ICOT Technical Report: TR-717

TR-717

Methodology of GOSEDAI

by

N. Sanechika, H. Oki, T. Akaosugi & S. Sei

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

Methodology of GOSEDAI

Noriaki Sanechika
Electrotecnical Laboratory

Hiroaki Oki
Future Technology Laboratory Inc

Takashi Akaosugi
Shinichi Sei
Institute for New Generation Computer Technology

Abstract

In this paper, we introduce a methodolgy to design a Go program, GOSEDAI: position recognition based on human's perception model; candidate move generation through CASE idea; move evaluation based on the positional variation; the final move decision based on the principle of "focus of attention."

1. Introduction

Full_width search paradigm has shown a brilliant performance in Chess programming. The approach, however, is not effective enough for the game of Go because of its prohibitive large scale of search space. Some innovative methods are required for Go to achieve the same success as that of Chess. Currently Go playing is one of problems that even people with an average ability can behave far better than machines..

To challenge the problem, we, a group in the fifth generation project, have developed a Go program, GOSEDAI[1].
By the way, GOSEDAI = 碁世代 = Go Generation = GOG..
GOSEDAI introduced the following methods:

① " Progressive deepening " recognition of a board position.
② Candidate move generation based on Go specific view points (CASE).
③ Move evaluation by the positional variation based on "deiri"idea.
④ The final move decision based on "focus of attention" principle.

Above four steps in this order form the move decision process of GOSEDAI.

2. Progressive deepening recognition of a position

The term "progressive deepening" was first used by de Groot[2] to represent the idea of selectively extending the main continuation of interest in Chess play. We have noticed that the underlying notion of this term can be extended to a more universal principle of human's recognition process. Clearly, the term "iterative deepening" stems from the prior one and a limited version to $\alpha - \beta$ search. In this paper, we use this term in an more extended and broader sense.

In GOSEDAI, we tried the simulation of human's " progressive deepening" recognition process that seems to support human's excellent pattern recognition ability. In Artificial Intelligence areas, the search paradigm has been a unique universal problem solving methodology. The key operation of search is "generate and test" of candidate moves. Search is surely universal but it easily leads to a combinatorial explosion for most of problems.

Human often takes the " generate and test " approach to solve a problem with no algorithm known, but his way to apply it is never monotonous. Some features of the way are as follows:

- Candidate moves are not generated exhaustively.
- "Generate and test" operations are iterated many times.
- Information gathered through a " generate and test" step is used to control the next " generate and test" step.
- A task involved in a " generate and test" of a deep step is usually more complex or heavier than that of a shallow level.
- The cost of " generate and test " at each step does not vary so much. Here, the cost is the number of candidate moves x test cost.

We call the above process "progressive deepening" (P.D.) recognition. P.D. is a matter of daily life for everyone who is forced constantly to make decisions on various things surrounded by flooding information with incomplete knowledge. One makes most of these decisions subconsciously to adapt oneself to the environments. This decision mechanism is provided as a part of functions of sensory organs such as vision. When P.D. is processed instantly, it will be observed as "focus of attention", well known in psychology.
In GOSEDAI, we tried to realize P.D. for position recognition, especially for processing " group classification."

Data structure: We provided a rather simple data structure considering the overhead required for data modification caused by each turn.
The data structure is composed of five types of objects: point, string, linkage, group and family(territory). Each object has its own attributes. Point, string and linkage objects are somewhat low level concepts and it is easy to calculate their attributes. On the other hand, a group object has attributes related to "death and life" and family object has attributes related to territories such as solid territory and potential territory. So it requires considerable amount of cost to obtain accurate values of these attributes. Particularly the state of a group is usually determined in the context of relative relationship with surrounding groups. So it cannot be decided in one pass but only after reiterated passes, that is, P.D..

Fig.1 Classication of a group G

```
① if all strings in G  are dead, then  <completely dead> → exit.
② if G has more than two eyes, then  <completely alive> → exit.
③ if G has many inner points, then < area_stable>→ exit.
④ if G is in the open area, then   <extensible>.
⑤ if G is besieged loosely, then  <escapable>.
⑥ call death & life routine,
     if alive even at the second turn, then <self_alive> → exit.
     if dead even at the first turn, then <self_dead>.
     if alive at the first and dead at the scond, then <neutral>.
⑦ if G is in the semeai state, then call semeai routine.
     if win even at the second turn, then <semeai_win>→exit.
     if dead even at the first turn, then <semeai_lose>→exit.
     if win at the first and lose at the scond, then <semeai_neutral>
                                                                 →exit.
⑧ if G is not extensible and not escapable and not neutral
                                and besieged tightly  then <dead> → exit.
⑨ tentative
```

The classification procedure from ① to ⑨ is applied to each group. If any group turns out to be dead after a procedure, it is iterated with that new information.

3. Candidate move generation

GOSEDAI generates its candidate moves from several Go-specific view points(called CASE). A CASE is a pattern-driven move generating rule. It consists of a scope, an intention, candidate moves(or a move-tree) and local move value. We provided a few dozens of CASEs: group related CASEs, territory related CASEs and a few auxiliary CASEs. For example, [death & life] , [semeai] , [besiegement] are group related CASEs; [extension] , [enclosure] , [invasion] are territory related CASEs; [caputure] , [connection] , [shape] are auxiliary ones; [joseki] , [edge] are knowlege oriented CASEs.

Candidate moves are generated at various timing, that is, before, during and after a position recognition. The reason move generating timing is so irregular is that heavy tasks should be postponed as late as possible and then it could be canceled if it turns out to be unnecessary. A CASE move is proposed only looking within its scope. Therefore sometime a CASE move becomes meaningless owing to the state of outside the scope. For frequently occurring troubles, we can cope with them by providing exceptional check mechanism.

Move values served by a CASE are meaningful only when they are compared among moves generated in the same CASE because each CASE has its own value unit.
The role of an auxiliary CASE is to compensate and to enhance other main

CASEs: when a move A from an auxiliary CASE and a move B from other main CASE happen to occupy the same point or to be close to each other, the value of B is enhanced or the point position of B is adjusted.

4. Evaluation of candidate moves

All CASE moves have to be reevaluated by the common value system because each CASE has its own value unit. It is reasonable to use "moku" (point) as value unit. That is, the value of a move is measured by the estimated number of points that is secured by the move. An estimated value of a move is composed of the follonwng components.

First component: the secured profit by the move.
Second component: the newly generated advantageous points
Third component: the lost points

Actually, the contents of each component varies according to the type of a CASE.

・group related moves: [death & life] , [semeai] , [besiegement]
1st component: the size of the target group x the variation of survival probabilities
2nd component: the size of neighboring opponent groups x the variation of survival probability
+ the variation of potential territory

・territory related moves: [extension] , [enclosure]
1st component: the variation of target territory(potential territory)
2nd component: the size of neighboring opponent groups x the variation of survival probability

・knowledge oriented moves: [joseki] , [edge] , [yose]
The value of a knowledge move is given directly by the designer.

A survival probability is a value of the experimentally designed function in term of vitality of a group. A vitality parameter is a linear combination of the inner area size and the degree of siege of the group, revised in consideration of the results of group classification. We show a relationship between vitality, state of the group and survival probability in Fig. 2.

Fig.2 Vitality and Survival probability

vitality	state of a group	survival probability
0	completely dead	0
1～ 3	almost dead	:
4～10	critical	50
11～19	tense	:
20～24	unstable	:
25～34	almost stable	100
35～40	stable	100
41～	completely stable	100

Evaluation of a potential territory is obtained by converting the potential value to the equivalent number of points(mokus).

We have still several problems about move evaluation;

- an effective means often contains more than one move.
- a problem of sente and gote (planning problem)
- how to find out effective means
- how to evaluate a means

5. The final move decision

If candidate moves are generated and evaluated, then it seems natural to select a move with the maximal value as the final move. There are some problems: the value of a move is not so accurate as expected and some candidate moves are not so proper.

We can obtain a more accurate move by adjusting these uncertain moves from a more macroscopic strategic point of views. If you can focus battle fields on the board, then you can concentrate yourpower to those places and can avoid the disturbance from outside the places. As a method to focus the battle fields, we introduced a measure parameter, urgency. An urgency parameter is designed differently for groups and territories.

An urgency of a group is a value of a function in terms of the vitality, which is designed experimentally. An urgency for a territory is a value of another function designed in a similar manner in terms of the size of the territory. Besides urgency, we can use the position of the opponent's last move and the global judgement of the situation of the game(keiseihandan).

We have various ways to combine these elements, but the point is to employ the concept of " focus of attention."

6. Conclusion

Through a Go programming, we have tried to model human's decision making process in the situation surrounded by flooding data, with limited information processing capacity and limited knowledge. This function is called " progressive deepening" or " focus of attention ", which we experience in our dairy life. Learning is one of most important human functions that we could not introduce in GOSEDAI.

We can expect an AI system truly worthy of its name by combining the P.G.function and learning ability.

References

[1] Sanechika,N. etal." A Technique of Go Playing System "GO Generation"", ICOT Technical Memorandum:TM-0618. 1988

[2] de Groot,A.D.,"Thought and Choice in Chess", Mouton, The Hague, 1965.