

TR-719

囲碁システム「碁世代」の仕様

実近 憲昭（電総研）、清 慎一、
赤尾杉 隆、瀧 和男、吉川 貞行、
古岡 勉、村沢 好信、内田 俊一、
沖 廣明、大島 正行、小木曾 督、
水野 勇一、坂本 淳

December, 1991

© 1991, ICOT

ICOT

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

(03)3456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

囲碁システム「碁世代」の仕様

実近 憲昭*

工業技術院 電子技術総合研究所

清 慎一 赤尾杉 隆 瀧 和男 吉川 貞行[†] 吉岡 勉[†] 村沢 好信 内田 俊一
(財) 新世代コンピュータ技術開発機構

沖 廣明 大島 正行 小木曾 督 水野 勇一 坂本 淳
(株) 未来技術研究所

1991年11月9日

1 はじめに

本仕様書は、第五世代コンピュータプロジェクトにおける囲碁対局システム「碁世代」の仕様について記述したものである。

囲碁対局システム「碁世代」開発の目的は、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の基本的な問題に対する新しいアプローチを探るとともに、大規模知識処理システムの並列化方式の研究を行うことである。

本システムの開発は、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ともに、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として推進することとした。中期の研究開発では、いくつか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、改良を加え棋力の向上を試みた。

また、昭和63年度からは、システムの並列化に着手し、まず本システムの詰め碁サブシステムの並列化を行った。昭和64年度からは対局システムの並列化に着手し、逐次対局システムと並列対局システムを融合した実験システムを試作し並列化の方針を固めた。平成2年度から本格的な並列対局システムの開発に着手し現在に至っている。また、これらシステムの並列化の作業と並行して、引き続きシステムの見直し、改良を行い棋力向上に努めた。

*現 (株) AI 言語研究所

[†]現 NTT データ通信株式会社

2 囲碁のルール

囲碁のルールの厳密な定義は簡単でなく、また国によって異なるルールも存在するが、原則となるルールは以下のように簡明である。

1. 交互着手

2人のプレイヤーがそれぞれ黒石と白石を持ち、19×19の格子点の上に石を交互に置いていく。

ただし、パスすることによって、石を置かずに相手に着手を譲ることもできる。

2. 除去

縦横に隣り合う同色の石の集団の縦横全てが敵の石に取り囲まれた時、囲まれた石は盤上から除去される。

例えば図 2.1 で × の位置に黒石が置かれると、白石は除去され、それぞれ矢印の右側の配置になる。

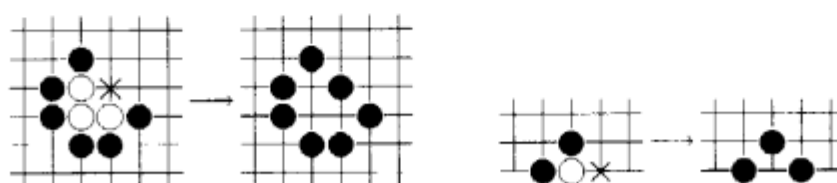


図 2.1 : 除去の例

3. 自殺着手禁止

着手した石が除去される状態になる場合は着手できない。ただし、着手によって他の石を除去できる場合は除く。

例えば、図 2.2 の左側の 2 例の × の点に黒石を置くことはできないが、右側の 2 例では白石が除去できる為 × の点に黒石を置くことができる。

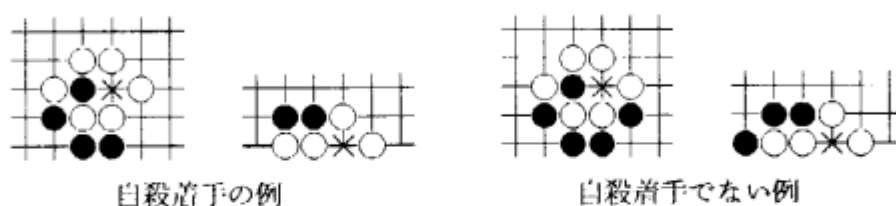


図 2.2 : 自殺着手禁止の例

4. 同形再現着手の禁止

着手によって石を除去する場合、石を除去した盤上の石の配置が 1 手前に現れた配置と同じになってはいけない。例えば、図 2.3 の左側の図の × の点に黒石を置いて、右側の図の配置になった直後は、右側の図の × の点に白石を置くことはできない。

このような形を劫(こう)と呼び、このルールによって永久にゲームが終わらない状態を回避している。

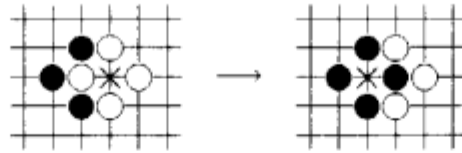


図 2.3：同形再現着手禁止の例

5. 勝敗

プレイヤーの両者が連続してパスをした時を終局とし、その時点で盤上に存在する石の数の多い方を勝ちとする。

3 プログラム構造

本システムは以下のようなプログラム構造となっている。

- 対局システム
 - 対局モード
 - * 対局設定
 - * 局面認識
 - データ構造更新
 - 局所探索
 - * 着手決定
 - 候補手列挙
 - 最終着手決定
 - 候補手フィルタ
 - 形勢判断
 - 実験解析モード
- 知識エディタ
- 評価用ツール
- 並列対局システム

3.1 対局システム

実際に対局を行う為のプログラムである。与えられた局面を認識し次の着手を決定する対局モードと、認識した結果を参照する為の実験解析モードとがある。

3.1.1 対局モード

1. 対局設定

対局を開始する前に以下のような対局方法を設定する。

- 対局者
先手後手それぞれの対局者として以下のいずれかを選択する。

- 人間
- コンピュータ
- RS232C
- PSI-NET

この設定によって、人間対コンピュータの対戦だけでなく、人間対人間、コンピュータ対コンピュータの対戦も可能である。また、RS232C や PSI-NET を選択することにより、それぞれの通信機能を備えた異なるコンピュータとの対戦も可能である。

- 手合い
互先、先、又は九子以下の置き碁による対戦のいずれかを選択する。互先を選択した場合のコミは五目半である。

- 打音有無
打音したことを示す音の有無を選択する。
- 形勢表示有無
コンピュータが認識している形勢を1手毎に表示するかどうかを選択する。

2. 局面認識

着手によって変化した局面の認識を行う。局面の認識は盤上の抽象対象に対して段階的に行われる。必要に応じて捕獲、連結といった局所探索が呼ばれる。

3. 着手決定

認識された局面に対して、いろいろな候補手知識が駆動され、候補手が生成される。各候補手にはいったん評価値が付けられるが、候補手フィルタによって再評価が行われる場合もある。また、形勢判断を行い、その結果によって着手が変更されることもあるが、基本的には最大の評価値合計を持つ点が着手として選ばれる。

4. インクリメンタル更新と全面更新

局面認識や着手決定の中の以下の処理は、最終着手によって引き起こされる変化部分のみ更新を行うインクリメンタル更新を行うことによって処理時間の節約をはかっている。インクリメンタルに更新できる処理は以下の処理以外にもあるが、それ程処理時間が多く掛からない処理やインクリメンタルな更新の記述が大変な処理については全面更新を行っている。また、アグイのあった時やマッタした場合は全ての処理が全面更新される。

- 点データ構造の更新
着手した点の各種属性と着手した点に隣接する点のダメ数などの属性を更新する。
- ポテンシャルの更新
着手した点に対して、ポテンシャルボタン知識(付録参照)によって与えられる影響範囲にある点のポテンシャル属性を更新する。
- 連データ構造の更新
着手した点によって新たにできる連の生成とその連に融合された古い連の削除を行う。また、着手した点に隣接する敵連のダメ数などの属性を更新する。
- 連の死活属性の更新
連の死活属性は捕獲探索によって求められる。連の死活属性は着手した点のケイマ範囲内にある連とその連に隣接する敵連について更新する。
- 結線のデータ構造の更新
着手した点によって新たにできる結線の生成を行う。
- 結線の連結度属性の更新
結線の連結度属性は連結探索と切断/連結ボタン知識(付録参照)によって求められる。結線の連結度属性は着手した点のケイマ範囲にある結線と死活属性が変化した連の点を端点とする結線について更新する。
- 定石候補手更新
定石候補手は着手のあった隅について、その隅が既に定石はずれの状態になっていない場合に更新する。

- ダメ点候補手更新

着手した点のケイマ範囲の連のダメ点について更新する。

- フトコロ拡大 / 縮小候補手更新

着手した点が4線以下の点の場合、着手した辺の着手した点を中心に2路以内の範囲についてそれまでマッチングしていた候補手を削除し、新たにボタンマッチング処理を行う。

- 模様接点候補手更新

着手によってポテンシャル値が変化した点及びその点に隣接する点について更新を行う。

3.1.2 実験解析モード

対局モードにおいて入力待ちの状態の時、及び局面編集ツールの任意の時点から実験解析モードへ移行できる。実験解析モードでは、局面認識によって更新された各種データ構造の内容や、着手決定によって求められた各種候補手や形勢が参照できる。また、各種局所探索を駆動し、その探索の過程や結果など見ることができる。

3.2 知識エディタ

本システムにおける知識の一部はデータベース化され、対局中に必要に応じて適宜参照される。知識エディタはこれらデータベース化された知識の人力及び管理を容易にする為のツール群である。知識エディタには現在、ポテンシャルエディタ、定石エディタ、手筋エディタが用意されている。

3.3 評価用ツール

囲碁対局システムのようなプログラムでは試行錯誤的な改良が余儀なくされている。評価用ツールは、改良した効果をできる限り正確に評価する為のツール群である。評価用ツールには自動テスト機能、棋譜棋力判定ツール、遠隔対局機能などがある。

自動テスト機能は、局面認識、局所探索、着手決定や形勢判断に関するいろいろな問題を予め作っておき、それらの問題に対する正答数によって改良の効果を評価する機能である。棋譜棋力判定ツールは、改良前と改良後のプログラムでどのような着手の差が表われたかを見ることができる。また、遠隔対局機能は、改良前のプログラムやパラメータを一部変更したプログラムとの対戦や、他の囲碁ソフトとの対戦を行うことによって総合的な棋力の評価を行う為の機能である。

3.4 並列対局システム

現在の対局システムは、順次マルチPSI上の並列対局システムへの移行を進めている。並列対局システムでは、現状の逐次対局システムと比べて、より多くの知識を処理することが可能となる。現時点では逐次対局システムと同程度の機能と一部並列対局システム固有の処理が実現されているのみであるが、今後新たな処理を随時導入していく予定である。

4 データ構造

盤上の石の配置から様々な意味のある形態の対象を抽出し、各対象の状態を認識していくことにより局面の認識を行う。これら意味のある形態の対象として、表 4.1 のような 5 種類の対象を考える。

表 4.1 : データ構造内の対象

対象	定義	意味	主な属性
点	19×19 の全ての格子点	石の配置	色, ポテンシャル値, 隣接点
連	隣接する同色の石の極大集合	捕獲の対象	石数, ダメ数, 種石 / 非種石, 死活
結線	同色の石, または石と盤端との間の仮想的な線	領域の境界 連結の対象	結線種別, 連結度
群	強連結した同色の石の極大集合	死活の対象	領域, 中地, 眼数, 包囲度, 強度, 重要度, 手数, 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	地の対象	領域, 辺長, 中地, 包囲度, 強度, 重要度

4.1 点データ構造

石の配置を認識する為に使われる。

盤上の全格子点に対応する点オブジェクトから構成されている。

4.1.1 点オブジェクトの属性

1. 色

点の色 (黒 / 白 / 空)。

2. 座標

点の X 座標と Y 座標で, 左上隅が (1,1) である。

3. 基準座標および基準座標変換コード

基準座標 (X,Y) と基準座標への変換方式のコード (後述詳細)。

4. 高さ

盤端からの距離 (盤端の時が 1)。

$$\text{高さ} = \min(\text{基準座標 } X, \text{基準座標 } Y)$$

5. 隣接点

その点に隣接する点の集合。

6. ダメ点

隣接点に属する点の内, 空の点の集合。

7. ダメ数

ダメ点に属する点の数.

8. ポテンシャル値

盤上の石 (黒 / 白の点) はその周囲の点に、距離に反比例するような勢力を及ぼすと考えられる.

ポテンシャル値とは、周囲の石からの勢力を合算したものである (最大 100) .

なお、単純に合算したものと、死群の石からの勢力を無視したものとの 2 種類のポテンシャル値がある (後述詳細) .

9. 連

連に属している点の場合は、その連.

10. 結線

結線の端点である場合は、その結線.

11. 遮断結線

結線の遮断点に属している場合は、その結線 (4.5.2 参照) .

黒結線 / 白結線のそれぞれ別々に保持している.

同色の 2 つ以上の結線の遮断点である時は、より強い結線を優先する.

12. 群

群の領域に属している点の場合は、その群.

13. 族

族の領域に属している点の場合は、その族.

4.1.2 基準座標, 基準座標変換コード

碁盤を図 4.4 のように 8 等分し、図の中の 0 の部分を基準部と呼ぶことにする.

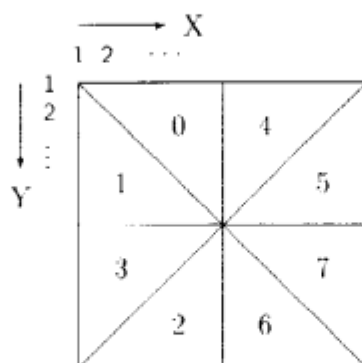


図 4.4 : 基準座標変換コード

基準座標とは、盤上の各点に対して、その点と対称の位置関係にある基準部内の点の座標のことである.

また、図 4.4 で 8 等分された部分の中の数字は各部分内の点における基準座標変換コードを示している。盤上の各点の座標を (X,Y) とする時、基準座標変換コードと基準座標との関係を表 4.2 に示す。

表 4.2：基準座標への変換方法

基準座標変換コード	基準座標 X	基準座標 Y
0	X	Y
1	Y	X
2	X	路数 + 1 - Y
3	路数 + 1 - Y	X
4	路数 + 1 - X	Y
5	Y	路数 + 1 - X
6	路数 + 1 - X	路数 + 1 - Y
7	路数 + 1 - Y	路数 + 1 - X

4.1.3 ポテンシャル値, ポテンシャル関数

ポテンシャル関数は、盤上の各点での黒 / 白の勢力の強さ (ポテンシャル値) を表現するものである。

盤上に置かれた石は、その周囲の点に距離に反比例するような影響を及ぼすと考える。ただし、隅や辺との位置関係が影響してくる為、人間の感覚に合うよう、個々に「ポテンシャルボタン知識」(付録参照) を用意した。ポテンシャルボタン知識は、ポテンシャルボタンエディタというツールによって容易に変更できる。

盤上に置かれた各石について、黒石であれば正の、白石であれば負のポテンシャル値をボタンに従い周囲の点に加算する。それらの合計値が各点のポテンシャル値となる。

4.2 連データ構造

石のアタリ、ヌキといった状態を認識する為に使われる。

互いに隣接する同色の石の集合である連オブジェクトから構成されている。

4.2.1 連オブジェクトの属性

1. 色
連の色 (黒 / 白)。
2. 点
連に属する点の集合。
3. 石数
連に属する点の数。
4. ダメ点
連に隣接する空点の集合。
連に属する点のダメ点の和集合と同じ。

5. 死活, 脱出点, 捕獲点

ダメ数3以下の連に対して黒番 / 白番で捕獲探索を駆動し, 表4.3 に従い死活を求める. ダメ数4以上の連の死活は生とする.

表 4.3 : 連の死活

連側手番での 捕獲探索結果	連の敵側手番での捕獲探索結果	
	成功	失敗または不明
成功 または 不明	中立	生
失敗	死	生

連側手番での捕獲探索結果が成功または不明の時の解の手を脱出点とし, 連の敵側手番での捕獲探索結果が成功の時の解の手を捕獲点とする.

6. 種石 / 非種石

図4.5 のようなキリチガイの状態があり, 4つの石の連の死活が全て生か中立の時, 各点の連を種石と呼ぶ.

種石以外の連を非種石と呼ぶ.

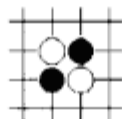


図 4.5 : キリチガイの状態

4.3 探索用点, 連データ構造

点と連のデータ構造を, 探索処理向けに改良したものが探索用点, 連データ構造である. 探索用点, 連データ構造によって点, 連オブジェクトの更新に要する処理時間及びメモリを大幅に削減できる.

通常の点, 連データ構造と探索用の点, 連データ構造を1つにまとめることは可能であり, 望ましいことであるが, 現状は別々に管理している. つまり, 対局における打碁時には両方のデータ構造を同じように更新し, 探索における仮想的な打碁時には探索用の点, 連データ構造のみ更新している.

4.3.1 探索用点, 連オブジェクトの属性

探索用点, 連オブジェクトでは, 点オブジェクトと連オブジェクトを別々に管理していない. 連の属性は, 連の石の中で最後に打碁された石に対応する点オブジェクトの中に保持されており, 最終打碁以外の点オブジェクトには, その点が連の最終打碁であった時の連の属性が保持されている. また, このような連の中の石の履歴は, 親点, 子点という属性によって管理されている.

1. 色, 座標, 標準座標, 標準座標変換コード, 高さ, 隣接点, ダメ点, ダメ数
点オブジェクトと同様.

2. 親点

親の点.

3. 子点

子の点の集合.

4. 子数

子点に属する点の数.

5. 連石数

連に属する石の数.

6. 連ダメ数

連に属する石に隣接する空点の数.

7. 連ダメ点

連ダメ数が3以下の時のみ, 連に属する点に隣接する空点の集合.

連のダメ点の管理をダメ数3以下に制限したのは以下の理由による.

- 可変長のデータの管理は処理が重く, また, メモリの無駄使いの原因となる.
- 現状の探索中に連のダメ点の位置が必要になる処理は, ダメ数が3以下の場合がほとんどである.

ただし, まれにダメ1以上の場合でもダメ点の位置が必要になる場合があるので, その場合は子点の属性から算出するメソッドが用意されている.

4.3.2 更新方法

探索用点, 連オブジェクトの更新方法を図 4.6 を例に説明する. 図 4.6 の左図は6個の白石から成る連を示している. 石の中の数字はその順序で打着されたことを示している.

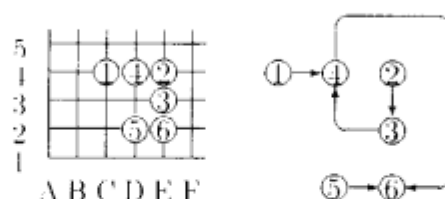


図 4.6: 探索用点, 連オブジェクトの更新方法

また, 図 4.6 の右図は, 各点の親(点)の様子を現わしている. 打着時にその点に隣接する味方石がある時, それぞれの味方石の連の中で最後に打着した石が子(点)になり, 打着した点がそれらの子の親となる. 例えば今6の石を打着したとすると, 1~4の石からなる連の中の最後に打着された石である4の石と5の石が6の石の子となり, 6の石は4と5の石の親となる.

打着した時の探索用点, 連オブジェクトの更新処理の内容を以下に示す.

- 打着点の色をセットする。
- 打着点の全ての隣接点のダメ数を1減らすとともに、ダメ点から打着点を除く。
- 打着点に隣接する敵石がある場合、その敵石を含む連の中で最後に打着された石の連ダメ数を1減らすとともに、連ダメ数が3以下の時は連ダメ点から打着点を除く(もしくは再算出する)。
- 打着点に隣接する味方石がある場合、その味方石を含む連の中で最後に打着された石の親点に打着点をセットするとともに、その石を打着点の子点に追加する。
- 打着点の子点の属性から、打着点の連石数、連ダメ数、連ダメ点の属性を算出しセットする。

また、探索のバックトラックの時やアゲ石の処理の場合に必要となる、打着した石をハガす時の更新処理の内容を以下に示す。

- 打着点の全ての隣接点のダメ数を1増やすとともに、ダメ点に打着点を追加する。
- 打着点に隣接する敵石がある場合、その敵石を含む連の中で最後に打着された石の連ダメ数を1増やすとともに、連ダメ数がまだ3以下の場合は、連ダメ点に打着点を追加する。
- 打着点に隣接する味方石がある場合、その味方石を含む連の中で最後に打着された石の親点をリセットするとともに、打着点の子点をリセットする。
- 打着点の色を空にする。

4.3.3 性能評価

通常の点と連のデータ構造を使った場合と探索用の点、連データ構造を使った場合の消費時間と消費メモリを比較した(表4.4)。

測定値は、コンピュータ同士19路盤の1回の対局で生じた捕獲探索における読みの中で生じた約5万手分の平均値である。

表4.4：通常の点と連データ構造と探索用点、連データ構造の性能比較

データ 構造	CPU消費時間 / 1手			消費ヒープメモリ / 1手		
	打着時	ハガシ時	合計	打着時	ハガシ時	合計
通常	3.38 ミリ秒	3.70 ミリ秒	7.08 ミリ秒	91 ワード	89 ワード	180 ワード
探索用	1.39 ミリ秒	0.56 ミリ秒	1.95 ミリ秒	0 ワード	0 ワード	0 ワード
改善率	2.4 倍	6.6 倍	3.6 倍	∞	∞	∞

4.4 結線データ構造

連結する可能性のある同色の石と石または石と盤端との関係を認識する為に使われる。

盤上に存在する図4.7の結線ボタンによって定義される結線オブジェクトから構成されている。ただし、図4.7の結線ボタンで×印のマークのついた位置は味方石でないことを条件とする。また、二線足、三線足、四線足の結線は、石の高さがそれぞれ2、3、1以上であることを条件とする。

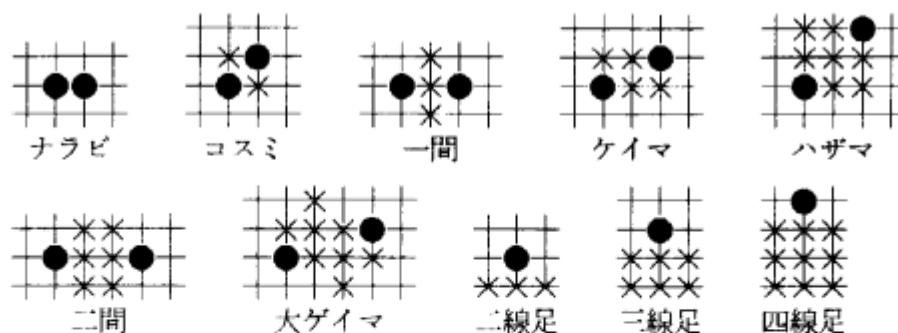


図 4.7：結線パターン

4.4.1 結線オブジェクトの属性

1. 色
結線の色 (黒 / 白)。
2. 両端点
結線を構成する 2 つの点。
3. 結線種別
結線の種類 (ナラビ / コスミ / 一間 / ケイマ / 二間 / 大ゲイマ / ハザマ / ……)
4. 連結度
結線の両端点が連結する可能性の度合 (強結線 / 弱結線 / 連結不能結線)。
局所探索の結果やボタン知識によって決定する (後述詳細)。
5. 群
群に属している結線の場合は、その群。

4.4.2 連結度

結線の連結度は以下のように決定する。

- ナラビの場合
常に強結線とする。
- コスミでキリチガイになっている時
キリチガっている 2 つの敵石の連の死活より表 4.5 に従い連結度を決定する。
- 上記以外のコスミ、一間、ケイマの場合
黒番 / 白番で連結探索を駆動し、探索の結果より表 4.6 に従い連結度を決定する。
ただし、結線側手番の探索における失敗とは探索の結果が“ツキスケ”の時をいい、以外は成功とする。また、結線の敵側手番の場合は、探索の結果が“連結”の時を失敗といい、以外は成功とする。

表 4.5：キリチガイのコスミ結線の連結度

敵石の連 の死活	もう一方の敵石の連の死活		
	死	中立	生
死	強結線	強結線	強結線
中立	強結線	強結線	弱結線
生	強結線	弱結線	連結不能結線

表 4.6：コスミ、一間、ケイマ結線の連結度

結線側 手番	結線の敵側手番	
	成功	失敗
成功	弱結線	強結線
失敗	連結不能結線	連結不能結線

- 二間、大ゲイマ、ハザマ、二線足、三線足、四線足の場合

切断 / 連結候補手ボタンで、かつ切断 / 連結候補手に空点がある時 (付録「切断 / 連結候補手ボタン知識」参照) 弱結線とし、切断 / 連結候補手が全て空点でない時、連結不能結線とする。

また、切断 / 連結候補手ボタンでない時は強結線とする。

4.5 群データ構造

連結しそうな石の集団の状態を認識する為に使われる。

強連結結線でつながる同色の石の集合で定義される群オブジェクトから構成されている。

ただし、強連結結線とは以下の結線をいう。

- ナラビ、コスミ、一間、ケイマ、二線足、三線足、四線足の強結線
- 両端点の高さが4線以下の二間、大ゲイマの強結線
- 切断 / 連結候補手ボタン [二線足 (')] (付録「切断 / 連結候補手ボタン知識」参照) の二線足の弱結線

群データ構造の更新は2つのフェーズで行われる。最初のフェーズで死となった群の隣接敵群を融合して新しい1つの群を生成する処理が次のフェーズで行われる。同時に死群のポテンシャル値をリセットする処理が行われる。

4.5.1 群オブジェクトの属性

群オブジェクトは以下のような属性を保持している。

1. 色

群の色 (黒 / 白)。

2. 領域
群の強連結結線で囲まれた閉領域内の点の集合 (後述詳細) .
3. 領域サイズ
領域に属する点の数.
4. 自己石
領域に属する自己側の石の集合.
5. 石数
自己石の数.
6. サイズ
領域の点と領域の点の外側2路範囲内の点の数の総和. 群の死によって得られる相手側の利得 (自己側の損失) の指標となる.
7. 中立領域
領域に隣接し後1手で群の領域となるような領域の集合 (後述詳細) .
8. 中地
領域内の地の数 (後述詳細) .
9. 眼数
領域内の眼の数 (後述詳細) .
10. 中手
眼数が自己手番では2眼で相手手番では1眼になる時 (1.5眼と呼ぶ), 自己手番で2眼にする / 相手手番で1眼にする手 (後述詳細) .
11. 包囲度
相手側の石による包囲の度合い (後述詳細) .
12. 群ダメ点
群の包囲度を計算する為の群のダメ点の集合 (後述詳細) .
13. 包囲結線
包囲している相手側の結線の集合 (後述詳細) .
14. 包囲点
包囲している点の集合 (後述詳細) .
15. 包囲群
包囲している相手側の群の集合 (後述詳細) .
16. 強度
群の強さ (後述詳細) .

17. 重要度

1 手手をいれる価値や、群の死活が周囲に与える波及効果 (後述詳細)。

18. 手数

自己石のダメ数の総和、攻め合いの有無のチェックに使われる。

19. 分類

死活 / 攻め合いの問題を検出する為の群の分類 (後述詳細)。

4.5.2 領域, 中立領域, 中地

領域とは群の石と強連結結線の遮断点によって囲まれた (群の石や強連結結線の遮断点も含む) 点の集合のことである。ここで遮断点とは結線の両端点を結ぶ仮想的な線を表現するもので、図 4.8 のように定義する。ここで大きな黒丸が結線の端点を表わし、小さな黒丸が遮断点を表わしている。

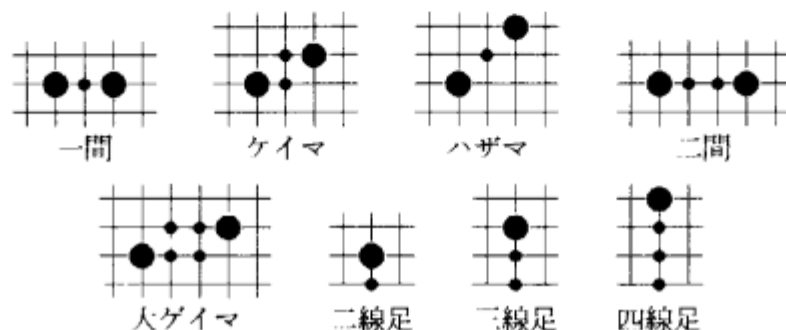


図 4.8 : 結線の遮断点

中立領域とは、群の領域内の点と 1 つの強連結結線でない結線の遮断点によって囲まれた (群の領域内の点は含まず、結線の遮断点は含む) 点の集合のこと、後 1 手掛けると領域になるような閉領域のことである。図 4.9 が閉領域の例である。大きな黒丸が群の石を表わし、大きな黒丸と小さな黒丸を合わせた点の集合が群の領域を表わしている。×印のマークが群の中立領域を表わしている。

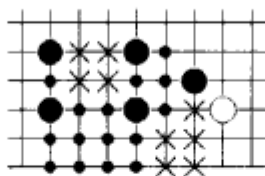


図 4.9 : 群の領域と中立領域

また、群の中立領域をその領域内の点の数の大きい順に並べ、1 番目、3 番目、5 番目、... に大きい中立領域を先手中立領域と呼び、2 番目、4 番目、6 番目、... に大きい中立領域を後手中立領域と呼ぶ。

中地とは、領域 (もしくは中立領域) 内の内点の数のことであり、先手中地と後手中地の 2 種類がある。ここで、内点とはある閉領域内の点の内、その閉領域以外の点には隣接していない点のことである。先手中地と後手中地は以下のように定義する。

先手中地 = 領域と先手中立領域の和集合からなる領域の内点の数

後手中地 = 領域と後手中立領域の和集合からなる領域の内点の数

先手中地とは群側手番で想定される地の大きさを表わし、後手中地とは群の敵側手番で想定される地の大きさを表わしていると考えられる。また、単に中地と呼ぶ時は、後手中地のことである。

4.5.3 眼数, 中手

眼数は群内の眼の数のことであり、中手は眼数を効果的に増減させる手のことである。この処理は重い為、現在包囲度3以下で中地7以下の群についてのみ算出している。眼数および中手は以下のよう算出する。

1. 眼形を求める。

群の領域内で全て群の石によって囲まれた閉領域を眼形とする。眼形が2つ以上あれば眼数を2とし、眼形が1つもなければ眼数0とする。どちらの場合も中手はなしである。眼形が1つであれば以下の処理を行う。

2. 眼形の次数コードを求める。

眼形の次数コードとは眼形内の各点の次数の組み合わせのことである。次数とは、その点の隣接点の内同じ眼形内にある点の数のことである。

3. 急所数を求める。

眼形の次数コードより表 4.7 に従い急所数を求める。

表 4.7 : 眼形の急所数

眼形サイズ	次数コード	急所数	眼形サイズ	次数コード	急所数
1	(0)	0	5	(4, 1, 1, 1, 1)	1
2	(1, 1)	0	6	(2, 2, 2, 2, 1, 1)	4
3	(2, 1, 1)	1		(3, 2, 2, 1, 1, 1)	3
4	(2, 2, 1, 1)	2		(3, 3, 1, 1, 1, 1)	2
	(3, 1, 1, 1)	1		(3, 3, 2, 2, 1, 1)	2
	(2, 2, 2, 2)	0		(3, 2, 2, 2, 2, 1)	2
5	(2, 2, 2, 1, 1)	3		(3, 3, 2, 2, 2, 2)	2
	(3, 2, 1, 1, 1)	2		(4, 2, 1, 1, 1, 1)	2
	(3, 2, 2, 2, 1)	1		(4, 2, 2, 2, 1, 1)	1

4. 急所を求める。

眼形内の点の内次数の多い点から順に急所数分の点を急所の点とする。

5. 眼数を求める。

急所の点の内既に敵石が置かれている点の数を求め、表 4.8 に従い眼数を求める。

6. 中手を求める。

眼数が2眼または1眼の時は中手はなしとし、眼数1.5眼の時、急所内の空点を中手とする。

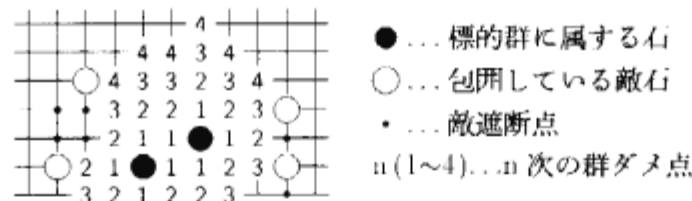
表 4.8 : 眼形の眼数

急所数	急所内の敵石数	眼数
3 以上	-	2
2	2	1
	1	1.5
	0	2
1	1	1
	0	1.5
0	-	1

4.5.4 包囲度, 群ダメ点, 包囲結線, 包囲点, 包囲群

包囲度とは, ある群 (または族) が敵石によってどのくらい包囲されているかの度合を示すもので, 数値の小さいほど強く包囲されていることを示す. 包囲度が 0 の時を特に完全包囲と呼ぶことがある. 包囲度は以下のように定義する.

標的の群に属する各石のダメ点を 1 次の群ダメ点と呼び, n ($1 \sim 3$) 次の群ダメ点の 1 路外側のダメ点を $n+1$ 次の群ダメ点と呼ぶ. ただし, その点が敵遮断点 (4.5.2 参照) である場合と, 4 次の群ダメ点で高さが 1 の場合は除く. 例として図 4.10 のような石の配置における 4 次以下の群ダメ点を示す.

図 4.10 : n 次の群ダメ点

4 次の群ダメ点を単に群ダメ点と呼び, 群ダメ点から包囲度を決定する. 群ダメ点をナラビかコスミで互いにつながっている極大集合のグループに分ける. このグループを群ダメ点グループと呼ぶ. 3 個以上の要素からなる群ダメ点グループについて, その群ダメ点の数の総和を群の包囲度とする. 図 4.10 の例では全ての群ダメ点は 1 つの群ダメ点グループに属し, 包囲度は 6 となる.

また, 包囲結線とは, 標的群を包囲している敵結線のことで, 3 次以下の群ダメ点に隣接する敵遮断点の属する結線とする. 包囲点とは, 標的群の近傍の敵石もしくは味方石のことで, 3 次以下の群ダメ点に隣接する標的群以外の群に属する石と包囲結線の端点の和集合とする. 包囲群は, 標的群の近傍の敵群のことで, 敵包囲点が属する群のこととする.

4.5.5 強度

群の強度とは, 群に所属する石全体の強さを表わすもので, 以下のように定義する.

1. 基本式

基本となる式は以下のように簡単である。

$$\text{強度} = \frac{\text{中地} \times 5}{(\text{中地強度})} + \frac{\text{包囲度} \times 2.5}{(\text{脱出強度})}$$

ここで5と2.5の定数は以下のようにして決めた。様々の棋譜に表われたいろいろな局面について、その局面の群の強度を人間の感覚で評価した。人間の石の強さに対する感覚を数値で表現する為に、表4.9を目安とした。

表 4.9：石の強さに対する感覚と強度との関係

石の強さに対する感覚	強度
生	40 ~ 50
ほぼ生	30 ~ 40
生きているがもう1手掛けるのも有力	20 ~ 30
中立(敵手番なら死、味方手番なら生)	10 ~ 20
ほぼ死	1 ~ 10
死	0

人間の感覚から求めた強度と前述の式による強度の差が最小になる定数を求めることによって、前述の5と2.5の定数が決定された。

2. 中地強度の補正

中地強度をより精密にする為、以下の補正を行う。

(a) 中地の補正

中地とは群内の地の大きさを示すもので、群の領域内の点の内境界の内側の点の数によって決めている(4.5.2 参照)。ただし、境界が全て石である場合は1目の地としての価値は充分あるが、境界がまだ空点である場合は多少目減りさせた方が実情に合っている。この為、中地としてカウントされる点(内点)が空点の境界と接している場合は、1目ではなく0.5目として中地にカウントする。また、2つの空点の境界と接している場合はさらにその半分の価値とする(以下同じ)。

(b) 眼数による補正

同じ中地であっても、後1手で2眼になるような状態(4.5.3 参照)である場合は、群の強度に与える影響は大きいと考えられる。従って、眼数が1.5眼の場合は中地強度として10を加えることにする。

(c) フトコロ手による補正

中地はほとんど確定した地の大きさを表わすが、将来中地が増える可能性の有無も考慮した方が群の強度はより正確になる。

フトコロ拡大候補手(6.1.8 参照)は群の中地を増やす手のことで、候補手評価値はその手によって増える中地の大きさによって決定されている。群にフトコロ拡大候補手がある場合、以下のフトコロ強度を中地強度に加算する。

$$\text{フトコロ強度} = \min(\text{フトコロ拡大候補手評価値}, 10)$$

3. 脱出強度の補正

同じ包囲度であっても、脱出路が1つの場合と複数の場合では、脱出できる可能性に差が出てくる。脱出路が1つであれば敵からの1手で完全に封鎖される可能性があるが、脱出路が複数あれば1手で封鎖することは難しくなる。

3個以上の要素からなる群ダメ点グループ(4.5.4 参照)が複数ある場合、脱出強度は以下の式によって決める。

群ダメ点グループが複数(N 個)ある時の

$$\text{脱出強度} = N \times 10$$

4. 分類による補正

群の各種属性によって、戦略的見地から群の分類が行われる(4.5.7 参照)。群を分類する際隣接する群の強度が問題となるが、その強度は以上の処理によって求められた強度が使われる。

群の分類処理後、その結果によって以下のように群の強度が補正される。

- 完全死群の場合
群の強度は0とする。
- 眼数2の安全群の場合
群の強度は50とする。
- 上記以外の場合
群の強度は最大45とする。
- 死群、仮死群の場合
群の強度は最低1とする。
- 脱出可能群、攻め合い注意群の場合
群の強度は最低10とする。

4.5.6 重要度

群の重要度として以下の2種類がある。

1. 手入れ重要度

もう既に十分強く、死ぬ可能性のないような石を攻める(/守る)手の価値は低い。また、既にほとんど死んでいる石に手を入れることも同様に価値は低い。石の強さがその中間の状態の時に、その石を攻める(/守る)手の価値が高くなる。このような石を攻める(/守る)手の価値を手入れ重要度と呼び、表4.10のような関数によってその値を決定する。この手入れ重要度は様々な群の攻防候補手の評価値を決定する為に使われる。

2. 種石重要度

種石とは、その石の死活がその石の周囲の相手側の石の死活に強く影響しているような状態の石のことを言う。即ち、種石が死ぬことによってその周囲の弱かった相手側の石が全て安全になるので、種石の死活に関する候補手の評価値は高くする必要がある。種石の死活に関する価値を種石重要度と呼び、以下のような関数によってその値を決定した。ここで、 T_i 及び S_i はそれぞれ対象としている群の包囲群の強度とサイズである。

$$\text{種石重要度} = \frac{1}{15} \sum_i (30 - T_i) \cdot S_i$$

標的群の強度	0	1	2	3	4	5	6	7	8	9	10
手入れ重要度	0.0	7.1	10.0	12.0	13.6	14.9	16.0	16.9	17.7	18.3	18.9
	11	12	13	14	15	16	17	18	19	20	
	19.3	19.6	19.8	19.9	20.0	19.9	19.8	19.6	19.3	18.9	
	21	22	23	24	25	26	27	28	29	30	
	18.3	17.7	16.9	16.0	14.9	13.6	12.0	10.9	9.8	8.7	
	31	32	33	34	35	36	37	38	39	40	
	7.7	7.0	5.9	5.2	4.6	3.8	3.3	2.8	2.3	1.9	
	41	42	43	44	45	46	47	48	49	50	
	1.6	1.2	1.0	0.8	0.6	0.4	0.2	0.1	0.0	0.0	

表 4.10：手入れ重要度関数

4.5.7 分類

群の各種属性より、戦略的見地から以下の順に群の分類を行う。

- 群の石が全て死連の時

完全死群とする。

- 眼数 2 以上または包囲度 4 以上または中地 11 以上の時

安全群とする。

- 援軍ありまたは脱出可能の時

脱出可能群とする。

ただし、援軍ありとは、群の包囲点の中に強度 10 以上の同色群に属する点がある時をいう。
また、脱出可能とは、包囲結線の中に以下の結線がある時をいう。

- 二間、大ゲイマ、四線足の強結線
- 二線足以外の弱結線

- フトコロ手ありまたは眼数 1 以上または中地 8 以上の時

仮死群とする。

ただし、フトコロ手ありとは、群の石を基点とするフトコロ拡大候補手がある時をいう。

- 中地が 4 以上の時

死群とする。

- 攻め合い状態の群である時

攻め合い注意群とする。

ただし、攻め合い状態とは、包囲群の中に以下の条件を全て満たす群が存在する時をいう。

- 安全群, 脱出可能群, 完全死群以外の群であること
- 自群, 包囲群ともに1つの連の群である時は以下の条件を満たすこと

$$\text{自群の実質手数} \geq \text{包囲群の実質手数}$$

自群, 包囲群どちらかが1つの連の群でない時は以下の条件を満たすこと

$$\text{自群の実質手数} \geq \text{包囲群の実質手数} + 3$$

- 上記いずれでもない時

完全死群とする。

なお, 完全死群があった場合は, その包囲群を融合した新しい群の生成処理が行われるとともに, 完全死群に属する各石のポテンシャルがリセットされる。

4.6 族データ構造

模様や厚みといった概念を認識する為に使われる。

ポテンシャル値7以上(白の場合はポテンシャル値-7以下)で互いに隣接する点の極大集合で定義される族オブジェクトから構成されている。

4.6.1 族オブジェクトの属性

- 色
族の色(黒/白)。
- 領域
族を構成する点の集合。
- サイズ
領域に属する点の数。
- 自己石
領域に属する自己側の石の集合。
- 石数
自己石の数。
- 辺長
領域に属する高さが一線の点の数。
- 中地
領域に属する点の内, その点のコスミ範囲にある点のポテンシャル値が全て7以上(白の場合は-7以下)の点の数。
- 包囲度, 族ダメ, 包囲結線, 包囲点
群オブジェクトと同様。

- 強度

$$\text{強度} = \min(1.6 \times \text{包囲度} + 0.8 \times \text{中地} + \text{確定地強度}, 50)$$

$$\text{確定地強度} = \begin{cases} 10 \text{ (中地} \leq 5 \text{ の時)} \\ 10 \times \max(5 - \frac{\text{サイズ}}{\text{石数}}, 0) \text{ (以外)} \end{cases}$$

- 手入れ重要度

群オブジェクトと同様

5 局所探索

局面認識において各種対象の状態を認識する方法として、パタンの知識を利用することと探索によって求めることが考えられる。一般に前者はコストは小さいが正確さに欠け、反対に後者は正確であるがコストが大きい。従って、通常はパタンの知識を使って処理し、一定時間内処理できる程問題が煮詰まってくれば探索によって求めている。現在本システムでは、表 5.11 のような 6 種類の局所探索が用意されている。

表 5.11 : 局所探索の種類

種類	標的	問題	結果
シチョウ	ダメ数 2 以下の連	標的連がシチョウか	成功 / 失敗
捕獲	ダメ数 3 以下の連	標的連が捕獲可能か	成功 / 失敗
連結	線足以外の結線	標的結線が連結可能か	連結 / キリチガイ / ツキヌケ
詰め碁	完全包囲の群	標的群は生か	生 / 死 / コウ
眼数	サイズ 4 以下の地	標的地に 1 眼できるか	成功 / 失敗
多重標的捕獲	ダメ数 2 以下の連 を含む複数の連	いずれかの連が捕獲可能か	成功 / 失敗

5.1 探索アルゴリズム

本システムで用意されている全ての局所探索は、完全に評価できる局面まで探索を行う完全探索(読み切り探索)であり、探索のアルゴリズムはアルファベータ法を使っている。

また、手順前後が頻繁に生じる探索では、一度探索した局面と結果を記憶しておき、同じ局面が発生した時にその結果を使うことで無駄な探索をしないようにしている。

5.1.1 探索における局面記憶の効果

探索における局面記憶は、探索の途中で生じた局面と評価値を記憶しておき、同じ局面が再び現れた時に記憶しておいた評価値を使うことによって、無駄な探索を減らす為に行われる。

碁世代のいろいろな局所探索に局面記憶を導入してみたところ、詰め碁や眼数などの探索では、探索の手数を大幅に削減できた。また、シチョウ、捕獲、連結といった探索では性能に大きな変化が見られなかった為、現在局面記憶は行っていない。以下では、詰め碁探索を例にして、探索における局面記憶の方法とその効果について述べる。

5.1.1.1 局面記憶の方式

詰め碁探索における局面記憶の方式について述べる。

- ハッシュテーブル

局面の石の配置をキーとし、その局面の繰り上がり評価値をデータとするハッシュテーブル上に局面を記憶する。

手番が攻め番 / 守り番のそれぞれの場合について、別のテーブル(ハッシュインデクスは、現在各 5,000 エントリー用意している)に記憶する。

- キー

盤面の石の配置 (盤上の格子点の黒 / 白 / 空の状態を該当する 2bit 上に表現) とコウの情報からなるストリングデータをキーとする。

- ハッシュ関数

標的とする群の近傍にある数点の石の配置を数値化した値を、ハッシュインデクスエントリー数で割った剰余をとる。

また、記憶する局面として、評価値が繰り上がってきた時だけでなく、子ノードに進む時の局面も記憶しておくことによって以下のような効果が得られる。

碁では、劫以外に、何手も掛けて元の局面に戻ること (長生や循環劫など) が稀に起こる。局面記憶を採用する以前の探索では、このような同形再現を認識することは難しかった。この為、一定の深さを越えても終端局面に達しない場合に、そのような同形再現が生じたと認識していた。子ノードへ進む時の局面も記憶することによって、そのような同形再現を即座に認識できるようになる。探索の深さによる間接的な認識と比較して、より正確でかつ無駄な探索を減らす効果が得られる。

5.1.1.2 局面記憶による探索性能の効果

図 5.11 の 4 つの詰め碁問題 (全て黒先) に対して、局面記憶をしない場合とする場合の、探索に要する手数と時間の測定結果を表 5.12 である。

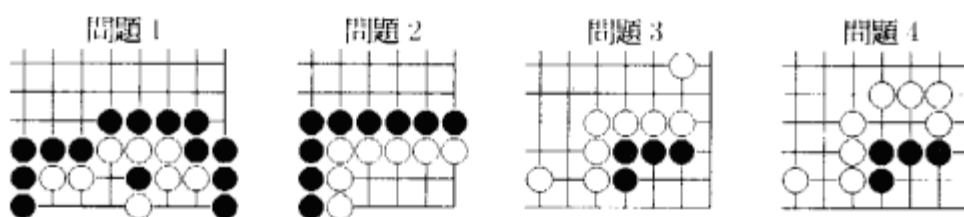


図 5.11：局面記憶の性能測定に使った詰め碁問題

表 5.12：局面記憶をしない場合とする場合の探索性能の測定結果

問題 番号	局面記憶なし (A)		局面記憶あり (B)		性能向上率 (A)/(B)	
	探索手数	探索時間	探索手数	探索時間	探索手数	探索時間
問題1	652 手	1.8 秒	319 手	1.2 秒	2.0 倍	1.5 倍
問題2	2,645 手	10 秒	877 手	3.7 秒	3.0 倍	2.7 倍
問題3	5,221 手	11 秒	1,105 手	3.9 秒	4.7 倍	3.6 倍
問題4	80,910 手	237 秒	5,382 手	21 秒	15.0 倍	11.3 倍

表 5.12 から分かるように、局面記憶の導入によって数倍程度の探索性能が得られた。また、探索手数を多く必要とする問題ほど、性能向上率が上がることが分かった。

5.1.2 キラーヒューリスティック

キラーヒューリスティックとは「現局面に似た局面で以前調べたことがある場合、その局面における最良手(キラー手と呼ぶ)は、現局面において最良手になりやすい」という考え方であり、既にチェスなどのプログラムで、その効果が認められている。

碁世代では、詰め碁探索を例にキラーヒューリスティックの効果を測定したが、特に効果が認められなかったため、現在キラーヒューリスティックは使っていない。以下に、詰め碁を例にしたキラーヒューリスティックの実験結果について述べる。

1. 実験したキラーヒューリスティック

チェスプログラムでは、様々なキラー手が提案されているが、碁世代で実験したキラー手は以下の2通りである。

- 直前に探索された兄弟ノードにおける最良手。
- 直前に探索された子ノードにおける最良手。

前者は、チェスプログラムなどでも採用されている代表的なキラー手である。チェスを例に説明してみよう。相手側がある手を打った局面において、例えば相手側のビショップをただ取りにする手を発見し、その手によって相手側の手を反駁した場合、相手側が元の局面で他の手を打った局面においても、なおビショップをただ取りできる手が残っているなら、まずその手を探索するという考え方である。

この考え方が囲碁においても有効であるかどうかを予測するのは難しい。「2の一」のように位置だけに依存するような急所が残されているような局面についてはある程度有効かもしれないが、多くの場合、碁の急所は相手の打筋によって都度変化すると考えられるからである。

後者のキラー手は、チェスプログラムなどの例にはない。多分初めての試みだと思われる。この手は、ある手を打ってみてダメだったら、その手を反駁した相手側の手の位置に打ってみるというもので、人間が通常使っているヒューリスティックである。この考え方は「敵の急所は我が急所」という囲碁の格言にも通じるものがある。

2. キラーヒューリスティックの効果

図5.11(5.1.1参照)の4つの詰め碁問題に対して、キラー手なしと2通りのキラー手を導入した場合の、探索に要した手数と時間の測定結果が表5.13である。

表5.13から分かるように、測定した詰め碁の問題に関する限り、キラー手の導入による探索性能の向上は見られなかった。

5.2 シチョウ探索

シチョウ探索は、アタリの連続によって、標的とする連が捕獲可能かどうかを完全探索するもので、以下の用途に使われる。

- 捕獲などの探索における終端局面判定や着手の生成の為に
- 各種パタン知識の条件として

表 5.13：キラーステューティックの導入による探索性能の測定結果

問題 番号	キラース なし (A)	キラース 1(B)	キラース 2(C)	性能向上率	
				(A)/(B)	(A)/(C)
問題 1	319 手	324 手	392 手	0.98	0.81
	1.2 秒	1.3 秒	1.7 秒	0.92	0.71
問題 2	877 手	895 手	870 手	0.98	1.01
	3.7 秒	3.9 秒	4.0 秒	0.95	0.93
問題 3	1,105 手	1,401 手	1,090 手	0.79	1.01
	3.9 秒	5.0 秒	4.1 秒	0.78	0.98
問題 4	5,382 手	6,489 手	6,213 手	0.83	0.87
	21 秒	25 秒	26 秒	0.81	0.81
平均				0.90	0.93
				0.87	0.86

5.2.1 標的対象

以下の条件を満たす連を標的とする。

- 攻め番でダメ数 2 以下の連
- 守り番でダメ数 1 以下の連

5.2.2 終端局面評価関数

局面が以下のいずれかの条件を満たす時、終端局面と判定し、その終端局面の評価値を返す。手数の制限は三劫などの循環劫対策である。

- 守り番で標的連のダメ数が 2 以上の時

評価値 = 10 (成功)

- 攻め番で標的連のダメ数が 1 の時

評価値 = 10 (成功)

- 攻め番で標的連のダメ数が 3 以上の時

評価値 = -10 (失敗)

- 探索の深さが 100 手を越えた時

評価値 = 0 (不明)

- 探索の総手数が 300 手を越えた時

評価値 = 0 (不明)

5.2.3 強制手

以下の手がある局面では、その1手のみ探索する。

- 攻め番の時、守り側の打着によって標的連のダメ数が1以上になる手がある時、その手。

5.2.4 着手生成

強制手がない時、以下の着手について着手評価値の大きい順に探索する。ただし、複数の着手が同じ点になる時は1つとし、着手評価値は各着手評価値の内最大の評価値とする。また、同じ評価値が複数ある時は、盤上の位置がより高い方の手を優先し、高さも同じ場合はより中央の手を優先する。

- 守り番の時

- 標的連の包囲連をヌク手がある時その全ての手

着手評価値 = 4

標的連のダメ点

着手評価値 = 3

- 攻め番の時

- 標的連のダメ点

着手評価値 = その点へ守り側が打着した場合の標的連のダメ数

5.2.5 性能評価

図5.12の4つのシチョウ問題についてシチョウ探索の性能を測定した。数字のついた白石が標的となっている石を表わし、その数字が問題の番号を表わしている。全て黒番で標的の石が捕獲できるかどうかの問題であり、答えは全て捕獲可能である。

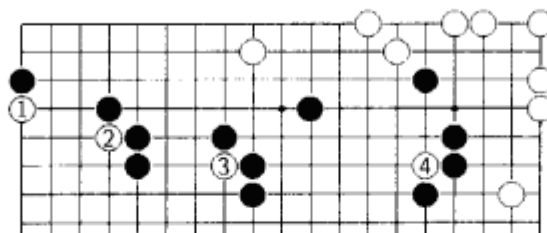


図 5.12：シチョウの問題

問題1は辺に沿って一直線に読みが進む問題であり、問題2と3は左下方に斜めに読みが進むシチョウの典型的な問題である。問題4は試行錯誤を繰り返しながら、いろいろな方向に読みが進む問題で、手数も長く、人間の上級者でも最後まで読み切るには骨が折れる問題である。4つのシチョウ問題について、シチョウ探索に要した手数と時間の測定結果が表5.14である。

測定結果の表から、1手当たりの探索時間は探索手数の多い問題ほど増えていることが分かる。問題2や3のような斜めに読みが進むシチョウ問題について、探索手数と1手当たりの探索時間の関係を調べると正比例していることが分かる。

表 5.14：シチョウ探索性能の測定結果

問題	探索手数 (N)	探索時間 (T)	1 手当たりの 探索時間 (T/N)
問題 1	30 手	109msec	3.63msec
問題 2	12 手	34msec	2.83msec
問題 3	28 手	94msec	3.36msec
問題 4	66 手	267msec	4.05msec

探索手数 (N) と 1 手当たりの探索時間 (T/N) の関係を, 15 種類のいろいろな手数のシチョウ問題の測定結果に対する直線補完によって求めると, 以下の式が得られる.

$$T/N \text{ (msec)} \simeq 0.028N + 2.535$$

探索手数が増えることによって 1 手当たりの探索時間が増える原因としては, 守り側の着手生成時の「標的連の包囲連をヌク手」を求める処理が考えられる. この処理は標的連の各石の隣接点を調べる為, 標的連の石数が増えるに従って処理時間が増える. 1 直線に進むような探索では, 探索中の標的連の平均の石数は探索の手数に比例する為, 1 手当たりの探索時間が探索の手数に比例して増えることになる.

5.3 捕獲探索

捕獲探索は, 標的とする連が捕獲可能かどうかを完全探索するもので, 以下の用途に使われる.

- 連の死活判定の為
- 捕獲探索候補手の生成の為
- 連結などの探索における終端局面判定や着手の生成の為
- 各種ボタン知識の条件として
- 最終着手決定時のフィルタとして

5.3.1 標的対象

- ダメ数 3 以下の連を標的とする

5.3.2 終端局面評価関数

局面が以下のいずれかの条件を満たす時, 終端局面と判定し, その終端局面の評価値を返す.

- 守り番で標的連のダメ数が 4 以上の時

$$\text{評価値} = 10 \quad (\text{成功})$$

- 攻め番で標的連のダメ数が 1 の時

$$\text{評価値} = 10 \quad (\text{成功})$$

- 攻め番で標的連のダメ数が2で、シチョウ探索によって標的連が捕獲可能の時

評価値 = 10 (成功)

- 攻め番で標的連のダメ数が2または3で $N1 \times 2 + N2$ が4以上の時

評価値 = -10 (失敗)

ただし、標的連の包囲連の内、ダメ数が1の連の個数を $N1$ 、ダメ数が2の連の個数を $N2$ とする。

- 攻め番で標的のダメ数が4以上の時

評価値 = -10 (失敗)

- 探索の深さが20手を越えた時

評価値 = 0 (不明)

- 探索の総手数が200手を越えた時

評価値 = 0 (不明)

5.3.3 強制手

以下のいずれかの手がある局面ではその1手のみ探索する。

- 守り番の時

- 標的連のダメ数が1の時、そのダメ点もしくはアタリの包囲連をスク手。
- 守り側の打着によって標的連のダメ数が4以上になる手がある時、その手。
- サイズ2以上でアタリの包囲連がある時その包囲連をスク手。
- 2つ以上の包囲連に対して両アタリする手がある時、その手。
- サイズ2以上でダメ数2の包囲連をシチョウ探索で捕獲可能の時、その捕獲手。

- 攻め番の時

守り側の打着によって標的連のダメ数が5以上になる手がある時、その手。

5.3.4 着手生成

強制手がない時、以下の着手について着手評価値の大きい順に探索する。ただし、複数の着手が同じ点になる時は1つとし、着手評価値は各着手評価値の内最大の評価値とする。また、同じ評価値が複数ある時は、盤上の位置がより高い方の手を優先し、高さも同じ場合はより中央の手を優先する。

- 守り番の時

- ダメ点

標的連のダメ点(ただし、その点への守り側打着によって標的連のダメ数が1の時は除く)。

$$\text{着手評価値} = \text{ダメ評価点} + \text{ダメ点ボーナス点}$$

ただし、

$$\text{ダメ評価点} = \text{ダメ基礎点} \times \text{ダメ増分度}$$

$$\text{ダメ基礎点} = \frac{10}{\text{標的連のダメ数}}$$

ここで、ダメ増分度とはその点への守り側の打着によって増える / 減る標的連のダメ数のことで、最大を2とする。

また、ダメ点ボーナス点とは、そのダメ点が包囲連との共有ダメ(「6.1.5 ダメ点候補手」参照)の時、共有ダメボーナス点となり、ナラビダメの時はナラビダメボーナス点となり、以外は0点となる。

共有ダメボーナス点は以下の式で与えられる。

$$\text{共有ダメボーナス点} = \text{共有ダメ基礎点} + \text{包囲連ダメ増分度} \times 0.1$$

ここで、共有ダメ基礎点とは表5.15の関数によって求められる。また、包囲連ダメ増分度とはそのダメ点への攻め側の打着によって増える / 減る包囲連のダメ数のことである。

表 5.15：共有ダメ基礎点関数

包囲連のダメ数	1	2	3	4以上
共有ダメ基礎点	3.0	2.5	2.0	1.5

ナラビダメボーナス点は以下の式で与えられる。

$$\text{ナラビダメボーナス点} = \text{ナラビダメ基礎点}$$

ここで、ナラビダメ基礎点とは表5.16の関数によって求められる。

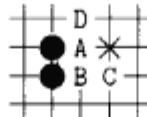
表 5.16：ナラビダメ基礎点関数

包囲連のダメ数	1	2	3	4以上
ナラビダメ基礎点	3.0	2.5	2.0	1.5

- 一間点

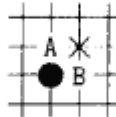
標的連のダメ数が2で、標的連(黒石)に対して図5.13のような一間(×)の手がある時、その手。

$$\text{着手評価値} = \begin{cases} 15.0 & (\text{その手の高さが2以上でBが空点の時}) \\ 1.0 & (\text{上記以外}) \end{cases}$$



A, x は空点で B は空点か白で C か D が黒であること

図 5.13：標的連に対する一間点



A, B, x は空点であること。

図 5.14：標的連に対するコスミ点

・ コスミ点

標的連 (黒石) に対して図 5.14 のようなコスミ (x) の手がある時, その手.

$$\text{着手評価値} = \begin{cases} \text{ダメ基礎点} + 2.5 & (\text{その手の高さが2以上の時}) \\ 1.0 & (\text{高さが1の時}) \end{cases}$$

包囲連ヌク手

アタリの包囲連をヌク手.

$$\text{着手評価値} = 6.0$$

アタリでキリを入れる手

包囲連に対して, アタリでキリを入れる手がある時, その手. (ただし, その点への守り側の打筋によってできる連のダメ数が1の時は除く)

$$\text{着手評価値} = 10.0$$

包囲連シチョウで捕る手

高さが2で, 石数が1で, ダメ数が2の包囲連でシチョウ探索で捕獲可能の時, その包囲連をシチョウで捕る手.

$$\text{着手評価値} = 8.0$$

パスの手

以上の処理で生成された全ての着手の最大着手評価値が負の時, パスの手.

$$\text{着手評価値} = 0.0$$

● 攻め番の時

ダメ点

守り番と同様.

一間点

守り番と同様.

- コスミ点

守り番と同様.

- 包囲連ツグ手

高さ2で、石数1で、アタリの包囲連のダメ点の高さが1の時、そのアタリの包囲連をツグ手.

$$\text{着手評価値} = 6.0$$

- 包囲連シチョウから逃げる手

アタリの包囲連をツグ手がシチョウ探索で捕られない時、そのアタリの包囲連をツグ手.

$$\text{着手評価値} = \begin{cases} 30.0 & (\text{包囲連の石数が2以上の時}) \\ 6.0 & (\text{包囲連の石数が1の時}) \end{cases}$$

- 元をツグ手

以上の処理で生成された全ての着手の内、最大着手評価値が負で、最大着手評価値の着手によってできる連のダメ数が1の時、最大着手評価値の着手の位置をそのダメ点の位置へ移動する。(ただし、その点への着手によってできる連のダメ数が1の場合には除く)

5.3.5 性能評価

図 5.15 の4つの捕獲問題について捕獲探索の性能を測定した. ×印のマークのついた白石が標的となっている連を表わしている. 全て黒番で標的連が捕獲できるかどうかの問題であり、答えは全て捕獲可能である.

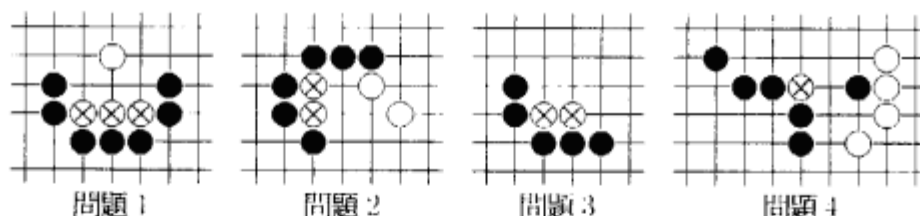


図 5.15 : 捕獲の問題

問題1は有名な「鶴の巣ごもり」の問題で、問題2は「ハサミツケ」、問題3は「ゲタ」の問題である。これらの問題は手数も短く、1手目の正解手を発見できれば後は簡単な問題であり、10級程度の棋力があれば解ける問題である。問題4は、「カク」から「コルミシチョウ」に持っていく多少複雑な問題で、手数も長いので、5級程度の問題であると思われる。4つの捕獲問題について、捕獲探索に要した手数と時間の測定結果が表 5.17 である。

1手当たりの探索時間は問題によってかなり異なる。これは、捕獲探索における終端局面の評価や着手生成の処理の中で呼ばれているシチョウ探索の処理時間が問題によってかなり異なることが最大の原因であると思われる。

また、1級から10級程度の捕獲問題50問について、その正解率や手数を調べた(表 5.18)。不正解となった問題は7問あったが、内4問は誤った解を出し、残り3問は探索手数が200手を越えた為不明となった。

誤った解を出したそれぞれの原因を以下に示す。

表 5.17：捕獲探索性能の測定結果

問題	探索手数 (N)	探索時間 (T)	1 手当たりの 探索時間 (T/N)
問題 1	5 手	59msec	11.8msec
問題 2	9 手	141msec	15.7msec
問題 3	15 手	124msec	8.3msec
問題 4	39 手	469msec	12.0msec

表 5.18：捕獲問題 50 問での捕獲探索性能の測定結果

正解率	86%
平均探索手数	33 手
合計探索時間	26 秒

- 敵石をスクことによって根元をツグような攻め側の手が生成されない。
- コウの手を単に打着禁止手として処理している為、コウにせず捕獲できる問題であっても、コウになる手の方を正解とする場合がある。
- ダメ数 3 以上の隣接敵連に対する攻め側の手が生成されない。
- 標的連とコスミでつながっている石のダメをツメるような攻め側の手が生成されない。

今後誤答を出した原因を取り除くような改良を加えることで正解率は高くなると思われるが、反対にそのような改良によって全体の探索時間が増えることも考えられるので、改良にあたっては充分注意が必要である。

不明となった 3 問について、探索手数の制限をなくして探索させたところ、それぞれ 202 手、363 手、759 手で解が出た。探索手数が多く掛かった原因としては、標的連のダメを詰める手の評価値に問題があるようである。ダメを詰める手の価値を算出する時にダメを詰める効率のみで決めている為、打った手がすぐ取り返されて結局ダメを詰めたことにならないような手の価値も高くなってしまい、今後このような点を改良することで探索手数を減らすことができる。

5.4 連結探索

連結探索は、標的とする結線が連結可能かどうか完全探索するもので、連結可能でない場合には、“キリチガイ”で切れるか“ツキヌケ”で切れるかも求める。

5.4.1 標的対象

線足以外の全ての結線。

5.4.2 終端局面評価関数

局面が以下のいずれかの条件を満たす時、終端局面と判定し、その終端局面の評価値を返す。

- 攻め番で端点が一つの連結連になった時

評価値 = 10 (連結)

- 両端点の連結連の距離 (互いの連結連の一番近い2点の距離) が $\sqrt{5}$ (ケイマ) で、結線と垂直に敵石が2つ並んだ時

- 攻め番

評価値 = 10 (ツキヌケ)

- 守り番

評価値 = -10 (ツキヌケ)

- 両端点の連結連の距離が $\sqrt{4}$ (一間) で、結線と垂直に敵石が3つ以上並んだ時

- 攻め番

評価値 = 10 (ツキヌケ)

守り番

評価値 = -10 (ツキヌケ)

- 両端点の連結連の距離が $\sqrt{2}$ でギリチガイバタンの時

攻め番

1. どちらかの端点の連が敵手番で捕獲可能

評価値 = 10 (ツキヌケ)

2. 1以外で、両方のギリ石が敵手番で捕獲可能、または どちらかのギリ石が味方手番で脱出不能

評価値 = -10 (連結)

3. 上記以外

評価値 = 5 (ギリチガイ)

- 守り番

1. どちらかの端点の連が味方手番で脱出不能

評価値 = -10 (ツキヌケ)

2. 1以外で、どちらかのギリ石が敵手番で捕獲可能

評価値 = 10 (連結)

3. 上記以外

評価値 = -5 (ギリチガイ)

- 探索の深さが20手を越えた時

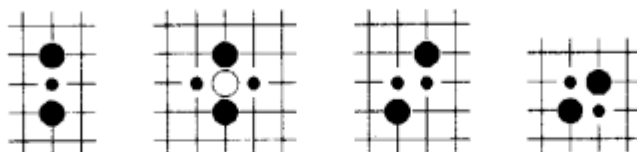
評価値 = 0 (不明)

- 探索の総手数が300手を越えた時

評価値 = 0 (不明)

5.4.3 強制手

両端点の連結線の距離が一間、ケイマ、キリチガイの時、以下に示す点が空点ならその点を強制手とする。



5.4.4 着手生成

強制手がない時、以下の着手について結線の両端点の中心に近いものから順に候補手を生成する。

- 端点を斜辺とする四角形内 (一間、二間の時はその左右一路まで) のすべての点。
- 上記の範囲内にアタリの敵連がある時、そのダメ点。

5.4.5 性能評価

図 5.16 の 4 つの連結問題について連結探索の性能を測定した。× マークのついた白石が標的となっている結線を表わしている。全て黒番で標的の結線が切れるかどうかの問題であり、問題 1 は切斷不能であり、問題 2 と 3 はキリチガイ、問題 4 はツキシケという答えになる。

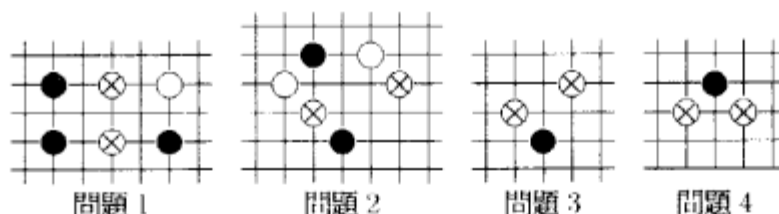


図 5.16 : 連結の問題

4 つの連結問題について、連結探索に要した手数と時間の測定結果が表 5.19 である。

表 5.19 : 連結探索性能の測定結果

問題	探索手数 (N)	探索時間 (T)	1 手当たりの 探索時間 (T/N)
問題 1	19 手	0.197 秒	10.4msec
問題 2	113 手	13.2 秒	92.3msec
問題 3	6 手	0.927 秒	155msec
問題 4	12 手	0.095 秒	7.9msec

探索に要する時間のほとんどは、終端局面評価の為の捕獲探索の処理に要する時間である。従って、1 手当たりの探索時間は問題によって大幅に異なる。探索結果の正解 / 不正解も捕獲探索の結果の正確さに大きく依存している。

連結探索は時間が掛かり過ぎる為、現在は制限して使っている。今後、終端局面評価や着手生成時に使っているヒューリスティックを改良することによって、探索の性能を向上させることが必要である。

5.5 詰め碁探索

詰め碁探索は、標的とする完全包囲の群の死活を完全探索するもので、標的群に対する生 / 死 / コウの結果を求める。ただし、詰め碁探索は現在その駆動条件を検討中の段階であり、対局モード中に呼ばれることはない。実験解析モードでのみ駆動できる。

5.5.1 標的対象

完全包囲の群。

5.5.2 コウの為の特殊処理

詰め碁では結果がコウになる問題がしばしばある。結果がコウになる問題とは、探索の過程でコウが生じ、コウの勝ち負けで生死の結果が決まる場合のことである。コウ勝ちを条件として成功する場合より無条件に成功する方が当然望ましいので、コウ勝ちを条件とする以外成功する手順のない時のみ、コウという結果を返す必要がある。この為、以下のようなコウの為の特殊処理を行う。

- コウの手の着手は最後に生成する。
- 直前の守り側の手がコウの手の時のみ、攻め側にもバスの着手を生成し、その直後の守り側の着手ではバスの手を生成しない。
- コウを着手した場合、その局面に対する繰り上がり評価値が 10 (成功) の時は 5 (コウ) にする。

5.5.3 終端局面評価関数

局面が以下のいずれかの条件を満たす時、終端局面と判定し、その終端局面の評価値を返す。ただし、要石とは、標的群の自己石の内要石度の最も高い石の連のことであり、探索駆動時に求めておく。要石度とは以下の式で定義する。

要石度 = その石の連のダメ数 - その石と標的群の重心との間の距離

ここで、標的群の重心とは、標的群の自己石の座標を平均したものとする。

- 攻め番で要石のダメ数が 1 の時

評価値 = 10 (成功)

- 探索の深さが 30 手を越えた時

評価値 = 0 (不明)

- 探索の総手数が 10,000 手を越えた時

評価値 = 0 (不明)

5.5.4 着手生成

攻め側 / 守り側の打てる点の範囲を探索駆動時に求めておく。

- 守り側の打てる点

標的群の完全包囲の内側の点 (包囲結線の石と包囲結線の遮断点によって囲まれた内側の点) とする。

- 攻め側の打てる点

守り側の打てる点と守り側の打てる点に隣接する空点との和集合とする。

また、探索駆動時に打てる点を有望な点の順にソートしておく。打てる点の有望度は以下のよう
に決める。ただし、複数の手が同じ点になる時は評価値を合算する。また、同じ評価値が複数ある
時は、盤上の位置がより高い方の手を優先し、高さも同じ場合はより中央の手を優先する。

- 外ダメ点

攻め側の打てる点で守り側の打てる点でない時

$$\text{有望度評価値} = 0$$

- 壁点

守り側の打てる点で外ダメ点に隣接している時

ダメ場

空点の内点 (壁点以外の守り側打てる点) に隣接していない時

$$\text{有望度評価値} = 0$$

- 非ダメ点

上記以外の時

$$\text{有望度評価値} = 10$$

- 内点

壁点以外の守り側打てる点の時

$$\text{有望度評価値} = \text{ダメ基礎点} + \text{位置基礎点}$$

ここで、ダメ基礎点とは、その点の隣接点の内守り側の石以外の点の数を N_d とする時、
表 5.20 の関数によって求められる。

また、位置基礎点は、その点の標準座標に従い、表 5.21 の関数によって求められる。

表 5.20 : ダメ基礎点関数

N_d	0	1	2	3	4 以上
ダメ基礎点	0	0	1	7	15

表 5.21 : 位置基礎点関数

その点の標準座標	(1,2)	(2,2)	以外
位置基礎点	2	1	0

表 5.22 : 要石のダメ点有望度評価値関数

要石のダメ数	1	2	3	4以上
有望度評価値	10	5	2	1

- 要石のダメ点

攻め側の打てる点についてその点が要石のダメ点である時、表 5.22 の関数によって有望度評価値を求める。

以上で求めた打てる点を使って、以下の着手を以下の順に生成する。

- 打てる点

各手番に該当する打てる点の内、空点でかつコウ点でない手をソートされた順に生成する。

- パス

直前に打たれた敵の着手がパスでない時、パスの手を生成する。

ただし、守り番の時はその局面でそれまで生成された打着禁止手以外の着手の数が 2 の時のみとし、攻め番の時は直前に打たれた敵の着手がコウ点の時のみとする。

- コウ点

コウ点があればコウを取り返す手を生成する。

5.5.5 性能評価

図 5.17 の 4 つの詰め碁問題について詰め碁探索の性能を測定した。全て黒番で包囲されている石が生きるかどうかの問題であり、問題 1~3 の答えは死、問題 4 の答えは生である。

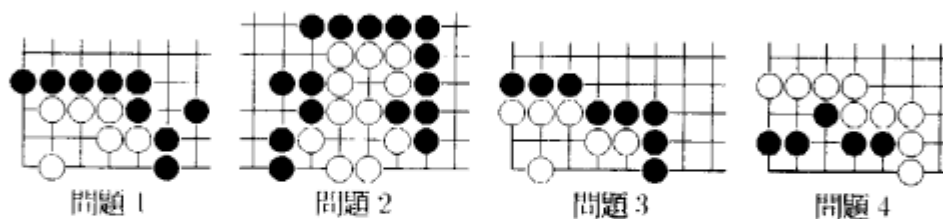


図 5.17 : 詰め碁の問題

問題 1 は 2 の一に置く問題で、問題 2 はサガリで欠眼に導く問題である。問題 3 はウツテガエシをねらってホウリコム問題で、問題 4 は石の下の問題である。4 つの詰め碁問題について、局面記憶あり (5.1.1 参照) の詰め碁探索に要した手数と時間の測定結果が表 5.23 である。

1 手当たりの探索時間が、探索手数が多いほど少ないのは、探索駆動時に行われる前処理が関係していると思われる。特に打てる点を有望な手の順にソートする処理は比較的重い処理で、測定した 4 つの問題で平均 0.3 秒程掛かっている。

表 5.23：詰め碁探索性能の測定結果

問題	探索手数 (N)	探索時間 (T)	1手当たりの 探索時間 (T/N)
問題 1	386 手	2.04 秒	5.3msec
問題 2	462 手	2.51 秒	5.4msec
問題 3	591 手	3.03 秒	5.1msec
問題 4	1,572 手	7.22 秒	4.6msec

5.6 眼数探索

眼数探索は、標的となる地の点に完全一眼ができるかどうか完全探索するもので、群の眼数計算等の為に使われる。ただし、眼数探索は現在その駆動条件を検討中の段階であり、対局モード中に呼ばれることはない。実験解析モードでのみ駆動できる。

5.6.1 標的対象

サイズ 1 以下の地 (閉領域) を対象とする。

5.6.2 終端局面評価関数

局面が以下のいずれかの条件を満たす時、終端局面と判定し、その終端局面の評価値を返す。ただし、眼形対象領域とは、対象となる地の点とその地の点に隣接し、かつ単純な欠眼にならない全ての空点とし、完全一眼とは、ダメ 2 以上の一つの連で囲まれた領域とする。また、単純な欠け目にならない空点とは、下図の × の点で敵石が A の点には 1 つもなく、B の点にはたかだか 1 個の時とする。ただし、敵石はダメ 3 以上かダメ 2 の場合はシチョウで捕獲されないことを条件とする。



- 守り番で眼形対象領域に完全一眼ができた時

評価値 = 10 (成功)

- 攻め番で眼形対象領域に完全一眼ができた時

評価値 = -10 (失敗)

- 守り番で眼形対象領域に単純な欠け目にならない空点がなくなった時

評価値 = -10 (失敗)

- 攻め番で眼形対象領域に単純な欠け目にならない空点がなくなった時

評価値 = 10 (成功)

- 探索の深さが 20 手を越えた時

評価値 = 0 (不明)

- 探索の総手数が 10,000 手を越えた時

評価値 = 0 (不明)

5.6.3 着手生成

以下の着手について着手評価値の大きい順に探索する。ただし、複数の着手が同じ点になる時は 1 つとし、着手評価値は各着手評価値の内最大の評価値とする。また、同じ評価値が複数ある時は、盤上の位置がより高い方の手を優先し、高さも同じ場合はより中央の手を優先する。

- 守り番

コスミ範囲の手

眼形対象領域の点からコスミ範囲にある全ての空点 (評価値は探索駆動時に以下の評価値を算出しておき、探索中は評価値の再計算をしない)。

壁点 守り番の候補手以外の点に隣接する空点

着手評価値 = 6

内点 壁点以外の空点

着手評価値 = $D \times 2$

ただし、 D は攻め側が打着してできる連のダメ点から壁点を除いた点の数で最大 4 とする。

敵石を抜く手

眼形対象領域の点からコスミ範囲にある全てのダメ 1 の敵石をヌク手

着手評価値 = 10

- 攻め番

コスミ範囲の手

守り番と同様

外点

眼形対象領域の点からコスミ範囲にある全ての空点に隣接する空点

着手評価値 = 2

敵石を抜く手

守り番と同様

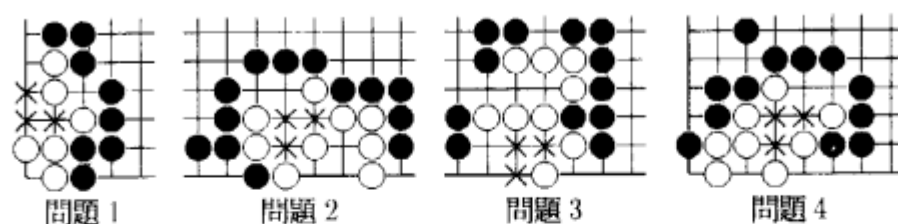


図 5.18：眼数探索の問題

表 5.24：眼数探索性能の測定結果

問題	探索手数 (N)	探索時間 (T)	1手当たりの 探索時間 (T/N)
問題 1	46 手	0.86 秒	18.7msec
問題 2	83 手	1.13 秒	13.6msec
問題 3	94 手	0.97 秒	10.2msec
問題 4	360 手	3.38 秒	9.4msec

5.6.4 性能評価

図 5.18 の 4 つの眼数問題について眼数探索の性能を測定した。全て黒番で × のマークのついた点の範囲内に眼ができるかどうかの問題であり、答えは全て眼ができない (半眼になる) である。

4 つの眼数問題について、局面記憶あり (5.1.1 参照) の眼数探索に要した手数と時間の測定結果が表 5.24 である。

1 手当たりの探索時間が、探索手数が多い程少ない傾向が見られるのは、探索駆動時に行われる一定時間の前処理の為と思われる。

5.7 多重標的捕獲探索

多重標的捕獲探索は、複数の連を標的とし、いずれかの連が捕獲可能か完全探索するものである。ただし、この多重標的捕獲探索は多重の標的 / 意図を扱う探索の 1 つの簡単な試みの段階であり、対局モード中に呼ばれることはない。実験解析モードでのみ駆動できる。

5.7.1 標的対象

任意の複数の同色連を対象とする。ただし、実験解析モードから呼び出す時は、操作を簡単にする為、ある 1 つの連を指定することによって、その連に隣接する全ての敵連を標的対象にするようにしている。

5.7.2 終端局面評価関数と着手生成

以下のように局面が終端かどうかを判定する。終端局面である場合はその終端局面の評価値を返し、終端局面でない場合は着手を生成する。

- 守り番でアタリ (ダメ数が 1) の標的連が複数ある時

- 各アタリの標的連をツグ手(アタリの標的連のダメ点とアタリの標的連に隣接するアタリの敵連のダメ点の和集合)の積集合が空集合の時

$$\text{評価値} = -10 \quad (\text{失敗})$$

- 上記以外の時

各アタリの標的連をツグ手の積集合を着手とする。

- 守り番でアタリの標的連が1つの時

- アタリの標的連に対する守り番のシチョウ探索の結果が失敗(逃げられない)の時

$$\text{評価値} = -10 \quad (\text{失敗})$$

上記以外の時

シチョウ探索で成功した手を着手とする。

- 守り番で上記いずれでもない時

$$\text{評価値} = 10 \quad (\text{成功})$$

- 攻め番でアタリの標的連が1つ以上ある場合

$$\text{評価値} = 10 \quad (\text{成功})$$

- 攻め番でダメ数2の標的連が1つもない時

$$\text{評価値} = -10 \quad (\text{失敗})$$

- 攻め番で上記いずれでもない時

全てのダメ数2の標的連のダメ点を着手とする。ただし、各着手について以下の着手評価値を算出し、着手評価値の大きい順に生成する。同じ着手評価値の着手が複数ある時は、盤上の位置がより高い方を優先する。

$$\text{着手評価値} = \sum \frac{10}{\text{隣接する標的連と同色の連のダメ数}}$$

5.7.3 性能評価

図 5.19 の3つの多重標的捕獲探索問題について多重標的捕獲探索の性能を測定した。×印のマークのついた白石が標的となっている連を表わしている。全て黒番で標的連のいずれかが捕獲できるかどうかの問題であり、答えは全て捕獲可能である。

問題1はアタリでキリをいれてから両アタリに持っていく問題で、問題2はアタリの連続から最後にシチョウにする問題である。問題3は問題2と同様アタリの連続から最後にシチョウにする問題であるが、問題1と2が1手打ったことによってできたダメ数2の敵連に対して次々アタリに掛けていくのに対し、離れた別々のアタリを組み合わせでシチョウに持っていく問題になっている。3つの多重標的捕獲問題について、多重標的捕獲探索に要した手数と時間の測定結果が表 5.25 である。

1手当たりの探索時間は問題によってかなり異なる。これは捕獲探索と同様、探索の中で呼ばれているシチョウ探索の処理時間が問題によってかなり異なることが最大の原因であると思われる。

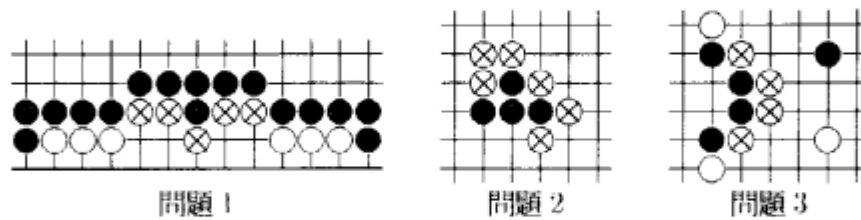


図 5.19：多重標的探索の問題

表 5.25：多重標的捕獲探索性能の測定結果

問題	探索手数 (N)	探索時間 (T)	1手当たりの 探索時間 (T/N)
問題 1	3 手	14msec	4.7msec
問題 2	7 手	107msec	15.3msec
問題 3	17 手	72msec	4.2msec

6 着手決定

認識された局面に対して、いろいろな候補手知識が駆動され、候補手が生成される。各候補手にはいったん評価値が付けられるが、候補手フィルタによって再評価が行われる場合もある。また、形勢判断を行い、その結果によって着手が変更されることもあるが、基本的には最大の評価値合計を持つ点が着手として選ばれる。

6.1 候補手列举

局面認識の処理によって得られたいろいろな対象の状態に応じて候補手知識が駆動され、表 6.26 のような 12 種類の候補手が生成される。

表 6.26：候補手の種類

候補手の種類	対象	手の意図
定石候補手	隅の石の配置	定石の手
辺点候補手	辺に沿った石の配置	ヒラキ、ツメ、ワリウチ等の手
打ち込み候補手	隅の石の配置	三々へのウチコミ / 防御の手
捕獲 / 逃亡候補手	中立の連	石を捕る / 逃げる手
ダメ点候補手	接触する黒 / 白の連	ハネ、ノビ、オシ等の石の競り合いの手
切断 / 連結候補手	弱結線	石をキル / ツナグ手
包囲 / 脱出候補手	小さい包囲度の弱群	弱石を包囲 / 脱出する手
フトコロ拡大 / 縮小候補手	フトコロを拡げる余地のある弱群	弱石のフトコロを拡げる / 狭める手
死活 / 攻め合い候補手	1.5 眼の弱群や攻め合い注意群	中手の手や攻め合いで相手のダメを詰める手
分離 / 連絡候補手	分離できる大きな族	石を分離 / 連絡する手
地模様候補手	弱い境界線を持つ族	模様を地にする / 地になるのを防ぐ手
模様接点候補手	接触する黒 / 白の族	模様の勢力を拡げつつ相手の模様の勢力を制限する手

6.1.1 定石候補手

盤上の四隅について定石が進行中であれば、定石候補手を生成する。

定石は 1 つの木構造をしたファイル上に納められており、知識エディタツールの 1 つである定石エディタ (7.2.2 参照) によって作成される。現在、239 変化図 (定石木のリーフの数) の定石データが登録されている。

1. 候補手の位置

進行中の定石ノードの全子ノードの位置とする。

ただし、子ノードにバスの着手 (手抜き) がある場合は、バスのノードの全子ノードの位置に反対手番の候補手を生成する。

2. 候補手評価値

各ノードには評価値に対する基礎点のデータが登録されている。また、その時の局面に応じてその基礎点を微調整する為の情報も登録されている場合もあり、その場合は調整条件文と調整点が登録されている。候補手の評価値は該当ノードの基礎点と調整点によって以下のよう求められる。

$$\text{評価値} = \text{基礎点} + \text{調整点}$$

ただし、調整点は調整条件文が真の時だけ加算するもので、その値は正負どちらの場合もある。調整条件文は、現在局面の任意の点のポテンシャル値を指定した定数と比較する論理演算子のみが許されている。

3. 定石進行中の管理

定石進行中とは、各四隅の石の配置が定石木のリーフ以外のノードの石の配置と同じで、かつ定石スコープ内に他の石のない場合をいい、それ以外を“定石はずれ”の状態という。いったん“定石はずれ”となったらその隅については定石候補手は生成されない。定石スコープとは各四隅の $n \times n$ の点の範囲で、 n は以下の式によって求める。

$$n = \text{路数} / 2 - 1$$

6.1.2 辺点候補手

3線もしくは4線の石が、辺上で L 間 ($3 \leq L \leq 12$) で対峙する場合、以下の2種類の辺点候補手を生成する。

1. 布石の手

対峙している石がともに強い場合 (群の強度 > 20) は、布石の手を生成する。

(a) 候補手の位置

i. 辺方向

対峙している2石の中央の位置とする。

L が偶数の場合は対峙している2石の内弱い方へ寄せる。

ただし、 L 間で異色対峙している場合は石の強弱によらず黒の候補手は黒石から2間の位置 (白石から1間の位置)、白の候補手は白石から2間の位置 (黒石から1間の位置) とする。

ii. 高さ

3線もしくは4線である。

対峙している2石の色と高さによって決める (表 6.27)。

iii. 候補手位置補正

ただし、同色対峙で対峙 L 間内の5線もしくは6線に敵石がある場合は、その敵石の2路下の位置に補正する (図 6.20 の左図)。

また、その敵石と対峙しているどちらかの石とが辺方向に1路しか離れていない場合は、さらに1路中央寄りの位置に補正する (図 6.20 の右図)。

(b) 候補手評価値

L の大きさによって決める (表 6.28)。

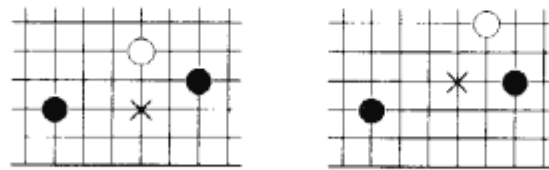
ただし、同色対峙で対峙 L 間内の5線もしくは6線に敵石がある場合は求めた評価値をさらに2倍する。

表 6.27 : 布石の手の高さ

対峙して いる一方 の石	対峙しているもう一方の石			
	黒		白	
	3 線	4 線	3 線	4 線
黒	3 線	4/3	4/3	3 線
	4 線	4/3	3 線	3 線
白	3 線	3 線	3/4	3/4
	4 線	3 線	3 線	3 線

4/3... 黒の候補手は 4 線で白の候補手は 3 線.

3/4... 黒の候補手は 3 線で白の候補手は 4 線.



×... 補正後の候補手の位置.

図 6.20 : 布石の手の位置補正

(c) 特殊処理

同色対峙の布石の手の敵側の候補手は辺点候補手と呼ばず、打ち込み候補手 (6.1.3 参照) と呼ぶ.

2. ヒラキ / ツメの手

対峙している石のどちらかが弱い場合 (群の強度 ≤ 20) ヒラキ / ツメの手を生成する.

(a) 候補手の位置

i. 辺方向

対峙している 2 石の内より弱い方の石に対して、味方側の候補手 (ヒラキ) は 2 間の位置、敵側の候補手 (ツメ) は 1 間の位置とする.

ただし、L が 3 間の時は中央の位置とする.

ii. 高さ

対峙している 2 石の内より弱い方の石と同じ高さとする.

(b) 候補手評価値

評価値 = 対峙している候補手の内より弱い方の石の群の手入れ重要度 + 4

表 6.28 : 布石の手の評価値

L	3	4	5	6	7	8	9	10	11	12
評価値	8.0	12.0	16.0	16.4	16.8	17.1	17.4	17.6	17.8	18.0

6.1.3 打ち込み候補手

盤上の四隅に対して三々への打ち込みの手を生成する。三々以外の一般的打ち込みの手については辺点候補手生成処理で行われる (6.1.2 参照)。

1. 候補手の位置

隅の 4×4 の範囲に星 (4, 4) の位置のみ着手があり、星以外の点は全て空点の場合、三々 (3, 3) の位置に打ち込み候補手を生成する。

2. 候補手評価値

$$\text{評価値} = 20$$

6.1.4 捕獲 / 逃亡候補手

盤上の中立の連に対して捕獲 / 逃亡候補手を生成する。

1. 候補手の位置

連の捕獲点の位置に捕獲候補手を生成し、脱出点の位置に逃亡候補手を生成する。

2. 候補手評価値

$$\text{評価値} = 10 + (\text{連の石数} - 1) \times 2 + \text{連の所属する群の種石重要度}$$

3. 愚形補正

逃亡候補手 (連の脱出点の位置) の着手がアキ三角の愚形になる時、着手の位置を補正する。着手位置の補正は、連の脱出点を求めた時の捕獲探索で枝刈りされた残りの手と愚形となった脱出点のコスミ範囲内の空点との和集合の点について、愚形以外の脱出点が見つかるまで捕獲探索をやり直すことによって行われる。ただし、愚形手のコスミ範囲の点については以下の条件を付加する。例えば、図 6.21 のような場合 A の手は D が白石でなく、B の手は D、E が、また C の手は E、F が白石でないこととする。また、探索のやり直しの際、盤上の位置がより高い方の手を優先し、高さが同じ場合はより中

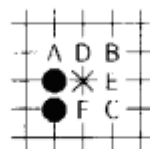


図 6.21 : 愚形補正時の探索条件

央の手を優先して探索する。

4. 捕獲 / 逃亡位置最適化

捕獲探索は最初に成功した手のみを求めている。これは処理時間の節約の為であるが、反面同じ意図を満足するより良い手を発見できない場合が生じる。捕獲 / 逃亡位置最適化は、処理時間がある程度抑えながら、そのような手を見失わないようにする為の処理で、以下のように行われる。

捕獲 / 逃亡候補手のケイマ範囲の点に対して、捕獲 / 逃亡候補手以外の全ての候補手の評価値が元の点より大きい点を求める。求めた点について評価値の大きい点から順に、その点でも同じ捕獲 / 逃亡の意図が満足するか捕獲探索によって求める。満足する点が見つかれば捕獲 / 逃亡候補手をその点に移す。

図 6.22 の例では A の点に逃亡候補手が生成される。捕獲 / 逃亡位置最適化の処理により、B の点でも逃亡の意図を満足し、かつ B の手は三線足の連結の価値があることが発見され、A の点の逃亡候補手は B の点へ移動することになる。

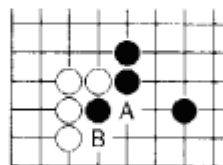


図 6.22：捕獲 / 逃亡位置最適化の例

6.1.5 ダメ点候補手

黒 / 白の石が互いに近接する場合、それぞれの石の勢力に関する攻防手として、ダメ点候補手を生成する。

囲碁用語で言うハネやノビ / オシの手になる。

1. 候補手の位置

図 6.23 のような黒 / 白の石の配置がある場合の × の位置で、(a) をナラビダメ、(b) を共有ダメと呼ぶ。ただしナラビダメでかつ共有ダメとなる場合はナラビダメが優先される。

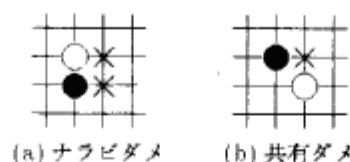


図 6.23：ダメ点候補手パターン

2. 候補手評価値

評価値 = ダメ点の価値 + ボーナス点

3. ダメ点の価値

ダメ点の価値は、種石係数を l 、ダメ数 n の連のダメ基礎点を $v(n)$ 、ダメ増分度を δ 、愚形係数を g とする時、以下の式で定義する。ただし、ナラビダメの場合は手番側の石についてダメ点の価値を求め、共有ダメの場合は黒 / 白両方の石のダメ点の価値を合計 (複数の組み合わせがある場合は最大値) する。

$$\text{ダメ点の価値} = l \times v(n) \times \delta \times g$$

$$t = \begin{cases} 2 & (\text{種石の時}) \\ 1 & (\text{種石以外の時}) \end{cases}$$

連のダメ数 (n)	1	2	3	4	5	6 以上
ダメ基礎点 ($v(n)$)	10	5	3.3	2.5	2	1

$$\delta = \min(\text{ダメ点打着後の連のダメ数} - \text{打着前の連のダメ数}, 2)$$

$$g = \begin{cases} 0 & (\text{共有ダメ点着手時アキ三角の形になる時}) \\ 1 & (\text{以外}) \end{cases}$$

4. ボーナス点

(a) ナラビダメボーナス点

ナラビダメに対しては図 6.24 のようにハネ (a) とノビ (b) の 2 種類の候補手を生成するが、そのどちらを優先するかを決めるのが、このナラビダメボーナス点である。

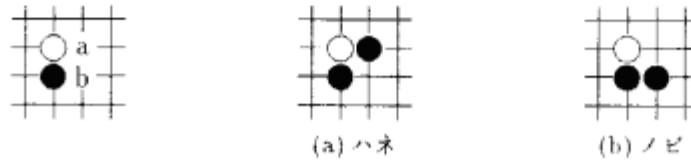


図 6.24 : ハネとノビ

i. ナラビダメボーナス点の位置

対象としている連 (図 6.24 の黒石) のダメ数を Df 、隣接する敵連 (図 6.24 の白石) のダメ数を De 、ハネによってできる連 (図 6.24 の a) のダメ数を Da 、敵がハネた場合にできる連 (図 6.24 の b) のダメ数を Db とする時、

$$\min(Df, Da) \geq \min(De, Db) \quad \text{ならば} \quad \text{ハネの位置 (図 6.24 の a)}$$

$$\min(Df, Da) < \min(De, Db) \quad \text{ならば} \quad \text{ノビの位置 (図 6.24 の b)}$$

とする。

ii. ナラビダメボーナス点

対象としている石の高さが 2 線以上でかつ以下のいずれかの条件を満たす時 8 点とし、以外は 2 点とする。

- $Df \geq 2$ でかつ $Db \leq 2$ の場合。
- $Df = De = 3$ でかつ $Da = Db = 3$ または 4 でかつ隣接する黒 / 白の石数がともに $Da - 1$ の場合。

6.1.6 切断 / 連結候補手

石の切断 (キリ) や連結 (ツナギ) の候補手を生成する。盤上の全ての弱結線 (4.4.2 参照) が対象となる。ただし、以下の場合には除外する。

- 結線の両端点が同じ群に所属する時

- 結線の両端点が同じ族に所属し、かつその族の中地が4以上の時
- 以下の処理で求めた切断 / 連結候補手の点が結線の敵側 (のぞき側) の群に所属している時

対象とする弱結線に対して以下のように候補手を生成する。

1. コスミでキリチガイになっている時

(a) 候補手の位置

切断候補手 (結線の敵側手番のみ) はキリチガっている敵中立連の脱出点の位置とし、連結候補手 (結線側手番のみ) はキリチガっている敵中立連の捕獲点の位置とする。

(b) 候補手評価値

$$\text{評価値} = 15$$

2. 上記以外のコスミ、一間、ケイマの場合

(a) 候補手の位置

切断候補手は結線の敵側手番の連結探索によって得られた次の1手の位置とし、連結候補手は結線手番の連結探索によって得られた次の1手の位置とする。

(b) 候補手評価値

黒番 / 白番での連結探索の結果より表 6.29 に従い決定する。

表 6.29 : コスミ、一間、ケイマの連結 / 切断候補手評価値

結線側手番	結線の敵側手番	
	ツキヌケ	キリチガイ / 不明
連結	30 点	15 点
キリチガイ / 不明	22.5 点	10.5 点

3. 二間、大ゲイマ、ハザマ、二線足、三線足、四線足の場合

(a) 候補手の位置

「切断 / 連結候補手ボタン知識」(付録 参照) により決定される位置とする。

(b) 候補手評価値

「切断 / 連結候補手ボタン知識」により決定される評価値とする。

6.1.7 包囲 / 脱出候補手

盤上の中立群 (強度が1以上35未満の群) に対して包囲 / 脱出候補手を生成する。

1. 候補手の位置

(a) 援軍との分離 / 連絡の手

包囲度3以下で援軍あり (4.5.7 参照) の時、援軍との分離 / 連絡の手を生成する。援軍の石と最も近い群の石を求め、その2石の中央の位置とする。ただし、援軍の石が複数ある場合は群の強度の大きい方を優先する。

(b) 包囲 / 脱出の手

上記以外で包囲度が3～10の時、包囲 / 脱出の手を以下のようにして生成する。

- 群の群ダメ点グループ (4.5.4 参照) 毎に以下のように候補手を生成する。
- 群ダメ点グループ内の群ダメ点が6個以上ある時はグループ内の中心寄りの5点を求める。
- 求めた5個以下の点の重心を求め、その重心に最も近い群の石を求める。この石を基点と呼ぶ。
- 基点からの一間 / コスミ / ナラビの点の内、重心に最も近い点を求める。
- 求めた点が一間の時、以下の補正を行う。
 - 一間の中間の点が敵石に隣接している時
一間の中間の点 (ナラビ) の位置に補正する。
 - 基点の石の横に敵石がある時
基点の石の連のダメ数が3以上で敵石の連のダメ数が2以下であれば、敵石へのハネ (コスミ) の位置に補正する。

(c) 包囲網強化 / 突破の手

上記いずれでもなく脱出可能 (4.5.7 参照) の群の時、包囲網強化 / 突破の手を生成する。

- 群の包囲結線に二間、大ゲイマ、四線足の強結線がある時
その結線の遮断点の位置とする。
- 群の包囲結線に二線足以外の弱結線がある時
その結線に対する連結 / 切断候補手 (6.1.6 参照) の位置を包囲網強化 / 突破の位置とする。

2. 候補手評価値

$$\text{評価値} = \text{群の手入れ重要度}$$

3. 包囲 / 脱出候補手の再評価

包囲 / 脱出の手と包囲網強化 / 突破の手については、実際に打って見て以下の条件を満足しない場合は候補手を生成しない。

- 候補手の位置を起点として包囲度算出の処理 (4.5.4 参照) を行い、包囲度が5以上になるか、または包囲点に自群以外の味方群に所属している点があること。

6.1.8 フトコロ拡大 / 縮小候補手

盤上の中立群に対して包囲 / 脱出候補手が生成されない時にフトコロ拡大 / 縮小候補手を生成する。ただし、現在はヨセ候補手も兼ねているので、中立群でない場合にも候補手は生成される。

1. 候補手の位置

「フトコロ候補手拡大 / 縮小ボタン知識」 (付録参照) により決定される。

2. 候補手評価値

対象としている石の群が中立群である時、

$$\text{評価値} = \text{群の手入れ重要度}$$

上記以外の時、

$$\text{評価値} = V_f$$

ただし、 V_f は「フトコロ候補手拡大 / 縮小ボタン知識」により決められる評価値である。

6.1.9 死活 / 攻め合い候補手

盤上の中立群に対して、包囲 / 脱出及びフトコロ拡大 / 縮小のいずれの候補手も生成されない時、死活 / 攻め合い候補手を生成する。

1. 候補手の位置

(a) 中手の手

群の眼数が 1.5 の時、群中手の位置とする (4.5.3 参照)。

(b) 攻め合いの手

群が攻め合い注意群である時 (4.5.7 参照)、攻め合い対象の敵群の石の全てのダメ点の位置とする。

2. 候補手評価値

$$\text{評価値} = \text{群の手入れ重要度}$$

6.1.10 分離 / 連絡候補手

盤上のサイズ 30 以上の族に対して分離 / 連絡候補手を生成する。分離 / 連絡候補手とは、その手によって 1 つの族が複数族に分離する (またはその手を防ぐ) ような手のことである。

1. 候補手の位置

ポテンシャル分離点の位置とする。ただし、ポテンシャル分離点の算出方法は「馬の背による近似」を使う。

ポテンシャル分離点は、族の敵側手番では分離候補手、味方手番では連絡候補手となる。

2. 候補手評価値

連絡候補手の場合、

$$\text{評価値} = \min(21 - \text{族の強度}/2, 10)$$

分離候補手の場合、

$$\text{評価値} = \text{連絡候補手の評価値} \times 0.8$$

3. ポテンシャル分離点

ポテンシャル分離点とは、1 つの族を複数の族に効果的に分離する手を、盤上のポテンシャルの分布から求めようとするものである。

ポテンシャルを標高と考えると、盤上の各石はそれぞれ 1 つの山と考えることができる。すると族は、一定の標高以上でつながっている山の集まり (山脈) と考えることができる。山脈全体を次第に沈下させていくと、1 つの山脈がいずれ複数の島となる。複数の島になる寸前の海拔 0m 地点をポテンシャル分離点と考える。

つまり、山脈のポテンシャル分離点の地表を削れば、それが最も効果的に 1 つの山脈を複数

の山脈に分ける方法だと考えられる。

ただし、以上のような方法でポテンシャル分離点を求めるには、族の点のポテンシャル値を少しずつ下げていって、その都度複数の族になっているかを検査する必要があるが、その処理のコストは大きい。

4. 馬の背による近似を使ったポテンシャル分離点算出方法

ポテンシャル分離点は、ポテンシャルを標高と考えた時、言わば山脈の馬の背における鞍部の位置になると考えられる。従って、族内の点の中から、その点の周囲のポテンシャルの配置が馬の背の形をしている点を求めることは、ポテンシャル分離点を求める1つの近似となると考えられる。

馬の背の形をした点を求める為に以下の処理を行う。

- 族の各点についてその点を中心に3×3の範囲のポテンシャル分布を求める。
- 図 6.25 の左図のポテンシャル分布のように、中心の点の周囲に中心の点より高いポテンシャルの点の集まりと、中心の点より低いポテンシャルの点の集まりが2個ずつ交互に分布している場合、この中心の点は馬の背の形をしていると考える。

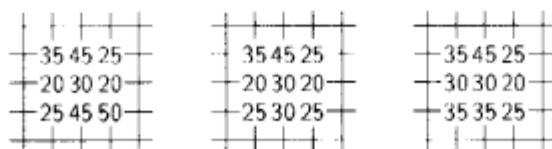


図 6.25：馬の背の形の点の例

- 図の中央の図と右図では、中心の点のポテンシャル値と等しいポテンシャルの点が周囲に存在している。この場合、正確にはポテンシャルの等しい点の方向の先のポテンシャル分布を考慮しなければ、馬の背の形をしているかどうか判断できないが、現在この2つの例の場合もそのままポテンシャル分離点としている。
- ただし、以下の点は馬の背の形をしていてもポテンシャル分離点からは除外している。
 - ・ 右に隣接している点
 - ・ 高さが2線以下の点

6.1.11 地模様候補手

地模様候補手は模様を地にしようとしたり、逆に地になるのを防ぐような手である。ただし、現在模様の存在を認識しておらず、一定の距離の同色の2石がある時に、取り合えずその間に打つような手を生成する。運良くその背後に模様があった時には良い手となる可能性があるが、それ以外は無駄な手を生成している場合もある。

盤上の強度が20を越える群について以下のような候補手を生成する。

1. 候補手の位置

- (a) 地へ侵入する手または侵入を防ぐ手
包囲 / 脱出候補手の包囲 / 脱出の手 (6.1.7 参照) と同様にして求める。
- (b) 地の包囲網を強化 / 突破する手
群の石の周囲の石の配置が図 6.26 のボタンとなる時、ボタンによって求められる位置。

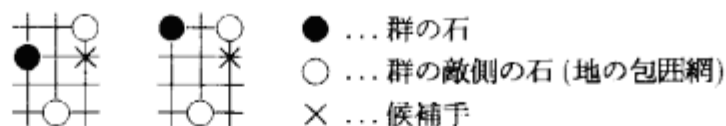


図 6.26：地の包囲網を強化 / 突破する手のボタン

2. 候補手評価値

$$\text{評価値} = \min(\text{敵族のサイズ}/2, 15)$$

ただし、敵族は地へ侵入する手または侵入を防ぐ手の場合、群の群ダメグループの群ダメ点が入っている敵族のことで、地の包囲網強化 / 突破の手の場合は、ボタン上の敵石が入っている族のことである。敵族が複数ある時はそれらの族の平均サイズの $1/2$ を敵族のサイズとする。

6.1.12 模様接点候補手

模様接点候補手とは、囲碁用語の「両ゲイマ」の位置のように、自分の模様を広げつつ相手の模様を制限するような手である。

1. 候補手の位置

高さが u 線以上の点の内、コスミ範囲内にポテンシャル 1 以上の点と -1 以下の点を持ち、コスミ範囲内は全て空点でかつケイマ範囲に黒 / 白の石のある点を模様の接点と呼び、ケイマ範囲内の黒 / 白の石のことを模様の起点と呼ぶ。

黒 / 白の模様の起点の組み合わせの内、2 点間の距離が大ゲイマ以内の時の模様の接点を候補手の位置とする。

2. 候補手評価値

$$\text{評価値} = m(Sb) + m(Sw)$$

ただし、 Sb とは黒の模様の起点が所属する族のサイズであり、 Sw は白の模様の起点が所属する族のサイズである。また、 $m(S)$ は以下で定義される関数である。

$$m(S) = \begin{cases} 0 & (S \leq 10 \text{ の時}) \\ (S - 10)/8 & (10 < S \leq 50 \text{ の時}) \\ 5 & (S > 50 \text{ の時}) \end{cases}$$

6.2 最終着手決定

列挙された各種候補手より以下の手順で着手が決定される。

1. 死群フィルタ

完全死群となった群に関する候補手を取り除く。

2. 無駄手フィルタ

周囲の局所的な石の配置から明らかに無駄と分かる候補手を取り除く。

3. 定石フィルタ

定石進行中の隅については、定石候補手と辺点候補手以外の候補手を取り除く。

4. 合算

同一の点における全ての候補手の評価値を合算し、評価値合計を求める。

5. 最大評価値合計算出

最大評価値合計の手を求める。ただし、最大評価値合計が0以下の時はパスを着手とする。

6. 自殺手フィルタ

最大評価値合計の手(パス以外)を打ったと仮定してその手が相手番で捕獲可能か調べる。捕獲可能の場合はその点の評価値合計をリセットし、最大評価値合計算出の処理以降をやり直す。捕獲不能であればその手を着手とする。

7. 投了

形勢判断の結果が、投了回数以上不利であれば、上記処理によらず投了を着手とする。ただし、現在投了をむやみにしないよう、投了回数は361目が設定されている。

6.2.1 死群フィルタ

完全死群となった群に関する候補手を取り除く。死群フィルタの対象となる候補手と調べる必要のある群を表6.30に示す。

表 6.30：死群フィルタの対象となる候補手

死群フィルタの対象となる候補手	完全死群かどうか調べる対象となる群
辺点候補手	対峙している2石が所属するそれぞれの群
捕獲 / 逃亡候補手	対象とする連が所属する群
ダメ点候補手	ダメ点ボタンとなる黒 / 白2つの連が所属するそれぞれの群
切断 / 連結候補手	対象とする結線の両端点及びノゾいている石が所属するそれぞれの群
包囲 / 脱出候補手	対象とする群
フトコロ拡大 / 縮小候補手	フトコロボタンの各石が所属するそれぞれの群
死活 / 攻め合い候補手	対象とする群

6.2.2 無駄手フィルタ

敵からの攻撃を守る手の場合、虎口点のように敵が打ってもすぐ取り返せるような位置を守ることは無駄である。敵が打ってできる連のダメ数が1になるような点については以下の候補手を取り除く。

- 切断 / 連結候補手
- 死活 / 攻め合い候補手

6.2.3 定石フィルタ

定石とは隅の局所的な範囲において、いろいろな点を考慮した結果の最善手を教えるものである。従って、盤上の四隅の内定石進行中の隅については定石候補手以外の候補手は取り除く。ただし、辺点候補手は布石における候補手である為取り除かない。

定石フィルタの範囲は定石進行中の隅における $n \times n$ の点の範囲で、 n は以下の式によって求める。

$$n = \text{路数} / 2 - 1$$

6.2.4 自殺手フィルタ

最終着手決定で求められた最大評価値合計の手を実際に打ってみて、着手の反対手番で着手した石が捕獲可能かどうかを捕獲探索によって調べる。捕獲可能な場合はその手以外の手で再度最終着手決定の処理が行われる。

6.2.5 形勢判断

群の死活と死群処理後のポテンシャル値によって形勢判断を行う。黒 / 白の形勢は以下の式によって求める。

$$\begin{aligned} \text{黒の形勢} = & \text{黒の確定地} - \text{白の確定地} + \text{黒の模様等価目数} \\ & - \text{白の模様等価目数} - \text{コミ} - \text{置き石数} \end{aligned}$$

$$\text{白の形勢} = - \text{黒の形勢}$$

ここで、黒の確定地とは以下のいずれかの条件を満たす点の数のことである。

- 完全死群でない黒の群の領域内の黒石及び空点
- 完全死群の白の群の石
- ポテンシャル値が 100 以上の点
- 隣接する全ての点のポテンシャル値が 35 以上の点

同様に白の確定地とは以下のような点の数のことである。

- 完全死群でない白の群の領域内の白石及び空点
- 完全死群の黒の群の石
- ポテンシャル値が -100 以下の点
- 隣接する全ての点のポテンシャル値が -35 以下の点

また、模様とは確定地以外の点の内ポテンシャル値が 0 でない点のことで、ポテンシャル値が正の場合黒の模様と呼び、負の場合は白の模様と呼ぶ。模様の等価目数はその点のポテンシャルの絶対値に対して表 6.31 により決定する。

表 6.31 : 模様等の等価目数関数

ポテンシャル値	～75	～60	～50	～45	～40	～30	～25	～20	～15	～10	～3	～1
等価目数	1.00	0.97	0.90	0.83	0.77	0.63	0.55	0.45	0.33	0.21	0.10	0.02

7 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、試行錯誤的な改良を余儀なくされている本システムのような囲碁プログラムにおいては、見掛け以上に重要な部分である。開発ツールは大きく以下の3種類に分類される。

- 実験解析モード

対局の任意の時点で局面認識や着手決定の処理で得られた結果を参照したり、局所探索を駆動させてその結果や探索過程を見ることができる。

また、各種知識で使われているパラメータを一部変更して対局にどのような影響が表われるかを確認することもできる。

- 知識エディタ

本システム内のデータベース化された知識を編集する為のツール群である。

- 評価用ツール

改良による効果を正確に評価する為のツール群である。

7.1 実験解析モード

対局モードにおける着手入力時、及び局面編集ツールの任意の時点で、実験解析モードに移行できる。実験解析モードでは以下の機能が提供される。

- データ構造表示

現在の局面に対する各種データ構造及びそれぞれの対象の属性を表示する。

- 局所探索手順表示

現在の局面に対して選択された局所探索を実行させ、その探索の過程や結果などを表示する。

- 形勢表示

現在の局面に対する形勢を表示する。

- パラメータのチューニング

対局システムにおける各種囲碁知識の中に使われている様々のパラメータに対する参照 / 更新の処理を行う。

7.1.1 データ構造表示

選択されたデータ構造の種類に従い、以下のような表示を行う。

- ポテンシャル

盤上の全ての点のポテンシャル値を表示する。正のポテンシャル値の場合は黒字に白ヌキの数字で表示し、負の場合は白地に黒色の数字で表示する。数字がポテンシャル値の絶対値を表わす。

また、通常は死群処理前のポテンシャル値が表示されるが、選択により死群処理後のポテンシャル値も表示できる。

- 点

盤上の全ての点の黒 / 白 / 空の状態が表示される。

- 連

盤上の全ての連が数字付きの黒石もしくは白石によって表示される。同じ数字が同じ連に所属する石であることを示す。

- 群

盤上の全ての群が数字付きの黒石もしくは白石によって表示される。同じ数字が同じ群に所属する石であることを示す。また、群の領域及び中立領域の空点に群の石と同じ数字が表示される。さらに、先手中立領域の点には左下がりの斜線の網掛け表示が行われ、後手中立領域の点には右下がりの斜線の網掛け表示が行われる。また、通常は死群処理後の群の状態が表示されるが、選択により死群処理前の群の状態を表示することもできる。

- 族

盤上の全ての族が数字付きの黒石もしくは白石によって表示される。同じ数字が同じ族に所属する石であることを示す。また、族の領域の空点に族の石と同じ数字が表示される。

- 結線

盤上の強結線と弱結線とが、小さい黒石もしくは白石と2つの石を結ぶ線によって表示される。線は実線と破線とがあり、実線が強結線を、破線が弱結線を示す。

また、各データ構造が表示されている状態で任意の対象を選択すると、選択された対象の属性が表示される。

- ポテンシャル

点と同様

- 点

盤上の任意の点を選択すると、その点のポテンシャル値、ダメ数、遮断結線などの属性が表示される。

- 連

盤上の任意の数字付きの石を選択すると、その点が所属する連のダメ数、石数、種石 / 非種石、死活などの属性が表示される。

- 群

盤上の任意の数字付きの石または数字のみの点を選択すると、その点が所属する群の石数、サイズ、中地、包囲度、眼数、強度、重要度、実質手数、分類などの属性が表示される。ただし、眼数については群の更新処理において眼数算出の処理が行われた群についてのみ表示される。

- 族

盤上の任意の数字付きの石または数字のみの点を選択すると、その点が所属する族の石数、サイズ、辺長、中地、包囲度、強度、重要度などの属性が表示される。

- 結線

盤上の任意の結線の両端点を選択すると、その2点を両端点とする結線の結線種別、連結度などの属性が表示される。

7.1.2 局所探索手順表示

選択された局所探索を指定された条件で駆動し、その探索の過程や結果などを表示する。各局所探索駆動時には以下の条件を指定する必要がある。

- シチョウ

標的とする連の位置、手番、及び表示 ID。ただし、手番が標的連の味方手番の時は標的連のダメ数は1、敵手番の時は標的連のダメ数は2であること。

- 捕獲

標的とするダメ数3以下の連の位置、手番、及び表示 ID。

- 多重標的捕獲

標的とする連が共通して隣接している敵連の位置、及び表示 ID。

- 連結

標的とする線足以外の結線の両端点の位置、手番、及び表示 ID。

- 詰め碁

標的とする完全包囲の群の位置、手番、及び表示 ID。

- 眼数

標的とするサイズ1以下の地の位置、手番、及び表示 ID。

選択された表示 ID の内容によって、以下の表示が行われる。

- 表示 ID = 0 の場合

探索中には何も表示せず、探索の終了時に探索の結果と次の1手(探索の結果が成功もしくは不明の時のみ)、及び探索に要した手数と処理時間が表示される。

- 表示 ID = 1 の場合

探索中の仮想的な石の打着 / ハガシの処理に対応して数字付きの石が表示される。数字は探索の深さを表わしている。探索終了時の表示は表示 ID が0の時と同じ。

- 表示 ID = 2 の場合

表示 ID が1の時と同様に探索の1手毎の表示を行い、さらに打着表示の都度進め方の問い合わせを行う。“ステップ”が選択されると、次の打着まで進み、また問い合わせが行われる。“スキップ”を選択し、スキップする手数が入力されると、その手数分表示 ID が1の時と同様の表示がされ、その手数分の表示が終了すると、また問い合わせが行われる。“終了”を選択すると探索が終了する。探索終了時の表示は表示 ID が0の時と同じ。ただし、途中で“終了”を選択した場合結果は不明が返される。

- 表示 ID = 3 の場合

探索は行われず、1 手目の探索の着手候補の位置が数字付きの石で表示される。数字は着手候補の優先順位を示す。

- 表示 ID = 4 の場合

表示 ID が 3 の場合とほとんど同様だが、表示される数字付き石の数字が優先順位でなく、算出された着手評価値の値を示す。ただし、着手評価値を 10 倍した整数部分のみが表示される。

7.1.3 候補手表示

盤上の全ての点の候補手評価値合計を表示する。通常はその局面の手番における評価値合計が表示されるが、選択によりその局面の反対手番の評価値合計を表示させることもできる。また、評価値合計でなく、特定の候補手のみを表示することもできる。

表示は盤上の点にその点の評価値合計の整数部の数字を表示することで行われる。この状態で盤上の数字の位置を選択すると、候補手の明細が表示される。候補手の明細は、その点に生成された全ての候補手の種類とその評価値が表示される。

7.1.4 形勢表示

現在の局面における形勢を表示する。黒 / 白の確定地、模様等個目数、コミ、置き石数などの明細も表示する。

また、確定地と認識された点の位置を小さい石で表示する。

7.1.5 パラメータのチューニング

対局システムにおける各種囲碁知識の中のパラメータに対する参照 / 更新の処理とそれに伴う処理を行う。以下の機能がある。

- パラメータの参照 / 更新

対局システムにおける各種囲碁知識の中のパラメータに対する参照 / 更新の処理を行う (付録「チューニング可能なパラメータ」参照)。

- 再更新

更新されたパラメータで再更新を行わせる。選択できる再更新処理の種類には群更新、族更新、形勢判断、全面更新の 4 種類があり、全面更新では局面認識から最終着手決定までの全ての処理が全面更新によって行われる。

- ロード

チューニング可能なパラメータの内容は全てファイルに保存できる。保存しておいたパラメータファイルのファイル名を指定することによって、そのファイルのパラメータの内容にリセットされる。

- セーブ

チューニングしたパラメータの内容を指定したファイル名のファイルに保存する。この時、“parameter.init”というファイル名で保存しておく、次に対局システム駆動時には自動的にそのファイルのパラメータの内容がロードされる。“parameter.init”というファイルがない時はデフォルトの値 (付録「チューニング可能なパラメータ」参照) が使われる。

7.2 知識エディタ

本システムにおける知識の一部はデータベース化され、対局中に必要に応じて適宜参照される。これらデータベース化された知識は以下のような知識エディタによって編集される。

- ポテンシャルエディタ

ポテンシャルは、石がその周囲の点に及ぼす影響度を求める為のものである。周囲への影響度は基本的に石からの距離に反比例すると考えられるが、辺の近傍の点では特殊な事情があり一律に決められない。そこで、人間の感覚を目安に試行錯誤的に決定することにした。ポテンシャルエディタはそのような試行錯誤に要する手間を簡単にする為のツールである。

- 定石エディタ

囲碁の定石は数が多く、また年々新しい定石も増えている。

定石エディタは、定石の入力や管理を容易にする為のツールである。

- 手筋エディタ

囲碁には、局所的な石の配置に対して有効な手を教える手筋と呼ばれる知識が数多く存在する。

手筋エディタは、そのような手筋の入力や管理を容易にする為のツールである。

ただし、手筋エディタはまだ完成して間もなく、実際に手筋エディタで編集された知識を対局システムで利用されるのはこれからの作業である。現在、ワタリ、ヨセといった手筋から順に活用していく予定になっている。また、現在プログラムによって実現されているダメ点、切断 / 連結、フトコロなどの候補手知識も、今後は当ツールによって統一的に管理されることが考えられている。

7.2.1 ポテンシャルエディタ

ポテンシャルエディタによって以下の内容が編集される。現在登録されている内容については付録の「ポテンシャルボタン知識」参照。

1. ポテンシャルボタン種類の登録

ポテンシャルのボタンは辺との位置関係によって異なる。盤上の石の配置に対応して、異なるポテンシャルボタンの種類を登録する。

2. ポテンシャルボタンの編集

各ポテンシャルボタンの種類に対して、ポテンシャルの分布ボタンを編集する。石の周囲の点にポテンシャル値の数字を入力していくことによって編集を行う。

7.2.2 定石エディタ

定石エディタによって以下の内容が編集される。

1. 定石ノードの追加 / 削除

定石は1つの大きな木構造のデータによって実現されている。定石の各1手は木のノードに対応している。定石に手を追加する場合は、ルートノードからその手を打つ前の局面に到るまでの手順に従って木をたどり、その局面のノードに対して追加したい手の位置のノードを追加することによって行われる。

2. 評価値の設定

定石の各ノードには評価値に関する情報を設定する必要がある。この情報によって対局時に定石候補手が生成される。評価値に関する情報には、基礎点と基礎点を微調整する為の情報とがあり、基礎点は必ず設定する必要がある。

基礎点を微調整する情報とはその時の局面の周囲状況に応じて基礎点を微調整する為のもので、調整条件文と調整点の情報とからなる。調整条件文が真の時には、基礎点に調整点(正負あり)を加算した値が候補手の評価値となる。調整条件文は、現在盤上の任意の点のポテンシャル値と定数との比較のみができる。

3. 手抜き登録

定石は基本的に黒 / 白交互に打つ手順の知識により構成されているが、相手が手抜きした場合の続けて打つ手も登録できる。

7.2.3 手筋エディタ

手筋エディタによって以下の内容が編集される。

1. 適用場所

手筋を登録する場所として、隅 / 辺 / 中央のいずれかを選択する。

2. キーボタン

手筋が生成される石の配置を、適用場所(隅 / 辺 / 中央)に対応した碁盤の格子点上に登録する。各格子点には以下の種類の状態が登録できる。

黒, 白, 空, 黒 / 白, 黒 / 空, 白 / 空, 黒 / 白 / 空

3. 手順木

キーボタンがマッチングした場合に、候補手を生成する為の情報である。木構造の形式であり、手筋の各手が木のノードに対応している。キーボタンが表示されている碁盤上に1手ずつ石を置きながらその位置を登録していく。木の各ノードには位置とともに以下の情報を登録する。

- 評価値

手筋候補手を生成する時の評価値。

- 属性

手筋候補手として生成する時の条件。この条件が真の時のみ候補手を生成する。
例えば、以下のような条件が登録できる。

(例1)...(3, 4)の点のポテンシャル値は15以上であること

(例2)...(5, 6)の群の強度は(4, 3)の群の強度より小さいこと

(例3)...(7, 8)の連に対する黒番の捕獲探索の結果が0(不明)以上であること

このように候補手生成時の局面に対して、データ構造の内容を比較したり、局所探索を駆動してその結果を比較することができる。

4. 適用条件

該当の手筋を適用する為の条件を登録する。以下の2種類がある。

- 局相

以下の内の任意個を選択する。例えば、中盤、終盤、形勢不利を選択した時は「中盤以降で形勢不利の場合」のみ適用することになる。

序盤 / 中盤 / 終盤, 形勢不利 / 有利

- 意図

以下のような手筋の意図を選択する。例えばウチコミが選択されている時、対局システムにてウチコミの手を生成したい時にその意図が選択されている手筋ボタンが適用される。

ウチコミ、ヨセ、捕獲、ニゲ、キリ、ツナギ、包囲、脱出、……

7.3 評価用ツール

囲碁対局システムのようなプログラムでは試行錯誤的な改良が余儀なくされている。従って、改良した場合はその効果をできるだけ正確かつ効果的に調べる必要がある。現在、その為のツールとして以下のものが用意されている。

1. 自動テスト機能

予め用意しておいた問題を解かせて、その正解率や所用時間を表示する局面編集ツール(後述)内の機能である。与えることのできる問題には、捕獲や詰め碁などの探索問題、群の強度などの局面認識の問題、次の1手などの着手決定の問題などがある。

2. 棋譜棋力判定ツール

予め用意された棋譜の手とコンピュータが考えた手に対して人間がどちらの手が良いかを判断していくことによって、棋譜の手とコンピュータの手との相対的評価を行うツールである。

例えば、改良前のコンピュータ同士の棋譜を取っておき、改良後のプログラムで当ツールを走らせると、改良したことの影響として以前と違う手を打つようになった局面が表示される。各局面について改良前と改良後の手でどちらが良い手であるかを人間が評価していけば、改良の効果が評価できる。ある局面でより良い手を打つように改良しても、その改良の為に別の局面で悪い手を打つようになることはしばしば起きる。そのような状況を発見するのに便利である。

3. 棋譜編集ツール

棋譜を簡単に作成する為のツールである。1手目から順に人力していくのではなく、碁盤の上に数字を書いていく要領で人力していくので簡単である。1つの棋譜を5分くらいで入力できる。

4. 局面編集ツール

ある局面を簡単に作成する為のツールである。碁盤上に黒/白の石を適当に置いていく要領で局面が作成できる。詰め碁や次の1手の問題を作成する時に便利である。

5. 遠隔対局機能

改良の効果を調べる為には実際に対局させてみる事が確実な方法である。遠隔対局機能とは、異なるPS1同士、または他のコンピュータの囲碁ソフトと自動的に対戦させる為の、対

局システム内の機能の1つである。前者は古いバージョンと新しいバージョンのシステムを対戦させたり、パラメータを一部変えて対戦させる場合に便利である。後者は通信対局機能を備えた市販の囲碁ソフトと対戦させるのに便利である。

6. 消費時間 / メモリ測定機能

囲碁のようなプログラムでは消費時間を一定時間内に抑えることは重要である。本システムでは一局に要するコンピュータの消費時間を30分以内にするを目安としている。

消費時間を一定時間内に抑えるためには、導入する知識の量を制限する必要がある。従って、知識の処理に要する時間を減らすことは、導入できる知識の量を増やせることにつながる。棋力に関係してくる。

当ツールは実際の対局をしながら、個々の処理に要した時間とメモリを測定する対局システム内の機能の1つである。各知識の効果とそれに要する時間とを考え合わせて、改良の日安とするのに便利である。

7.3.1 自動テスト機能

自動テスト機能は局面編集ツールの1つの機能である。予め用意されたコマンドファイルの内容に従って自動テストを行い、結果をログ出力する。

コマンドファイルの内容は以下のコマンドを要素とするリストである。

1. 次の1手の問題実行

予め用意された複数の次の1手の問題をコンピュータに解かせ採点結果をログ出力する。採点の仕方は1つの問題毎に複数の解答例とその得点を用意し、コンピュータの手と解答例が一致した場合の得点を合計する。

コマンド形式：

```
sentaku mondai(FAs)
FAs = [FA1, FA2, ...]
FAi = { 局面ファイル名ストリング, As }
局面ファイル名ストリング = 局面編集ツールで作成されたファイルの
                             ファイル名(手番の情報も含まれている)
As = [A1, A2, ...]
Ai = { 次の1手の候補の座標ストリング, 得点整数 }
座標ストリング = X座標ストリング + Y座標ストリング
X座標ストリング = "A", "B", ..., "T"
Y座標ストリング = "1", "2", ..., "19"
```

2. 次の1手の問題実行2

予め用意された複数の次の1手の問題をコンピュータに解かせ採点結果をログ出力する。採点の仕方は1つの問題に正解手を1つ用意し、コンピュータの考えた最良の10個の手のいずれかがその正解手と一致した場合の数を合計する。

コマンド形式：

```
jiko_saiten(FAs)
FAs = [FA1, FA2, ...]
FAi = { 局面ファイル名ストリング, 次の1手の座標ストリング }
```

3. 詰め碁問題実行

予め用意された複数の詰め碁の問題を解かせ正解率をログ出力する。

コマンド形式：

```
tsumego(FAs)
FAs = [FA1, FA2, ... ]
FAi = {局面ファイル名ストリング, 標的群の座標ストリング,
       正解アトム, As}
正解アトム = iki/kou/shini
As = [A1, A2, ... ]
Ai = 正解となる次の1手の座標ストリング
```

4. 捕獲問題実行

予め用意された複数の詰め碁の問題を解かせ正解率をログ出力する。

コマンド形式：

```
hokaku(FAs)
FAs = 詰め碁問題実行コマンドと同様
```

5. 群強度判定問題実行

予め用意された複数の局面内の複数の群の強度について、人間の感覚との差の平均値や分散をログ出力する。

コマンド形式：

```
kyodo(FAs)
FAs = [FA1, FA2, ... ]
FAi = {局面ファイル名ストリング, As}
As = [A1, A2, ... ]
Ai = {標的群の座標ストリング, 人間の感覚による群強度整数}
```

6. 形勢判断問題実行

予め用意された複数の局面の形勢について、人間の感覚との差の平均値や分散をログ出力する。

コマンド形式：

```
keisei(FAs)
FAs = [FA1, FA2, ... ]
FAi = {局面ファイル名ストリング, 人間の感覚による形勢整数}
```

7. チューニング可能パラメータの値変更

チューニング可能なパラメータの値を変更し、コマンド内容をそのままログ出力する。

コマンド形式：

```
{パラメータ ID 関数 f, 変更するパラメータ値}
```

パラメータ ID 関数子 = パラメータ ID を関数子化したもの
(付録「チューニング可能なパラメータ」参照)
変更するパラメータ値 = 変更するパラメータの値の整数もしくは
実数もしくはそれらを要素とするベクタ
(例)
{search(max(hokaku)), {30, 300}}... 捕獲探索の最大探索深さ及び
手数をそれぞれ 30 手, 300 手とする.

8. コメント出力

人力のコメントをそのままログ出力する.

コマンド形式:

`comment(コメントストリング)`

9. 数字付き石表示

碁盤上に数字付きの石を表示する.

コマンド形式:

`display_number_stone(石の色, 石の中の整数, X 座標整数, Y 座標整数)`
石の色 = black/white

10. 数字表示

碁盤上に数字を表示する.

コマンド形式:

`display_number(表示する数字, X 座標整数, Y 座標整数)`

7.3.2 棋譜棋力判定ツール

以下のような手順によって、棋譜の手とコンピュータの考えた手とどちらが良いかを総合判定する.

1. 予め用意された棋譜のファイル名を入力する.
2. 棋譜の手と同じ局面でコンピュータ自身が考えた手が同じ場合は棋譜に従い次々に石を表示していく.
3. 棋譜の手とコンピュータ自身が考えた手が異なる場合 2 つの手を数字付きの石で表示する
石の中の数字は 1 と 2 で, 1 か 2 かは単に座標によって決める. 従って表示からはどちらの手が棋譜の手かあるいはコンピュータの考えた手かは人間には分からない.
4. 2 つの手を表示するとともに, 以下のいずれかを求めるメニューを表示し人間の人力を得る.
 - 1 の手の方が良い (1 の手の方に 3 点加算する)
 - 1 の手の方が少し良い (1 の手の方に 1 点加算する)
 - 同じ (加算なし)

- 2の手の方が少し良い (2の手の方に1点加算する)
- 2の手の方が良い (2の手の方に3点加算する)

5. 棋譜が全て終了したら総合計を出力する。

棋譜の黒番の手とコンピュータが考えた手の採点結果の合計の差と棋譜の白番の手とコンピュータが考えた手の採点結果の合計の差とを表示する。

7.3.3 棋譜編集ツール

以下の手順で棋譜ファイルを作成する。

1. 対局方法の設定

路数、置き石などの情報を入力する。対局システムで対局する時の対局設定と同じ内容である。

2. 碁盤上の棋譜の入力

碁盤上の棋譜を入力していく。カーソルが最初左上に現われる。その位置に棋譜の手を入力する場合その手は何手目に打たれたかの数字を入力すると、数字付きの石が表示される。石の色は自動的に決められる。入力が終わるとカーソルは1つ右に移動する。手を入力する必要のない場合はリターンキーのみ入力する。

3. 盤外の棋譜入力

1度打たれた石がアゲられて再度同じ位置に打たれた場合やパスした時は碁盤上に表現できない。それらの手は未入力となっている。碁盤上の棋譜入力が終了すると、未入力の手が表示されるので、それらの手がどの位置に打たれたかもしくはパスしたかを入力すると、入力した内容が盤外情報として碁盤の下に表示される。

4. セーブ

セーブを選択すると、入力された情報から指定されたファイル名の棋譜ファイルを作成し出力する。

7.3.4 局面編集ツール

以下の手順で局面ファイルを作成する。

1. 石の配置入力

碁盤上に黒 / 白の石を置いていく。また、×マークを盤上の格子点につけることもできる。×マークは表示だけの意味で点の認識には何の影響も与えない。

2. コメントと手番入力

局面に任意の名前を付けることができる。また、その局面が黒番か白番かを入力することができる。

3. セーブ / ロード

作成した局面をファイルにセーブしたり、セーブしたファイルをロードする。

4. コマンド

作成したコマンドファイルにしたがって、自動テストを行う (7.3.1 参照)。

7.3.5 遠隔対局機能

改良の効果を調べる為には実際に対局させてみるのが簡単かつ有効である。特にコンピュータ同士であれば自動的に対局させることができれば人間の手間はほとんど要らない。遠隔対局機能は2つのコンピュータ同士を通信を介して自動対局させる対局システム内の機能である。通信の手段としては、PSIのLANであるPSI-NETとRS232Cとがある。前者は本システムの古いバージョンと新しいバージョン同士の対局やパラメータを一部変えてPSI同士で対局させてみる時などに便利である。後者は通信機能を備えた市販のパソコンソフトとの対局に便利である。以下に2つの通信手段に対するプロトコルとメッセージの内容について述べる。

● PSI-NET

－ プロトコル

SIMPOS が提供する PSI ネットワークシステムの LIA ソフトに従う。

－ 通信メッセージ

送受信されるメッセージはストリング形式をとる。

* 初期送信 (対局設定送信)

1. 置き石数 “0” ~ “9”
2. コミ “0” ~ “9” (コミの整数部分のみ)
3. 路数 “09”/“13”/“19”
4. 遠隔側の手番 “m”(後手)/“s”(先手)

以上の項目を1つのストリングとしてまとめて送信する。

* 初期受信 (対局設定受信)

初期送信と反対の処理を行い、対局設定の情報を受け取る。また、コミに関しては1以上であれば0.5を加える。

* 着手の送信

打着の場合

1. 着手 ID. “move”
2. X座標 “01” ~ “19”
3. Y座標 “01” ~ “19”

その他の場合

1. 着手 ID. “pass”, “matta”, “end”

以上の項目を1つのストリングとしてまとめて送信する。

* 着手の受信

着手の送信と反対の処理を行い、着手の情報を受け取る。

● RS232C

プロトコル

1. 転送速度 2400bps
2. 文字長 7bit
3. パリティ・チェック方式 偶数チェック
4. ストップ・ビットの長さ 1bit

通信メッセージ

現在打着の送受信のみをサポートしている。送受信されるメッセージは6バイトのストリング形式をとる。

* 打着の送信

1. X座標 “01”～“19”(パスの場合は“00”)
2. displacement 空白文字
3. Y座標 “01”～“19”(パスの場合は“00”)
4. 終了ID. 改行文字

・送信手順

送信は1バイトずつ行われ、エコーバックを得た後次の1バイトを送信する。
終了ID.のエコーバック受信によって打着の送信を終了する。

* 打着の受信

打着の送信と反対の処理を行い、打着の情報を受け取る。

7.3.6 消費時間 / メモリー測定機能

“時間測定”を選択し対局を行うと、50手毎と対局の終了時にいろいろな処理に要した消費時間と消費ヒープメモリの量が表示される。

現在測定を行っている処理は、時間もしくはメモリを多く消費する以下の処理である。

● 処理全体(タイマ #0)

1. データ構造更新全体(タイマ #1)

- (a) 連の死活を判定する為の捕獲探索の処理(タイマ #5)
- (b) 結線の連結度を判定する為の連結探索の処理(タイマ #6)
- (c) 群の更新処理(タイマ #7)
- (d) 族の更新処理(タイマ #8)

2. 候補手列挙の処理全体(タイマ #2)

- (a) 候補手の初期化処理(タイマ #9)
- (b) ダメ点候補手処理(タイマ #10)
- (c) 切断 / 連結候補手処理(タイマ #11)
- (d) 分離 / 連絡候補手処理(タイマ #12)
- (e) 模様接点候補手処理(タイマ #13)
- (f) フトコロ拡大 / 縮小候補手処理(タイマ #14)
- (g) 捕獲位置最適化の処理(タイマ #16)

3. 形勢判断(タイマ #3)

4. 着手決定処理全体(タイマ #4)

- (a) 合算、無駄手フィルタ、定石フィルタ(タイマ #17)
- (b) 最終着手決定(タイマ #18)
- (c) 自殺手フィルタ(タイマ #19)

8 並列対局システム

並列対局システムには以下の特徴がある。

- 並列システムと逐次システムの融合

囲碁の対局に必要な処理には、逐次的な処理と並列に行える処理とが存在する。並列に実行可能な処理の大きな部分を並列マシン上で実行した結果と、逐次マシン上で実行した処理の結果により着手を決定する。

- 仕事の粒度で実行順序付けをする負荷分散方式

並列実行させる仕事の粒が大きいと、各プロセッサの仕事の終了時刻に大きなばらつきが生じる。その為、小さな仕事を大きな仕事の後に実行することによって、負荷を均等に分散する方式を採用した。

- 囲碁における並列性の抽出

並列マシンを利用することにより、逐次版システムより多くの知識を処理できる。現在の逐次版システムを強くする為に必要でかつ並列性の高い処理として、仮打ち再評価と探索処理の強化を選び、方式の設計を行った。

- プロセッサ割付け方法

大規模システムにおける協調型問題解決方法の一つの新しい方式として、遊軍方式とそれに基づくプロセッサへの割付け方を設計した。遊軍方式とは、囲碁における処理を、対局に最低限必要な処理と、さらに強くする為に必要だが多量の時間を要する処理とに分け、前者の処理を実行しつつ後者の処理も暇な時間を利用して行う方式である。

8.1 実験システム

並列対局システムの開発の初期段階として、逐次マシン上の対局システムと並列マシン上で一部の処理を行うプログラムとが通信を行って、次の着手を決める対局システムを実現した。

また、仕事の大きさによる動的な実行順序付けを行いプロセッサ間の負荷を均等にする負荷分散方式を試みた。

8.1.1 システム構成－並列システムと逐次システムの融合

囲碁の対局に必要な処理には、逐次的な処理と並列に行える処理とが存在する。そこで、並列に実行可能な大きな部分を並列マシン上のプログラムに実行させ、逐次マシン上で実行した処理の結果とマージさせて、着手を決定する構成とした(図 8.27)。

またこの実験システムは、並列実行する部分とそれまでの逐次版システムとを融合することによって一つのシステムとして機能するようになっているので、システムの並列化の効果を部分的に確認しながら作成することができる。

囲碁の対局を行う為の処理の流れは、図 8.28 の通りである。候補手生成の為には局面認識結果が必要であるというように、逐次的な処理の流れが存在する。逐次的に処理が進む部分に対しては並列マシン上で実行しても効果はない。それに対して、局面認識における個々の処理である、点の認識や連の認識などは盤面上に複数発生するので並列実行の効果が望める。同様に、候補手生成処理においても並列に処理できるものがある。実験システムでは、認識処理から連の生死を判断する捕獲探索と候補手生成の一つである消し候補手生成処理を切りだし、並列マシン上で実行させた。これらの処理は、囲碁における処理の中で重要かつ重い処理である。

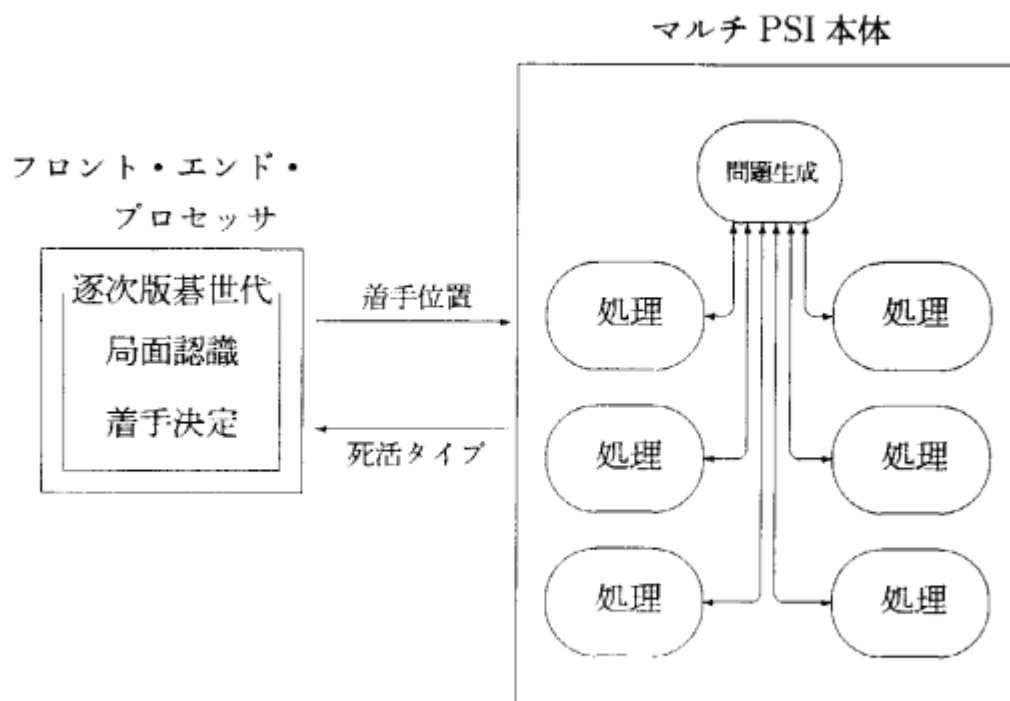


図 8.27：システムの構成

フロント・エンド・プロセッサである PSI には、逐次の囲碁システム「碁世代」をのせ、並列マシンであるマルチ PSI には連の捕獲探索と消し候補手生成のプログラムをのせた。逐次の囲碁システムはマルチ PSI と通信を行いながら、次の着手を決定する。

相手の着手から自分の着手を決めるまでのシステム全体の処理の流れは、次の通りである。

1. フロント・エンド・プロセッサは、相手の着手をマルチ PSI のマスタプロセッサに通信する。
2. マルチ PSI マスタプロセッサは着手によって生じた盤面上の複数の捕獲探索を暇なプロセッサに通信する。

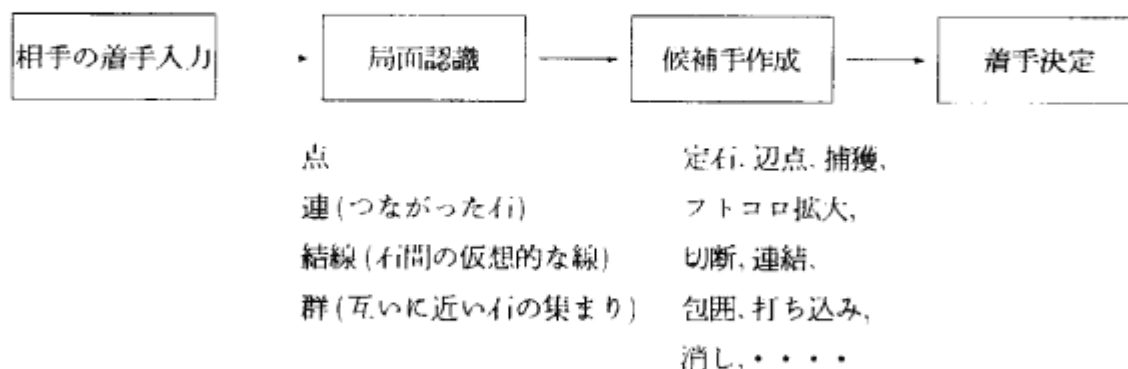


図 8.28：着手決定の流れ

3. 受け取ったプロセッサは捕獲問題を処理し、結果をマスタプロセッサに通信する。
4. マスタプロセッサは結果をフロント・エンド・プロセッサに通信する。
5. マルチ PSI から返ってきた捕獲探索の結果である死活タイプに基づき、ある程度つながっていると思われる石の認識処理などの高度な認識を行う。
6. その認識結果に基づき、候補手の生成を行う。
7. その間マルチ PSI はより多くの時間を要する捕獲探索問題と消し候補手生成処理を実行する。
8. そして、マルチ PSI から得られた捕獲探索の結果と、消し候補手を、フロント・エンド・プロセッサ側が出した候補手と比較検討して、最終的な着手を決定する。

この試作した並列版「碁世代」では、逐次版「碁世代」より約6倍多くの処理を行っているが、逐次版「碁世代」と実行時間はほぼ同じという結果が得られた。

8.1.2 仕事の粒度で実行順序付けする負荷分散方式の実験

並列実行において重要な問題の一つに、いかに均等にプロセッサに負荷を分散させるか、ということがある。並列実行させる処理の粒の大きさが予想できる場合には、粒の大きな仕事を小さな仕事より優先的に実行させる実験を行った。

並列マシンを使う場合、全てのプロセッサが同程度の忙しさであることが望ましい。その為、暇なプロセッサに対して、仕事をさせるのが動的負荷分散方式である。しかし、仕事の大きさが一律な場合はこれで良いが、大きさが一律でない場合は、仕事の大きさを考慮せずに暇なプロセッサに割当てていくと、各プロセッサの処理の終了する時刻が大きく異なる場合がある。あるプロセッサで最後に割当てられた処理の粒が大きく、他のプロセッサが全て暇なのにそのプロセッサが動いている為に全体としては処理が終わらない事が起き得る。そこで、仕事の終わる時間のばらつきを埋める為に、仕事の大きさを考慮して仕事を割付けることが重要であると考えた。大きな処理を初めに投げて、小さな処理は大きな処理を全て投げ終わった後に割付けると、大きな処理の谷間を小さな仕事が埋めていくことになり、全ての実行がほぼ同時期に終わる。この実験では、仕事を大きさによって2種類に分け、大きな仕事は投げる優先順位を高く、小さな仕事の優先順位を低くすることにより、それを実現させた。

大きな仕事としてはつながった石の生死を判断する捕獲探索を、小さな仕事としては消し候補手(敵の模様を侵略する手)の生成を例に取り、実験を行った。

捕獲探索問題は盤面上に複数発生する。ダメ数に関係なく全ての石について、この捕獲探索を行うのが望ましいが、この探索は多量の処理時間を要する為、ダメ数4以下の制限を設けている。逐次版システムではさらに条件がきつく、ダメ数が3以下の連についてのみ処理を行っているが、それでも一手の打済に要する処理時間の約4割を占めている大きな仕事である。

消し候補手は模様の地化を阻止する候補手であり、相手の模様の中で、石と石の間が大きくひらいている所を探し、その中間あたりに打つのが良いとされている。その為、各点が周囲の状況を見渡し「消し」として有効な点かどうかを判断して、消し候補手を生成している。これは、捕獲探索に比べると小さな処理であるが、盤面上の空点全てについて考慮するので、一つのプロセッサのみを使うと大量の時間がかかる。その為、この処理は逐次版システムでは扱っていなかった処理である。

以下に、ある囲碁の対局の中盤における実行の台数効果の測定結果を示す(表 8.32)。プロセッサが16台以上の時、捕獲探索の処理時間と捕獲探索 + 消し候補手の探索時間が同じなのは、消し候補手が捕獲探索の仕事の粒のばらつきを埋め切れなかった為である。

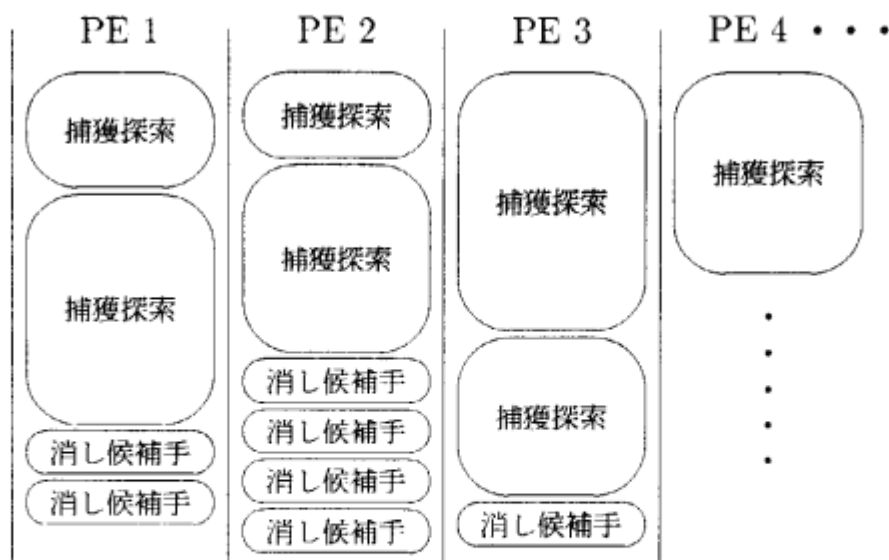


図 8.29：仕事の振り分けの様子

表 8.32：台数効果の測定結果

PE 台数	1	4	8	16	32	64
捕獲探索 (秒)	140	46.1	21.9	13.6	10.0	9.2
台数効果 (倍)	1	3.0	6.3	10.2	13.9	15.1
捕獲探索 + 消し候補手 (秒)	167	53.6	21.9	13.6	10.0	9.2
台数効果 (倍)	1	3.1	6.7	10.2	13.9	15.1

8.2 本格版システム

本格版システムは、前述の実験システムのような部分的な並列システムではなく、全て並列言語で書かれた本格的なシステムである。現在までに、本格版システムに取り入れる処理内容の検討、本格版システムにおけるプロセッサ割付け方法の検討、逐次版システムの KL1 化 (ESP から KL1 への書き換え) を行った。

8.2.1 囲碁における並列性の抽出

現在の逐次版囲碁システムの棋力はまだ目標の 5 級に達していない。それは、強くする為に必要な多くの処理がまだインプリメントされずに残っている為である。残っている理由は、一手にかける時間を、人間との対局で人間が我慢できる範囲に抑える為である。しかし、並列マシンを使うことにより取り込むことのできる処理が増えると期待される。そこで、現在の逐次版システムを見直した結果必要かつ並列性の高い処理の抽出作業を行った。

1. 仮打ち再評価

逐次版の見直し作業のうち、現在のシステムの弱点の一つに、その場面にあまり適していない手を打つことが挙げられる。それは、現在のシステムが放つ手は、部分的知識に基づいている為である。例えば、脱出のつもりで打っても、脱出の手的前方に相手の勢力があった為に、

脱出の意図を満足できなかったりする。それを解消する為に、一手仮打ちをしその部分の局所的な再評価を行うことにより、その手が本当に良い手かどうかを検証することを考えた。これにより、単純な先読みだけではカバーしきれない、候補手の局面との整合性を見ることができる。またこの作業は、部分的に行われるものであり、互いに独立な為並列実行による効果が高いと思われる。

再評価を取り込んだ時の処理の手順は以下の通りである。

- (a) 局面認識において、ターゲットとなる群 (互いに近いところにある石の集まり) や領域と、そのターゲットの意図を求める。
- (b) そのターゲットに関する候補手から評価値の高い打着候補位置を幾つか選びだし、第一次候補手とする。
- (c) 第一次候補手のうち、再評価が必要な候補手については仮打ちする。
- (d) 仮打ちした手の近傍のデータを再更新する。
- (e) 更新した結果が初めの意図を満足しているかを検証する。
- (f) 比較した結果により、その候補手の評価値を変更する。検証した結果、不適当となった候補手の評価値は低くなり、適当と認められたものは評価値が高くなる。
- (g) 再評価により補正された候補手と、再評価の対象とならなかった第一次候補手から、着手位置ごとの評価値の合計を計算し直し、評価値の高い点を着手とする。

ターゲットは、形勢により求める。これは、形勢によって戦略を変える為であり、それにより劣勢の時も常に勝ちの可能性を残し、優勢の時は最低でも勝ちに導くことができるようにする為である。

- (a) 局相が大きく不利？

有利ならば、ターゲットを特には設けない。(全てがターゲットとも言える。)
不利ならば、次へ。

- (b) 群の楽観評価 (自分の中立群を生き、敵の中立群を死と見なす) をし、それでも不利？

有利ならば、敵の中立群のうち最大の影響度のものをターゲットとする。
不利ならば、次へ。

- (c) 領域の楽観評価 (味方の模様地を自分の確定地、敵の模様地を中立領域と見なす) をし、それでも不利？

有利ならば、敵の模様地と味方の模様地の中で最大のものをターゲットとする。
不利ならば、敵の模様地の中で最大のものをターゲットとする。または投了。

求めたターゲットに対して意図を求める。石の意図は、その群の状態や領域の状態からほぼ一意に決まる。

- 中立状態の群

眼作り / 眼潰し、脱出 / 包囲、攻め合い

- 模様

模様の地化、模様の拡大 / 縮小、打ち込み

- 安全な群

地の拡大 / 縮小, 模様の拡大 / 模様の拡大阻止, 地作り / 地作り阻止

意図を満足させる候補手のうち、再評価が必要なものとそうでないものがある。以下に、再評価を必要とする意図を満足させる候補手を挙げる (表 8.33)。

表 8.33 : 候補手毎の再評価の基準

候補手	意図	達成度の確認方法
フトコロ拡大	地の拡大, 地作り	一次認識における中地との比較
フトコロ縮小	地の縮小, 地作り阻止	一次認識における中地との比較
群の眼形	眼作り, 眼潰し	完全生群になった, 完全死群になった
地模様	模様の地化	模様と地の計算
包囲 (脱出)	包囲, 脱出	包囲度
群の分離	脱出抑制	他の群との距離を計る
群の連結	脱出	他の群との距離を計る
辺点	地作り阻止	はさまれた群とはさむ群の強度
ダメ点	模様の拡大	模様の大きさ
模様の接点	模様の拡大	模様の大きさ

なお、以下の点が課題として残っている。

- 評価値補正の詳細設計
- 一手読みに関する問題点

一手後の局面では、安全な群 → 中立状態の群、中立状態の群 → 死んだ群、死んだ群 → 中立状態の群、中立状態の群 → 安全な群、に変わる境で形勢が大きく変化する。しかし、二手打てば、安全 → 中立 → 安全、のように元の状態に戻るならば、その形勢を大きく変える手は、いい手かどうか分からない。場合によっては、攻め合い状態で負けているのに先手で自駄目を詰めるなど悪手を打つことも考えられる。よって、一手後の形勢で判断するのは危険である。

- 形勢判断結果を使っている評価方式についての問題点

現在の形勢判断結果が悪い。(特に群の生死)
対局の序盤では形勢判断にあまり意味がない。形勢はある程度石が増えて来てから使うべき。

2. 探索問題により解決できる問題を増やす

一般にゲームプログラムでは、探索により問題を解決することの効果が大きい。これは、知識から答を導くよりも手間がかかるが、コンピュータはそのような単純な処理が得意である。そこで、探索により解決できる種類の問題を多く取り入れることにした。例えば、一つのターゲットに対しての捕獲探索は不成功に終るが、他のターゲットに対する利きを考慮すれば捕獲が成功するような、多重目的による探索などである。

8.2.2 プロセッサ割付け方法

本格版システムでは、大規模システムにおける協調型問題解決方式の一つとして遊軍方式を提案している。現在までに、その基本方式の検討と設計を行った。以下に、その方式とプロセッサへの割付けについて報告する。

遊軍方式は、処理をその性質から2種類に分け、暇なプロセッサにはより時間のかかる高度な処理を低い優先順位で実行させるものである。処理を、囲碁の対局に際して必要な処理及び最低限の囲碁の強さを保証する処理と、さらに強くなるには必要だが多くの時間を要するので一手にかかる時間内にいい結果が得られなくてもしょうがない処理の2つの処理に分けて考える。前者の仕事を本軍ジョブ、後者の仕事を遊軍ジョブとここでは名付けている。本軍ジョブは全てのプロセッサに分散し実行させるが、遊軍ジョブは暇なプロセッサのみが実行する。本軍ジョブの個数は囲碁を打つという一連の処理の流れの中で増減する。減った時点で、遊軍ジョブが動き出す為に、暇なプロセッサがなくなり台数効果が上がる。そのうえ、本軍ジョブの実行が優先され遊軍ジョブの実行結果を待たずに対局を進められるので、多くの機能を取り込んでいるが一手にかかる処理時間は増えない。

処理の流れは、以下の通りである。

1. 自分の番

各プロセッサは相手着手により盤面の更新を行い、本軍ジョブ、遊軍ジョブをマスタプロセッサに通信。

マスタプロセッサは本軍ジョブ、遊軍ジョブを暇なプロセッサに分配。

各プロセッサは与えられた仕事を処理し、結果をマスタプロセッサに通信する。

マスタプロセッサは、集まった結果を各プロセッサに通信し、盤面の更新を行わせる。

2. 相手考慮中(自分の着手決定後)

相手の打着によりサスペンドしていた遊軍ジョブの実行再開。

3. 相手着手入力

遊軍ジョブのスコープ内に、相手の着手が打たれたなら、そのジョブは抹殺する。

まだ送っていない遊軍ジョブは捨て、既に送った遊軍ジョブには消滅するよう命令を送る。

相手着手を各プロセッサに送る。

なお、各プロセッサには本軍ジョブ実行用の盤と遊軍ジョブ実行用の盤の2つを配置させる。本軍ジョブは、本軍ジョブを実行中でないプロセッサ(遊軍ジョブは実行中でもかまわない)に送り実行させる。遊軍ジョブは、本当に暇なプロセッサに送る。

9 現状の評価と今後の課題

本システムの開発は、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、5級の棋力を目標に開始された。昭和63年度には10級の棋力に達し、その後のシステムの見直し、改良により、1年に1級くらいずつ棋力が向上している。

しかし、現状の棋力はまた10級程度に落ちている。これは、今年度前半の改良の結果、細部の改良ではこれ以上の棋力の向上が望めないことが判明し、今年度後半から大幅なシステムの見直しを開始したが、現時点ではまだ見直し作業の途中段階で知識の中の不整合がいくつかある為である。

また、昭和63年度からシステムの並列化作業を進めており、今年度は本格的な並列対局システムを開発している。並列対局システムでは逐次対局システムと比べて大幅に処理能力が向上すると思われる。

処理能力の向上に伴う新たな知識処理の導入と前述の大幅なシステムの見直しの完了によって、今後棋力が大幅に向上することが望まれる。

A ポテンシャルパタン知識

● ポテンシャルパタンの種類

基準座標に従い、以下の15種類のボタンを用意する。

E	4	E	3	E	2	E	1	E	E	E	E	E	E	E
D	3	D	2	D	1	D	D	D	C	D	D			
	C	2	C	1	C	C	C	C	C	C				
		B	1	B	B	B	B	B	B	B				
				A	A	A	A	A	A	A				
					A	A	A	A	A	A				
						A	A	A	A	A				
							A	A	A	A				
								A	A	A				
									A	A				
										A				

● ポテンシャルボタン

ただし、黒石の点のポテンシャル値は100とする。

				5					
		6	7	7	7	6			
		6	7	10	12	10	7	6	
		7	10	19	35	19	10	7	
		5	7	12	35	●	35	12	7
		7	10	19	35	19	10	7	
		7	10	12	10	7	6		
			7	7	7	6			
				5					

ボタン A

		7	16	26	42	26	16	7	
		12	14	20	24	20	14	12	
		7	16	26	42	26	16	7	
		5	7	12	35	●	35	12	7
		7	10	19	35	19	10	7	
		6	7	10	12	10	7	6	
			6	7	7	6			
				5					

ボタン B

		14	22	26	42	26	16	7	
		22	21	26	24	20	14	12	
		26	26	33	42	26	16	7	
		42	24	42	●	35	12	7	5
		26	20	26	35	19	10	7	
		16	14	16	12	10	7	6	
		7	12	7	7	7	6		
				5					

ボタン B1

		13	17	29	47	29	17	13	
		13	17	29	47	29	17	13	
		5	7	18	42	●	42	18	7
		7	10	19	35	19	10	7	
		6	7	10	12	10	7	6	
			6	7	7	6			
				5					

ボタン C

		23	24	35	47	29	17	13	
		32	27	36	47	29	17	13	
		42	30	49	●	42	18	7	5
		26	20	26	35	19	10	7	
		16	14	16	12	10	7	6	
		7	12	7	7	7	6		
				5					

ボタン C1

		27	36	53	29	17	13		
		36	39	54	29	17	13		
		53	54	●	42	18	7	5	
		29	29	42	19	10	7		
		17	17	18	10	7	6		
		13	13	7	7	6			
				5					

ボタン C2

		14	20	38	70	38	20	14	
		5	13	19	45	●	45	19	5
		7	10	19	35	19	10	7	
		6	7	10	12	10	7	6	
			6	7	7	6			
				5					

ボタン D

		33	30	45	70	38	20	14	
		48	31	52	●	45	19	13	5
		26	20	26	35	19	10	7	
		16	14	16	12	10	7	6	
		7	12	7	7	7	6		
				5					

ボタン D1

394877382014
 5457●4519135
 29294219107
 1717181076
 1313776
 5

ボタン D2

5780382014
 80●4519135
 384519107
 20191076
 141376
 5

ボタン D3

5142254●5422145
 710193519107
 6710121076
 67776
 5

ボタン E

493461●5422145
 2620263519107
 161416121076
 7127776
 5

ボタン E1

5766●5422145
 29294219107
 1717181076
 1313776
 5

ボタン E2

89●5422145
 384519107
 20191076
 141376
 5

ボタン E3

●5422145
 5419107
 221076
 1476
 5

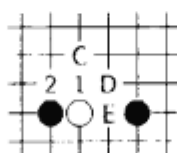
ボタン E4

B 切断 / 連結候補手パターン知識

- … 結線の端点
- … 結線をノゾいている敵石 (ただし、死連の石でないこと)
- × … 黒白共通の候補手
- b … 黒側の候補手
- w … 白側の候補手
- 数字 … 黒白共通の条件つき候補手

[二間 a]

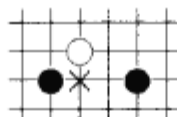
白石を A, A に隣接する黒石を B とする。



評価値 = 15 点

- 1, 2, C, D, E に白石がなく、かつ 1 の点の高さが 2 以上の場合
 - 1 … A のダメ数 ≥ 3 かつ B のダメ数 ≤ 2 の時のみ候補手とする。
 - 2 … 上記以外の場合のみ候補手とする。
- 上記以外の場合
 - 1, 2 とも候補手としない。

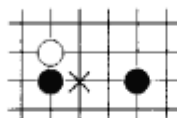
[二間 b]



評価値 = 15 点

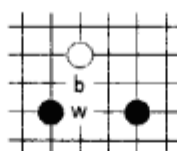
[二間 c]

白石に隣接する黒石を A とする。



- A のダメ数 ≥ 4 の場合
 - 黒側候補手評価値 = 5 点
 - 白側候補手評価値 = 0 点
- 上記以外の場合
 - 黒側候補手評価値 = 15 点
 - 白側候補手評価値 = 0 点

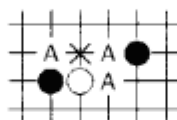
[二間 d]



評価値 = 15 点

[大ゲイマ a]

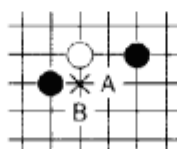
A が全て白石でない時のみ候補手とする。



評価値 = 18 点

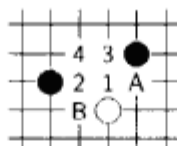
[大ゲイマ b]

A が白石でない時のみ候補手とする。



- B が黒石の場合
評価値 = 5 点
- 上記以外の場合
評価値 = 18 点

[大ゲイマ c]



- A, B が白石でない場合

候補手の位置	1	2	3	4
黒側評価値	15	15	15	15
白側評価値	15	15	18	15

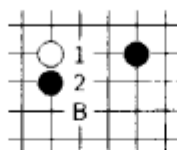
- B が白石で A が白石でない場合
1, 2, 4 が候補手で評価値は 15 点
- A が白石で B が白石でない場合

候補手の位置	1	2	3
黒側評価値	15	15	15
白側評価値	15	15	18

- A, B が白石の場合
3 が候補手で、黒側評価値は 15 点で白側評価値は 18 点

[大ゲイマ d]

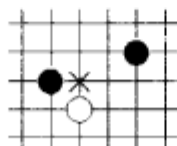
白石に隣接する黒石を A とする。



- 1 … B が黒石の場合のみ候補手とする。
- 2 … B が黒石でない場合のみ候補手とする。

- A のダメ数 ≥ 3 の場合
評価値 = 10 点
- 上記以外の場合
評価値 = 20 点

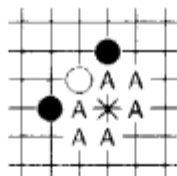
[大ゲイマ e]



評価値 = 15 点

[ハザマ a]

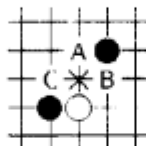
A が全て白石でない場合のみ候補手となる。



評価値 = 10 点

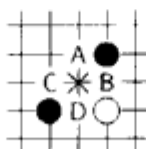
[ハザマ b]

A または B が白石でなく、C が白でない場合のみ候補手となる。



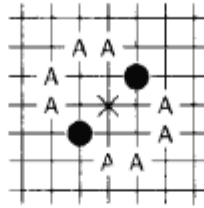
[ハザマ c]

A または B が白石でなく、C または D が白でない場合のみ候補手となる。



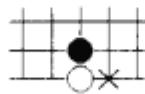
評価値 = 15 点

[ハザマ d]



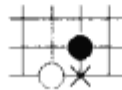
- A に白石がある場合
評価値 = 15 点
- 上記以外の場合
評価値 = 5 点

[二線足 a]



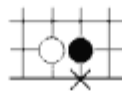
評価値 = 5 点

[二線足 b]



評価値 = 5 点

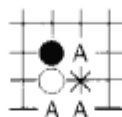
[二線足 c]



評価値 = 5 点

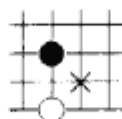
[三線足 a]

A が全て白石でない場合のみ候補手となる。



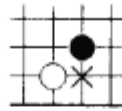
- $\times$ が 2 線以上の場合
評価値 = 15 点
- 上記以外の場合
評価値 = 1 点

[三線足 b]



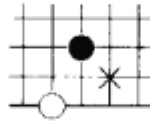
評価値 = 15 点

[三線足 c]



評価値 = 15 点

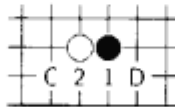
[三線足 d]



評価値 = 15 点

[三線足 e]

白石を A とし、黒石を B とする。

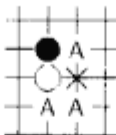


評価値 = 15 点

- C が白石で D が黒石でない場合
1 が候補手となる。
- A のダメ数 ≥ 3 , B のダメ数 = 2 の場合
1 が候補手となる。
- 上記以外の場合
2 が候補手となる。

[四線足 a]

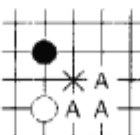
A が全て白石でない場合のみ候補手となる。



- $\times$ が 3 線以上の場合
評価値 = 15 点
- 上記以外の場合
評価値 = 1 点

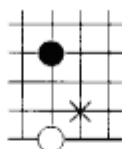
[四線足 b]

A が全て白石でない場合のみ候補手となる。



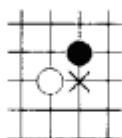
評価値 = 15 点

[四線足 c]



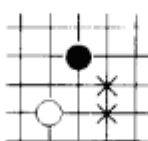
評価値 = 15 点

[四線足 d]



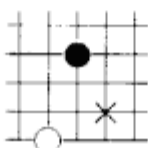
評価値 = 15 点

[四線足 e]



評価値 = 15 点

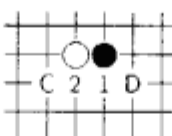
[四線足 f]



評価値 = 15 点

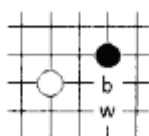
[四線足 g]

白石を A とし、黒石を B とする。



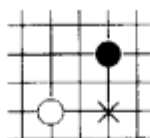
- C が白石で D が黒石でない場合
1 が候補手となる。
- A のダメ数 ≥ 3 , B のダメ数 = 2 の場合
1 が候補手となる。
- 上記以外の場合
2 が候補手となる。

[四線足 h]



評価値 = 20 点

[四線足 i]



評價值 = 20 点

C フトコロ拡大 / 縮小候補手パターン知識

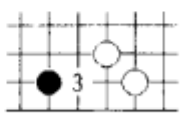
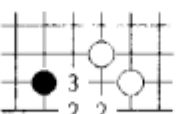
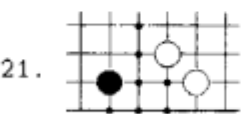
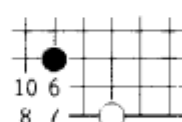
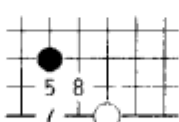
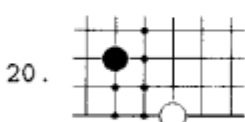
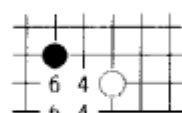
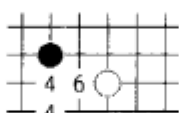
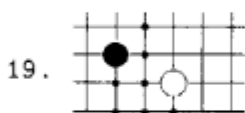
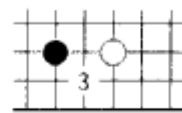
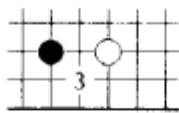
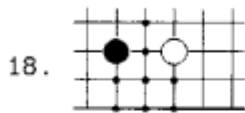
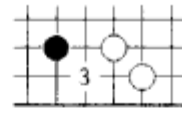
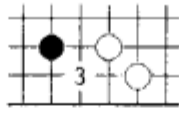
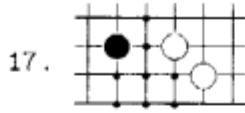
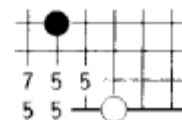
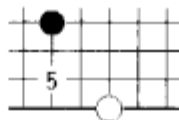
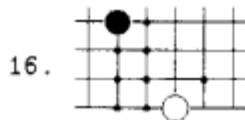
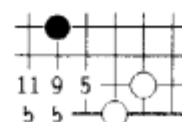
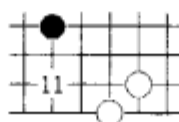
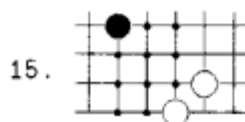
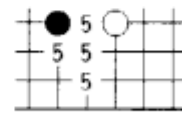
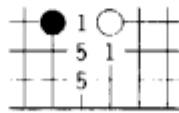
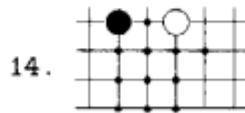
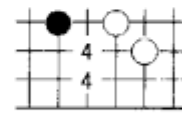
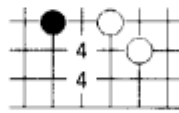
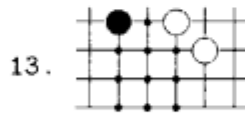
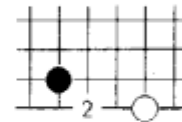
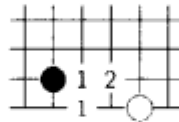
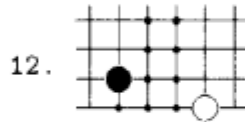
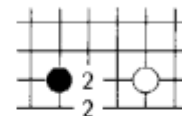
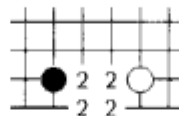
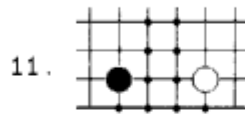
碁世代が知識として持っているフトコロ拡大 / 縮小候補手のパターンについて列挙する。

	パターン	黒からのフトコロ 拡大候補手	白からのフトコロ 縮小候補手
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			

ボタン

黒からのフトコロ
拡大候補手

白からのフトコロ
縮小候補手



	ボタン	黒からのフトコロ 拡大候補手	白からのフトコロ 縮小候補手
22.			
23.			
W のどちらか / 両方が白の時			
24.			
25.			

D チューニング可能なパラメータ

チューニング可能なパラメータの種類とデフォルト値を以下に示す。

- データ構造関連

- 群関連

- * 攻め合い境界値

- デフォルト値 = 10

- * 死群境界値

- デフォルト値 = 7

- * 強度係数

- デフォルト値 = {1.6, 0.8, 6.0, 5.0, 10.0}

- * 弱石境界値

- デフォルト値 = 20

- 族関連

- * 強度係数

- デフォルト値 = {1.6, 0.8, 6.0, 5.0, 10.0}

- * 中地境界値

- デフォルト値 = 7

- * 族境界値

- デフォルト値 = 7

- 局相関連

- 模様等価回数関数

- デフォルト値 = {0.02, 0.10, 0.21, 0.33, 0.45, 0.55, 0.63, 0.77,
0.83, 0.90, 0.97, 1.00}

- 候補手関連

- 辺点候補手関連

- * 基礎点関数

- デフォルト値 = {8.0, 12.0, 16.0, 16.4, 16.8, 17.1, 17.4, 17.6,
17.8, 18.0}

- * 攻め基礎点

- デフォルト値 = 1.0

- ヨセ候補手関連

- * 攻め基礎点

- デフォルト値 = -1.0

- 捕獲候補手関連

- * 基礎点

- デフォルト値 = 10.0

- * 取り切り点

デフォルト値 = 1.0

- ダメ点候補手関連

- * ボーナス点 [1 線]

デフォルト値 = 1.0

- * ボーナス点 [2 線以上]

デフォルト値 = 8.0

- * 基礎点関数

デフォルト値 = {1.0, 5.0, 3.3, 2.5, 2.0, 1.0}

- 切断 / 連結候補手関連

- * キリチガイ基礎点関数

デフォルト値 = {30.0, 23.0, 20.0, 17.0, 15.0}

- 包囲 / 脱出候補手関連

- * 基礎点関数

デフォルト値 = {0.0, 7.1, 10.0, 12.0, 13.6, 11.9, 16.0, 16.9,
17.7, 18.3, 18.9, 19.3, 19.6, 19.8, 19.9, 20.0,
19.9, 19.8, 19.6, 19.3, 18.9, 18.3, 17.7, 16.9,
16.0, 14.9, 13.6, 12.0, 10.9, 9.8, 8.7, 7.7,
7.0, 5.9, 5.2, 4.6, 3.8, 3.3, 2.8, 2.3, 1.9, 1.6,
1.2, 1.0, 0.8, 0.6, 0.4, 0.2, 0.1, 0.0, 0.0}

- 模様接点候補手関連

- * 評価値係数

デフォルト値 = {10.0, 50.0, 5.0}

- 着手決定関連

投了日数

デフォルト値 = 361

- 探索関連

最大手数

- * シチョウ

デフォルト値 = {100, 300}

- * 捕獲

デフォルト値 = {20, 200}

- * 連結

デフォルト値 = {20, 300}

- * 詰め碁

デフォルト値 = {30, 10000}

- * 眼数

デフォルト値 = {20, 10000}

- ゲーム設定関連

- コンピュータ着手意図表示

デフォルト値 = あり / なし (起動システムによって異なる)

- 人間着手意図表示

デフォルト値 = あり / なし (起動システムによって異なる)

- 解析中データ対象指示

デフォルト値 = あり / なし (起動システムによって異なる)

- 分散 PE 台数 (マルチ PSI の FEP として立ち上げられた時のみ有効)

デフォルト値 = マルチ PSI の接続 PE 台数