TR-720

The Specifications of "GO Generation"
— Draft —

by
N. Sanechika, S. Sei, T. Akaosugi, K. Taki,
T. Yoshioka, Y. Murasawa, S. Uchida, H. Oki,
M. Ohshima, M. Ogino, Y. Mizuno & J. Sakamoto

December, 1991

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

The Specifications of "GO Generation" −Draft−

Noriaki Sanechika[1]
Electrotechnical Laboratory of the Agency of Industrial Science and Technology

S. Sei T. Akaosugi K. Taki S. Yoshikawa[2]
T. Yoshioka[2] Y. Murasawa S. Uchida
Institute for New Generation Computer Technology

H. Oki M. Ohshima M. Ogiso Y. Mizuno J. Sakamoto
Future Technology Laboratories Inc.

1991 November

[1]currently at AI Language Research Institute Ltd. (AIR)
[2]currently at NTT Data Communications Systems Co.

1 Preface

This specification is written for the Go playing system GOG (Go Generation) in the Fifth Generation Computer Systems Project.

The purpose of this research is to pose basic search problem solving problems, to resolve ambiguities, to transact exceptions, to develop cooperative problem solving in the AI field, and to study the parallel system of large information processing.

GOG has been developed as part of the intermediate stage of the Fifth Generation Computer Systems Project. A working group, the AI scholars' organization, has been established as an adviser, and this research has been conducted jointly with it and the Electrotechnical Laboratory of the Agency of Industrial Science and Technology at MITI. We have been trying to simulate the human player with some themes of AI. In February 1987, a prototype that could finish the game was developed. Since 1987, we have make a better version of GOG and have tried making progress with the game skill.

In 1988, we started to program a parallel processing GOG. First, we started with a system for the ending problem. In 1989, we began to program a real parallel system. We produced an experimental system which unifies a sequential playing system and a parallel playing system. As a result, we decided on a policy of conversion to a parallel processing system. Since 1990, we developed a full parallel processing GO playing system. Also, we have contined to polish and improve the playing skill of the system.

2 Rules of "GO"

It is difficult to strictly difine the rules of GO, especially since the rules differ between countries. The principle, however, is as follows.

1. Make A Move In Turn

   The two players alternately place their respective black or white stones upon board intersections on a 19×19 grid. A player may pass his turn without placing a stone.

2. Remove

   A stone or a solidly connected group of stones of one color is captured and removed from the board when all the intersections directly adjacent to it are occupied by the enemy. In examples, in Fig.2.1, when Black places a stone at ×, White stones will be removed. The right figure shows the result.

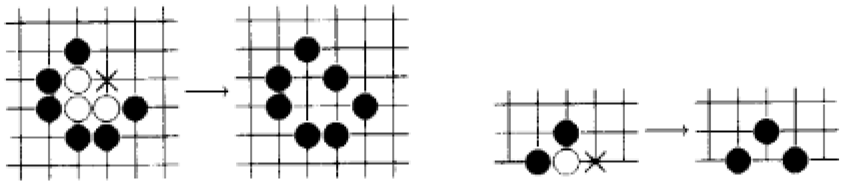
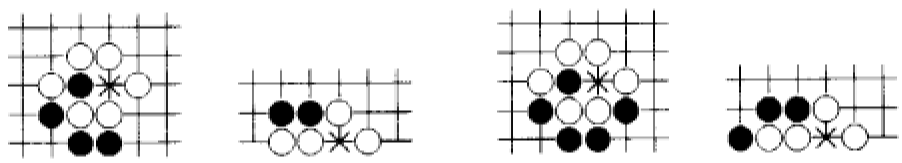


Fig. 2.1 : Example: Removing Stones

3. No-Self-Capturing Move Regulation

   A player cannot place a stone where it will be removed immediately, except when enemy stones are captured also. For example, in Fig.2.2, Black may not place a stone at × in the two left examples. However, Black may place a stone at × in the right two examples.



Example: Self-Capturing Move   Example: Not Self-Capturing Move

Fig. 2.2 : Example: No-Self-Capturing Regulation

4. No-Immediate-Recapture in Ko

   Any move which will re-create the board as it was two moves previously is illegal. For example in Fig.2.3, immediately Black places a stone at × in figure other left, White may not place a stone at × in the right figure.
   This rule is called "Ko" and prevents the game from becoming stalemated.

5. The End of The Game

   The game is over when both players pass their turn. The player with the most territory is the winner.

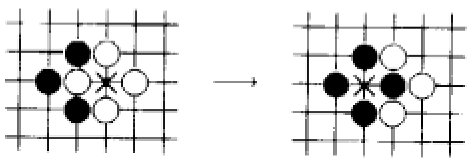


Fig. 2.3 : Example: The Ko Rule

3 System Structure

This system is provided with the following system structures.

- Playing System
  - Game Mode
    - * Game Set Up
    - * Recognition of Position
      Revision of Data Structure
      Local Area Search
    - * Decision Making for Next Move
      Enumeration of Candidates
      Final Decision for Next Move
        Candidate Filter
        Situation Judgement
  - Analysis Mode
- Knowledge Editor
- Grading Tools
- Parallel Playing System

3.1 Playing System

This is the actual GO playing program. The program has a Game Mode, which recognizes the current game board situation and decides the next movement from that, and an Analysis Mode, which examines the results of recognition.

3.1.1 Game Mode

1. Game Set Up

   Before starting the game, the following set up options can be changed.

   - Player

     Select one of the following for each player.

     - human
     - GOG
     - RS232C
     - PSI NET

     By changing this option, human vs computer, human vs human, and computer vs computer games can be played. If RS232C or PSI-NET is selected, the game can be played against other computers with those communication devices.

   - Handicap

     Handicaps from no handicap to a 9 stone handicap can be selected. When no handicap is selected, Komi will be 5.5 stones.

- Sound On/Off

  Provides a sound effect to the indicate placing of a stone.

- Indication of Game Status

  The computer's recognition of the development of a game can be displayed.

2. Recognition of Position

   The program recognizes the positioning of the game pieces as they change with moves. The recognition made through abstract objects on the game board. The local area search, such as Capture and Linkage, will be called, as need arises.

3. Decision Making for Next Move

   The candidates for the next move are created by using information of the candidate on the recognaized positioning. All candidates are evaluated once, but in some cases, the candidates are reevaluated by the candidate filter. Also as the game develops and changes, the next move is decided according to the process of the game. Generally, though, the candidate with the highest evaluation basically becomes the next move.

4. Incremental Revision and Overall Revision

   The transaction times for the recognition of position and the following are saved by applying incremental revision, which is impremented depend on the changes made by the last move. There are more transactions which the incremental revision can be applied to. This isn't applied to transactions which don't require long transaction times or are difficult to encode by applying this revision. When a stone is enclosed by the enemy and removed, or a player asks to undo, overall revision will be implemented.

   - Revision of Point Data Structure

     All attributes of the point where a stone is placed on and attributes of adjacent points, such as "DAME", will be revised.

   - Revision of Potential

     According to the potential pattern knowledge (see APPENDIX), the potential attribute of points which are influenced by the move will be revised.

   - Revision of String Data Structure

     The new string which is presented by the move will be created and the string which is fused into the new string will be deleted. Also, attributes of the string, such as the enemy string's "DAME", which is adjacent to a point where a stone is placed, will be revised.

   - Revision of Life And Death Attribute of String Data Structure

     The life and death attribute of the string is revised by the capture search. The string within the knight move area from the move and the enemy string which is adjacent to those will be revised.

   - Revision of Linkage Data Structure

     The new linkage which is presented by the move will be created.

- Revision of Strength of Linkage Data Structure

  The strength of the linkage is presented by the connection search and the connection/cut pattern knowledge (see APPENDIX).

- Revision of JOSEKI Candidate

  JOSEKI candidates are revised when the move made at the corner and this corner is still in JOSEKI positioning.

- Revision of DAME Candidate

  The DAME of the string within the knight move area from the move will be revised.

- Revision of FUTOKORO Expand/Reduce Candidate

  When the move is made at lower than the 4th line, the candidate which is within 2 blocks from this move and is nominated by the pattern matching will be deleted and the pattern matching will be implemented.

- Revision of Sphere's Contact Point Move Candidate

  The point for which the value of the potential is changed by the move and its adjacent points will be revised.

3.1.2 Analysis Mode

When the system is waiting for any input from the player in Game Mode or is at the Board Situation Editor, the Analysis Mode can be executed. In Analysis Mode, the contents of all data structures which were revised by the recognition of position and all candidates which were nominated by the decision for the movement and the development of the game can be examined. Also all local area search routine can be executed and the process of each routine and its result can be examined.

3.2 Knowledge Editor

Part of the knowledge for this system is applied as a data base. This data base will be used any time the need arises. The knowledge editor is a tool for inputting and managing the data base. This editor is composed of a potential JOSEKI and TESUJI editor.

3.3 Evaluation Tool

The trial and error method is essential for the improvement of a program, such as the GO playing system. This evaluation tool is a set of tools which evaluate the improved part of the system as accurately as possible. The tool is composed of an auto-test, game record rating tool, remote playing function.

The auto-test function has prepared problems for the recognition of position, local area search, decision for movement, and recognition of game status. This tool evaluates the improvements in the system for those problems. The game record rating tool shows the changes of moves from before improvement and after it. Remote playing function is used for playing with the system before improvement with different options or with other GO playing programs. It evaluates the system's total skill and strength in GO.

3.4 Parallel Playing System

The system is being converted to a parallel one running on Multi-PSI. The parallel system will be able to handle more information than the original sequential one. The current version of the parallel system has only the same level of functions the sequentyial version has and some parts of the parallel system's own process, however, the new transactions will be implemented in the future.

4 Data Structure

Dia. 4.1 : Objects of Data Structures

Object	Definition	Function	Main attribute
Point	Board configuration	Stone's location	Color,contact point, potential, candidates
String	A unit which is alive or dead simultaneously	Unit of capture	Number of stones, DAME points, assigns life and death
Linkage	Supposed line between two points of the same color or between a stone and the board edge	Border of territory unit of connection	Type of linkage, type of connection
Group	A unit of the same color whose stones are strongly connected	Unit of life and death	Number of stones, territory size,vitality, rate of siege
Family	A unit of the same color whose potential has more than specified value	Unit of territory	Number of stones, length,territory size, strength,importance

4.1 Point Data Structure

This data structure is used to the recognize the position of stones.
It is composed of the point objects which correspond to entire board grid intersections.

4.1.1 Attributes of Point Object

1. Color

   The color of a point(Black/White/Empty).

2. Coordinates

   The ordinate and abscissa of Point. The top-leftmost point is represented as (1,1).

3. Basic Coordinates and Conversion Code to Basic Coordinates

   The basic coordinates (X,Y) and the code which specifies how to convert to the basic coordinates (see the following details).

4. Height

   The distance from the edge of the board (the edge is 1).

   $$\text{Height} = \min(\text{Basic Coordinates X}, \text{Basic Coordinates Y})$$

5. Adjacent Point

   The collection of points adjacent to the point.

6. DAME

   The collection of points which are adjacent to the point and have no stones on them yet.

7. Total of DAME

   The total number of stones belonging to DAME.

8. Potential Value

   A stone (Black/White) is considered to give an effect which is inversely proportional to the distance on nearby points. The potential value is attained by adding up all effects from nearby stones (Max. 100). There are two different kind. One is just the sum of all, another is done with the consideration of the dead stones group.

9. String

   If the point is on the string, this attribute has its string's "Ren" object.

10. Linkage

    If the point is on the end of the linkage, this attribute is linked to "Kessen".

11. Block Linkage

    If the point is on the block point of the linkage, this attribute is linked to "Kessen". This attribute holds the Black and White linkage objects separately. If there are more than two same colored linkages, the strongest linkage takes precedence.

12. Group

    If the point belongs to the group, this attribute has its group's "Gun" object.

13. Family

    If the point belongs to the family, this attribute has its family's "Zoku" object.

4.1.2 Basic Coordinates/Conversion Code to Basic Coordinates

Divide the game board into eight parts as Fig. 4.4 and settle 0s as bases.

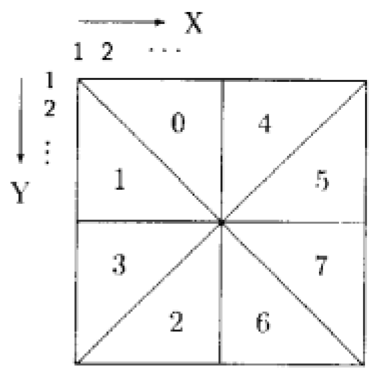


Fig. 4.4 : Conversion Code to Basic Coordinates

Basic Coordinates are the coordinates of the symmetrical point of each point on the game board.

In Fig. 4.4, the number within each of eight divided parts specifies the conversion code for its point. Dia. 4.2 explains the relation between the basic coordinates and the conversion code when each point on the board takes (X,Y).

Dia. 4.2 : Conversion to Basic Coordinates

Conversion Code	Basic Coordinates X	Basic Coordinates Y
0	X	Y
1	Y	X
2	X	Board Size+1−Y
3	Board Size+1−Y	X
4	Board Size+1−X	Y
5	Y	Board Size+1−X
6	Board Size+1−X	Board Size+1−Y
7	Board Size+1−Y	Board Size+1−X

4.1.3 Potential Value, Potential Function

The potential function represents the strength of Black/White at each point on the board. We define the stone of the board as giving an effect which is inversely proportional to the distance on nearby points. To adjust it to human sense, we prepare "Potential Pattern Knowledge"(see APPENDIX) for the influence of the corner and board edge. This data base is easily fixed by the potential pattern editor.

The placed Black stone's potential value is added to the adjacent points' as the positive and the White's is added as the negative. These totals become each point's potential value.

4.2 String Data Structure

This data structure is used for the recognition of capture (ATARI and NUKI).

It is composed of the string objects which are collections of the same colored stones adjacent to each other.

4.2.1 Attributes of String Object

1. Color

   The string color(Black/White).

2. Point

   The collection of point objects belonging to String.

3. Total of Stone

   The total number of stones belonging to String.

4. DAME

   The collection of points which are adjacent to the string and do not contain a stone yet. This is the same as the entire DAME of points which belong to the string.

5. Life and Death, Escape Point, Capture Point

   Following Dia. 4.3, Life and Death is determined by applying the capture search in Black/White turn to the string which has 3 or less DAMEs. If the string has 4 or more DAMEs, its Life and Death is determined as alive.

Dia. 4.3 : Life and Death of String

Result of Capture Search for string's color	Result of Capture Search for string's opposite color	
	Success	Failure or Unsettle
Success or Unsettled	Neutral	Alive
Failure	Dead	Alive

   If the capture search for the string's color succeed or unsettled and returned its move, the move becomes the Escape Point. If the capture search for the string's opposite color succeed and returned its move, the move becomes the Capture Point.

6. TANE ISHI/ NON-TANE ISHI

   When strings are crosscutting each other like Fig. 4.5 and all 4 stones' Life and Death of its strings are alive or nutral, these strings are determined as TANE ISHI.
   The strings except TANE ISHI are determined as NON-TANE ISHI.

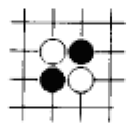


Fig. 4.5 : Crosscut

4.3 Point and String Data Structures for Search

The point and string data structures for search are improved on the point and string data structures for search process. By these data structures, a lot of time for revising point and string object and memory will be saved.

It is better to use the same data stractures for ordinaly use and for search but currentaly these are separated. For the move in the game, both type of data stractures are revised. For the imaginary move in the search process, only data structures for search are used.

4.3.1 Attributes of Point and String Object for Search

The point object for search and the string object for search are united and managed as the point and string object for search. The point object of the last placed stone within the string includes attributes of the string. And point objects of other stones within the string includes attributes of the string when the stones are the last placed stone. Such a history of stone is saved in attributes called the parent point and the child point.

1. Color, Coordinates, Basic Coordinates, Conversion Code to Basic Coordinates, Height, Adjacent Point, DAME, Number of DAME
   same as the point object.

2. Parent Point

   The parent point of the string.

3. Child Point

   The collection of child point.

4. Total of Child Point

   The total of child stones.

5. Total of String's Stone

   The total of stones belong to the string.

6. Total of String's DAME

   The total of adjacent liberties to the string's stones.

7. String's DAME

   If the total of string's DAME is 3 or less, this attribute has the collection of adjacent liberties to the string's stones.
   The reason why the collection will be saved if the total is 3 or less is followings;

   - The management of variable length data costs a lot of time and memory.
   - In the most of cases when it is necessary the position of the string's DAME, the total of string's DAME is 3 or less.

   However, only sometimes, the necessity of the total of string's DAME arises. For this, the method which calculates it from the attribute for the child point is prepared.

4.3.2 Revision Method

Using Fig. 4.6 as an example, the followings explain how to revise the point and string object for search. The left figure in Fig. 4.6 shows a White string whith 6 stones. The each number of stones explains when the stone is placed.

Also the right figure in Fig. 4.6 shows relations among each stones' parent stone. If a move is made and its adjacent stone is the same color as the move, each stone which belongs to the string and is the latest move among its stones becomes its child and the move becomes a

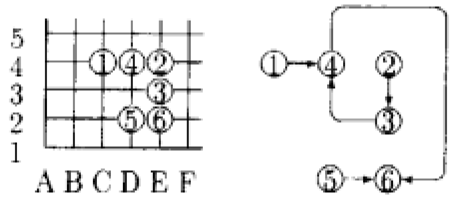


Fig. 4.6 : Revising Method of Point and String Object for Search

parent of children. For example, if a move is made at 6, 4 which is the latest move in a string (1~4) and 5 becomes its children.

Followings are details of the revision for the point and string object for search when a move is made.

- Set the color of move.
- Reduce 1 from all total of adjacent stones' DAME and remove the move frome DAME.
- If the opponent stone adjacent to the move exists, reduce 1 from the total of string's DAME which is holded the latest move of the string with the adjacent stone. And if the total is 3 or less, remove the move from the string's DAME. (or re-calculate it.)
- If the friend stone adjacent to the move exists, define the move as the parent stone of the latest move within the string which includes the adjacent stone and add the adjacent stone to children stones of the move.
- From attributes of the move's children stones, re-calculate attributes (the total of string's stone, total of string's DAME and string's DAME) and re-set these.

Also followings are detailes of the revision for removing a current move when the process, the backtrack in search or removing stones, occurs.

- Add 1 to all total of adjacent stones' DAME and add the move to DAME.
- If the opponent stone adjacent to the move exists, add 1 to the total of string's DAME which is holded the latest move of the string with the adjacent stone. And if the total is 3 or less, add the move to the string's DAME.
- If the friend stone adjacent to the move exists, reset the parent stone of the latest move within the string which includes the adjacent stone and reset children stones of the move.
- set empty as the move's color

4.3.3 Evaluation of Performance

We tested the cost of time and memory for each using the ordinaly point and string datat structures and using point and string data structure for search. (Fig. 4.4) The measurement is the average of about 50,000 moves for the capture search in a game (a computer against a conputer, 19×19 board).

Dia. 4.4 : Comparison between Ordinaly Point and Data Structures and Point and String Structure for Search

Data Structure	CPU Time/Move			Heap Memory/Move		
	Move	Remove	Total	Move	Remove	Total
Ordinaly	3.38 msec	3.70 msec	7.08 msec	91 word	89 word	180 word
For Search	1.39 msec	0.56 msec	1.95 msec	0 word	0 word	0 word
Rate of Improvement	× 2.4	× 6.6	× 3.6	∞	∞	∞

4.4 Linkage Data Structure

This data structure is used for the recognition of relations between stones which are the same color and might be connected, and a stone and the edge of board.

It is composed of existing linkage objects, which are defined as the linkage pattern in Fig. 4.7, on the game board. Except, in Fig. 4.7, the friend stone must be placed at ×. Also the heights of the perpendicular linkage to the edge of board from 2nd, 3rd and 4th line (perpendicular 2, 3, 4) must be more than 2, 3 and 4.

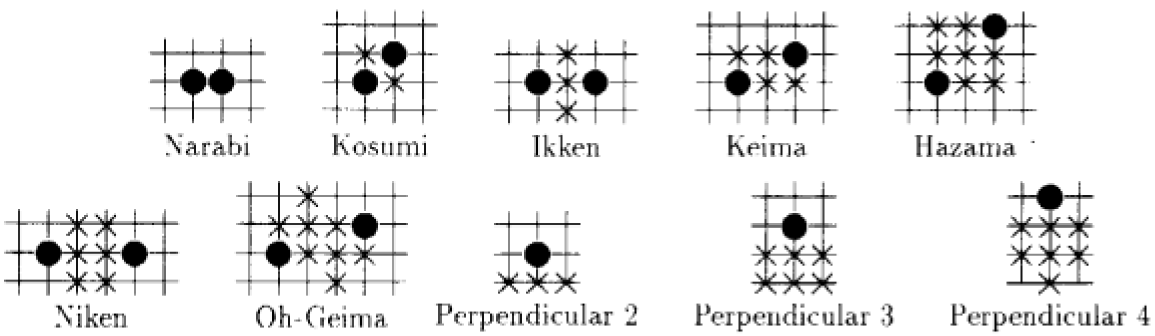


Fig. 4.7 : Linkage Pattern

4.4.1 Attributes of Linkage Object

1. Color

   The color of linkage(Black/White).

2. Edge Points

   The both points which are the edge of linkage.

3. Variety of Linkage

   The varieties of linkage. (Narabi/Kosumi/Ikken/Keima/Niken/Oh-Geima/Hazama/······)

4. Strength of Connection

   The rate of possibility that the both edge of linkage will connect. (Strong Linkage/Weak Linkage/Unconnectable Linkage)

5. Group

If the linkage belongs to the group, this attribute has its group object.

4.4.2 Strength of Connection

The linkage's strength of connection is decided as follows;

- Narabi

  Always the strong linkage.

- Kosumi and Crosscut

  From the life-and-death of crosscutting 2 enemy stones' strings and Fig. 4.5.

Dia. 4.5 : Crosscutting Kosumi Linkage's Strength of Connection

Life-and-Death of Enemy Stone's String	Life-and-Death of Another Enemy Stone's String		
	Alive	Neutral	Dead
Alive	Strong Linkage	Strong Linkage	Strong Linkage
Neutral	Strong Linkage	Strong Linkage	Weak Linkage
Dead	Strong Linkage	Weak Linkage	Unconnectable Linkage

- Other Kosumi, Ikken, Keima

  From the result of connection search in Black/White turn and Fig. 4.6.

Dia. 4.6 : Kosumi, Ikken and Keima Linkage's Strength of Connection

Turn for Linkage's Friend Side	Turn for Linkage's Enemy Side	
	Success	Failue
Success	Weak Linkage	Strong Linkage
Failure	Unconnectable Linkage	Unconnectable Linkage

The search for the linkage's friend side will fail, if the result of search is "break through". In other cases, it will succeed.

- Niken, Oh-Geima, Hazama, Perpendicular 2, 3, and 4

  If the linkage matches with the Cut/Connect Candidate Pattern and there is vacant position among its candidate positions, the linkage is determined as a weak linkage. And if every these candidate positions are occupied, it is determined as a unconnectable linkage. If it dosen't match this pattern, it is determined as a strong linkage.

4.5 Group Data Structure

This data structure is used for the recognition of the group which is formed by stones nearly connected.

It is composed of the group object which is formed by the same colored stones of strong connected linkage.

The strong connected linkage is as follows;

- Strong linkages of Narabi, Kosumi, Ikken, Kima, and Perpendicular 2, 3, and 4
- Strong linkages of Niken(but the height of both edge are 4 or less) and Oh-Geima
- Weak linkage of perpendicular 2 in Cut/Connect candidate pattern [Perpendicular 2 (C)] (see APPENDIX "Cut/Connect candidate pattern")

The rivison of the group data structure will be done in 2 phases. The group will fuse the adjacent enemy group, which was determined as the dead group in the first phase, in the second phase.

4.5.1 Attributes of Group Object

The group object has following attributes.

1. Color

   The color of group (Black/White).

2. Territory

   The collection of points which are enclosed with the group's strong connected linkages (see the details that follow).

3. Size of Territory

   The total of points belong to the territory.

4. Stone

   The collection of friend stones within the territory.

5. Total of Stone

   The total of friend colored stones.

6. Size

   This is a total of points whitin the territory and outer 2 paths from territory. And it is used for the outline of the opponent's gains by the death of group.

7. Neutral Area

   The collection of the area which is adjacent to the territory and will become the territory after one more move is made (see the details that follow).

8. Interior

   The total of the territory's interriors (see the details that follow).

9. Number of Eyes

   The number of eyes within the territory (see the details that follow).

10. Nakate

   If a player plays this point first, he gets two real eyes and lives. If his opponent plays the point, he will be reduced to only one real eye (see the details that follow).

11. Rate of Siege

   The rate of the opponent's siege (see the details that follow).

12. Group's DAME

   The collection of group's DAMEs for the calculation of the siege's rate (see the details that follow).

13. Surrounding Linkage

   The collection of the opponent's linkages which surrounds the group (see the details that follow).

14. Surrounding Point

   The collection of points which surround the group (see the details that follow).

15. Surrounding Group

   The collection of the opponent's groups which surrounds the group (see the details that follow).

16. Strength

   The strength of the group (see the details that follow).

17. Importance

   The merit of a move which protects the group and the effect of the group's life-and-death (see the details that follow).

18. Moves

   The total of DAME of each stone of group. It is used to detect the capturing race.

19. Classification

   The classification of the group is used to detect the life-and-death/capturing race probrem (see the details that follow).

4.5.2 Territory, Neutral Area, Interior

The definition of territory is the collection of the point which is surrounded by group's stones and strong connected linkages' block point. The block point expresses the imaginary line between both linkage's edges. Fig 4.8 is its definition. In the figure, each large black circle expresses a edge of linkage, and small one dose a block point.

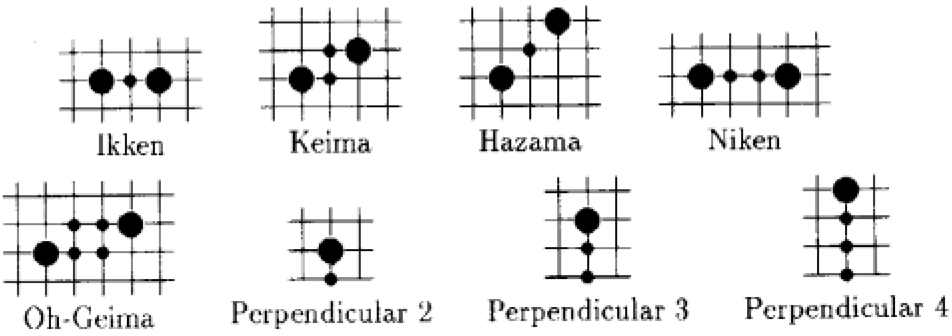


Fig. 4.8 : Block Point of Linkage

The definition of neutral area is the collection of the point which is within the territory of the group and is surrounded by block points of linkages which are not strong connected linkages (except points within territoy of the group and block points of linkages). In another words, the neutral area is the closed area which will become a territory, if one more move which will protects the group is made. Fig. 4.9 is an Example for the closed area. Each large black circle expresses a stone of the group, and the collection of the large and small black circle expresses the territory of the group. × s express the neutral area.

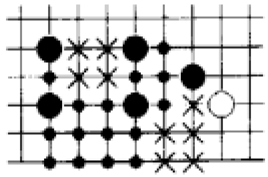


Fig. 4.9 : Territory of Group and Neutral Area

Odd orders, in reducing order of point size, which is within the neutral area of the group are defined as the first move's neutral area and even orders are defined as the second move's neutral area. The interior is the total of points within the territory (or the neutral area). There are also the first move's interior and the second move's interior. These points are within the closed area and only adjacent to points within the closed area. The followings are definitons of the first move's interior and the second move's interior.

The First Move's Interior

= Total of points within the territory and the first move's neutral area

The Second Move's Interior

= Total of points within the territory and the Second move's neutral area

The first move's interior expresses the interior in the group's turn. And the second move's interior dose the interior in the group's opponent's turn. When you see "interior" without any words, it means the second move's interior.

4.5.3 Number of Eyes, Nakate

The number of eyes means how many eyes there are within the group. The Nakate is the move which increases and reduces eyes efficiently. The process which searches the number of eyes costs a lot. So currently it will be implemented only for the group where the siege's rate is 3 or less and interior is 7 or less. The following shows how to compute the number of eyes and Nakate.

1. Search for eyes

   The closed area which is surrounded stones of the group and is within the territory of the group is declared as the eye. If there is 2 eyes, the number of eyes is 2. If there is none, it is 0. Either case, there is no Nakate. If ther is only one eye, followings are proceeded.

2. Determine the degree code for eyes

   The degree code of eye corresponds to the degree of each point within a eye. The degree expresses the total of points which are adjacent to each point within a eye.

3. Calculate total of entire vital points.

   Calculate the total of entire vital point with the degree of eye and Fig. 4.7.

Dia. 4.7 : Vital Points of Eye

Size of Eye	Degree Code	Vital Points
1	(0)	0
2	(1, 1)	0
3	(2, 1, 1)	1
4	(2, 2, 1, 1)	2
	(3, 1, 1, 1)	1
	(2, 2, 2, 2)	0
5	(2, 2, 2, 1, 1)	3
	(3, 2, 1, 1, 1)	2
	(3, 2, 2, 2, 1)	1

Size of Eye	Degree Code	Vital Point
5	(4, 1, 1, 1, 1)	1
6	(2, 2, 2, 2, 1, 1)	4
	(3, 2, 2, 1, 1, 1)	3
	(3, 3, 1, 1, 1, 1)	2
	(3, 3, 2, 2, 1, 1)	2
	(3, 2, 2, 2, 2, 1)	2
	(3, 3, 2, 2, 2, 2)	2
	(4, 2, 1, 1, 1, 1)	2
	(4, 2, 2, 2, 1, 1)	1

4. Search vital points

   Based on the point which has the largest degree of eye in the eye, A number, a total of vital points by the degree code, of vital points are difined.

5. Calculate Number of Eyes

   From the number of points which are already occupied by the enemy within vital points, the number of eyes are calculated with Fig 4.8.

6. Search Nakate

   If there are 2 eyes or 1eye, there is no Nakate. If there are 1.5 eyes, Nakate is vacant points within vital points.

Dia. 4.8 : Number of Eyes

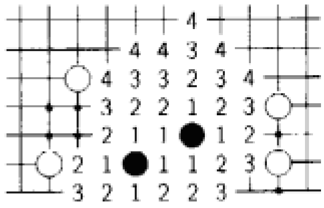
Vital Points	Enemy Stones within Vital Points	Eyes
3 or more	–	2
2	2	1
	1	1.5
	0	2
1	1	1
	0	1.5
0	–	1

4.5.4 Rate of Siege, Group's DAME, Surrounding Linkage , Surrounding Point, Surrounding Group

The rate of siege expresses the extent of enemy's siege which is surrounding the group. The small number of the rate means how strong the siege encloses the group. If the rate is 0, the siege is called a perfect siege. The rate is defined as follows.

DAMEs of each stone of the target group are called 1st group's DAME. DAME whichi is a block outer from Nth(1∼3) DAME is called N-1th DAME.

When the point which is recognized as DAME is a block point, and when it is 4th DAME and its height is 1, these cases are exceptions. Following Fig. 4.10 is a example of 4th or lower DAME.



● ...Stones of Target Group
○...Surrounding Enemy Stones
• ...Enemy's Block Point
n(1∼4)...Nth Group's DAME

Fig. 4.10 : Nth Group's DAME

4th Group's DAME is just called Group's DAME and the rate of siege is determined by this. Group's DAMEs are devided into groups which are connected to each other by Narabi or Kosumi. These groupes are colled the group of Group's DAME. If the group is composed of 3 or more elements, the number of its Group's DAME becomes the rate of siege. In Fig. 4.10, every Group's DAME belongs to A group of Group's DAME, and the rate of siege is 6.

The surrounding linkage is the enemy linkage which is surrounding the target group. It is a linkage which the enemy's block point adjacent to 3rd or lower Group's DAME belongs to. The surrounding point is the enemy or friend stone nearby the target group. It is a collection of stones, which belong to the group except the group adjacent to the 3rd or lower Group's DAME, and the edge of surrounding linkage. The surrounding group is the enemy group nearby the target group. The enemy's surrounding point belongs to this.

4.5.5 Strength

The strength of group expresses the strength of group's entire stones and is defined as follows.

1. Basic Formula

   The basic formula is as simple as follows.

$$\text{Strength} = \underset{\text{(Strength of Interior)}}{\text{Interior} \times 5} + \underset{\text{(Possibility of Escape)}}{\text{Rate of Siege} \times 2.5}$$

   The constants, 5 and 2.5, are decided by the following reasons. We had a person estimated the strength of a group in various board situations and several games. To realize the human's evaluation of the strength of stone, Fig. 4.9 is set up a standard.

   Dia. 4.9 : Strength of Stone and Human Evaluation of It

Sence toward Strength of Stone	Strength
Alive	40 ~ 50
Nearly Alive	30 ~ 40
Alive, but playing a move would be better	20 ~ 30
Neutral (Alive in friend's turn, Dead at enemy's)	10 ~ 20
Nearly Dead	1 ~ 10
Dead	0

   The constants are arranged to minimize the differeces between the strength by the human sence and the formula.

2. Adjustment for Strength of Interior

   For still more accuracy, the following adjustments are done with the strength of interior.

   (a) Adjustment for Interior

   The interior expresses the size of territory within the group and is the number of points within group's territory. (see 4.5.2) If entire boundary points are occupied by stones, it is worth 1 point. But, if there is any vacant point, it is more realistic to reduce some points from the interior. Thus when the point will be counted as the interior is adjacent to an vacant boundary point, it will be counted as 0.5 point for the interior. When it is adjacent to two vacant boundary points, it will be counted as 0.25 point. (and so on)

   (b) Adjustment by Eyes

   In the case which one more move will make an interrior with 2 eyes (see 4.5.3), the interior is considered that it gives larger influence to the group than others. From this reason, if the interior's number of eyes is 1.5, 10 is added to its strength of interior.

(c) Adjustment by FUTOKORO

The interior expresses size of nearly definite territory. But the strength of group becomes much accurate, if the interior is considered the possibility of its future increase. FUTOKORO Expand candidate (see 6.1.8) is the move will enlarge the group's interior. The candidate's valuation is decidecd by its size of increased interior. If FUTOKORO Expand candidate exists in the group, following strength of FUTOKORO is added to the strength of interior.

$$\text{FUTOKORO's Strength} = \min(\text{FUTOKORO Expand Candidate's Valuation}, 10)$$

3. Adjustment for Possibility of Escape

There is the difference in ecapability between an escape and some escapes with the same rate of siege. If there is a escape, an enemy's move might block it. However, if there are some, it is difficult. If there are some group of Group's DAME with 3 or more DAMEs (see 4.5.4), the possibility of escape is decided as follows. If there are some group of Group's DAME,

$$\text{Possibility of Escape} = N \times 10$$

4. Adjustment by Classification

By all attributes of group, the group is classified from a strategical point of view. (see 4.5.7) The strength of group which will be a problem for the classification is computed by the following procedure. After classification procedure, the strength of group will be adjusted by its result as follows.

- Complete-Death Group
  0
- Safe Group with 2 Eyes
  50
- above others
  Maximum strength is 45.
- Dead and Apparent-Death Group
  Minimum strength is 1.
- Escapable and Capturing-Race-Warning Group
  Minimum strength is 10.

4.5.6 Importance

The group has following two items of important.

1. Importance of Interference

If the move attacks (or protects) the stone which is strong enough and will survive, this move is not important. Also interfering with the stone nearly dead has no inmportance. If the strength of stone is moderate, the merit of move, even to attack or protect, increases. We decide to call these merit of interfering with the stone the importance of interference, its value is computed by the formula in Dia. 4.10. This inportance of interference is used for evaluation of various group's fightting candidates.

Strength of Target Group	0	1	2	3	4	5	6	7	8	9	10
Inportance of Interference	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	

Dia. 4.10 : Inportance of Interference Function

2. Importance of TANE ISHI

   TANE ISHI is the stone the life and death of which it has the great influence on enemy stones' life and death nearby. Because the death of TANE ISHI makes all enemy stones which were weak nearby be safe, the evaluation's value for the life and death must take higher. We decide to call the assessment of TANE ISHI's life and death the importance of TANE ISHI, its value is computed by the formula below. Each T_i and S_i expresses the group's rate of siege and its size.

$$\text{Inportance of TANE ISHI} = \frac{1}{15}\sum_i (30 - T_i) \cdot S_i$$

4.5.7 Classification

By all attributes of group, the group is classified from a strategical point of view.

- If entire stones of the group belong to dead strings

  Complete-Death Group

- If the group has 2 or more eyes, its rate of siege is 4, or higher or its interior is 11 or more

  Safe-Group

- If there is reinforcements or it is escapable

  Escapable-Group

The existance of reinforcements is determined, when there is the point which belongs to the frend colored group, the strength of which group is 10 or more, and is the group's surrounding point. The group is determined as escapable, if there are any likages below within the group's surrounding linkages.

- Strong Linkage (Nikken, Oh-Geima, and Perpendicular 4)

Weak Linkages except Perpendicular 2

- If there is FUTOKORO move, the group has an eye or more, or its interrior is 8 or more

  Apparent-Death Group

  The existance of FUTOKORO move is determend, when there is FUTOKORO Expand candidate based on the group's sotnes.

- If the group's interior is 4 or more

  DEAD Group

- If the group is in the capturing race

  Capturing-Race-Warning Group

  The capturing race is when there are any groups satisfies followings.

  - Groups except Safe, Escapable, and Complete-Death Group
  - If both the group and its surrounding group are a string, a following should be satisfied.

    Substantial Moves for Group ≧ Substantial Moves for Surrounding Group

  - If the group or its surrounding group is not a string, followings should be satisfied.

    Substantial Moves for Group ≧ Substantial Moves for Surrounding Group + 3

- above others

  Complete-Death Group

  If the Complete-Death group exists, a new group, it together with its surrounding group, are created and potential of stones belong to the Complete-Death group are reseted.

4.6 Family Data Strucure

This data structure is used for the general idea, such as the recognition of the sphere of influence and the wall.

It is composed of the family object which is the entire collection of points, the potential ,of which it is 7 or higher (if White, its potential is -7 or less) and is adjacent to each other.

4.6.1 Attributes of Family Object

- Color

  The color of family (Black/White).

- Territory

  The collection of points belong to the family.

- Size

  This is a total of points belong to the family's territory.

- Stone

  The collection of friend stones within the territory.

- Total of Stone

  The total of friend colored stones.

- Length of Side

  The total of stones belong to the territory and are on the edge line.

- Interior

  The number of points, belong to the territory and entire Kosumi points of whichit have 7 or higher (if White, -7 or less) as their potential.

- Rate of Siege, Family's DAME, Surrounding Linkage, and Surrounding Point

  same as the group objet's.

- Strength

$$\text{Strength} = \min(1.6\times \text{ Rate of Siege } + 0.8\times \text{ Interior } + \text{ Strength of Definite Territory }, 50)$$

$$\text{Strength of Definite Territory} = \begin{cases} 10 \text{ (If Interior } \leqq 5 \text{)} \\ 10 \times \max(5 - \dfrac{\text{Size}}{\text{Number of Stone}}, 0) \text{ (above others)} \end{cases}$$

- Inportance of Interference

  same as the group object's.

5 Local Area Search

For the method of regnizing every objects in the recognition of position, using the pattern knowledge and getting results by searching are considered. In general, the former saves the cost but is inaccurate. The latter is accurate but wastes the cost. Consequently, a problem is solved by using the pattern knowledge first. Then if the problem boiled down to one can be finished withn specified time, the search is started. For this system , 6 local area searches (Dia. 5.11) are prepaired.

Dia. 5.11 : Variety of Local Area Search

Category	Target	Question	Result
Ladder	String with 2 or less DAMEs	String might be caught in Ladder	Success/Failure
Capture	String with 3 or less DAMEs	String might be caught	Success/Failure
Connection	Linkage except Perpendicular	Linkage might be connected	Connect/Crosscut/ Break Through
Tsumego	Completely surrounded group	Group is alive	Alive/Dead/Ko
Eyes	Territory with 4 or less as size	Territory might have an eye	Success/Failure
Multi Target Capture	Strings including one with 2 or less DAMEs	Any string might be captured	Success/Failure

5.1 Search Algorithm

All local area search in this system are complete searches (read-all search), proceed to search until reaching the phase which is evaluated completely, and the Alpha-Beta method is applied to them.

For the search often face to the sequence of moves can be swaped its order, it memorizes the positioning once searched and its result. If the same positioning appears later, the search uses the results fro not wasting the cost.

5.1.1 Effect of Memorizing Board Positioning on Search

The memorizing board positioning is what memorizes the positioning on the phase appears during search and its evaluation. If the same positioning appears later, the search uses memorized evaluation for saving unnecessary search.

Applying the memorizing board positioning to many GOG's local area searches, a major reduction of the search's cost is found in the search for the end game and eyes. In the search for ladder, capture, and connection, the search didn't show any notable changes. Now these searches are not applied this method.

5.1.1.1 Procedure of Memorizing Board Positioning

Followings are explanation for the procedure of memorizing board positioning.

- Hash Table

  Using the positioning of stone on the board as the key for hashing, the carried valuation of the positioning is appointed to its data.

  The positioning is memorized on the hash table.

  For each turn (Black/White), it is done on different tables (the hash index is prepared for 5,000 entries).

- Key

  The key for hashing is created frome the positioning of stone on the board (the situation of entire intersections (Black/White/Empty) is express on corresponding 2 bits.

- Hashing Function

  The function calculate the surplus of the number divided by entries of hash index.

By memorizing the board positioning, not only when the valuation is carried up, but also when the search goes through child node, the following effect can be gotten.

In Go, the former board position is rarely recreated after spending some moves; such as endless Ko. Before applying the memorizing board positioning, it is difficult to recognize the re-created board positioning. Thus, if the game is still continuing at the specified depth of search, the system decides the re-creation is occured. By memorizing the positioning when blanching to the child node, the re-creation is easily detected.

Comparing with the indirect detection by the depth of search, this method is much accurate and reduces unnecessary search.

5.1.1.2 Effect of Memorizing Board Positioning for Search Performance

For 4 Tsumego problems (all problems, Black plays first) in Fig. 5.11, Dia. 5.12 shows moves and elapsed times for each problems with the memorizing board positioning and without it.

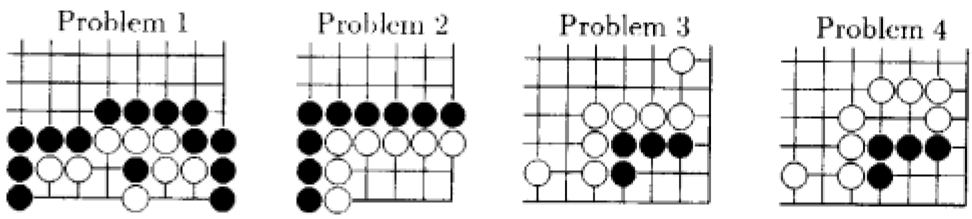


Fig. 5.11 : Tsumego Problems for Measuring Memorizing Board Positioning's Performance

As you see in Dia. 5.12, the performance of search grows couples of times by the memorizing board positioning.

And the problem needs more moves to solve it, the performance appears to grow more.

Dia. 5.12 : Results with Memorisinig Board Positioning and without it

Problem No.	No Memorizing(A)		Memorizing(B)		Growth (A)/(B)	
	Moves	Time	Moves	Time	Moves	Time
Problem 1	652 moves	1.8 sec	319 moves	1.2 sec	× 2.0	× 1.5
Problem 2	2,645 moves	10 sec	877 moves	3.7 sec	× 3.0	× 2.7
Problem 3	5,224 moves	14 sec	1,105 moves	3.9 sec	× 4.7	× 3.6
Problem 4	80,910 moves	237 sec	5,382 moves	21 sec	× 15.0	× 11.3

5.1.2 Killer Heuristic

Killer Heuristic is the theory 'If the search was done in the past board situation which is similer to the current board situation, the best move for the past's might be the one for the current's.'; the move is called 'Killer Move'. This theory is being used for chess, and it effect is recognized. We tested applying the theory to GOG, we couldn't found any improvement. So, GOG is not using the theory. The followings are the result of experiment for killer heuristic on Tsumego problem.

1. Experimention of Killer Heuristic

   On chess programming, many killer moves are being proposed. We tested following two moves on GOG.

   - The best move in previously searched brother branch
   - The best move in previously searched child branch

   The former is a representive killer move, is adapted to chess program. Take chess for explaining these.

   When the player find a move captures his opponent's bishop without any risk after opponent played a move, and the move also refutes opponent's privious move: If opponent played a move at a different point and the move , the player found, is still valid, the search start at the move.

   Estimating what this theory is effective for Go or not is quite difficult. About the situation with the move, dipends on only its position such as 'at the 2-1 point', it might have some effects. But most cases it isn't, because the vital point will change move by move in Go.

   Chess program doesn't have the later,so this might be the first experiment. This move is, playing a tentative move at somewhere and then place a stone at the point which refutes the tentative move as the player's move. It is usually used by human player. This theory is correponding to a proverb in Go, 'My opponent's vital point is my vital point.'.

2. Effectiveness of Killer Heuristic

   Dia. 5.13 shows the resoult of elapsed time and moves for the search in Fig. 5.11 (see 5.1.1), Tsumego problems, with killer heuristic and without it.

   As you see in Dia. 5.13, for Tsumego problem, killer heuristic has no effect on the search performance.

Dia. 5.13 : Result of Search Performance with Killer Heuristic

Problem No.	Without Killer Move (A)	Killer Move 1 (B)	Killer Move 2 (C)	Growth (A)/(B)	Growth (A)/(C)
Problem 1	319 moves 1.2 sec	324 moves 1.3 sec	392 moves 1.7 sec	0.98 0.92	0.81 0.71
Problem 2	877 moves 3.7 sec	895 moves 3.9 sec	870 moves 4.0 sec	0.98 0.95	1.01 0.93
Problem 3	1,105 moves 3.9 sec	1,401 moves 5.0 sec	1,090 moves 4.1 sec	0.79 0.78	1.01 0.98
Problem 4	5,382 moves 21 sec	6,489 moves 25 sec	6,213 moves 26 sec	0.83 0.84	0.87 0.81
Average				0.90 0.87	0.93 0.86

5.2 Shicho Search

Shicho search is used for searching the target string can be captured by sequence of ATARI, and its uses are follows.

- For the judgement of the last phase and creation of the move in the searches, such as capture.
- For conditions of various pattern knowledges.

5.2.1 Target Object

The target is the string satisfies followings.

- The string with 2 or less DAMEs in offense's turn
- The string with 1 or less DAMEs in defense's turn

5.2.2 Last Phase Evaluation Function

If the board positioning satisfies at least one of followings, it will be detarmined as the last phase of search and return its valuation. The limitation of move is the countermeasure for endless ko fights, such as 3 ko.

- Defense's turn and target string has 2 or more DAME.

$$\text{Move's Valuation} = 10 \qquad \text{(Success)}$$

- Offense's turn and target string has 1 DAME.

$$\text{Move's Valuation} = 10 \qquad \text{(Success)}$$

- Offense's turn and target string has 3 or more DAMEs.

$$\text{Move's Valuation} = -10 \quad \text{(Failure)}$$

- If depth of search exceeds 100 moves.

$$\text{Move's Valuation} = 0 \quad \text{(Unknown)}$$

- If total moves for search exceeds 300 moves.

$$\text{Move's Valuation} = 0 \quad \text{(Unknown)}$$

5.2.3 Compulsory Move

If any of following moves are seen on the board, the search will be done just for these.

- In offense's turn, lay a move at where the target string will have 4 or more DAMEs by defense's move at here.

5.2.4 Producing Move

If any compulsory move dosen't exist, following moves are searched in the large order of their valuations. If different moves placed at the same position, the valuation is the most largest valuation of these. Also if different moves have the same valuation, the one at highest position on the board, and if it is, further more, the same the one at center most is given the priority.

- in Defense's turn
  - Every move breaks through the target string's surrounding string, if it exists.

$$\text{Move's Valuation} = 4$$

Target string's DAMEs

$$\text{Move's Valuation} = 3$$

- in Offense's turn
  - Target string's DAMEs

$$\text{Move's Valuation} = \text{Target string's DAME after opponent played at here}$$

5.2.5 Evaluation of Performance

We tested 4 Shicho problems in Fig. 5.12 for the performance of Shicho search. Numbered white stones express target stones, the number does its ploblem number. These problems test Black can capture the target string or not, and all answers are Black can capture them.

Problem 1 is a typical problem which a player must read the capturing race, runs parallel with the edge of board, and Problem 2 and 3 are the one, runs diagonally to the bottom left. Problem 4 is difficult. Reading it must be done by many trials and errors, and it runs to many ways and takes many moves; even professionals have been known to make mistake. Dia. 5.14 shows their results, moves and elapsed times, by Shicho search.

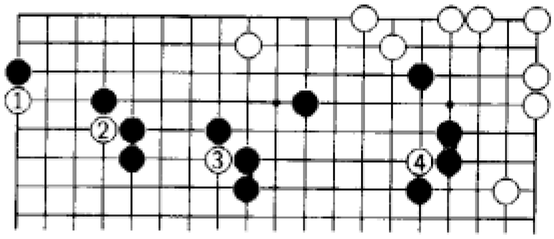


Fig. 5.12 : Shicho Ploblems

Dia. 5.14 : Result of Testing Shicho Search

Problem	Moves (N)	Elapsed Time (T)	Elapsed Time per Move(T/N)
Problem 1	30moves	109msec	3.63msec
Problem 2	12moves	34msec	2.83msec
Problem 3	28moves	94msec	3.36msec
Problem 4	66moves	267msec	4.05msec

From these results, it is clear that the problem needs more moves have the larger elapsed time per move. Examinning such problem 2 and 3, reading proceeds diagonally, the elapsed time per move is in proportion to the number of moves. Considering the relationship between the moves by search (N) and the elapsed time per the number of moves (T/N), and directory adjusted it by 15 results of Shicho problems, take defferent number of moves from others, a following function is obtained.

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

The reason the increase in the elapsed time per move by increase of move might be the procedure for producing the move, breaks through the target string's surrounding string, in defense's turn. Because this procedure examines adjacent points of entire target string's stones, its processing time will increase as increase of target string's stones. For the search, runs straight ahead, the number of the target string's stones in the search process will increase in proportion to the number of moves by search and therefore the elapsed time per move increases in proportion to the number of move by search.

5.3 Capture Search

The capture search searches entire possiblities of capturing the target string, and this is used for following uses.

- Decisiding string's life-and-death
- Producing Capture Search Candidate
- Determining the last phase of search, such as for connection, and producing the move

- As the condition for various pattern knowledge
- As the filter for final decision making of move

5.3.1 Target Object

- String with 3 or less DAME

5.3.2 Last Phase Evaluation Function

If the board positioning satisfies at least one of followings, it will be detarmined as the last phase of search and return its valuation.

- Defense's turn and target string has 4 or more DAME.

  Move's Valuation = 10 (Success)

- Offense's turn and target string has 1 DAME.

  Move's Valuation = 10 (Success)

- Offense's turn and target string has 2 DAMEs, and the result of Shicho search says that the target string can be captured.

  Move's Valuation = 10 (Success)

- Offense's turn, the target string has 2 or 3 DAMEs, and $N1 \times 2 + N2$ is 4 or more.

  Move's Valuation = −10 (Failure)

  Among targe string's surrounding strings, the toatal number of these with 1 DAME is expressed as $N1$ and the one of these with 2 DAMEs is done as $N2$.

- Offense's turn and the target string has 4 or more DAMEs.

  Move's Valuation = −10 (Failure)

- Depth of search exceeds 200 moves.

  Move's Valuation = 0 (Unknown)

- Total moves for search exceeds 200 moves.

  Move's Valuation = 0 (Unknown)

5.3.3 Compulsory Move

If any of following moves are seen on the board, the search will be done just for these.

- in Defense's turn
  - If the target string has 1 DAME, play a move at this DAME or where a part of target string's surrounding string is captured.
  - Play a move at where the target string will have 4 or more DAMEs by defense's move at here.
  - If the size of surrounding string is 2 or more and it has ATARI, play a move at where this suurounding string is captured.
  - Play a move at its ATARI where is belongs to 2 or more different surrounding strings.
  - Play a move at where the surrounding string, size of which is 2, with 2 DAMEs is captured by Sicho.
- in Offense's turn
  - The move, if opponent plays at here, the target string will have 5 or more DAMEs.

5.3.4 Producing Move

If none of compulsory move exists, following moves are searched in order of their valuations. If different moves placed at the same position, the valuation is the most largest valuation of these. Also if different moves have the same valuation, the one at highest position on the board, even it is the same the one at center most is given the priority.

- in Defense's turn
  - DAME

    The target string's DAME (except when opponent plays at this DAME and the target string will have 1 DAME).

$$\text{Move's Valuation} = \text{DAME's Valuation} + \text{DAME's Bonus}$$

following as:

$$\text{DAME's Valuation} = \text{DAME's Base Point} \times \text{DAME's Increment}$$

$$\text{DAME's Base Point} = \frac{10}{\text{Number of Target String's DAME}}$$

DAME's increment expresses the number of increased/decreased target string's DAME by offense's move at here and its maximum is 2. Also DAME's bonus does what Multi-Owner's DAME 's bonus when this DAME belongs to the target string and its surrounding string, Side-by-Side DAMEs, is being alongside of defense's DAME, 's bonus, or 0 with others. Multi-Owner's DAME's bonus is computed by followings.

Multi-Owner's DAME's Bonus = Multi-Owner's DAME's Base Point+Surrounding String

Multi Owner's DAME's base point is computed by the function in Dia. 5.15. Also Surrounding String's DAME's Increment expresses the number of increased/decreased surrounding string's DAME by defense's move at here.

Dia. 5.15 : Multi-Owner's DAME's Base Point Function

Surrounding String's DAMEs	1	2	3	4 or more
Multi-Owner's DAME's Base Point	3.0	2.5	2.0	1.5

Side-by-Side DAMEs' bonus is computed by followings.

Side-by-Side DAMEs' bonus = Side-by-Side DAMEs' Base Poit

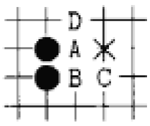
Side-by-Side DAMEs' base point is computed by the function in Dia. 5.16.

Dia. 5.16 : Side-by-Side DAMEs' Base Poit Function

Surrounding String's DAMEs	1	2	3	4 以上
Side-by-Side DAMEs' Base Poit	3.0	2.5	2.0	1.5

- Ikken Point

The move, if the target string has 2 DAMEs and player may play at 2 blocks strait away from (Black's) target string, Ikken move, as Fig. 5.13.



A.× are vacant, B is vacant or White, C or D is Black

Fig. 5.13 : Ikken Point against Target String

$$\text{Move's Valuation} = \begin{cases} 15.0 & \text{(If move's height is 2 or heigher and B is vacant)} \\ 1.0 & \text{(other than above)} \end{cases}$$

- Kosumi Point

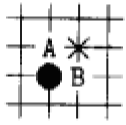
The move, if the player may play at a point diagonaly adjacent to (Black's) target string, diagonal move (Kosumi move), as Fig. 5.14.

$$\text{Move's Valuation} = \begin{cases} \text{DAME's Base Point} + 2.5 & \text{(If move's height is 2 or higher)} \\ 1.0 & \text{(other than above)} \end{cases}$$

- Capturing Move against Surrounding String

The move captures the surrounding string which will enclose the target string by one more move.

Move's Valuation = 6.0



A,B,× must be vacant.

Fig. 5.14 : Kosumi Point against Target String

- Cut Move at ATARI

  The move cuts the surrounding string and is played at the surrounding string's DAME (except when the offense plays at this DAME and the target string will have 1 DAME).

  Move's Valuation = 10.0

- Shicho Move against Surrounding String

  If the surrounding string, height. the nomber of stones, and DAMEs of which it are 2, 1, and 2, can be captured by shicho, the first move of this sequence of shicho.

  Move's Valuation = 8.0

- Pass

  If the largest move's valuation in all of above takes minus. pass the turn.

  Move's Valuation = 0.0

- in Offense's turn

  - DAME

    same as in Defense's turn.

  - Ikken Point

    same as in Defense's turn.

  - Kosumi Point

    same as in Defense's turn.

  - Joint Move for Surrounding String

    If the target string's height and the number of stones are 2 and 1, and its ATARI's height is 1, the move at where the move will complete the connection of string.

    Move's Valuation = 6.0

  - Escape Move from Shicho

    The joint move for surrounding string, if shicho search cannot capture it.

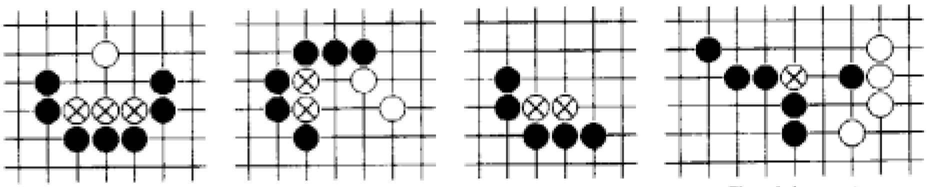
$$\text{Move's Valuation} = \begin{cases} 30.0 & \text{(If surrounding string has 2 or more stones)} \\ 6.0 & \text{(If surrounding string has 1 stone)} \end{cases}$$

- Joint Move for Future Capture

  Among all moves above, if the highest valuation is given as minus and the target string will have 1 DAME by the move hold this valuation, re-position the move at the DAME (except if the target string will have 1 DAME, after playing the move at new position).

5.3.5 Evaluation of Performance

We tested 4 capture problems in 5.15 for the performance of capture search. White stones marked with × express the target string. These problems test possiblities of capturing target string in Black's turn, and all answers are Black can capture it.



Problem 1 Problem 2 Problem 3 Problem 4

Fig. 5.15 : Problems for Capture

Problem 1 is classic examples of 'oiotoshi' known as 'crane's nest'. 2 and 3 are for clumps and nets, 'hasamituke' and 'geta' in Japanese. These problems take just few moves and just require player as 10 kyu, therfore if the first move is found, rest of all moves are easily found. Ploblem 4 is little complicated and should be lead from 'Kake' to 'Loose Radder', therefore, it require the player as 5 kyu. Dia. 5.17 shows their results, moves and elapsed times, by Capture search.

Dia. 5.17 : Result of Testing Capture Search

Problem	Moves (N)	Elapsed Time (T)	Elapsed Time per Move(T/N)
Problem 1	5 moves	59msec	11.8msec
Problem 2	9 moves	141msec	15.7msec
Problem 3	15 moves	124msec	8.3msec
Problem 4	39 moves	469msec	12.0msec

Dipending on the problem, each elapsed time par move is quite differnt from others. The main reason of this is shicho search, is called by the last phase evaluation and the move creation during capture search, takes different time for each problem.

We also examined the correct rate and moves for 50 capture problems; these problems are for 1 - 10 kyu. (Dia. 5.18) . The system didn't solve 7 problems correctly; 4 are incorrect, and rest of 3 exceeds limits of number of search moves, answers are obscure.

Followings are reasons of incorrect answers.

- The sequence of moves which removes the enemy stone and connect the frend stones have not been considered.

Dia. 5.18 : Result of Capture Search's Performance by 50 Capture Problems

Correct Rate	86%
Average Search Moves	33 moves
Total Search Time	26 sec

- Currently ko move is just recognized as a violation move, therefore the system returns ko move as an answer for the problem which can be solved by capture.
- The offense's move for adjacent strings with 3 or more DAMEs have not been produced.
- The offense's move decreases DAMEs of a stone which is connected with the target string by kosumi have not been produced.

By improving on these the, hereafter, the correct rate will increase. However it will increase the search time also. Therefore it must be carefully done.

3 unsettled plobrems are solved by removing the limit of the number of move; each one is solved with 202, 363, and 759 moves. The reason why these problems take many moves is considered the valuation for the move decreases the target string's DAMEs might be a problem.

The computation for the valuation is just decieded by the efficiency of decreasing that. Therefore, when the move decreased the DAME but the opponent regains the DAME, the valuation is given higher score.

5.4 Connection Search

The connection search searches entire possibilities of complete connection of the linkage's both edge, means the linkage becomes a string or not, and if it's not possible, the search determines it is disturbed by crosscuts or breaking through.

5.4.1 Target Object

All linkages except perpendiculars.

5.4.2 Last Phase Evaluation Function

If the board positioning satisfies at least one of followings, it will be detarmined as the last phase of search and return its valuation.

- Offense's turn and the edges become a connected string.

  Move's Valuation = 10 (Connect)

- If the distance between both edges' connected strings (between 2 nearest points to each other connected strings) is $\sqrt{5}$ (Keima) and 2 enemy stones line up vertically against the linkage.

  in Offense's turn

  Move's Valuation = 10 (Break Through)

- in Defense's turn

  Move's Valuation = −10 (Break Through)

- If the distance between each edge's connected string is $\sqrt{4}$ (Ikken) and 3 enemy stones line up vertically against the linkage.
  - in Offense's turn

    Move's Valuation = 10 (Break Through)

  - in Defense's turn

    Move's Valuation = −10 (Break Through)

- If the distance between each edge's connected string is $\sqrt{2}$ and crosscuts pattern is found in this area.
  - in Offense's turn
    1. Either one of edge's string can be captured in opponent's turn.

       Move's Valuation = 10 (Break Through)
    2. In offense's turn, except 1, enemy's both cutting stones can be captured or either one of these cannot escape.

       Move's Valuation = −10 (Connect)
    3. other than aboves

       Move's Valuation = 5 (Crosscuts)
  - in Defense's turn
    1. Either one of edge's string cannot escape in offense's turn.

       Move's Valuation = −10 (Break Through)
    2. In opponent's turn, except 1, both cutting stones cannot escape or either one of these can be captured.

       Move's Valuation = 10 (Connect)
    3. other than aboves

       Move's Valuation = −5 (Crosscuts)

- If the depth of search exceeds 20 moves.

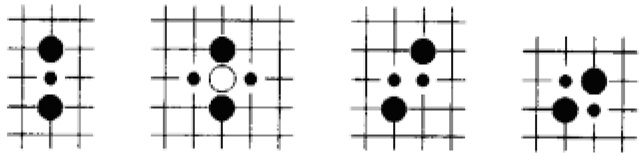
  Move's Valuation = 0 (Unknown)

- If total moves of search exceeds 300 moves.

  Move's Valuation = 0 (Unknown)

5.4.3 Compulsory Move

If the position of both edges' connected strings (of 2 nearest points to each other connected strings) is Ikken, Keima, or Crosscuts, and specified points as follows are empty, the move at these points are determined as compulsory moves.



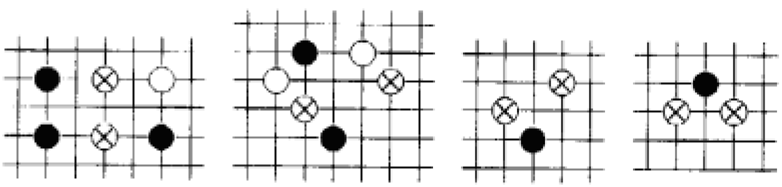
5.4.4 Producing Move

If none of compulsory move exists, following moves are detarmined as a candidate for the move in order of one's distance from the center of linkage's both edges.

- Entire points within the square created by linkage's both edges (if position of these edges is Ikken or Niken, expand the square to an outer block in both way).
- If the enemy string with ATARI exists within area specified above, this ATARI.

5.4.5 Evaluation of Performance

We tested 4 connection problems in 5.16 for the performance of connection search. White stones marked with × express the target linkage. These problems test possiblities of cutting target linkage in Black's turn; the answer for problem 1 is impossible, problem 2 is crosscuts, and problem 3 is breaking through.



Problem 1 Problem 2 Problem 3 Problem 4

Fig. 5.16 : Problems for Connection

Dia. 5.19 shows their results, moves and elapsed times, by Connection search.

Most of time costs for search are spent by the last phase recognition by capture search. Therefor each problem's time costs per move is quite different from other's. Also the result of this search is relying on the accuracy of capture search.

The connection search need time to excess, therfore, currently, it is not used. Herenow it is necessary to improve the performance of this search by improving the huristic used for the last phase recognition and move production.

Dia. 5.19 : Result of Testing Connection Search

Problem	Moves (N)	Elapsed Time (T)	Elapsed Time per Move (T/N)
Problem 1	19 moves	0.197sec	10.4msec
Problem 2	143 moves	13.2sec	92.3msec
Problem 3	6 moves	0.927sec	155msec
Problem 4	12 moves	0.095sec	7.9msec

5.5 Tsumego Search

Tsumego search searches completely the life-and-death of target group completely surrounded; its result will be alive, dead, or ko. However this search's start condition is still in study, so it won't be called from game mode; only it can be done from analysis mode.

5.5.1 Target Object

Completely surrounded group

5.5.2 Special Arrangement for Ko

In Tsumego, ko often becomes a problem's answer. The problem takes ko as its answer is the one of the life-and-death depends on the issue of ko when the search founds ko. As the answer for search, it is prefered one succeeded in search without any condition to one done by winning in a ko fights. Therefor ko will be returned as an answer only when the search succeeds by only winning in a ko fights. For this, following special arrangements are done for ko.

- Ko move is produced very last.
- Only when defense's previous move was ko, pass is produced as offense's move then pass won't be produced as the defense's move.
- If ko move is played and the accumulated valuation of the board positioning at this moment is scored 10, means success, the valuation will be replaced by 5, means ko.

5.5.3 Last Phase Evaluation Function

If the board positioning satisfies at least one of followings, it will be detarmined as the last phase of search and return its valuation. We define KANAME ISHI, means important stone, as the string which is scored with the most highest rate of KANAME ISHI among target string's stones, and it will be computed when the search starts. The rate of KANAME ISHI is computed by a following formula.

Rate of KANAME ISHI = Stone's DAMEs−Distance between Stone and Target Group's Center of (

The target group's center of gravity is the average of target group's entire stones' coordinates.

- in Offense's turn and KANAME ISHI has 1 DAME.

Move's Valuation = 10 (Success)

- Depth of search exceeds 30 moves.

Move's Valuation = 0 (Unknown)

- Total moves of search exceeds 10,000 moves.

Move's Valuation = 0 (Unknown)

5.5.4 Producing Move

The area possible to play a move in offense's/defense's turn is computed when the search starts.

- Point defense may play at

  Points within completly surrounded target group's interrior, means inner points surrounded by their surrounding linkage's stones and block points.

- Point offense may play at

  All points defense may play at and adjacent vacant points to these.

These points are sorted in order to their possibilties. This possibility is computed as follows. If defferent moves are played at the same point, each move's valuation for this point are added together. If more than one valuation take the same value for different points, the one, positioned at the highest on board and even the height is the same, the one positioned the closest to the center of board, will be selected.

- Outer DAME

  Point offense may play at but defense doesn't.

  Possiblity's Valuation = 0

- Wall Point

  Points defense may play at and is adjacent to outer DAME.

  - DAME Area

    Points not adjacent to vacant Inner Point.

    Possiblity's Valuation = 0

  - Non-DAME

    above others

    Possibilty's Valuation = 10

Dia. 5.20 : DAME's Base Point Function

Nd	0	1	2	3	4 or more
DAME's Base Point	0	0	1	7	15

- Inner Point

  Points defense may play at but not Wall Point.

  $$\text{Possibility's Valuation} = \text{DAME's Base Point} + \text{Position's Base Point}$$

  DAME's base point is computed by the function shown in Dia. 5.20; Nd expresses the number of stones which is this point's adjacent points except defense's. Also position's base point is computed by the function shown in Dia. 5.20 following by the point's standard coordinates.

Dia. 5.21 : Position's Base Point Function

Standard Coordinates	(1,2)	(2,2)	others
Position's Base Point	2	1	0

- KANAME ISHI's DAME

  If the point offense may play at and is KANAME ISHI's DAME, its possibility's valuation is computed by the function in Dia. 5.22.

Dia. 5.22 : KANAME ISHI's DAME's Possiblity Function

KANAME ISHI's DAMEs	1	2	3	4 or more
Possibility's Valuation	10	5	2	1

Using points found by aboves, following moves are created by following their order.

- Point may play at

  For each player, the move at where is vacant but isn't ko point among points, he may play at, are created by followin sorted order.

- Pass

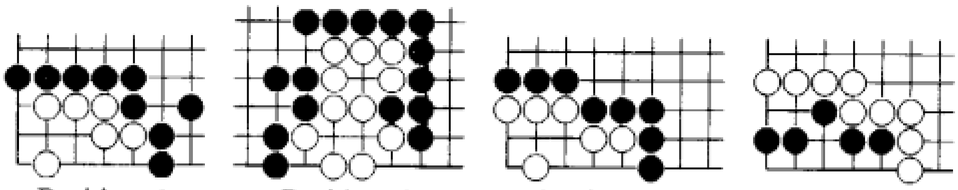
  Only when opponent's previous move wasn't ko, pass is produced. It is only provided when 2 moves, except the prohibited move, have been created at this board positioning in defense's turn, and when opponent's privious move was played at ko point in offense's turn.

- Ko Point

  If a point, endless cycles of capture and re-capture occurs by playing a move at,

5.5.5 Evaluation of Performance

We tested 4 Tsumego problems in 5.17 for the performance of tsumego search. These problems test surrounded stones may survive or not in Black's turn, and answers for problem 1~3 are survive and for 4 is dead.



Problem 1 Problem 2 Problem 3 Problem 4

Fig. 5.17 : Problems for Tsumego

Problem 1's answer is playing a move at 2-1, and 2 should be lead from the move decending to the edge of board to false eye. In 3. Black should use throw-in move preparing for snapback, 'uttegaeshi' in Japanese. In 4, a move will be found in where once Black's stones were removed. Dia. 5.23 shows their results, moves and elapsed times, by Tsumego search with memorizing board positioning (see 5.1.1).

Dia. 5.23 : Result of Testing Tsumrgo Search

Problem	Moves (N)	Elapsed Time (T)	Elapsed Time per Move(T/N)
Problem 1	386 moves	2.04sec	5.3msec
Problem 2	462 moves	2.51sec	5.4msec
Problem 3	591 moves	13.03sec	5.1msec
Problem 4	1,572 moves	7.22sec	4.6msec

The reason why total moves for search increaces, its search time decreases is considered it is related to preparation at when search is started. Specially sorting process costs a lot; in tested 4 problems, it took 0.3 sec on their average.

5.6 Real Eye Search

Real eye search searches if the target territory could have real eyes, and it is used for calculation of counting real eyes, and etc. However this search's start condition is still in study, so it won't be called from game mode; only it can be called from analysis mode.

o

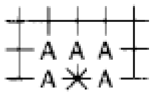
5.6.1 Target Object

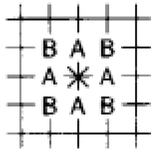
The target is the territory, closed area, size of which is 4 or less.

5.6.2 Last Phase Evaluation Function

If the board positioning satisfies at least one of followings, it will be detarmined as the last phase of search and return its valuation. The target territory for eyes are points within the territory and vacant points, are adjacent to the territory and won't become a simple false eye. And perfect one eye is a territory surrounded with 2 or more DAMEs.

Furthermore vacant point won't become a simple false eye, × in picture below, is defined as one dosen't have any enemy stones at As or dose at only one of Bs; enemy stone has 3 or more DAMEs, or dose 2 DAMEs and can be captured by Shicho.





- in Defense's turn, perfect real one eye appears in target territory for eyes.

  Move's Valuation = 10 (Success)

- in Offense's turn, perfect real one eye appears in target territory for eyes.

  Move's Valuation = −10 (False

- in Defense's turn, all vacant points won't become a simple false eyes disappears from target territory for eyes.

  Move's Valuation = −10 (False)

- in Offense's tern, all vacant points won't become a simple false eyes disappears from target territory for eyes.

  Move's Valuation = 10 (Success)

- Depth of search exceeds 20 moves.

  Move's Valuation = 0 (Unknown)

- Total moves of search exceeds 10,000 moves.

  Move's Valuation = 0 (Unknown)

5.6.3 Producing Move

Following moves are searched in the large order of their valuations. If different moves have the same valuation, the one at highest position on the board, and if it is, further more, the same the one at center most is given the priority.

- in Defense's turn

- Moves in Kosumi range

  All vacant points in Kosumi range from points in target territory for eyes; valuation was computed before the search starts, and it won't be re-computed during the search.

  Wall Point Vacant points adjacent to points except defense's candidates.

  $$\text{Move's Valuation} = 6$$

  Inner Point Vacant points except wall points.

  $$\text{Move's Valuation} = D \times 2$$

  D is the number of DAMEs of string, will be created by offense's move and includes its stone, except wall points and its maximum is 4.

- Move removes enemy stone.

  All enemy stones with 1 DAME in Kosumi range from points in target territory for eyes.

  $$\text{Move's Valuation} = 10$$

- in Offense's turn

  - Moves in Kosumi range

    same as in defense's turn

    Outer Point

    All vacant points adjacent to vacant points in Kosumi range from points in target territory for eyes

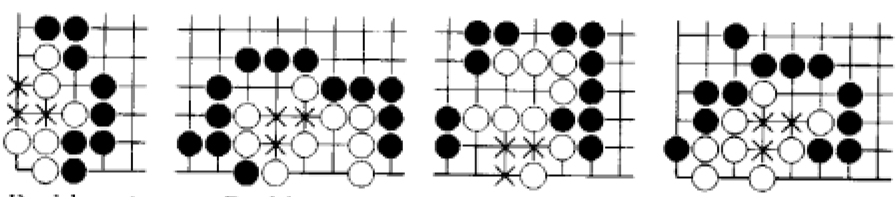
    $$\text{Move's Valuation} = 2$$

  - Move removes enemy stone.

    same as in defense's turn.

5.6.4 Evaluation of Performance

We tested 4 eyes problems in 5.18 for the performance of real eye search. These problems test possiblities of capturing target string in Black's turn, and all answers are Black can capture it.



Problem 1 Problem 2 Problem 3 Problem 4

Fig. 5.18 : Problems for Real Eye Search

Dia. 5.24 shows their results, moves and elapsed times, by Real Eye search with memorizing board positioning (see 5.1.1).

The reason why it seems its elapsed time is shorter when its search moves are fewer seems to be caused by pre-transaction at its starts.

Dia. 5.24 : Result of Real Eye Search

Problem	Moves (N)	Elapsed Time (T)	Elapsed Time per Move (T/N)
Problem 1	46 moves	0.86sec	18.7msec
Problem 2	83 moves	1.13sec	13.6msec
Problem 3	94 moves	0.97sec	10.2msec
Problem 4	360 moves	3.38sec	9.4msec

5.7 Multi-Target Capture Search

The multi-target capture search aims at some strings at the same time and searches these completely for which one of these might be captured. However this search is just as an experiment for the search treats multiple targets and purposes for these, therofore it won't be called from game mode; only it can be done from analysis mode.

5.7.1 Target Object

Objects are some specified same colored strings. However when calling it from analysis mode, for simplify its operation, if a string is specified, the search aims at entire enemy strings adjacent to specified string.

5.7.2 Last Phase Evaluation Function and Producing Move

The last phase of this search is determind as follows. When the search reaches the last phase, the search returns its valuation, and when haven't yet, the search produces the move.

- in Defense's turn, there are some target strings with 1 DAME (means the string is ATARI).
  - If no move, connects between frend strings, is common among entire target strings being ATARI.

    Move's Valuation = −10 (False)

    other than above

    A common move, connects between friend strings, among entire target string being ATARI is detarmined as the move.
- in Defense's turn, only one target string is ATARI.

    If the result of shicho search for the target string, is ATARI, in defense's turn is false (can't escape).

    Move's Valuation = −10 (False)

  - other than above

    The move returned as success by shicho search is determined as the move.
- in Defense's turn, other than above

$$\text{Move's Valuation} = 10 \quad \text{(Success)}$$

- in Offense's turn, some target strings are ATARI.

$$\text{Move's Valuation} = 10 \quad \text{(Success)}$$

- in Offense's turn, no target string has 2 DAMEs.

$$\text{Move's Valuation} = -10 \quad \text{(False)}$$

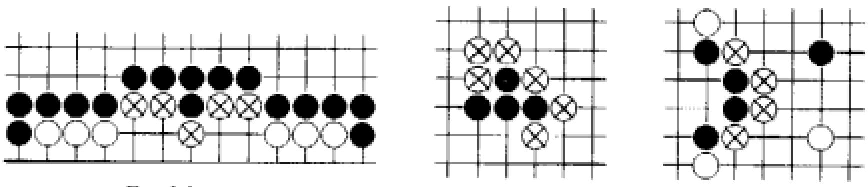
- in Offense's turn, other than above.

  All DAMEs of taget strings with 2 DAMEs are determined as moves. The valuation is computed for each move, and these moves are produced in large order to their valuations. If more than one valuation takes the same value for different points, the one positioned highest on the board will be selected. If the height is the same, the one positioned closest to the center of the board will be selected.

$$\text{Move's Valuation} = \sum \frac{10}{\text{DAMEs of target string's same colored adjacent strings}}$$

5.7.3 Evaluation of Performance

We tested 3 multi-target capture search problems in Fig. 5.19 for the performance of multi-target capture search. White stones, are marked with $\times$, express target stones. These problems test Black can capture the target string or not, and all answers are Black can capture them.



Problem 1 Problem 2 Problem 3

Fig. 5.19 : Problems for Multi-Target Capture Search

Problem 1 should be lead from cutting at ATARI to both player's ATARI, and 2 should be done from ATARI race to shicho. 3 is just like 2, from ATARI race to shicho, however in case of 1 and 2, enemy strings having 2 DAMEs after 1 move played are attacked continuously at their ATARI, in 3, shicho is started with combining separated ATARIs. Dia. 5.25 shows their results, moves and elapsed times, by Multi-Target Capture search.

Dipending on the problem, each elapsed time par move is quite differnt from others.

Dia. 5.25 : Results of Testing Multi-Target Capture Search

Problem	Moves (N)	Elapsed Time (T)	Elapsed Time per Move(T/N)
Problem 1	3 moves	14msec	4.7msec
Problem 2	7 moves	107msec	15.3msec
Problem 3	17 moves	72msec	4.2msec

6 Decision for Movement

For the recognized board situation, move candidates are produced by various knowledge of move candidates. At once each one of move candidates is evaluated, sometimes, however, it will be re-evaluated again by the move candidate filter. Also as the result of situation judgement, the move might be chandged, however, the point scored with the largest valuation, basically, will be selected as the move.

6.1 Enumeration of Candidates

Corresponding to each one of valious objects' situations, are computed by situation judgement, the candidate knowlwdge is executed and 12 move candidates in Dia. 6.26 are produced.

Dia. 6.26 : Varieties of Move Candidate

Varieties of Move Candidates	Object	Purpose
JOSEKI	a standard pattern of good play in the corner	JOSEKI move
Edge	a pattern along with the edge	extention, pincer, side's mid-point, and etc
Invasion	a pattern in the corner	invade/protect at 3-3
Capture/Escape	a neutral string	capture/escape for a stone
DAME	touched Black/White strings	stones' competition, such as hane, nobi, oshi, and etc
Cut/Connect	a weak linkage	cut/connect stones
Enclose/Escape	a weak group scored with low siege rate	
FUTOKORO Expand/Reduce	a weak group with expandable FUTOKORO	expand/reduce for a weak stone
Life-and-Death/ Capturing Race	a weak group with 1.5 real eyes or Capturing-Race-Warning group	a vaital point for an eye or filling opponent's DAME
Separete/Contact	a big seperatable group	separeting/contactting stones
Sphere	a family with weak border	obtain/prevent sphere as the territory
Spheres' Contact Point	touched Black/White families	expand own sphere and prevent expantion of enemy's

6.1.1 JOSEKI Move Candidate

Produce a JOSEKI move candidate, if JOSEKI is proceeding in conners of board.

JOSEKI is stored on the tree structured files created by JOSEKI editor, one of knowledge editor tools. Currentaly 239 varieties of diagram (leaves of JOSEKI tree) are recorded as JOSEKI datas.

1. Position of Move Candidate

   Positions of progressing JOSEKI node's entire children nodes

   If a child node has pass as the move, move candidates as opponent's are produced at entire its children nodes' position.

2. Valuation of Move Candidate

   Each nodes stores the base point for the valuation. In some cases, nodes store the information for the base point adjustment corresponding to its phase. The Valuation of move candidate is computed with the node's base point and adjustment point as follows.

   $$\text{Valuation} = \text{Base Point} + \text{Adjustment Point}$$

   The adjustment point will be added if the condition for the adjustment is matched; the point might take either plus or minus. As the adjustment condition, only the comparison between the potential value at any position and specified constant is allowed.

3. Management when JOSEKI is in progress

   If the stone positioning in each 4 conner of the board matches the one at JOSEKI tree's node except its leaves and no other stone is found within JOSEKI scope, it is detarmined that JOSEKI is in progress. Othere cases, it is detarmined as out of JOSEKI. Once it became in out of JOSEKI, JOSEKI move candidate won't be produced for this conner. JOSEKI scope is $n \times n$ area in each 4 conner; n is calculated as follows.

   $$n = \text{Size of Board}/2 - 1$$

6.1.2 Edge Move Candidate

If stones on the 3rd or 4th line is facing to each other and the distance between them is L ($3 \leqq L \leqq 12$), following 2 edge move candidates are produced.

1. Strategic Move

   If both facing stones have a strong status (group's strength>20), produce a strategic move.

   (a) Position of Move Candidate

      i. Side Direction
      Center of facing 2 stones.
      If L is even number, shift it to the weaker stone.
      If 2 defferent (Black/White) colored stones are facing to each other and take distance of 4 blocks; Black's move candidate takes 2 blocks away from Black stone (1 from White) and White's move candidate takes 2 from White (1 from Black).

ii. Height

On the 3rd or 4th line.

It is decided by color of facing stones and their height (Dia. 6.27).

Dia. 6.27 : Height of Strategic Move

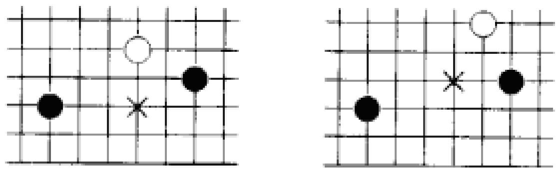
on one hand of facing stones		on the other hand of facing stone			
		Black		White	
		3rd	4th	3rd	4th
Black	3rd	4/3	4/3	3rd	3rd
	4th	4/3	3rd	3rd	3rd
White	3rd	3rd	3rd	3/4	3/4
	4th	3rd	3rd	3/4	3rd

4/3···Black's on 4th, White's on 3rd

3/4···Black's on 3rd, White's on 4th

iii. Adjustmemt for Position of Move Candidate

For above all, If the same colored stones are facing to each other and the enemy stone exists between them and on the 5th or 6th line, the move position is moved to 2 blocks down from the enemy stone (left figure in Fig. 6.20).
And if the destance of side direction between the enemy stone and one hand of facing stone is only 1 block, the position is shifted 1 block to the center (right figure in Fig. 6.20).



×···Position of Move Candidate after Adjustment.

Fig. 6.20 : Adjustment for Position of Strategic Move

(b) Valuation of Move Candidate

It is decided by size of L (Dia. 6.27).

However if the same colored stones are facing to each other on the 5th or 6th line, its valuation is doubled.

Dia. 6.28 : Valuation of Strategic Move

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

(c) Special Treatment

The enemy's strategic move for the same colored facing stones is, except called as the edge move candidate, done as the invasion move (6.1.3).

2. Extention/Pincer Move

   If either one of facing stones is weak (group's strength ≦ 20), produce an extention/pincer move.

   (a) Position of Move Candidate

      i. Side Direction

         From weaker one of 2 facing stones, the friend's move candidate (extention) is positioned at 2 blocks away and enemy's (pincer) is done at 1 block away.
         If L is 3, it is positioned at the center.

      ii. Height

         The same height as weaker one of 2 facing stones.

   (b) Valuation of Move Candidate

$$\text{Valuation} = \text{the weakest one of facing stones 's Importance of Interference} + 4$$

6.1.3 Invasion Move Candidate

For each 4 cornner of the board, produce an invesion move at 3-3. The general invesion move, except one at 3-3, is handled during the process for producing the edge move candidate (see 6.1.2).

1. Position of Move Candidate

   If the move exists only at a star (4, 4) within 4 × 4 area, means the rest of all points is vacant, produce an invesion move candidate at 3-3.

2. Valuation of Move Candidate

$$\text{Move's Valuation} = 20$$

6.1.4 Capture/Escape

Produce a capture/escape move candidate for the neutral string on the board.

1. Position of Move Candidate

   Produce a capture move candidate at the string's capture point and an escape move candidate at its escape point.

2. Valuation of Move Candidate

$$\text{Valuation} = 10 + (\text{String's Stones} - 1) \times 2 + \text{String's Group's Inportance of TANE ISHI}$$

3. Bad Shape Adjustment

   If an escape move candidate forms an empty triangle, a bad shape, adjust the move's postition. The adjustment for the move's position is done by retrying the capture search on the rest of moves after cutting branches off as the capture search for the string's escape point, and vacant points within Kosumi area from the escape point in the bad shape until non-bad shaped escape point is found. For points within kosumi area from the bad shape move, following conditions are added. In Fig. 6.21, as an example, A is played, only if D is Black. Also B and C are played, if D and E, and B and C are Black. When the search is being retried, the highest positioned one of moves, if some moves are

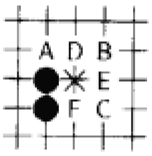


Fig. 6.21 : Search Conditions of Bad Shape Adjustment

positioned on the same hight, the most closest one, will be searched first;.

4. Capture/Escape Postion Optimumzation

   The capture search returns the first succeeded move only. It's just for saving the processing time, but using this way can't found the best move with the same purpose as the first one's. The capture/escape position optimumzation manages not to loos such a move with saving its processing time and it will be done as follows.

   > For points within Keima area from capture/escape move candidate, find points with larger valuation In order of large valuation, the capture search is tried at these points for if these points saticefy the purpose of capture/escape. If the point satisfies all of above is found, re-position the capture/escape move candidate at there.

   In an example 6.22, an escape move candidate is produced at A. By this optimumzation, B saticifies purpose of escape also and a merit of connection is found at B; therefore escape move move candidates from A to B.

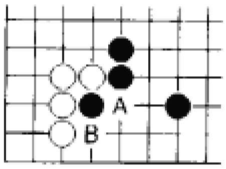


Fig. 6.22 : Example for Capture/Escape Postion Optimumzation

6.1.5 DAME Candidate

If Black/White stones are close positioned each other, DAME candidate is created as offense/diffence move for each sphere of influence. In GO, this is called Hane or Nobi/Oshi move.

1. Position of Move Candidate

   If such positioning of Black/White stones as ×s in Fig. 6.23 are found, (a) is called Side-by-Side DAMEs and (b) is Multi-Owner's DAME. If the positioning could be detarmined as both DAMEs, it is rather detarmined as Side-by-Side DAMEs.

(a)Side-by-Side DAMEs (b)Multi-Owner's DAME

Fig. 6.23 : Pattern of DAME move candidate

2. Valuation of Move Candidate

$$\text{Valuation} = \text{Worth of DAME} + \text{Bonus Point}$$

3. Worth of DAME

   The worth of DAME is computed by the following formula; t is as TANE ISHI coefficient, v is as DAME's base point of the string with n DAMEs. δ is as its DAME's increment, and g is as bad shape coefficient. Except if the candidate is Side-by-Side DAMEs, this computation is done for the current player's DAMEs. And if it is Multi-Owner's DAMEs, its result is the total of DAME's worth on both Black/White side; if there are many combinations, maximum values are used.

$$\text{Worth of DAME} = t \times v(n) \times \delta \times g$$

$$t = \begin{cases} 2 & \text{(if DAME is TANE ISHI)} \\ 1 & \text{(others)} \end{cases}$$

String's DAMEs(n)	1	2	3	4	5	6 or more
DAME's Base Point($v(n)$)	10	5	3.3	2.5	2	1

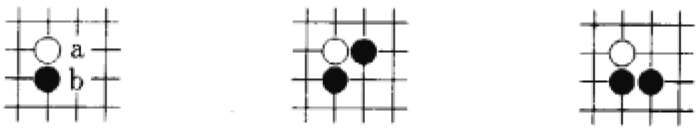
$$\delta = \min(\text{String's DAMEs after the move} - \text{String's DAMEs before the move}, 2)$$

$$g = \begin{cases} 0 & \text{(if play a move at Multi-Owner's DAME, an empty triangle is made)} \\ 1 & \text{(others)} \end{cases}$$

4. Bonus Point

   (a) Side-by-Side DAMEs' Bonus Point

   For Side-by-Side DAMEs, such as Hane(a) and Nobi(b) In Fig. 6.24, 2 move candidates are produced. Rather which could be a candidate is diceded by Side-by-Side DAMEs' bonus point.



(a)Hane (b)Nobi

Fig. 6.24 : Hane and Nobi

i. Position of Side-by-Side DAMEs' Bonus Point
If Df is as the number of target string's DAMEs, Blacks in Fig.6.24, De is as the number of adjacent enemy string's DAMEs, Whites in Fig. 6.24, Da is as the number of the string, is created by Hane, 's DAMEs, and Db is as the number of the string, is created by the enemy's Hane, 's DAMEs,

$\min(Df, Da) \geqq \min(De, Db)$ therefore select the Position of Hane (a in Fig. 6.24)

$\min(Df, Da) < \min(De, Db)$ therefore select the Position of Nobi (b in Fig. 6.24)

ii. Side-by-Side DAMEs' Bonus Point
If the target stone's height is 2 or higher and it saticifies one of follows, the point is given 8. Others are 2.

- $Df \geqq 2$ and $Db \leqq 2$
- $Df = De = 3$, $Da = Db = 3 or 4$, and each adjacent Black/White Stones are counted $Da - 1$

6.1.6 Cut/Connect Move Candidate

The candidate for the cutting move or connecting move is created. Entire weak linkages (see 4.4.2) on the board are targeted. Followings are exceptions.

- If both edges of linkage belong to the same group.
- If both edges of linkage belong to the same family and the family's interior is 4 or more.
- If the point where Cut/Connect move candidate found by following processes is considered playing at belongs to the group of linkage's enemy.

The candidate moves for the target weak linkage are produced as follows.

1. If the linkage takes Kosumi positioning and is crosscutting.
   (a) Position of Move Candidate
   The cutting move candidate (only in the enemy of linkage 's turn) is positioned at crosscutting enemy's neutral string's escap point, and the connection move candidate (only in the linkage's turn) is positioned at crosscutting enemy's neutral string's capture point.

(b) Valuation of Move Candidate

Move's Valuation = 15

2. If the candidate is other Kosumis above, Ikken, or Keima.

   (a) Position of Move Candidate
   The cutting move candidate is positioned at the position of the next move by the connection search in the enemy's turn, and the connection move candidate is at the one in the linkage's side 's turn.

   (b) Valuation of Move Candidate
   Following Dia. 6.29, it is decided by the result of the connection search in Black/White's turn.

Dia. 6.29 : Valuation of Cut/Connect Move Candidate in Kosumi, Ikken, and Keima

	Enemy of Linkage 's turn	
Linkage Side's turn	Break Through	Crosscuts/Unknown
Connect	30 points	15 points
Crosscuts/Unknown	22.5 points	10.5 points

3. If Niken,Oh-Geima,Hazama, and Perpendicular 2, 3, and 4

   (a) Position of Move Candidate
   The position is decided by "Cut/Crosscut Move Candidate Pattern Knowlwdge" (see APPENDIX).

   (b) Valuation of Move Candidate
   The valuatio is decided by "Cut/Crosscut Move Candidate Pattern Knowlwdge" (see APPENDIX).

6.1.7 Enclose/Escape Move Candidate

The enclose/escape move candidate will be produced for the neutral group (its strength is from 1 to 35).

1. Position of Move Candidate

   (a) Cut/Connect Move against/toward Reinforcements
   If the rate of siege is 3 or lower (see 4.5.7), the cut/connect move against/toward reinforcements is produced. The position is the center of 2 stones; one is reinforcements and another is the stone nearest to the reinforcements among the neutral group. Except if the reinforcements has more than one stone, the one which has the highest strength is selected.

   (b) Enclose/Escape Move
   Except above, and if these rate of sige is 3 ~ 10, the enclosing/escape move will be produced as follows.

   - Produce the move candidate for each the group's the group of group's DAME as follows.

- If 6 or more group's DAME exist in the group of group's DAME, 5 points nearest to the center of group are selected.
- A point nearest to the center of selected 5 or less points' gravity amontg the neutral goup is searched. This point becomes the base point.
- The point nearest to the center of gravity is selected from the base point's Ikken, Kosumi, and Narabi point.
- If selected point is Ikken, the following adjustment is applied.
  - If the center of Ikken and the base point is adjacent to the enemy's stone. The position is removed to the center of Ikken and the base point (Narabi).
  - If the enemy stone exists at next to the stone of the base point. If a stone of the base point 's string's DAMEs are 3 or more and the enemy's string 's DAMEs are 2 or less, the position is removed to the enemy's stone 's Hane (Kosumi) point.

(c) Fortifying/Breaking Through Enclosure Move

Except above, if the neutral group os escapable (see 4.5.7), the fortifying/breaking through enclosure move is produced.

- If Nikken, Oh-Geima, or Perpendicular 4 exists on the surrounding linkage being surrounding the neutral group. The move is positioned at the block point of the surrounding linkage
- If the week linkage except Perpendicular 2 is exist. The move is positioned at the position of the connect/cut move candidate toward/against the week linkage.

2. Valuation of Move Candidate

$$\text{Move's Valuation} = \text{Group's Importance of Interference}$$

3. Re-Evaluation for Enclose/Escape Move Candidate

For the enclose/escape move and fortifying/breaking through enclosure move, the move candidate will produced, if following conditions are saticified after playing these moves actually.

- By the calculation for the rate of siege (see 4.5.4) starting from the position of the move candidate, the rate should be scored 5 or higher or the surrounding stone should include the point which is friend's side but doesn't belong to the surrounded group.

6.1.8 FUTOKORO Expand/Reduce Move Candidate

If the enclose/escape move is not produced as a candidate move against/toward the neutral group, FUTOKORO expoand/reduce move candidate will be produced. However currentaly this candidate is doubling as YOSE move candidate, therefore the candidate might be produced to the one which is not neutral group.

1. Position of Move Candidate

   The position is decided by "FUTOKORO Expand/Reduce Move Candidate Pattern Knowlwdge" (see APPENDIX).

2. Valuation of Move Candidate

   If the target group of stones is a nuctral group,

   $$\text{Move's Valuation} = \text{Group's Importance of Interference}$$

   Except above,

   $$\text{Move's Valuation} = V_f$$

   V_f is the valuation decided by "FUTOKORO Expand/Reduce Move Candidate Pattern Knowledge".

6.1.9 Life-and-Death/Capturing Race Move Candidate

If none of enclose/escape or FUTOKORO expand/reduce move is produced as a candidate move against/toward the neutral group, the life-and-death/capturing race move candidate will be produced.

1. Position of Move Candidate

   (a) Nakate Move

   When the group has 1.5 eyes, the candidate is positioned at the group's Nakde position (see 4.5.3).

   (b) Capturing Race Move

   When the group is a capturing-race-warning group (see 4.5.7), the candidates are positioned at entire DAMEs of stones which belong to the target enemy group of capturing race.

2. Valuation of Move Candidate

   $$\text{Move's Valuation} = \text{Group's Importance of Interference}$$

6.1.10 Separate/Contact Move Candidate

The candidate will be produced for the family with 30 or more as its size. The separate/contact move candidate is the one separates a family into more than one families (or defends from such a move).

1. Position of Move Candidate

   The move candidate is positioned at the potential's separating point. The potential's separating point is calculated by approximation of "Saddle". The potential's separating point becomes the separate move candidate in the family's enemy 's turn and does the contact move candidate in the family's friend 's turn.

2. Valuation of Move Candidate

as the contact move candidate

$$\text{Move's Valuation} = \min(21 - \text{Strength of Family}/2, 10)$$

as the separate move candidate

$$\text{Move's Valuation} = \text{Contact Move Candidate's Valuation} \times 0.8$$

3. Potential's Separating Point

The potential's separating point realizes the idea which is to figure out the point, separates a family effectively to more than one families, by the distribution of potential. If we suppose the potential to be the height above sea level, every each stone on the board is recognized as a mountain. Then the family is recognized as a chain of mountains with the same or higher height above sea level (a mountain range). If the mountain range is sunk, this one mountain range turns into more than one islands. We define the point just right before islands appear as the potential's separating point. From this, digging at the potential's separating point is realized as the best way to devide a mountain range into more than one. However, searching the potential's separating point by this method requires to check the familly, if it turns into more than one or not, and this check requires large costs.

4. Calculation for Potential's Separating Point by Approximation of "Saddle"

Thinking of the potential is the same as the height above sea level, the potential's separating point is expressed f as the saddle part of horse's back. Therefore searching the point with the potential around it is distributed like "Saddle" shape is accepted as the one of approximation of computing the potential's separating point.
The search processes for the "Saddle" shaped point are followings.

- For each point of the family, the potential's distribution, 3×3 wide, around the point is calculated.
- As the potential distribution map, the left in Fig. 6.25, if there are 2 groups, one is the group of points scored higher potential than the center of the potential distribution 's and another is the group of points scored lower potential than that, this center point is considered it is taking "Saddle" shape.

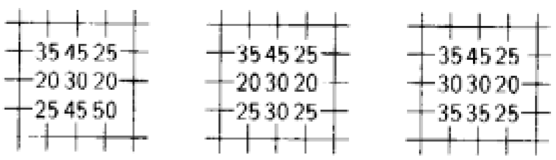


Fig. 6.25 : Example of "Saddle"

- In the midle and right figure, the point which has the same potential value as the central point does exists around the central point. Without an adjustment for the potential distribution ahead of points with the same potential, in such these cases, the right judgement of Saddle cannot be done; however in these 2 examples, the adjustment has not been used.

- In followin cases, except, even if Saddle is found, it will be treated as an exception.
  - Point adjacent to the stone
  - Point positions at 2 or less heigh

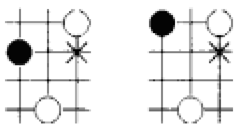
6.1.11 Sphere Move Candidate

The candidate develops MOYO and make it turn into Sphere, or privent such a delopment. However currentally our system does not recognize the sphere, hence it just produces the candidate at the midpoint between 2 same colored stones taking fixed distance. In this case, when the sphere really exists, the move almost becomes a good move. But if it doesn't, the move is definitelly a waste. For the group scored with 20 as its strength, following move candidates are produced.

1. Position of Move Candidate

   (a) Invading Sphere or Preventing Sphere from Invading
   This move is produced by the similar method for the enclose/escape move (see 6.1.7).

   (b) Fortifying/Breaking Through Enclosure



● ...Group's Stone
○ ...Group's Enemy 's Stone(Enclosure)
× ...Move Candidate

Fig. 6.26 : Patter of Fortifying/Breaking Through Enclosure

2. Valuation of Move Candidate

$$\text{Move's Valuation} = \min(\text{Size of Enemy Family}/2, 15)$$

   The enemy family is the one includes the group's group's DAME group 's DAME, if the move candidate is for invading sphere or preventing sphere from invading. Or it is the one includes the enemy stone on the pattern, if the candidate is for fortyfying/breaking theough its enclosure. If there are more than one families, the size of family is a half of the average of families' size.

6.1.12 Sphere's Contact Point Move Candidate

This move expands own sphere and also privant the expantion of its enemy's sphere such as "Ryo-Geima" in Japanese .

1. Position of Move Candidate

   Among points positioning at 5 or lower height, if the point is attended by points positioning within Kosumi range and scored 1 or more for their potential and there is no stone within its Kosumi range and are Black/White stones within its Keima range, it is defined as a sphere's contact point and Black/White 2 stones within its Keima range are defined as base points of sphere. Among valious convinations of base points of sphere, the one, the distance between its 2 stones is shorter than Oh-Geima distance, is selected as the move candidate.

2. Valuation of Move Candidate

$$\text{Move's Valuation} = m(Sb) + m(Sw)$$

Sd expresses the size of family including Black base point of sphere and Sw does the one including White base point of sphere And $m(S)$ is a function defined by followings.

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

6.2 Final Decision Making for Next Move

From enumerted various move candidates, the next move is selected by following procedures.

1. Dead Group Filter

   This filter removes candidates belonging to the dead group from the candidate list.

2. Useless Move Filter

   This filter removes obviously useless candidates measured by stones positioning around them from the candidate list.

3. JOSEKI Filter

   This filter removes candidates for the corner, except JOSEKI move and edge move candidate, if JOSEKI is in progress now at the corner.

4. Total

   This process adds up valuations of all candidates, the total of valuation, for the same point on the board.

5. Selecting Largest Total of Valuation

   This process selects a move candidate with the largest total of valuation on the board. If the total is 0, the move will be a pass.

6. Suicide Move Filter

   Assuming the move with the largest total of valuation is played, except a pass, This filter examines possibility of that the move is captured by enemy. If the move is possible to be captured, the total of valuation is cleared and later processes than selecting the largest total of valuation are re-processed. Otherwise, the move is selected as the next move.

7. Resign

   If the result of situation judgement exceeds the resign score, above processes will be nullified and a resign is selected as the next move. For preventing excessive resign, the resign score is currently 361.

7 Developing Tools

The developing tool is independent of main GO program, but such a program cannot avoid the trial and error for most of all improvement on it. Therefor the tool takes very important part in the entire program. The developing tool mainly divides into 3 parts.

- Analysis Mode

  At any time during the game, results of the recognition of position or the decision making for the next move or results or processes of the local area search can be refered in this mode. And changing value of various parameters in the system, its effects are examined.

- Knowledge Editor

  This editor is used to edit various knowledge as data base in the system.

- Evaluation Tool

  This tool is used for the evaluation of results of reprogramming the system.

7.1 Analysis Mode

When the system is waiting for the next move during Game Mode or any time when Board Positioning Editor is running, the mode of system can be switched to Analysis Mode. The following functions are supplied from this mode.

- Data Structure Indication

  This function displays various data structures and these corresponding attributes on the current board situation.

- Local Area Search Procedure Indication

  This function executs specified local area search selected by user and displays this search's process and result.

- Game Status Indication

  This function displays current status of each player.

- Parameter Tuning

  Using this function, vaious parameters in various knowledge of GO for this system can be displayed and modified.

7.2 Knowledge Editor

The knowledge about GO for this system is converted to the data base which will be refered during the game at any time the need arises. The knowledge as data base can be edited by following editors.

- Potential Editor

  The potential is used to measure the rate of a stone's influence on its around. The rate of influence on around is generally inversely proportional to the distance from the stone.

However, by certain reasons, it is not simply applied around the board side. With trial and error, we estimated the potential by setting up the human evaluation as a standard. This editor helps such trial and error and reduces the time cost for that.

- JOSEKI Editor

  GO contains huge varieties of JOSEK, standard patterns of good play in the corner, and new JOSEKIs are still found.
  This editor is used to simplify adding JOSEKIs and managing them.

- TESUJI Editor

  A large amount of knowledge called TESUJI, the most skillful move which gives an advantage in local situation, exists in GO.
  This editor is used to simplify adding TESUJIs and managing them.
  This editor, however, is just finished, therefore putting TESUJIs edited by this editor to the system's practical use is for later on. Watari, Yose are planed to use. The move candidate knowledge which is realized by the program, such as DAME, cut/connect, FUTOKORO, etc, will totally be managed by this tool.

7.3 Evaluation Tool

The trial and error method is essential for the improvement of a program, such as the GO playing system. From this reason, the improved system should be evaluated as accurately as possible. Currently, following tools are prepared.

1. Auto-Test Function

   This function, is a part of Board Positioning editor, evaluates the program by the result of the rate of correct answers as the program solves prepared problems and displays the elapsed time during that. The search problems, such as capture and Tsumego, position recognition problems, such as the group's strength, and decision making problems, such as the next move, can be used for this tool.

2. Game Record Rating Tool

   This tool is used to relatively compare prepared game record and computer's move by the programmer. Such as the program is remodeled, this tool displays the board situation which the program plays a move at different point comparing with the game recode of the program before remodeled. Judging for each board situation by human, the efficacy of remodeling can be evaluated. When a bad move in the certain board situation is corrected, the program often starts to play other bad move in other board situations. This tool is very useful to find such situations.

3. Game Record Editing Tool

   A game record can simply be edited by this tool. The editing is done by placing numbers on the board in any order. A game record could be finished within 5 minutes.

4. Board Situation Editing Tool

A board situation can simply be edited by this tool. The editing is done by placing Black/White stones on the board just as you like. This tool is useful to create problems for Tsumego.

5. Remote Playing Function

   For examining the efficacy of remodeling, actual playing is a reliable method. This function is used to play between different PSIs, or PSI and other computer's GO program. The former is useful for playing between old version program and new version one, and different tuned programs. The latter is useful for playing with the commercial GO program with the communication function.

6. Elapsed Time/Memory Usage Measurement Function

   For the programe, such as GO, to limit the playing time is very important. Finishing a game within 30 minutes is this system's target. For limiting a playing time, the amount of knowledge installed to the system is also limited. Hence limiting a playing time is strongly relating with the strength of the program. This tool measures elapsed time and memory usage of the program's each process. This tool is useful for remodeling with concerning a playing time and the efficacy.

8 Parallel Playing System

8.1 Intermediate System

The intermediate parallel GOG system consists of the parallel code that runs on the Multi-PSI and the sequential code that runs on the front-end processor (FEP).

In the intermediate system, one of the processors of the Multi-PSI serves as a manager processor, and the rest are worker processors. Each worker processor maintains a local copy of the board, and updates it each time the manager processor notifies a new move. This reduces interprocessor communication, since the master processor needs only to specify which string is the target of the search.

The system configuration is shown in Fig. 8.27. In this partly parallel GOG system, the front-end sequential GOG system notifies the enemy's moves to the Multi-PSI manager processor. The manager processor dispatches the capture search task to the worker processors. (The capture search tasks are generated only by the dangerous string in order to keep executing time short.) When a capture search is completed, a worker processor requests for a next capture search task. After all capture search tasks of dangerous strings have been dispatched, a particular kind of plausible move generation tasks are dispatched. Those are *keshi* candidate moves that may restrict the enemy's potential territories. The results are sent to the manager processor and then to the front end processor. After the front-end sequential GOG system have received the candidate moves from the Multi-PSI, it evaluates them with the candidates generated by itself. It then decides the next move.

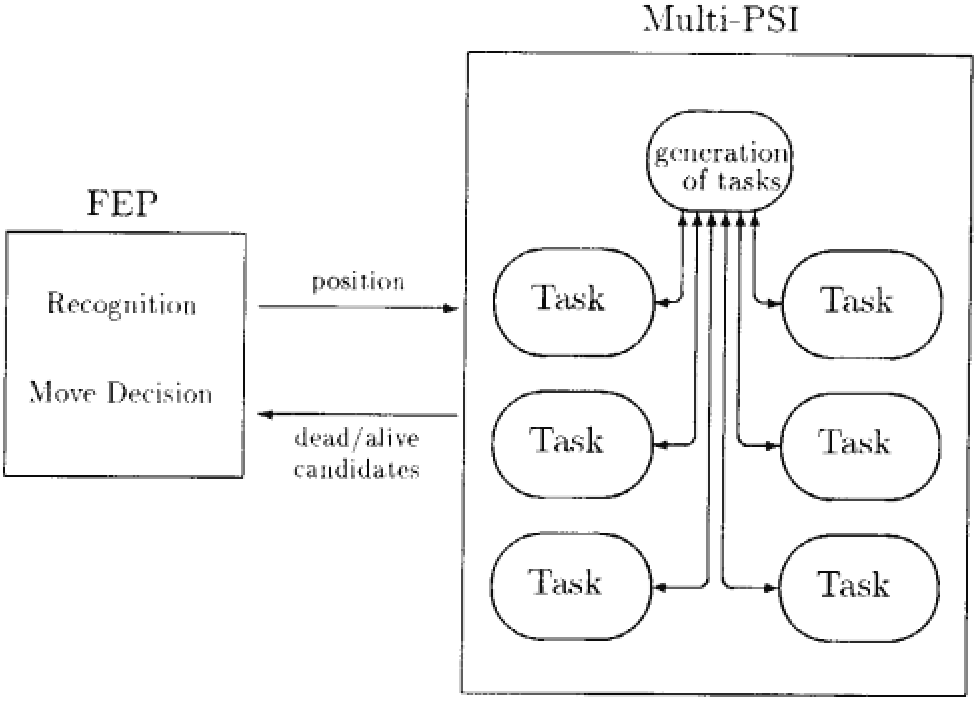


Fig. 8.27 : System Configuration

8.2 An Experiment of a Load Balance Technique

Good load balance between processors is key to high processor utilization. The dynamic load balancing technique is one of way to realize it. It is that the manager processor dispatches tasks to worker processor which are detected to be idle. But, even if we use the technique, load imbalance is caused by uneven granularity.

To alleviate this problem, we devised the following technique. At first, we classify all of tasks into two groups: a task group with larger granularity and a task group with smaller granularity. Larger tasks are given higher priority than smaller tasks. Therefore after all larger tasks have been dispatched, smaller tasks are dispatched. Then the smaller tasks tend to smooth out the load imbalance caused by uneven granularity of larger tasks (Fig. 8.28). This makes processor utilization rate higher.

We tested this technique on the experimental GOG system. In the intermediate system, the larger-grain task group consists of capture search tasks, and smaller-grain task group consists of keshi candidate move's, because the former tasks are usually much larger than the latter.

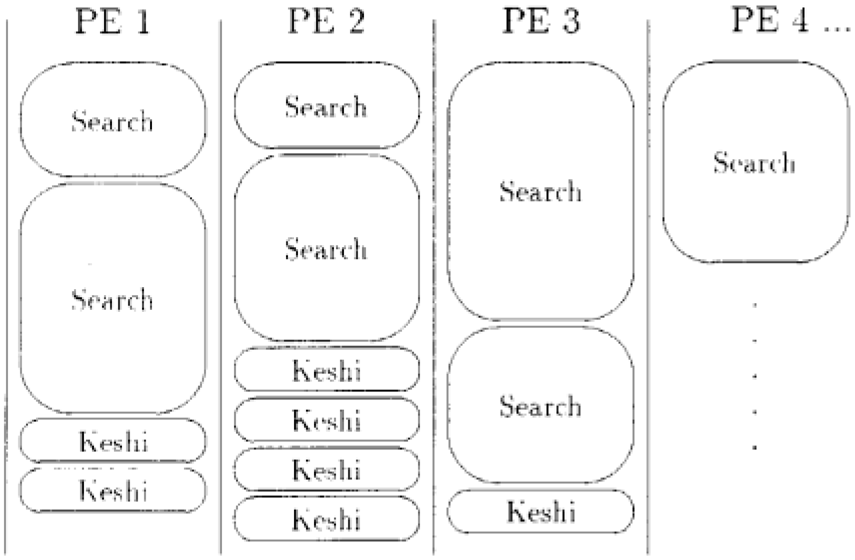


Fig. 8.28 : The smaller tasks tend to smooth out the load imbalance

In general, this technique can be employed when (relative) task sizes can be roughly estimated in advance.

9 Evaluation of Current System and Future Development

This system has been developed as a part of intermediate stage of the Fifth Generation Computer System Project and we aimed to reach 5 kyu as the system's strength. In 1988, the system's strength reached 10 kyu level. Since then, the strength improves 1kyu every year by remodeling. Currently, the strength, however, has downed to 10 kyu level. The reason is, after remodeling in the beginning of this year, we realized the system won't improve any further by changing details, therefore we started the system's full-remodeling in the second half of this year and it hasn't been finished yet. The full-remodeling is midway, because some adjustment should be done among knowledges.

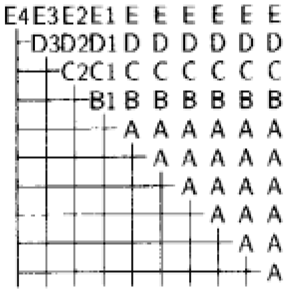
Since 1988, we have been converting the current system into a parallel implementation system, and in this year we start on a development of the full-parallel implementation GO system.

By adding more knowleges following the increase in computing power and finishing the full-remodeling explained in the first part of this section, we are expecting great improvement of the system's strength.

A Potential Pattern Knowledge

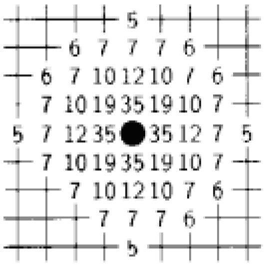
- Variations of Poteitial Pattern

  Based on the basic coordinates, following 15 variations are prepared.

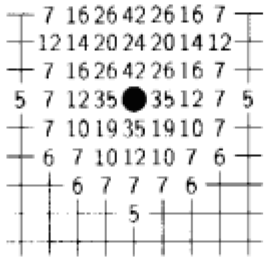


- Potential Pattern

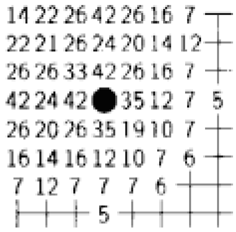
  The potential of point occupied by a Black stone is scored 100.



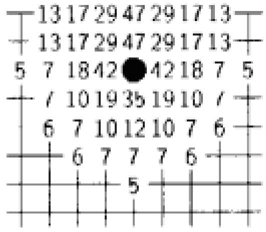
Pattern A



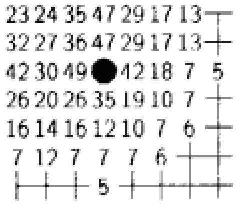
Pattern B



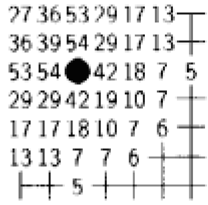
Pattern B1



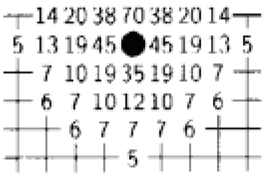
Pattern C



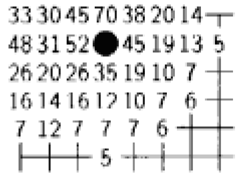
Pattern C1



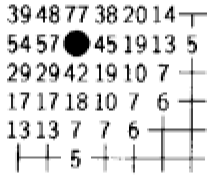
Pattern C2



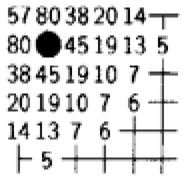
Pattern D



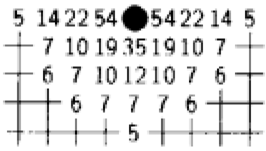
Pattern D1



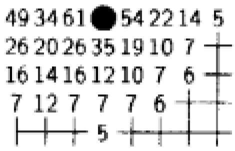
Pattern D2



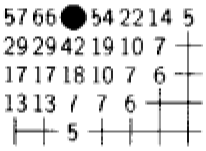
Pattern D3



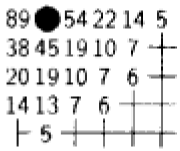
Pattern E



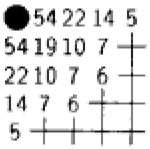
Pattern E1



Pattern E2



Pattern E3



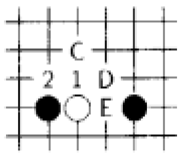
Pattern E4

B Cut/Connect Move Candidate Pattern Knowledge

● ···Linkage's Edge Point
○ ···Peeping Enemy Stone (except dead string's stone)
× ···Move Candidate for both side
b ···Black's Move Candidate
w ···White's Move Candidate
Number···Move Candidate with Condition for both Black/White

[Niken **a**]

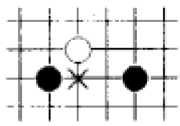
If A is a White stone, B is a Black stone adjacent to A,



Valuation = 15 points

- if no white stone is occuping at 1, 2, C, D, and E, a candidate will be
  1 ··· only if A's DAMEs ≧ 3 and B's DAMEs ≦ 2.
  2 ··· other than above.
- if a white stone is occuping 1, 2, C, D, or E,
  1, 2 won't be selected as candidates.

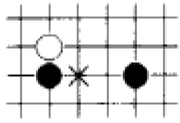
[Niken **b**]



Valuation = 15 points

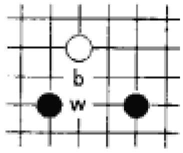
[Niken **c**]

If A is a Black stone adjacent to a White stone,



- if A's DAMEs ≧ 4,
  Valuation for Black's Move Candidate = 5 points
  Valuation for White's Move Candidate = 0 points
- other than above,
  Valuation for Black's Move Candidate = 15 points
  Valuation for White's Move Candidate = 0 points

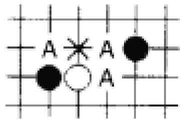
[Niken **d**]



Valuation = 15 points

[Oh-Geima(Large Knight Move) **a**]

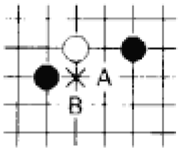
Only if all As are not occupied by White stones.



Valuation = 18 points

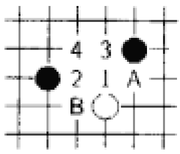
[Oh-Geima **b**]

Only if A is not occupied by a White Stone.



- if B is a Black Stone,
  Valuation = 5 points
- other than above,
  Valuation = 18 points

[Oh-Geima **c**]



- if A and B are not Wite stones,

Candidate's Position	1	2	3	4
Black's Valuation	15	15	15	15
White's Valuation	15	15	18	15

- if B is a White stone but A is not,
  1. 2. and 4 will be candidates scored with 15 points as their valuations
- if A is a White stone but B is not,

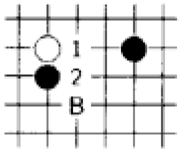
Candidate's Position	1	2	3
Black's Valuation	15	15	15
White's Valuation	15	15	18

- if A and B are White stones,

  3 will be a candidate scored with 15 points as Black's Valuation and 18 points as White's

[Oh-Geima **d**]

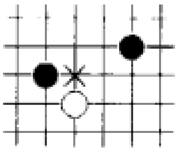
If A is a Black stone adjacent to a White stone, a candidate will be



1 ··· only if B is a Black stone
2 ··· only if B is not a Black stone

- if A's DAMEs ≧ 3,
  Valuation = 10 points
- other than above,
  Valuation = 20 points

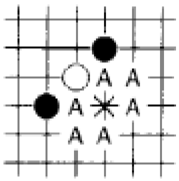
[Oh-Geima **e**]



Valuation = 15 points

[Hazama **a**]

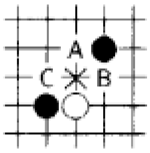
Only if all As are not occupied by White stones.



Valuation = 10 points

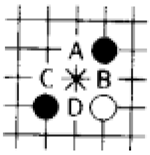
[Hazama **b**]

If A or B and C are not occupied by White stones.



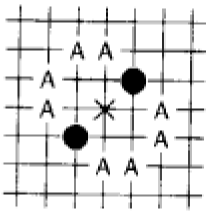
[Hazama **c**]

Only if A or B and C or D are not occupied by White stones.



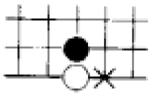
Valuation = 15 points

[Hazama **d**]



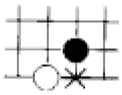
- if A is occupied by a White stone,
  Valuation = 15 points
- other than above,
  Valuation = 5 points

[Perpendicular 2 **a**]



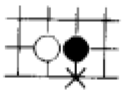
Valuation = 5 points

[Perpendicular 2 **b**]



Valuation = 5 points

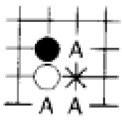
[Perpendicular 2 **c**]



Valuation = 5 points

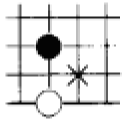
[Perpendicular 3 **a**]

Only if all As are not occupied by White stones.



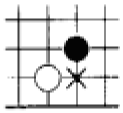
- if × positions at height 2 or higher,
  Valuation = 15 points
- other than above,
  Valuation = 1 point

[Perpendicular 3 **b**]



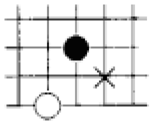
Valuation = 15 points

[Perpendicular 3 **c**]



Valuation = 15 points

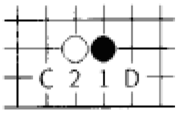
[Perpendicular 3 **d**]



Valuation = 15 points

[Perpendicular 3 **e**]

If A is a White stone, B is a Black stone,

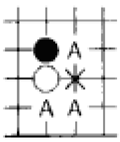


Valuation = 15 points

- if C is a White stone and D is not occupied by a Black stone,
  1 will be selected as a candidate.
- if A's DAMEs ≧ 3 and B's DAMEs=2,
  1 will be selected as a candidate.
- other than above,
  2 will be selected as a candidate.

[Perpendicular 4 **a**]

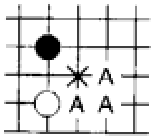
Only if all As are not occupied by White stones.



- if × positions at height 3 or higher,
  Valuation = 15 points
- other than above,
  Valuation = 1 point

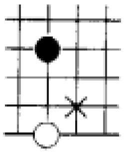
[Perpendicular 4 **b**]

Only if all As are not occupied by White stones.



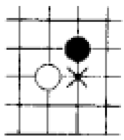
Valuation = 15 points

[Perpendicular 4 **c**]



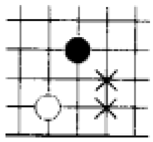
Valuation = 15 points

[Perpendicular 4 **d**]



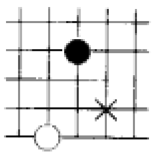
Valuation = 15 points

[Perpendicular 4 **e**]



Valuation = 15 points

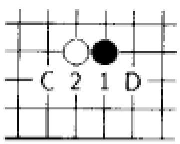
[Perpendicular 4 **f**]



Valuation = 15 points

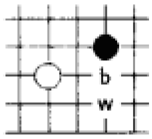
[Perpendicular 4 **g**]

If A is a White stone and B is a Black stone,



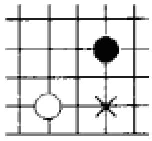
- C is a White stone and D is not occupied by a Black stone,
  1 will be selected as a candidate.
- if A's DAMEs ≧ 3 and B's DAMEs=2,
  1 will be selected as a candidate.
- other than above,
  2 will be selected as a candidate.

[Perpendicular 4 **h**]



Valuation = 20 points

[Perpendicular 4 **i**]



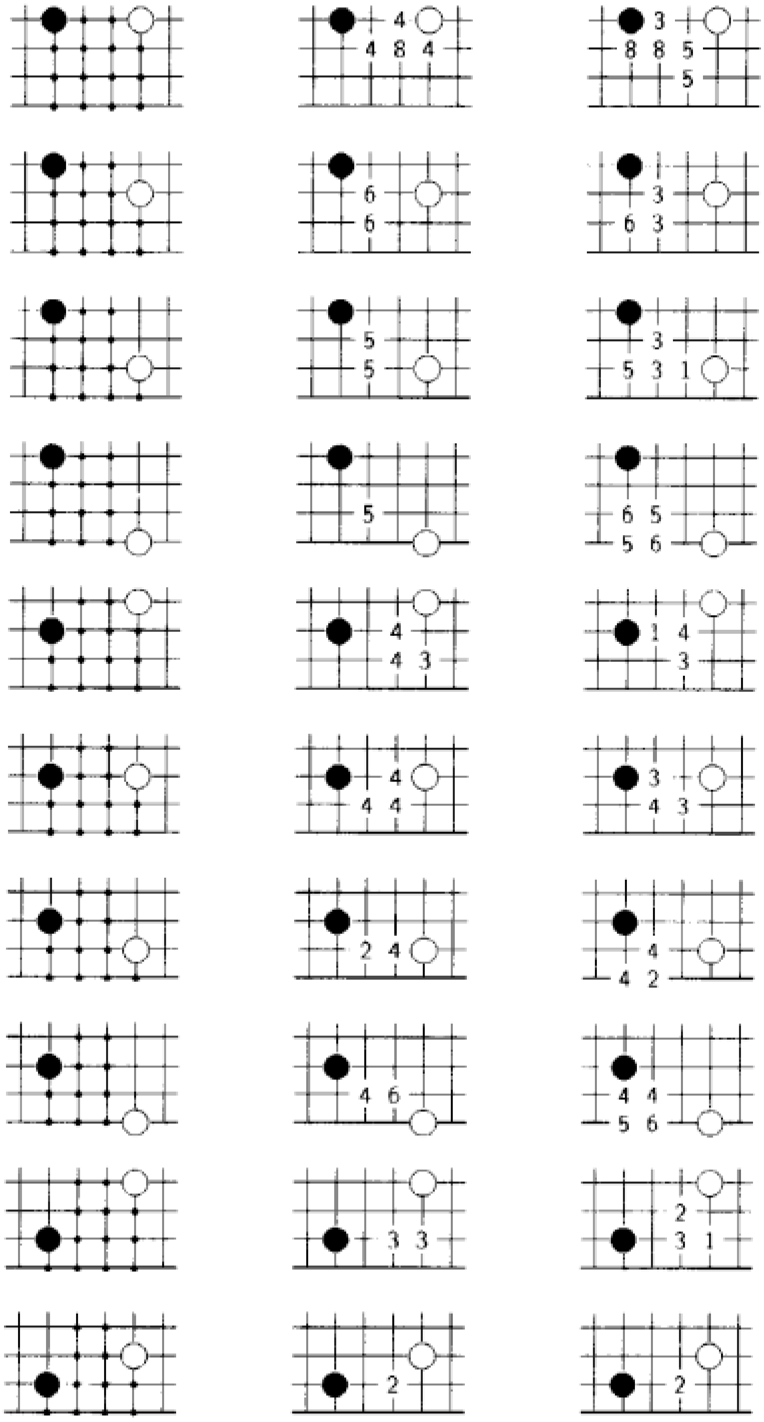
Valuation = 20 points

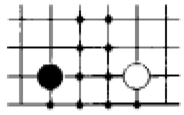
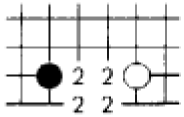
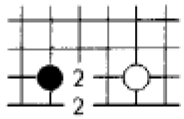
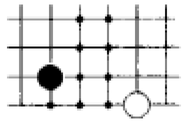
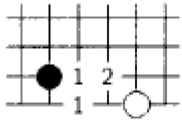
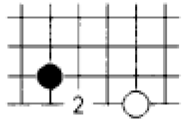
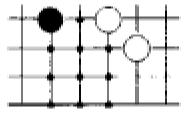
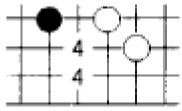
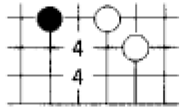
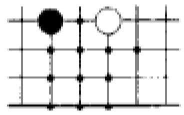
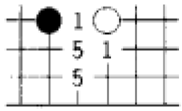
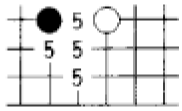
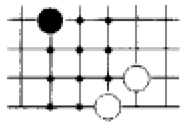
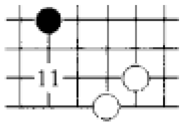
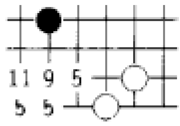
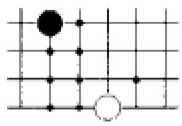
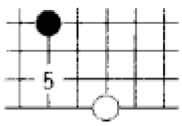
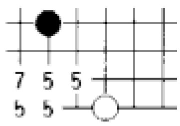
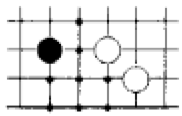
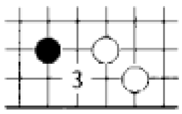
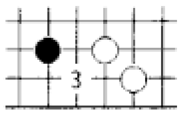
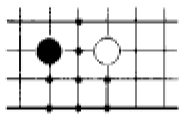
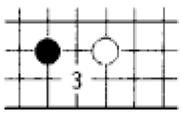
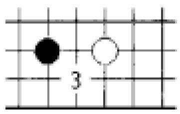
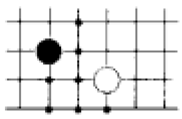
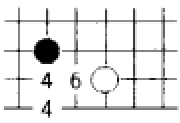
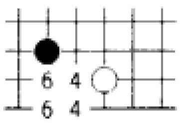
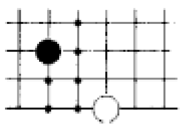
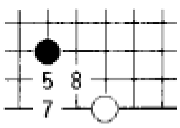
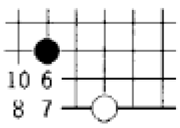
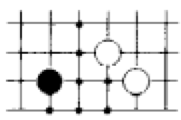
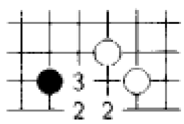
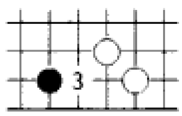
C FUTOKORO Expand/Reduce Move Candidate Pattern Knowledge

Followings are the knowledge of FUTOKORO expand/reduce move candidate GOG has.

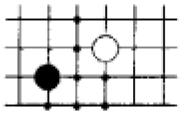
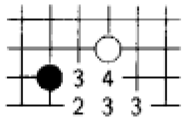
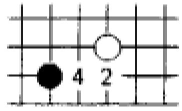
Pattern | FUTOKORO Expand Move by Black | FUTOKORO Reduce Move by White

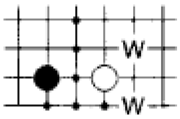
1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

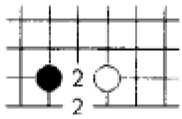
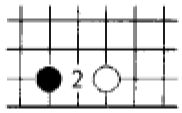


	Pattern	FUTOKORO Expand Move by Black	FUTOKORO Reduce Move by White
11.			
12.			
13.			
14.			
15.			
16.			
17.			
18.			
19.			
20.			
21.			

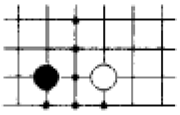
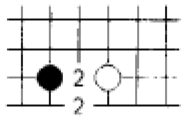
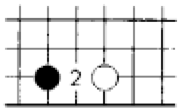
Pattern	FUTOKORO Expand Move by Black	FUTOKORO Reduce Move by White

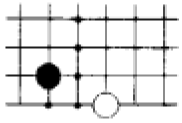
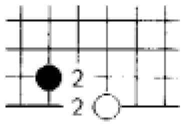
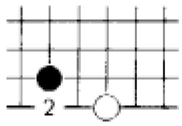
22.   

23. 

if one or both Ws are White.

24.   

25.   

D Tunable Parameter

Followings are various tunable parameters and their default values.

- Data Structures
  - Group
    - * Capturing Race Boundary

      Default Value = 10
    - * Dead Group Boundary

      Default Value = 7
    - * Strength Coefficient

      Default Value = {1.6, 0.8, 6.0, 5.0, 10.0}
    - * Weak Stone Boundary

      Default Value = 20
  - Family
    - * Strength Coefficient

      Default Value = {1.6, 0.8, 6.0, 5.0, 10.0}
    - * Interior Boundary

      Default Value = 7
    - * Family Boundary

      Default Value = 7
- Board Situation
  - Sphere Equivalent Size Function

    Default Value = {0.02, 0.10, 0.21, 0.33, 0.45, 0.55, 0.63, 0.77, 0.83, 0.90, 0.97, 1.00 }
- Move Candidate
  - Edge Move Candidate
    - * Base Point Function

      Default Value = {8.0, 12.0, 16.0, 16.4, 16.8, 17.1, 17.4, 17.6, 17.8, 18.0 }
    - * Attack Base Point

      Default Value = 4.0
  - Yose Move Candidate
    - * Attack Base Point

      Default Value = −1.0
  - Capture Move Candidate
    - * Base Point

      Default Value = 10.0

* Assure Capture Point

  Default Value = 1.0

- DAME Move Candidate
  * Bonus Point[Height 1]

    Default Value = 1.0
  * Bonus Point[Height 2 or higher]

    Default Value = 8.0
  * Base Point Function

    Default Value = {1.0, 5.0, 3.3, 2.5, 2.0, 1.0}
- Cut/Connect Move Candidate
  * Crosscuts Base Point Function

    Default Value = {30.0, 23.0, 20.0, 17.0, 15.0}
- Enclose/Escape Move Candidate
  * Base Point Function

    Default Value = {0.0, 7.1, 10.0, 12.0, 13.6, 14.9, 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 }
- Sphere's Contact Point Move Candidate
  * Evaluation Coefficient

    Default Value = {10.0, 50.0, 5.0}

• Decision Making for Next Move
  - Resign Score

    Default Value = 361

• Search
  - Maximum Move
    * Shicho

      Default Value = {100, 300}
    * Capture

      Default Value = {20, 200}
    * Connect

      Default Value = {20, 300}
    * Tsumego

      Default Value = {30, 10000}
    * Real Eye

Default Value = {20, 10000}

- Game Set Up

  Indication for Purpose of Computer's Move

  Default Value = Yes/No(depends on start up system)

  Indication for Purpose of Human Player's Move

  Default Value = Yes/No(depends on start up system)

  - Pointing Target Object in Analysis

  Default Value = Yes/No(depends on start up system)

- Dispersed PEs(only works when start uped as a FEP for Multi PSI)

  Default Value = Number of requiring Multi PSI's PEs