い問題解決状況に置かれたとき、人間は、先ずその問題に関係のありそうな要因を出来るだけ洗い出す。次にその要素間の因果関係を明らかにしようとする。この様にいくつかの同時生起、または同時進行する事象を有機的に結合させ統合処理する能力を持っている。

基においては、いくつかの味（可能性）を見ながら相手の出方に応じて手段を作っていく場合に、特にこの能力が発揮される。例えば、AとBという2つの目標があり、それぞれ単独に追求した場合はともに今一步で失敗に終わるような状況を考えてみよう。

このとき、Aを追求する過程で生じる環境の変化によって、Bが成功する条件が成立するようになる場合がある。即ち、まずAをしかけ相手がこれを防ぐ隙にBを達成させようとする発想が生まれる。このテーマは協調処理の真のあり方を示唆している。

以上のような複数個の人工知能的テーマに視点を置いて、人間プレイヤーのシミュレーションを試みた。われわれの碁プログラム「碁世代」は現在なお開発途上にあるが、以下はその中間報告である。

本研究は60年度後半から始まり、その年は各種ツールの開発・整備を行った。61年度は概念仕様を決定するとともに、その一部を試作することにより、プロトタイプ「碁世代（簡易版）」としてまとめた。62年度以降は、それらを元に本格的システム作りを開始し、局面認識の精密化、探索及び囲碁知識の拡充、知識間の協調問題解決の導入などのいくつかの改訂が行われ現在に至っている。

本システムは、第五世代コンピュータプロジェクトにおいて開発された推論マシン PSI (Personal Sequential Inference Machine) 上に、ESP (Extended Self-contained Prolog) で書かれている。その規模は現在約3万行になっている。

また、本プロジェクトは、囲碁プログラムの開発経験者及びそれに精通している情報科学関連の研究者で組織した委員会（棋士システム・ワーキング・グループ）の協力を得ながら、工業技術院電子技術総合研究所との共同研究として進められている。

2 人間の碁を打つ機能のモデル化

従来、囲碁のような完全情報二人ゲームのプログラムではアルファ・ベータ枝刈法によるゲーム木探索手法が主として用いられ開発されてきた。特にチェスを中心として研究が進められ、チェスプログラムとしては現在相当な成果を上げている。しかし碁にチェスと同じ探索指向型の手法を適用しようとする、碁の探索空間の規模がチェスと比べ桁違いに巨大なため、うまくいかないことが判明している。即ち、囲碁のプログラム化のためには、従来にはない本質的に新しい方法が求められたのである。

碁世代のアプローチは基本的には人間プレイヤーが通常行っている思考法を出来るだけ忠実にシミュレートしようとするものである。

人間プレイヤーの機能を大まかに分解すると、局面認識、探索、方針決定、手段構成となる。以下、これについて述べよう。

(1) 局面認識 (視覚情報処理機能)

人間が碁を打つ場合、視覚情報機能の果たす役割がかなり大きいといわれている。特に局面の認識段階においてその傾向が著しい。人間は石の配置を単に座標点の配置として捉えているわけではない。戦術的、戦略的に意味のある石の集団の形態を、経験的に、あるいは誰かに教えられて学習する。そしてそれらを用いて局面の認識を行っているのである。この「意味のある石の集団形態」は、人間が学習獲得する過程を反映して、階層的に形成される。

碁世代では、形態の対象として点、連、群、族を導入し、またその他に連結認識の対象として結線を定義した。

これらの諸概念の形成の過程について簡単に述べてみよう。まず知覚段階として「点」が最初にあるのは当然である。次にごく初心の段階では、石の捕獲が比較的分かりやすく、石を捕ったり、逃げたりすることが最大の関心事となる。そこで縦横連結した石の集団である「連」の概念が生まれる。更に経験を積むと、せっかく逃げた連が後でまとめて捕られたりすることがよくあることに気がつく。そこで最終的に、盤上に生き残れる石の集団としての「群」の概念が学習される。このとき群は比較的近接した連の集合から成っていることも理解される。群の概念に続くか、あるいは並行して、勢力圏、または領域に対応する「族」の概念も生まれる。これは、碁が最終的に地の大きさを競うゲームであることから、最初から分かっているようなものであるが、その重要性に気がつくのはかなり学習が進んだ後になるのが普通である。それは族が必ずしも地と直結しているわけではなく、むしろ、打ち込みを誘う大模様であったり、カラムやモタレといった碁における代表的な戦略の運用に関連した対象だからである。最後に挙げた「結線」の概念は、2つの同色石の距離が近接していれば連絡し易いし、敵から見れば切断しにくい。また連絡した石は、連絡前と比べると捕獲しにくい(強化される)、といった経験的事実から生まれたものである。また結線は領域の境界としても用いられる。即ち、無法に侵入してきた敵石は結線で外部との連絡を断たれる。

以上のような諸対象は、単に単体としてよりも相互の2次的な位置関係で初めて意味を持ってくる。例えば包囲関係、連絡や脱出の方向などである。これらの形態認識には人間の視覚的機能が大きく関与している。

(2) 探索 (事実の確認機能)

時折、局面において局所的に不安定な部分が発生する。例えば捕獲されそうな連、切断されそうな結線、死にそうな群、攻め合い状態にある2つの群などである。そしてこれらの部分の決着次第で局面の状態が大きく変動するときは、これらの結末を実際に先読みする必要がある。そのとき、先手番、後手番の二通りについて先読みし、後手番でも成功する時は成功と、先手番でも失敗する時は失敗として確定しており、単にデータ構造を更新するだけでよいが、中立のとき、即ち、先手番なら成功、後手番なら失敗のときは、実際に忙しい局面になり、その場所への着手の価値は高い。

このように、局面認識したり、関連した候補手を求めたりするのに、捕獲とか切断のように特定意図に限定した局所的な先読みが必要になる。先読み以外の方法、例えば特定のパターンを検出して処理するような方法では、様々な変化に対応しきれない。しかし、一般にこのような先読みは時間とメモリ（特に時間）についてコストが高くなる傾向にある。それでも、問題の規模を局所的に限定したり、問題が煮詰まるまで呼ぶタイミングを遅らせることなどでなんとか対処している。

(3) 方針決定（曖昧知識に基づく推論機能）

碁において必勝アルゴリズムが分かっている訳ではないので、方針決定において最善を確信する手段はない。いくつかの（当人にとって）妥当と思われる方針を取り上げ、ある程度の見通しを立て、最終的には好みに応じて決定している。ただし、なるべく実現の可能性が高く、かつその効果が期待できそうな（価値の高い）方針を選ぶ。しかしこれだけの条件では候補方針の個数はかなり多いことになる。実際には、それらの候補方針をすべて検討するようなことはほとんどなく、局面認識の結果として得られたいくつかの特徴によって想起される（当人が多用している）典型的戦法を土台にして、現状に合わせて修正を施したものを方針として採用しているようである。碁世代ではこれに基づいて、典型的状況とそれに関連した戦法を組み合わせ、新しくケース概念を導入した。

方針決定において、価値は大きいが実現手段があまり明確でない方針と、価値はそれほどでもないが、実現手段がハッキリしている方針とどちらを選ぶかは一種の賭になる。一般には、方針が定まった後に手段が具体化されると考えられているが、実際にはその区別はあまり定かではない。実現手段の裏付けが全くない方針を立てるのは時間の無駄であるし、そうかといって方針を立てる段階で詳細な手段を求めるのも時間が掛かりすぎる。人間が採用していると思われる妥当な解決策としては、方針と同時に暫定的手段を考えることである。ここで暫定的手段とは、候補方針の個数と見合った精度のもので簡単に求められるものでなくてはならない。即ち、個数が多いほど簡単なものにする。更に、候補となる方針の数が多い時はこれを一回のパスで1つに決定するのではなく、数回に分けて徐々に絞り込んでいくようである。このようにすれば方針が決定された時点で必要な精度の手段も求められていることになる。これはいわゆる認識の段階的深化の原理の1つの現れと見てよい。碁世代では採用したケース概念で用意した手段（候補手）は言わば段階的深化の第一段目の暫定的手段に相当している。

また方針が定まってもこれを実現する方法は一意には定まらず、常に最適化の可能性がある。更に手段を決定する過程で、最初の方針の枠外で良好な手段が偶然発見される場合もある。この場合にはあえて最初の方針には拘らないのもよくあることである。

(4) 手段構成（複合事象の有機的統合機能）

人間の手段構成の方法は、完全情報二人ゲームプログラムでよく用いられている標準的な方法としてのゲーム木探索法に比べるとかなり様子が異なっている。そのうちのいくつかの特徴を挙げてみよう。

- 目的／標的指向性が強い読み筋：

標準探索法における静的評価関数の最大化といった全局的かつ抽象的なものではなく、具体的に目標を定めそれを追求する形態のことが多い。

- 方式決定と手段構成の一体化：
(3)において述べた様に、方針を段階的に絞りながら、手段を精密化していく。
- 複数回の読み筋プロセスの同時進行とその融合：
現段階では成立条件がそろわない読み筋や、いくつかの結果に選択の余地のある読み筋いくつか組み合わせて、手段を組み立てる。
- 目的／標的の途中変更：
読みの過程で、現在の読み筋の価値を上回る目的又は標的が発見された場合、急遽それに乗り換える。

3 人間プレイヤのモデル化に基づく「碁世代」プログラム

2節で述べた人間の機能を実現するために碁世代では、図1に示すようなモジュール構成を導入した。例えば、局面認識、探索は、それぞれ局面解析モジュール、探索モジュールで行い、方針決定と手段構成はまとめて着手決定で行う。また、総合インタフェースモジュールは対局モード、編集モード、実験解析モードの切り替えを行う。

以下いくつかの例について重点的に説明しよう。また、総合インタフェースの内、編集モード、実験解析モードについては第4節の開発ツールで述べることにする。

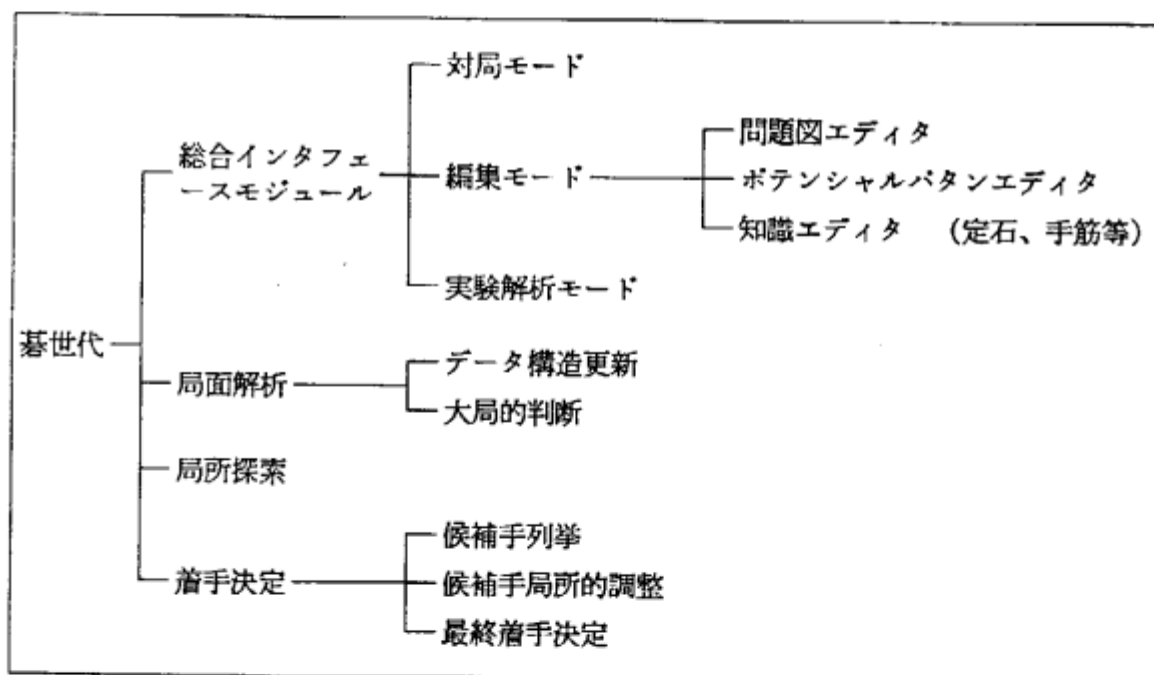


図 1: 碁世代モジュール構成

3.1 局面解析

3.1.1 データ構造更新

局面上の抽象対象は、そのままデータ構造を形成する。データ構造の更新は原則としてインクリメンタルな更新、即ち、最終着手によって引き起こされる変化部分のみに対して行われ、時間の節約をはかるが、場合によっては全面更新の方が楽な場合もある。

碁世代では2節の(1)で説明したように、点、連、群、族、結線の5種類のデータ構造を導入し、これにより局面を認識している。各データ構造の定義、機能及び主な属性を図2に示す。

名称	定義	機能	主な属性
点	19 × 19 の格子点	石の配置	色、ポテンシャル値、隣接点、各種候補手
連	隣接する同色の石の極大集合	捕獲の単位	石数、ダメ数、種石フラグ、死活タイプ
結線	同色の石、または石と盤端との間の仮想線	領域の境界 連結の単位	結線種別、連結種別
群	強連結した同色の石の極大集合	死活の単位	石数、中地、目数、包囲度、手数、強度、重要度、分類
族	ポテンシャル一定値以上で隣接する点の極大集合	地の単位	石数、辺長、中地、包囲度、強度、重要度

図 2: 局面データ構造とその属性

各対象はそれぞれ属性を有している。例として、群の属性の1つである包囲度を取り上げてみよう。これはある味方群が相手の石によってどの程度包囲されたかを示す量であり、もともと曖昧な概念であるが、群の強弱を決定する1つの要因となる。

図3の局面は、中央の黒3石が両側の白に囲まれかけている例である。上方にもう1つか2つ白に打たれば完全に包囲されそうである。人間は、このような状況をどうやって認識しているのだろうか。

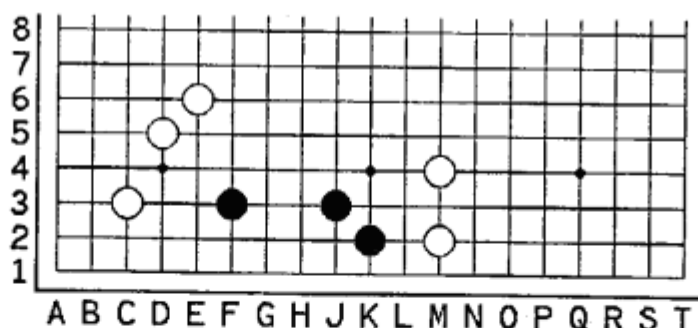


図 3: 包囲されかけている石の例

この方法を我々は図4のように考えた。群の各石から出発してそのダメ点へ次第に触手を伸ばしていく。ただし、この際一度通った点は2度通らず、石や敵結線の間の点（遮断点と呼ぶ。図6 参照）にぶつかったらそれ以上進めないことにする。つまり、囲まれそうな石の集団の各石から次第に視野を拡げていき、包囲網の空きがどれくらいあるか調べている。この時のダメ点を次の群ダメと呼び、4次の群ダメの数を包囲度と定義する（この4次の群ダメのことを単に群ダメと呼ぶこともある）。図の例では4次の群ダメは合計6個になるので包囲度は6となる。この数値が小さい程包囲度が強いことを示しており、包囲度0のことを完全包囲と呼ぶこともある。この時の（4次の）群ダメの点のリストは保存され、包囲／脱出の可能性やその候補手の位置を特定する時に利用される。（3.3着手決定参照）

包囲度の定義を図5に与えておく。図5の定義に基づき包囲度を算出するアルゴリズムをESP風に書いたものがアルゴリズム.1である。述語包囲度は、群のGoに対して、その群の包囲度を

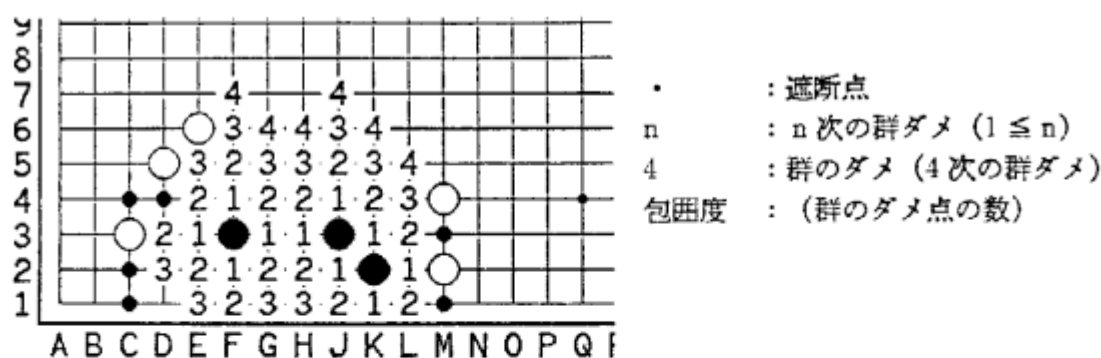


図 4: 包囲度算出例

包囲度	: 群のダメの数
群のダメ	: 4 次の群ダメ
n 次の群ダメ ($n \geq 1$)	: $n - 1$ 次の群ダメに隣接し、 $n - 1$ 次以下の群ダメでなく、空転で、敵遮断点でもない ($n = 4$ の場合は盤端でもない) 点の集合
0 次の群ダメ	: 群内の石の点

図 5: 包囲度の定義

求め E にユニファイする。述語群ダメは群 Go の次の群ダメの点のリストを Gds とユニファイする。述語群のダメのダメと述語群のダメによって、次の群ダメの点リストから $n+1$ 次の群ダメの点リストを求めている。

```

包囲度 (Go,E):- 群ダメ (4,Go,Gds),size_of_list(Gds,E);

群ダメ (N,Go,Gds):-
N>=1,! , 群ダメ (N-1,Go,Gds1), 群ダメのダメ (Gds1,N,Gds);
群ダメ (0,Go,Gds):- 群の石 (Go,Gds);

群ダメのダメ ([ ],_,[ ]):-!;
群ダメのダメ ([Gd|Gds],N,Gdsn):-
群ダメのダメ (Gds,N,Gdsn1), 隣接点 (Gd,AGds),
    群ダメ (AGds,N,Gdsn2),append(Gdsn2,Gdsn1,Gdsn);

群のダメ ([ ],_,[ ]):-!;
群のダメ ([AGd|AGds],N,[AGd|Gds]):-
    群ダメ条件 (AGd,N),!,
    群ダメフラグセット (AGd,N), 群のダメ (AGds,N,Gds);
群のダメ ([AGd|AGds],Gds):- 群のダメ (AGds,Gds);

群ダメ条件 (Gd,N):-
    群ダメフラグ (Gd,N0),N0 < N, 色 (Gd, 空),not(遮断点 (Gd)),

```

(N<=4,!;not(盤端(Gd))) ;

アルゴリズム 1: 包囲度算出アルゴリズム

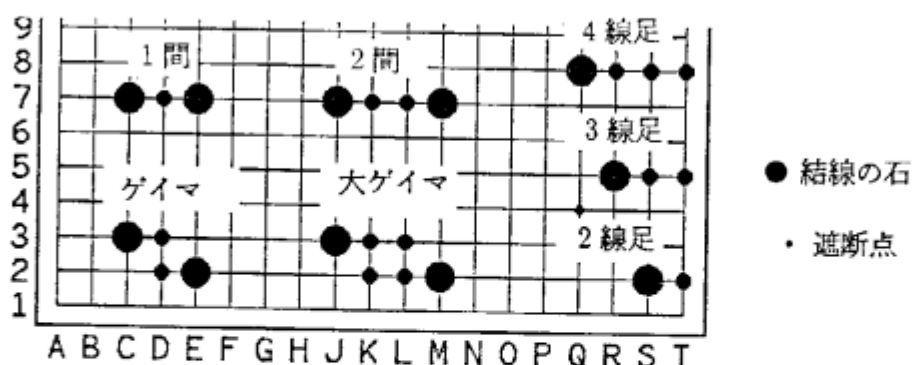


図 6: 遮断点

データ構造更新は1パスで終了しない場合がある。即ち、個々のデータ構造間の矛盾を調整したり、より深い認識に導いたりする為に、更に深い更新を行う。ここでは捕獲連調整と死群調整に関する例を取り上げてみよう（図7参照）。

名称	駆動条件	主な機能
捕獲連調整	捕獲可能な隣接異色連の発生	攻め合いを考慮した捕獲探索の駆動
死群調整	死群の発生	隣接敵群の融合、ポテンシャルのリセット

図 7: データ構造調整

捕獲連調整とは、局所的には認識できなかった連の捕獲の状況を少し視野を広げて見直すものである。死群調整は群が死んだことによって生ずる波及効果を他のデータ構造に反映させるものである。例えば、図8のような石の配置では中央の黒白の3石はともに死にであると判定されてしまう。隣接している黒白の石の一方が死ねばもう片方は生きである方が普通であり、ともに死にとなることはありえない。これは連の認識時に駆動される探索（3.2.1参照）は単一意図における局所的な探索であるので周囲との攻め合い関係についてはあまり深く読んでいないのである。このような場合、捕獲探索の視野をもう少し広げて、同程度の弱さの敵石があればそれを攻める手も全て探索するようにすればより正しくなる。図8の例ではこのより視野を広げた探索によって両方の石は中立（先手なら生きで後手なら死に）であることが判明する。

最初から視野を広げた探索にすればこのような手間は要らないがコストが掛かりすぎる。焦点を絞って必要とあれば段階的に視野を広げていく方が効率がよく実用的である。

次に図9の局面は死群調整の例である。一見右側の白の一团は弱そうに見えるが、中央の黒5石が死んでいるので安全である。つまり、白は2つの石に分かれているように見えるが一つの石であると考えた方が現実には近い。しかし我々はこれら石の集団の生き死にを群というデータ構造を対象に考えようとしている。ということは、群の集合を決定する時に群の属性が必要になってしまう。そこで形の上でまず群を定義しそれぞれの属性をもとめ、その結果死んでいる群があればその周囲の敵群を融合し再度群を定義し直している。また、この時の死群の各石によって生じていたポテンシャルなどの属性は同時にリセットされる。

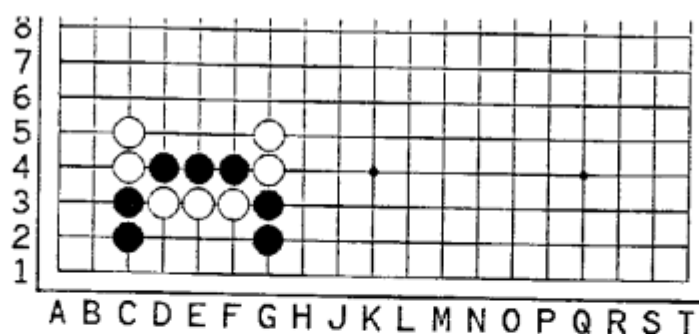


図 8: 捕獲連調整の例

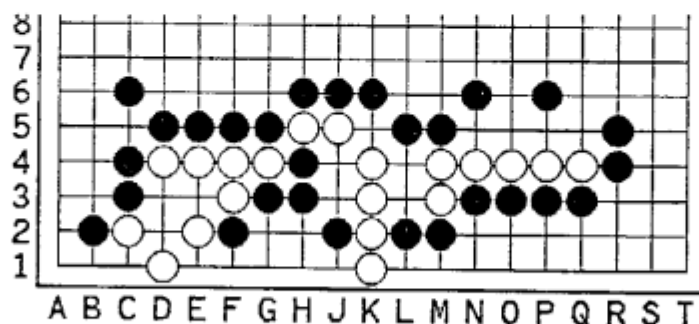


図 9: 死群調整の例

3.1.2 大局的判断

大局的判断は局面解析の1つの結果として得られるものである。ここで大局的判断とは、時相（序盤、中盤、終盤）、局相（模様、碁、戦いの碁、地の碁等）、形勢判断（優勢、互角、劣勢）等を指す。候補手の生成に、直接に用いられることはないが、大きく方向づけを行い無駄な候補手生成を抑制したりすることに役立つ。ここでは例として形勢判断を取り上げて説明する。

形勢判断の結果は主として攻勢、守勢の度合を決めるのに用いられる（3.3.3最終着手決定参照）。碁世代では地合い計算の仕方に関して、基本的に中国ルールを採用している。ここで中国ルールは、日本ルールのように、黒（白）石で囲まれた空点領域のみを計上するのではなく、境界も含めて、最終的に黒（白）の領域となるべき点の集合を全て計上する方式である。即ち黒（白）の領域とは、最終的に白（黒）が生存できない点の集合をいう。図10の局面を例にして説明しよう。

まずしっかり生きていると思われる石の集団とその中の地はそのまま計算して構わないであろう。また死んでいる石がそのまま相手側の領域とする（日本ルールのようにアゲハマの考えはない）。問題はそれ以外のまだ確定していない模様部分をどう扱うか、である。これに対する基本となるアイデアは、石の周囲に出来る影響力を特別に設計したポテンシャル関数を用いて近似的に表現し、盤上の全ての点についてポテンシャル値を求め、これをその点の石の占有確率（あるいは等価目数）に変換することである。なぜなら、ポテンシャル値が高い程その点に対する一方の色の勢力が強く、中国式に従えばその点が目となる可能性が強いからである。図11は以上の考えに基づいてその形勢判断の方法をまとめたものである。また、ポテンシャルに対する、その等価目数の関数は、図12のように決めた。

図10の局面に対して適用した結果をを図13に示す。図の中で○印は白地、●は黒地と判定された空点を表している。これと置いてある石の合計の目数が黒／白の地の目数になる。

それ以外の空点（+）のところはその点のポテンシャル値に応じた模様となっている。

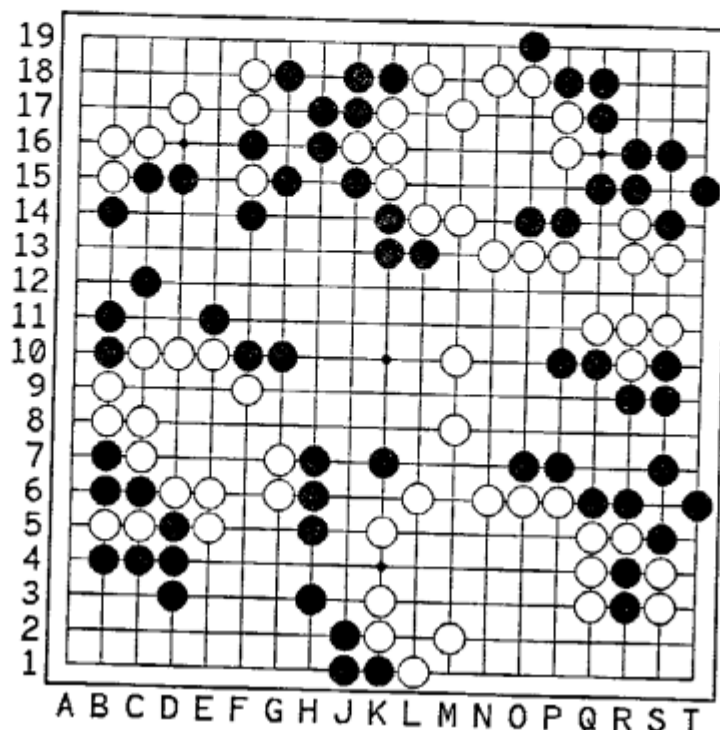


図 10: 形勢判断の為のある局面

3.2 局所探索

3.2.1 局所探索の機能と実現方法

局所探索を行う場面はいくつかあるが、プログラム作成の立場から見れば、これらを統一的に扱うことが出来る。即ち、探索局面と探索すべき目的（標的）、その時の手番を指定すれば問題状況は定まり、従って原理的には局面、目的（標的）、手番をパラメータとするツールを1つだけ設計すれば良い。しかし実際には目的によって候補手生成や終結判定のためのヒューリスティックがまちまちであるため目的毎に個別に作成した方が高い機能が得られる。目的毎の探索モジュールの種類を図14に示す。ここで標的となる対象のタイプは、例えば捕獲に対する連など目的毎に確定している。

探索アルゴリズムはアルファ・ベータ法を採用した。探索は読み切り型であり、結果は成功か失敗に分類される。一般に探索ルーチンの行使は高価であり、頻繁に使用すれば時間とメモリを浪費する原因となる。従って探索ルーチンを駆動する条件（図14）を設ける必要がある。この条件は探索の必要性をチェックすると同時に探索の範囲を限定することにも役立っている。チェック手続きは限定の目的を果たす範囲内で出来るだけ簡単なものが望ましい。アルゴリズム2.にネガマックス表現によるアルファ・ベータ探索アルゴリズムが与えてある。このアルゴリズムの中で候補手を生成する (generate) 部分とその局面を評価する (test) 部分のみが探索の目的毎に異なっている。逆に言えばこれらを定義することが探索全体を定義することになる。完全探索型の「シチョウ」とヒューリスティック探索型の「捕獲」を例にそれらを説明しよう。

黒の形勢 = (黒地の目数 + 黒模様 の等価目数)
 - (白地の目数 + 白模様 の等価目数)
 - 置石数 - コミの目数

白の形勢 = - 黒の形勢

ただし、黒地とは以下の点 (白地も同様)

- ・ 死群以外の黒石とその内点
- ・ 死群の白石とその内点
- ・ 隣接点のポテンシャル値が全て一定数以上の点
- ・ その点のポテンシャル値が 100 以上の点

また、模様とは地以外の点で、模様の等価目数とはそのポテンシャル値から等価目数関数 (図12参照) によって模様の価値を目数換算した値

図 11: 形勢判断の方法

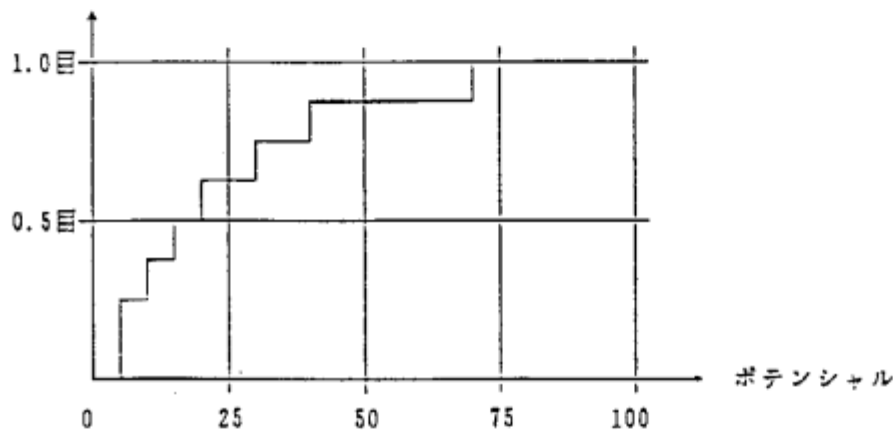


図 12: 等価目数関数

```

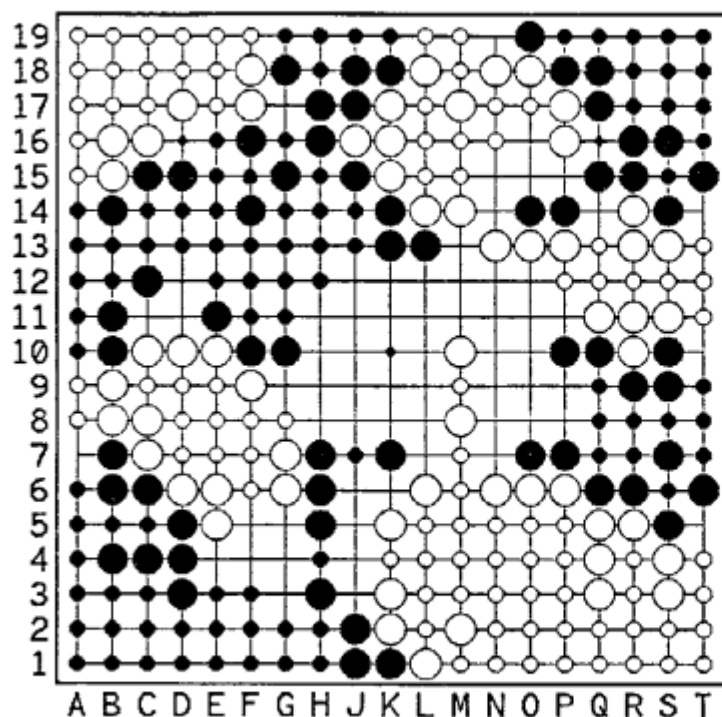
ネガマックス (P,_,_,V,_) :- 終結局面 (P,V),!;
ネガマックス (P,A,B,V,M) :- 候補手 (P,Ms), 最良候補手 (Ms,P,B,A,_,V,M);

最良候補手 ([ ],_,_,V,M,V,M) :- !;
最良候補手 ([M|Ms],P,B,V1,M1,V2,M2) :- 着手 (M,P,NewP),
    ネガマックス (NewP,-B,-V1,MV3,_), V3=-MV3,
    (M3>=B,! ,V2=V3,M2=M3;                                     % ベータカットオフ
    M3>=V1,! , 最良候補手 (Ms,P,B,V3,M,V2,M2);
    最良候補手 (Ms,P,B,V1,M1,V2,M2));
  
```

アルゴリズム 2: ネガマックス表現によるアルファ・ベータ探索アルゴリズム

(1) シチョウ

シチョウはアタリの連続で石を捕獲する手段のことである。例えば図15のように、攻め方はターゲット石の2つあるダメの一方を打つことによってアタリを掛け、守り方は残ったもう一方を打つことによってアタリから逃れる。その繰り返しの末、守



形勢判断

	黒	白
地	149 目	
模様	8 目	6 目

白半目有利です。

図 13: 形勢判断の例

目的	標的	駆動条件	結果
シチョウ	連	ダメ数 ≤ 3 の時	生き、死に
捕獲	連	ダメ数 ≤ 3 の時	生き、死に、不明
連結	結線	結線の周囲に敵石がノゾキにきた時	連結、切り違い、ツキスゲ
死活	群	完全包囲で包囲領域内の空点数 $\leq N$ の時	生き、コウ、死に
攻め合い	群対	死活駆動条件を満たす隣接異色群がある時	生き、コウ、死に

図 14: 局所探索モジュールの種類

り方にアタリから逃れる手がなくなり捕獲されるものである。図16や図17は囲碁用語ではそれぞれ「オイオトシ」、「ウツェガエシ」と呼ばれるが、これらもアタリの連続で捕獲できるので、シチョウの1種と言える。

シチョウの定義から、終結判定や候補手は必然的に決められる。図18がシチョウの終結判定である。終結局面における評価値の1はその手番にとっての「成功」を意味し、反対に-1が失敗を意味する。つまり、攻め番の時はダメ数2で守り番の時はダメ数1である限り探索が続き、その条件からはずれた時成功または失敗で終結する。

図19はシチョウの候補手である。

攻め方候補手の(1)は例えば図15の1、3、5手目のように守り方が打てば連のダメ数が4以上になる手である。この手がある時他の手へ打っても失敗するのは明らかである。(1)は(2)に含まれるので特に必要ではないが、あった方が効率はよい。

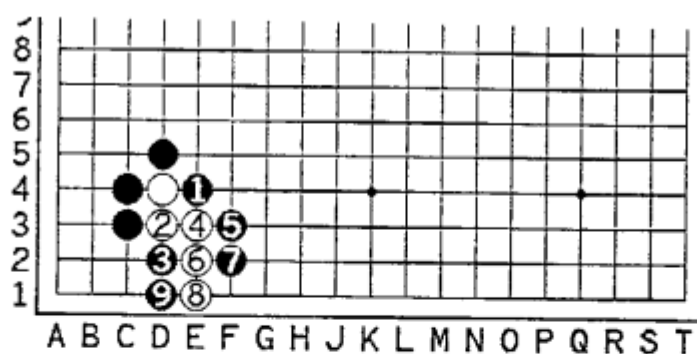


図 15: 典型的なシチョウ

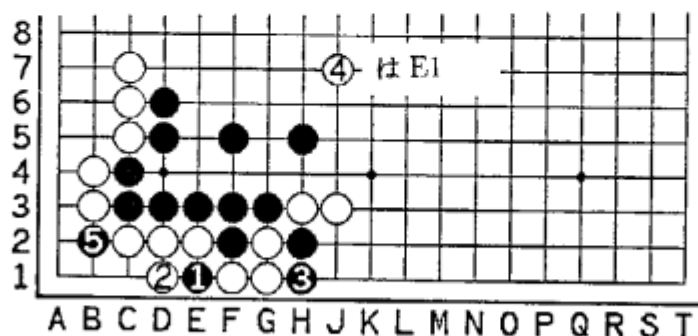


図 16: オイオトシ

これによってシチョウはほとんど一直線の読みとなる。また、守り方の候補手の(2)によって、守り方は逃げる手ばかりでなく敵石を捕ることによって自分を守る手も打つようになる。これによって前述のオイオトシやウツテガエシもシチョウと同様に解けるようになる。

(2) 捕獲

次に捕獲の終結判定と候補手について述べよう。捕獲は完全包囲を前提とする問題（詰碁または攻め合いの問題）以外の全ての問題を対象とする。

捕獲の問題の例を図20、図21、図22に示す。アタリの連続で石を捕るシチョウと比べて、制約は大幅に少ない。

これらの問題に対して爆発することなく妥当な（必ずしも正解でない）解を求めるには、人間が行っているのと同じようなヒューリスティックの導入が避けられない。即ち、終結判定を出来るだけ早くするヒューリスティックや、候補手の生成を極力押さえるようなヒューリスティックが必要となる。

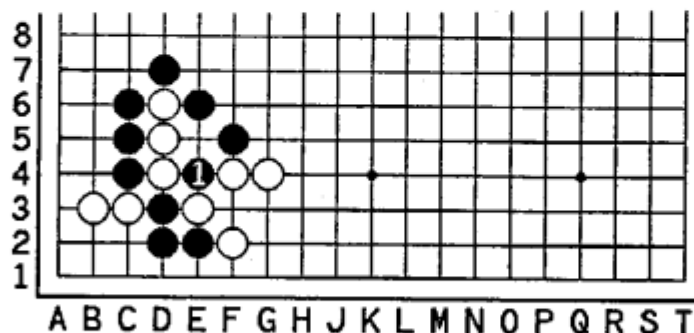


図 17: ウツテガエシ

次のいずれかの条件が満足される局面を終結局面とし、その評価値を返す	評価値
(1) 攻め番で標的連のダメ数 ≥ 3	-1
(2) 攻め番で標的連のダメ数=1	1
(3) 守り番で標的連のダメ数 ≥ 2	1

図 18: シチョウの終結判定

攻め方の候補手	守り方の候補手
(1) 守り方の打落によって標的連のダメ数が4以上なる点	(1) 標的連のダメ点
(2) 上記(1)の手がない時標的連のダメをツメる全ての手	(2) アタリの隣接敵連をヌク手

図 19: シチョウの候補手

図23に捕獲の終結判定のヒューリスティックを示す。この中で(1)は「標的連のダメ数が4以上になったら捕獲できない」としている。標的連の条件をダメ数3以下ではなく、もっと大きく設定した方が、当然、捕獲対象を広くカバーできるのであるが、例えばダメ数4以下とすると候補手の数はかなり増大し、解く所要時間は急激に増大するのに比べ、実用的に解ける問題の種類はそれほど増えない。(1)ではさらに制限を加えている。「ダメ数が3であっても、それを囲っている石の中にすぐアタリを掛けられるような石が3つ以上もあるようならその石は捕れない」としている。これも誤った解を出すことは余りなく、かつこれによって探索の効率がよく上がることが経験的に分かっている。また、これらによっても爆発の可能性がなくなる保証はない。(4)で一定の深さあるいは手数になっても解が求められない場合は「不明」を意味する評価値0を返し終結させている。

図24は捕獲の候補手ヒューリスティックを示している。候補手ヒューリスティックは強制手と通常手からなる。強制手が1つでもあればそれを候補手とする。なければ通常手の全てがその候補手評価値の大きい順に生成される。図20の「鶴の巣ごもり」の問題は強制手によって1手目が決められる。図21のゲタは通常手、図22のハサミツケは通常手によって正解手が候補手の1つとして生成される。どの問題もその後は通常手でダメをうまくツメていけば、シチョウに帰結する。なお、通常手の候補手評価値のダメ基礎点とは $10 \div (\text{標的連のダメ数})$ のことで、増分度とはその点へ

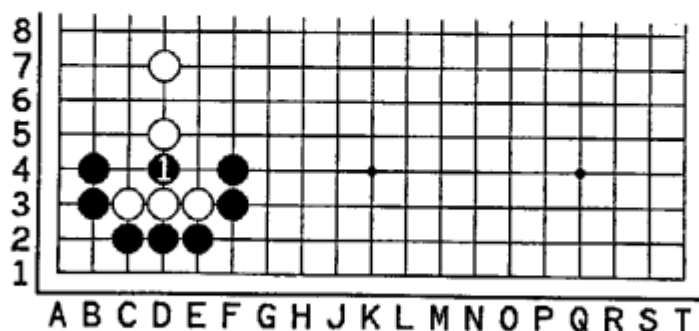


図 20: ツルの巣ごもり

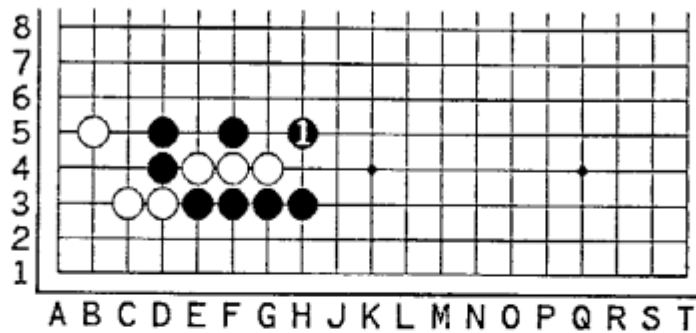


図 21: ゲタ

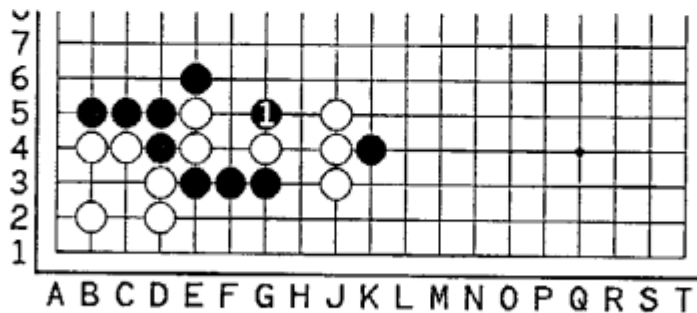


図 22: ハサミツケ

の打着によって増えるダメ数のことである。

3.2.2 局所探索に伴う問題点

局所探索に関しては、探索ルーチン自体の性能向上も重要であるが、それに劣らず大切なことは、ルーチンの活用の仕方である。以下に、いくつかの問題点について述べよう。

[1] 問題の検出と切り出し

問題の検出は先に述べた駆動条件のチェックで行なわれる。これは本格的に探索を行う前の調査の様なものである。問題がありそうであれば、問題のタイプに関する知識に基づき、問題を構成する要素を具体的に確定する。

[2] 探索結果の活用場面

探索の結果は様々な使われ方をするが、大きく2つの場合に分けられる。1つは、局面認識の補助手段として局面解析から呼ばれ、もう1つは、具体的に候補手を求める為着手決定のモジュールから呼ばれる。

次のいずれかの条件が満足される局面を終結局面とし、その評価値を返す	評価値
(1) 攻め番で標的連のダメ数 ≥ 4	-1
(2) 攻め番で標的連のダメ数 = 3 で、ダメ数 ≤ 2 の隣接敵連の数 ≥ 3	-1
(3) 攻め番で標的連がシチョウで取れる	1
(4) 探索の深さあるいは手数がある一定数以上に達した	0

図 23: 捕獲終結判定ヒューリスティック

攻め番ヒューリスティック	候補手評価値
【強制手】 (1) 守り方の打着によって標的連のダメ数が5以上なる手 (2) 標的連のダメ数3の時サイズ2以上のアタリの包囲連をツグ手 【通常手】 (1) 標的連のダメをツメる手 (2) アタリの包囲連をツグ手 (3) 標的連のコスミの手 (2 線以上) (4) 標的連から1 間の手 (2 線以上) ...	ダメ基礎点×増分度 9 10 8
守り番ヒューリスティック	候補手評価値
...	

図 24: 捕獲候補手ヒューリスティックの例

図25は捕獲を例にとった局所探索の係わる使用状況を示す。図の中の矢印は各モジュールが探索モジュールを呼びだしていることを示している。それぞれの目的/用途は以下の通りである。

- (1) 連の死活属性を求める（その結果は連の種石属性や群の分類属性決定時、及び捕獲/逃亡候補手列挙時に参照される）。
- (2) 隣接連間の矛盾をチェックし、もしあればより視野を拡げた探索を駆動する。
- (3) 終了判定や候補手の生成に利用する。
- (4) 切り違うまでの変化をボタンで処理し、切り違った連の死活を捕獲探索によって求め、その結果の総合判定により連結具合を求める。
- (5) 捨て石の場合を除き、着手候補を仮打ちした場合、それが捕獲されるようなら打たないようにする。
- (6) ある候補手が捕獲意図を持って提案された場合、その点の近傍に他のより高い価値の意図を持って、かつその捕獲の意図も満足する点があればそれに乗り換える。

また、これらの他に以下のものがある。

- (7) 捕獲練習問題の自己採点
- (8) 捕獲探索過程の表示

この例が示すように、他の種類のルーチンと呼んだり（シチョウルーチンを）、呼ばれたり（(簡易) 連結ルーチンから）することもある。

[3] 探索結果の管理

探索結果を一度利用しただけですぐ捨ててしまうこともある。また局面認識に用いられたような場合には、その認識した局面性質が持続する間は、認識するために必要はない。しかし、ゲームの進行に連れて探索結果を成立させる条件が変化すると、それ

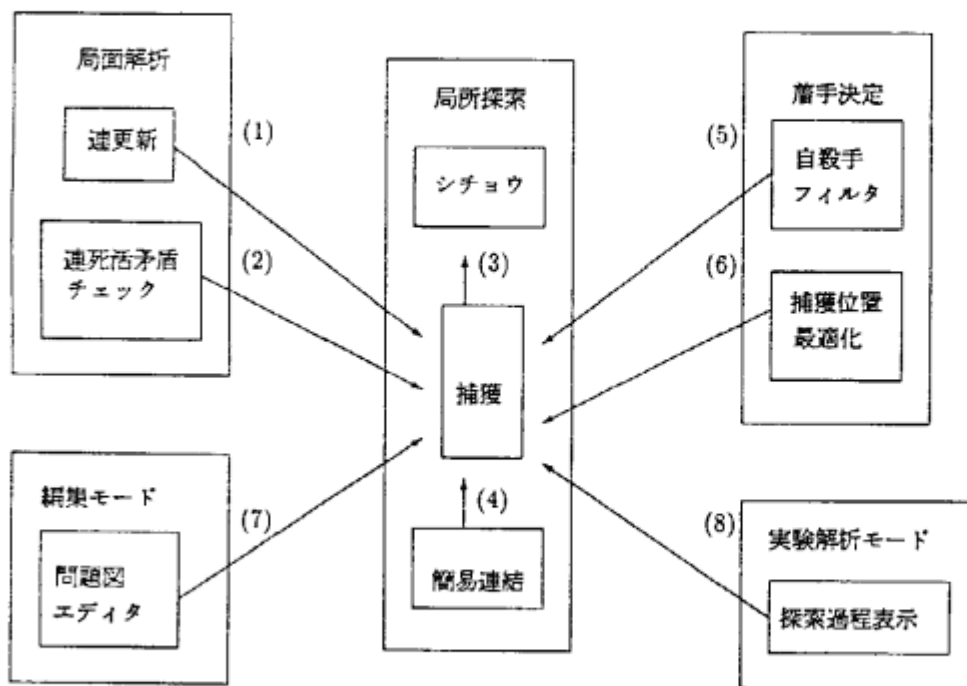


図 25: 捕獲探索に係わる処理の流れ

までの局面性質はもはや信用できないので読み直しが必要となる。それではどのような場合に、一度読んだ読み筋が破壊されるのであろうか。

これを調べようとする意外と面倒な問題を引き起こす。簡単に考えると、石の影響は大体その近傍に及ぶものであって最終着手の近傍に存在するデータ対象に関係したものだけ調べればよさそうである。しかしこのことは一般には成立しない。遥か離れたところで効くシチョウ当りはその典型的な例である。またある連の捕獲がセキ崩れを起こすような場合には、影響の範囲がかなり広範囲に渡り連鎖反応的に影響を受けることになるのが普通である。現在のところ、知識で補える範囲内で探索範囲を拡大している。

3.3 着手決定（方針と手段）

着手決定の流れは、一般的には、まず局面解析を行い、次にこれに基づき方針を立てる。そしてこの方針を実現する手段の可能性と効果の程度や副作用を含めた波及効果を検討し、更に手段を具体化精密化し最終的に着手を決定する。

図26は碁世代における意思決定の流れを示したものである。

3.3.1 候補手列举

与えられた問題解決の場で人間は判断の基準として、過去に遭遇した典型的状況に関する知識に基づき行動している、という仮定を設ける。「碁世代」ではこの典型的状況の概念を明確にする為に、ケースと呼ぶことにした。即ち、方針を決定する着眼点の選出は、碁の知識によって設定されたいくつかのケース（典型的状況）の観点から行なわれる。各ケースについてその特徴

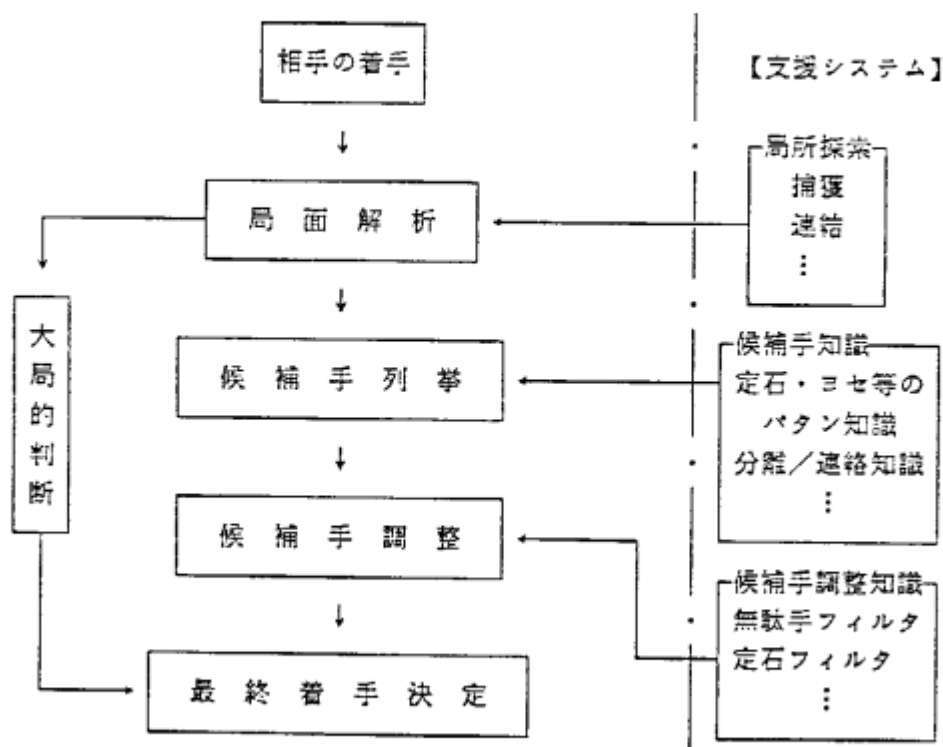


図 26: 基世代のモジュール構成と意思決定の流れの対応

を定義し、それにより現在どのようなケースが生じているか判定出来るようにする。また同時にこのケースが検出されたときこれを処理する手段としての候補手（以下これをケース候補手と呼ぶ）、及びその手の常識的な価値を示す評価法も与える。これにより（特に相手の）着手意図やその価値を理解出来るようになる。ケース候補手の種類を図27に示す。ケース候補手の例として包囲/脱出の手とフトコロ拡大/縮小の手について述べよう。

包囲/脱出の手は図4 のようにある程度囲まれていて、まだ完全には包囲されていない群について考慮される。図から明らかなように、逃げる方向としては（4 次の）群ダメの中心あたりがよさそうである。ただし余り遠くでは後で切断されるキズを残すことになるし、近すぎでは脱出の意味がうすい。そこで「逃げるは一問」という囲碁の格言が参考になる。即ち、以下のように決めた。まず、群ダメの中心座標が求められる。その位置が包囲/脱出点の1次近似となる。その点に最も近い群の点を（包囲/脱出の）基点と呼ぶ。基点より一問の点、コスミの点、あるいはノビの点の内、群ダメの中心座標に最も近い点を包囲/脱出点とする。ただし、その点が1問の点でかつノゾかれている場合には再度コスミかノビの手に補正される。図4の例では群ダメの中心座標が J6、基点が J3 となり、包囲/脱出点は J5 に決められる。また、その評価値については群の強度やサイズなどにより決められる。

つぎにフトコロ拡大/縮小候補手について説明しよう。フトコロ拡大/縮小の手には、例えば図28の×の点のような手である。これらの手は通常でも地を拡大/縮小する手であるので、それなりの価値はある。石の死活がからんでくると価値はより高くなる。例えば図中の H2 の点は現在ではそれほど価値はないが、左側の黒石の上方を白に包囲されると途端に価値はあがる。このようにフトコロ拡大/縮小の手は、そのボタンの検出については局所的に処理できるが、その価値は常に全局的な関係によって決められねばならない。従って、フトコロ拡大/縮小候補手は以下のように2段階の処理によって更新される。打着手後、まずその近傍の点についてフトコロ

ケース名称	特徴	内容
定石	隅のボタン	定石の手(変化図)
辺点	辺に沿ったボタン	ヒラキ、ツメ、ワリウチ等の辺上の手
ダメ点	接触する黒白のバタ	ハネ、ノビ、オシ等の石の競り合いの手
ウチコミ	隅の星のボタン	三々へのウチコミと防御
ヨセ	主に辺のボタン	ヨセの手
フトコロ拡大/縮小	主に辺のボタン、弱群	弱石のフトコロに関する手
模様接点	ボタン、隣接族	両ゲイマ等の模様の接点の手
捕獲/逃亡	ダメ以下の連	石をトル/ニゲル手
連結/切断	ノゾキのある結線	ノゾキに対するツギ/キリの手
包囲/脱出	小さい包囲度の弱群	弱石を包囲/脱出する手
分離/連絡	隣接する同色の弱群	石を分離/連結する手
地模様拡大/削減	ノゾキのある族	模様につたをしたり、侵入する手

図 27: ケース候補手

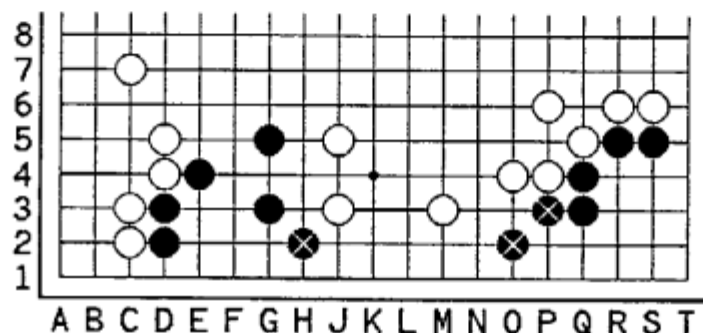


図 28: フトコロ拡大/縮小候補手の例

拡大/縮小の各種ボタンのマッチングを行い新たな候補手の追加/削除が行われる。つぎに盤面全体のフトコロ拡大/縮小候補手について評価値に変動がないかチェックされる。

フトコロ拡大/縮小の候補手知識は図29のように、候補手生成と評価値算出2つの規則とインスタンスデータからなる。候補手生成規則は図28の各ボタンも含めた多くの局所的ボタン規則からなる。評価値算出規則は評価値算出時の規則であり、局面解析によって得られる全ての情報を条件文に記述することができる。

3.3.2 候補手局所的調整

候補手調整は異種ケース間の不調和を局所的に調整してより効果を高めたり、または削除したりする機能である。一般にあるケースはそれに隣接したり、場合によっては混在するケースと独立に候補手を作成している。このことに起因する意図の不統一を調整するのが主な目的である。候補手調整には、図30のような5つの機能がある。例として死群フィルタと捕獲位置の最適化を見てみよう。

まず、死群フィルタの例(図31参照)では右と左に分離された形で白が黒に囲まれている。左の群はまだ死にきっていないので白はコシミの候補手を2カ所提示しているのに対し、右の群は死群と認識されたため、左群と同じパターンにも拘らず候補手は生成されない。尚この手の意味はヨセの意味と自己弱群を守るためのフトコロ強化の意味を兼ねており評価点もその合計と

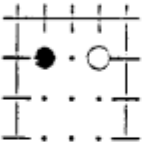
候補手生成規則	評価値算出規則
標準パターン1 :- 碁方石(-1, 1), 敵石(1, 1),  空点([(-1, 0), (-1, -1), (0, 1), (0, 0), (0, -1), (1, 0), (1, 1)]), 空端(0, -1), 候補手生成((0, 0), 1); 標準パターン2 :-, :	1 :- 基本点(黒白両方, 5), (弱石(-1, 1) ならば 攻守点(黒白両方, 10)); 2 :-, 候補手インスタンスデータ 基準座標 : H 2 色座標変換パターン : 0 候補手座標 : H 2 評価値算出法 : 1

図 29: フトコロ拡大/縮小の候補手知識

名称	主な機能
無駄手フィルタ	石の配置から候補手の意図の実現が無理と思われる手を取り除く
定石フィルタ	定石が進行しているその隅の他の候補手は取り除く
死群フィルタ	死群であれば意味を成さない候補手について死群になれば取り除く
自殺手フィルタ	着手決定された手が捕獲される場合意図的な場合を除いて取り除く
捕獲位置最適化	いくつかの手が同じ捕獲の意図を満たす場合 その中で最も状況に適した手を選ぶ

図 30: 候補手調整

なっている。

次に捕獲位置最適化の例(図32参照)では、手番は黒で D3 の 1 石を捕獲から守らねばならない状況である。捕獲ルーチンに問い合わせると最初 E3 の手を推奨してきた。一方この手の近傍を見ると D2 の点はフトコロ強化の意味があり、捕獲ルーチンはこの手でも逃げの目的は果たしていることを保証した。従ってニゲ評価点は D2 につけられ結局 E3 より D2 の点がより高い評価点で推薦されることになる。

3.3.3 最終着手決定

各ケースが互いに独立に挙げてきた候補手を候補手調整で局所的な調整を行い、最後に大局的な判断から次の一手に絞り込まれる。このプロセスはスコアテーブル(碁盤に相当する 19 × 19 マトリクステーブル。図33参照)を用いて実現される。

即ち、スコアテーブル上に各ケースからの候補手の評価値を記入する。このとき同じ点へ重複した記入があればその値は単純に加算される。そして原則として最高点を得た点を次の一手として決定する。図33のスコア表は白の候補手としての評価点である。例えば Q15 のカケは R16 にある黒の群を包囲する意味と自己模様を拡大する意味が複合している。またこの局面では D10 が最高の評価点となっている。

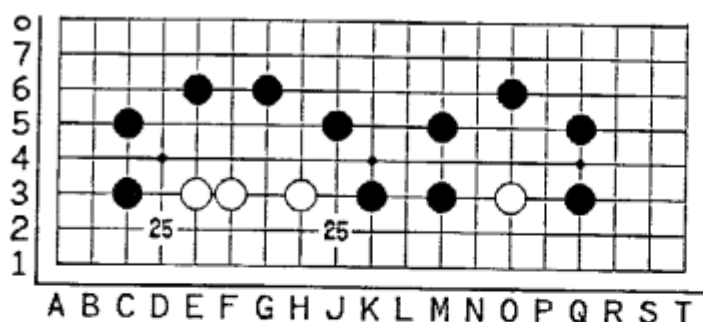


図 31: 死群フィルタ

候補手 D2(J2) の内訳	
フトコロ評価値	15.00
ヨセ評価値	10.00
合計	25.00

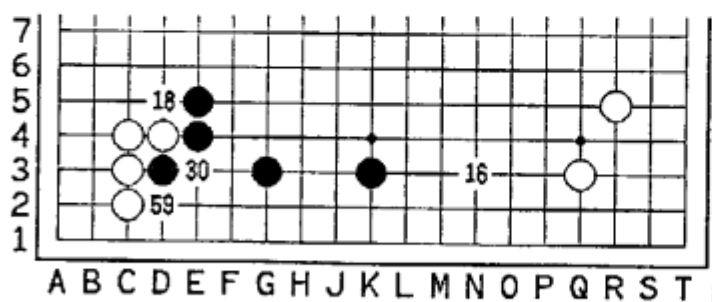


図 32: 捕獲位置最適化

候補手 D2 の内訳	
共有ダメ評価値	7.00
ツナギ評価値	15.00
ニゲ点評価値	23.60
フトコロ評価値	13.00
合計	58.60

スコア表の最大評価点を次の一手とするのはあくまで原則的であって、大局的判断観点または大局的戦略知識により上記の決定方式に修正が行われる場合がある。一般に形勢判断の結果優勢ならば守りの手でなるべく紛れの無い手を選び、逆に劣勢ならば紛れを求めて攻めの手を選ぶようにする。このとき局相によっても着手選択の範囲が限定される。

例えば模様の手、優勢ならば、自己模様の中心で相手の打ち込みを未然に防ぐ手を選ぶか、または相手模様を囲わせながら自己模様を囲う手を選ぶ。このとき例えば自己弱石があっても、それが捕られたと仮定しても形勢に関係の無い大きさであり、かつ逃げる方向が味方の大模様に向かう方向であれば優先度を下げることが行われる。弱石が捕られると形勢が逆転するほどの大きさならば極力味方模様に近付かないで生きる手を選ぶようにする。また特に相手が自己の最大の模様の中に打ち込んだ場合は一段落するまでその石を攻める事に専念する。

一方模様の碁でも劣勢の時は、相手模様に打ち込みを敢行したり、相手模様の境界から侵入をはかったりする。勝敗に関係しない程度の小さい弱群を攻め過ぎないようにする。

以上のような大局的判断を全ての場合について網羅することは困難であるが、典型的な場合について用意するだけでも相当棋力の向上を計ることが出来る。

4 開発ツール

開発ツールは、囲碁プログラム本体と直接の関係はないが、この整備如何によって開発期間を大幅に短縮出来たり、また間接的にプログラムの質を向上するのに大きく役立ち、見かけ以上に重要な部分である。特に囲碁プログラムは試行錯誤的に改良を加えることを余儀なくされており、プログラム自体も巨大化する傾向にあるので、開発ツールの良し悪しがプロジェクトの成功失敗の鍵を握るといっても過言ではない。総合インタフェースモジュールでは、対局モードと開発モード（実験解析モードと編集モードがある）の間を実行時に自在に行き来できるように設計

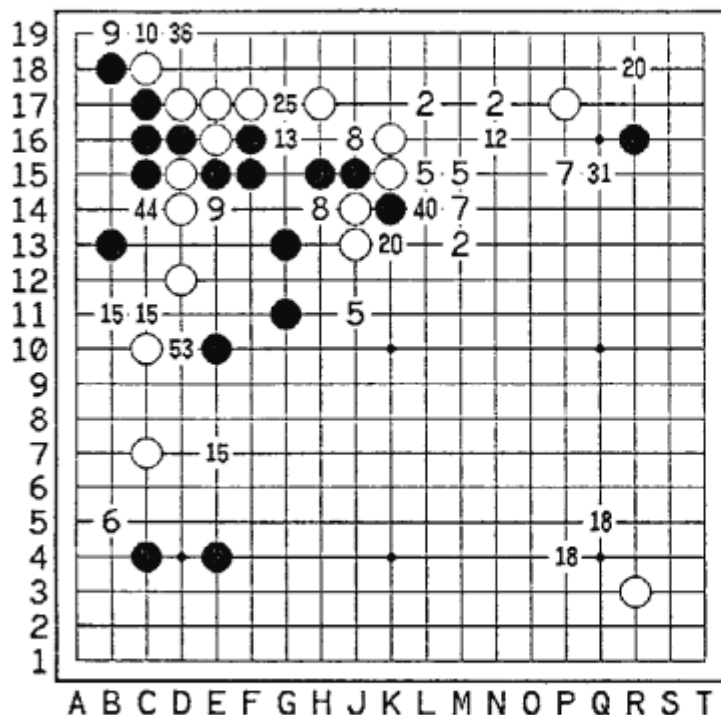


図 33: スコアテーブル

してある。これもデバッグの効率向上をはかる上で重要なことである。これまでの経験で得られた必要と思われるいくつかのポイントについて述べよう。

4.1 実験解析モード

これはいわゆるデバッグを行うモードである。対局中に問題が生じたときでもその状態を直接調べることが出来るように対局モードと実験解析モードはオンラインで任意に切り替えが出来る機能を持つことが望ましい。実験解析モードではデータ構造内の全ての対象及びその属性が覗ける peep 機能が必要である。更に問題を生じたサブモジュールの trace 機能も必要である。実験解析モードのモジュール構成を図34 に示す。

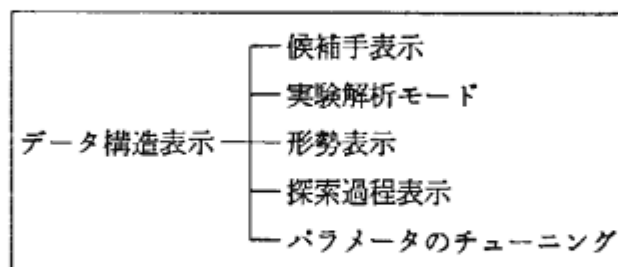


図 34: 実験解析モードモジュール構成

以下、各モジュールの機能について述べよう。

(1) データ構造表示

局面解析によって得られた各データ構造の分布やその属性を表示する。

(2) 候補手表示

候補手列挙及び候補手調整によって得られたスコアテーブルの内容を表示する。また、個々の候補手がどのような意図から成り、かつそれぞれの意図がどれぐらいの価値を持っているかを等価目数換算した値で表示する。

(3) 形勢表示

形勢判断によって得られた形勢を表示する。

(4) 探索過程表示

各種探索モジュールを呼びだしてその探索過程と結果を表示する。また、探索における候補手の優先順位も表示可能である。

(5) パラメータのチューニング

局面解析や候補手列挙などの知識は、多くのパラメータを含んでいる。これらのパラメータを妥当に設定することは知識の精度を上げる為に非常に重要である。これらのパラメータをダイナミックに変更して、局面の認識具合や候補手の変化の具合を見る事ができる。

4.2 編集モード

実験解析のための問題を作成したり、定石や手筋の知識を導入するためのエディタ等からなる。編集モードのモジュール構成を図35に示す。

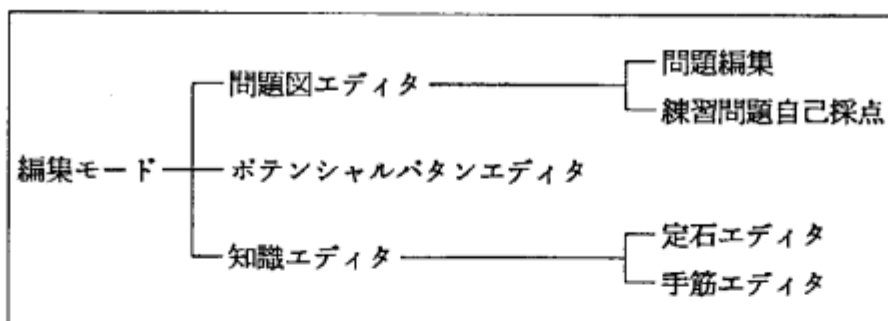


図 35: 編集モードモジュール構成

以下、各モジュールについて述べよう。

(1) 問題図エディタ

デバッグの為にいろいろな局面を作成したり、保存／再生したりする。また、それらの局面を問題として自己採点する機能がある。自己採点には、捕獲や詰碁などの探索問題や、「次の一手」などの着手決定の問題、また石の強弱の判定など局面認識の問題が登録できる。また、自己採点中に、内部パラメータを変更しながら解の変化を見ることが出来る。

(2) ポテンシャルボタンエディタ

ポテンシャルは石がその周囲に与える影響（厚み）を表すものであるが、隅や辺の影響を考えると、一律には決めにくい。本システムでは、石の盤上の位置に対応していくつかのポテンシャルのボタンを用意し、そのボタン毎のポテンシャル分布をエディタで自由に設定できるようにした。

(3) 知識エディタ

定石エディタと手筋エディタからなる。定石データは1つの大きな木構造から成っており、その分岐点に定石の選択条件（周囲の厚みやシチョウ関係など）設定できる。

手筋データはいくつかの独立した手筋のデータの集まりで、類似した適用条件毎にファイル化されている。個々の手筋データは、適用条件（時期、場所、石の配置など）と、マッチング時の手順（1手の場合もある）及びその評価値決定ルールから成る。

なお、実験解析モードや編集モードの表示例で、本文中に現れなかったものについては、いくつか巻末に付したので参照されたい。

5 現状の評価と問題点

第1節において人工知能のメインテーマとして5つのテーマを挙げた。最後の学習を除く4つのテーマの観点から碁世代がどのような問題を解決したか、また未解決のまま残しているかについて述べよう。

(1) 探索

最大のポイントは全局的な探索を捨てたことである。探索パラダイムが適用可能な問題空間には限界があり、チェスは恐らくその限界にあるものと推定される。チェスと同規模またはそれ以下の問題は探索パラダイムによって十分人間の最高のレベルに対抗できる能力を発揮出来るプログラムが開発できるであろう。囲碁はチェスの問題空間の規模を遥に越えている。

次のポイントは認識の手段として探索が不可欠であることを認め、その代わりに単意図、単一標的の局所的探索ルーチンに限定したことである。しかしこの様な限定は、やはりいくつかの限界を生み出す。当然のことながら二重ターゲット（図36）、二重意図の問題（図37）が処理できなくなる。また局所性を強調すると例えば図38の様な問題が解けなくなる。即ち、二重標的の問題は、黒 F1 にサガれば E3 の連か F4 の連のどちらかが捕れる問題である。二重意図の問題は、黒 B1 にサガれば隅での生きと右方への連結（ワタリ）のどちらかが成功する問題である。最後の問題は、D4 の連を捕ろうとしても読みの途中で一度はターゲット連のダメ数4以上になってしまう問題である。

第三のポイントは認識の段階的深化の考えを、組合せ爆発を回避するための資源有限な場合の普遍的なメカニズムとし、広範囲な探索が必要となる場面の前処理として積極的に導入を試み、局所的な探索のレベルに問題が分解されるまで探索ルーチンの適用を遅らせたことである。

(2) 曖昧

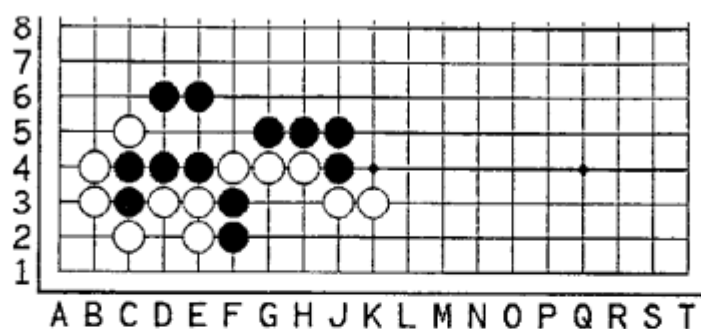


図 36: 二重ターゲット

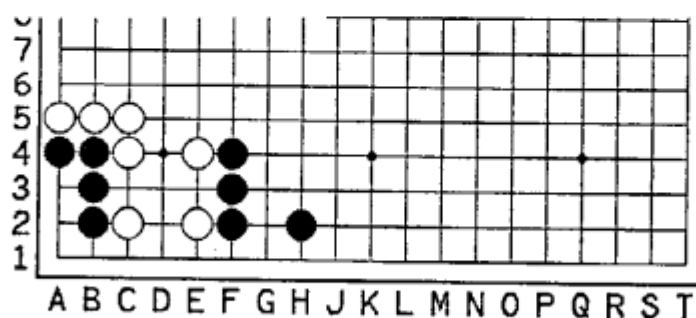


図 37: 二重意図

碁で用いられる概念は、正確な定義さえ困難な曖昧なものが大部分を占める。人間は定義抜きに以心伝的に修得し、そして同一概念に対する理解が個人毎に多少の食い違いがあっても、「実用上」はあまり問題とならない。ここで実用という意味は、碁の本を読んで理解したり、人に碁を教えるとき等を意味する。ましてブレイすること自体にはほとんど問題はない。

しかし一度プログラム化を行おうとすると、本来曖昧なものを無理矢理定義することを強えられる。例えば、群（族）の強度の問題をとりあげてみよう。人間にとって、強度の概念は本来曖昧であり、定性的相対的な意味しか持たない。即ち、着目している対象のみでなく、その周囲の石に配置、強弱関係によって定まる曖昧な量である。しかし「碁世代」ではこれがある具体的な数値で表示する必要があった。その為に便宜上、対象の最も大きそうな要因とみられる包囲度と領域サイズに着目し、

$$\text{強度} = a \times \text{包囲度} + b \times \text{領域サイズ}$$

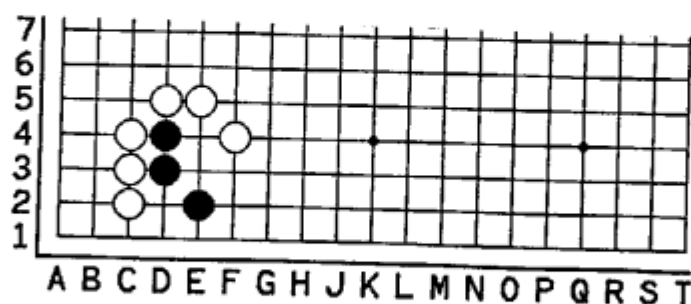


図 38: Dame数 4

なる関数で近似的に表現した。しかし、これはあくまで第1次近似である。従って、周臣群（族）の影響を考慮して、若干の補正を行っている。しかし、実際の碁に生ずる強弱の感覚からはまだかなりずれがある。

(3) 例外

例外処理を可能にするには、少なくとも例外、または非常事態の定義を明らかにしてプログラムがこれを認識できるようにしなければならない。例えば「形勢判断が不利でかつ定常処理ではこれを回復するよい手がないとき、または二か所以上で攻撃を受けこれを同時に防ぐ手が定常処理に無いとき」と定義すれば、非常事態のかなりの場合をカバーできる。

本来、相手の攻撃に全く気がつかないか、または甘くみて非常事態に至ることが多い。従って形勢が急激に変化した時も、非常事態検出の1つの方法である。しかしこのタイミングで非常事態を検出すると常に後手に回ることになるので相手の攻撃に対して手抜きして相手がもう一手仮打ちした図を想定し現在局面との形勢の変化を調べるのが妥当であろう。

(4) 協調

碁における有機的統合能力とは、狭く解釈すればいくつかのアジを組み合わせで新しく手段をつくり出す能力と考えてよい。

6 おわりに

囲碁システム、碁世代は、第五世代コンピュータプロジェクトの中期計画の一環として、昭和60年より開始された。ICOTでは、日本においては、欧米と比べこの種の研究の蓄積が極めて少ないこともあり、この分野の研究者を集めたワーキンググループ (CGS-WG) を組織するとともに、この分野での研究実績のある電子技術総合研究所との共同研究を行い、この研究を推進することとした。

昭和60年度においては、まず、その前半で、碁世代システムのベースとなったデータ構造やポテンシャル計算の方法などの基本方式を定めるとともに、PSI上にESPを用いてソフトウェア試作を行う方針を定め、基本となるソフトウェアモジュールの概念設計を行った。また、アタリ、アゲハマ、地計算などの機能を含む碁盤ツールを中心に、実際にソフトウェアの一部を試作した。

昭和60年度後半においては、次のステップとして、人間対人間の対局を追跡し、その局面をシステムの内部表現としてデータ構造の形で取込み、局面や局相の判定を行うことを目指した。このため、システムの基本構成要素となる対局モジュールのうち、入力、表示などを行うメインモジュール局面解析モジュール、局相判定モジュールなどの設計と試作を行った。これと、並行して、種々開発ツール、すなわち、実験解析ツール、手順木解析ツール、問題図作成ツール、ポテンシャル値チューニングツールなどの設計と試作を行った。

昭和61年度においては、人間と計算機との対局を可能とすることをめざし、捕獲、包囲などのアルゴリズムやその実装手法の検討を行った。また、死活の判定についても研究を行ったが、これは特に困難な問題であった。これらの機能を含む着手決定の簡易版プログラムが年度末に完成し、PSI上で動作するようになった。

昭和62年度には、データ構造および候補手知識に対する改良と機能の拡張を行ったほか、知識間の協調性の強化をはかった。また、詰碁の検討も行ったが、システムの改良、拡張は、計算量を増加させ応答速度の面での限界に近づいたため、碁世代システムへの組込みは行わなかつ

た。この詰碁については、別のシステムとして実装すると共に、並列推論マシン上での並列システムとしても試作を行っている。その後、PSI-Iの小型高速化版PSI-II上に移行し、システムのチューンアップを行った。

現在第五世代コンピュータプロジェクトでは並列推論マシンPIM (Parallel Inference Machine)を開発中であり、それに伴い本システムも並列化への移行作業を進めている。並列化によって期待される効果は、まず、処理能力の大幅な増大である。これまで、述べてきたように、本システムは局面の認識段階において局所的な探索を行うことはあっても、また、知識や認識結果間の協調が一種の試行錯誤的な振舞いをすることがあっても、コンピュータチェスなどと比較すれば全局的には1手読みのシステムと言わざるを得ない。今後、知識を増やしたり、洗練したり、また、局所探索をより充実させたりすることなどによって、多少とも棋力を向上させることは可能であろうが、一手読みでは飛躍的伸びは期待できないであろう。それでは、この全局的な先読みの実現を困難にしている最大の理由は何であろうか。それは、数手先の局面における形勢判断の処理時間である。碁において正確な形勢判断を行うには単に囲っている地の大きさだけでなく、個々の石の生死や強弱を正確に認識する必要があり、その為には今行っている1手読みのほとんどの処理が必要となる。即ち、全局的先読みを行おうとすれば、それによって展開される局面数倍に近い処理時間が必要となってくるのである。

並列化によって期待される効果は他にもある。現システムにおいてESPの記述能力は、囲碁知識及び知識間の協調を記述するのに大きく貢献した。と言うより、むしろそれがシステムの骨格を決定付けたと言ってもよい。このことは新たな並列環境が、より斬新なアイデアや新しい技術を生み出す可能性を示唆している。

このように、並列化による質的量的変化が碁世代をさらに一段と人間に近づけることが期待される。

7 謝辞

この囲碁システムが生まれる直接のきっかけを作られた横井俊夫氏（もとICOT第二研究室室長、現在、(株)電子化辞書研究所所長）、CGS-WGの主査の労をとられた鳥居宏治氏（大阪大学教授）、貴重なアドバイスを戴いたCGS-WGのメンバーの諸氏に感謝します。更に、CGS-TGのメンバーとしてプログラム作成の一部を担当して戴いた引地信弘氏（SRA）及び小島量氏（KKK）、協力戴いている三菱電機の関係者各位にも感謝しています。また、絶えずサポート戴いたETLの岡田義邦室長に感謝します。

参考文献

- [1] 実近. “知識指向型碁プログラムGO.1の戦略”、記号処理43-2、1987.11
- [2] N.Sanechika et al. “Notes on modelling and implementation of human player's decision process in the game of Go”, Bul.Electorec.Lab.Vol.45(1981)
- [3] 実近. “碁における石決定のプログラム化”、人工知能と対話技法研究会26-4、1982.6
- [4] Y.Mano. “An approach to Conquer Difficulties in Developing a Go Playing Program”, Journal of Information Processing, Vol.7, No2, July 1984
- [5] Reitman, W. & Wilcox B. “The structure and performance of the interim.2 Go Program.” Proceedings of the 6th international Joint Conference on artificial intelligence. Tokyo 1979.