Comparison of the Opening Strategies of Go: PON Based vs. Case Based

TAJIMA Morihiko, SANECHIKA Noriaki

National Institute of
Advanced Industrial Science and Technology
Tsukuba Central 2, 1-1-1, Umezono,
Tsukuba-shi, 305-8568 JAPAN
tajima.m@aist.go.jp, sanetika@sepia.ocn.ne.jp

Abstract. This paper deals with the problem of *fuseki.* It is a very difficult problem to make correct fuseki when the opening book cannot be applied. The method, which evaluates candidates by the estimation of groups based on the possible ommission number (PON), has been developed, and its effectiveness has been shown. This paper compares the method with a method based on *case* in order to reconfirm its effectiveness and shows that the combination of the case-based method and the PON-based method is promising for finding the best candidate.
Keywords: computer Go, evaluation function, fuseki, possible omission number (PON), case based

1 Introduction

The game of Go as a subject of computer games in the field of artificial intelligence bocomes the focus of attention. However, it is well known that it is very difficult to develop a human-expert level computer program to play Go by conventional search oriented techniques and that various problems should be solved [1]. It is pointed out that especially the understanding and the evaluation of positions are extremely hard for machine [2].

This paper deals with *fuseki*, the strategic placing of Go in the opening. In a sense, fuseki is the most difficult problem. In the end game, mathematical analysis can be employed. And in the middle game, conventional search techniques can be partially applied. Whereas in fuseki, it is necessary to deal with the vague arrangement of stones which seems to be effective in the later phases. And it is generally hard to evaluate the arrangement of stones because of the large time interval between the opening moves and the final result of the game. Most playing systems try to solve this problem by a database of opening book which is a kind of pattern knowledge. However, there is so huge variation of the board in the opening of Go that the opening book cannot cover all cases. Some heuristics is employed when the book is left, but there is no good heuristics.

In order to overcome the difficulty, Ricaud[3] proposed a method, which separates the real level and the abstract level. It generates candidate moves in the abstract level first, and then confirms them in the real level. We proposed another method based on the possible ommission number (PON). It is a method based on the idea that estimates the strength and the size of the dominated area of groups by their PON and makes the total of their products to evaluate the position generated by a candidate. We have reported the effectiveness of the method from the comparison between the method and a popular method based on a kind of influence function which expresses the influences of stones to their surroundings [4].

Generally it is difficult to evaluate a game playing system. Although it should be evaluated by real matches with many other playing systems after all, comparison of its strategy with other strategies is also needed. This paper compares our strategy with another strategy. Sanechika[5] proposed a method based on *case* and developed a playing system based on the method. The strategy divides a position into meaningful situations which are temporal (opening/ middle game/ end game) or spacial, recognises each of them as a typical case, makes each case recommend (a) candidate(s), and chooses the best candidate among the all candidates. This paper compares the opening strategy based on PON with that based on case and observes the difference between their characteristics of the strategies.

Below in this paper, the strategy or the system based on PON is denoted by "SP", and that based on case by "SC".

We show the outline of SC in Section 2, the outline of SP in Section 3, and our experimental results comparing SP with SC in Section 4. We discuss the results and characteristics of each strategies in Section 5, and conclude this paper in Section 6.

2 The case based strategy

SC enumerates candidate moves and chooses the best move among them on the case based strategy. Cases are also utilized to detect the intention of the opponent's moves. SC generates various kinds of candidates by making each case recommend (a) candidate(s) [5]. The spacial cases that SC has are the following:

- *fuseki* cases
  - book move
  - edge move
    a move on an n-th line ($n \leq 4$) exept corners
  - edge-boundary move
    a kind of edge move made between two edge-stones of different colours whose horizontal [1] distance is two or three.
- weak group cases
  - life & death move
  - *semeai* move (or battle move or capturing race move)
  - surrounding/blockade move
  - move connecting/disconnecting to/from remote group
  - move connecting/disconnecting to/from adjacent group
- region cases
  - *moyo* move (or potential territory move)
  - semi-territory move
    a move to a semi-territory
    - intrusion move
      a move reducing a semi-territory
    - invasion move
      a move jumping into a semi-territory
- contact fight cases
  - contact move
    a move to a liberty
  - capture/escape move
  - cut/connect move

[1] The word 'horizontal' here means 'parallel to the edge'.

Among them, the contact fight cases are used as auxiliary cases to the other cases. The principal objective of introducing the case based strategy is not covering all the cases or strictly dividing of the board, but neglecting the nonurgent cases in early stage by rough estimation in order to omit unnecessary calculation, concentrate on urgent cases, and increase the accuracy of making moves. However, we utilize only the aspect of candidate generation of the strategy in the experiment of this paper, since the principal objective is not directly related to this paper

Each candidate is given a value of evaluation. Each value is optimised as much as possible by various situations being considered. It is the sum of the default value given by the correspoinding case and the bonus point according to a situation(s). As for situations, the arrangement and the degree of life of surrounding groups, etc. are considered. In principle, the strategy of candidate evaluation makes the best effort in observing the local view. The candidate with the highest score is selected among all candidates generated.[2]

The opening phase of SC is from the beginning to the 30th ply at most. The case named "opening case" limits the candidates to the moves on corners and edges (from 1st to 4th lines) where life can easily be secured. It does

[2] In the real playing system, a candidate chosen as the best move might be the candidate other than that of the highest value, since some strategic procedures are also appended in the system.

not include the moves for the purposes like capturing race and *moyo* expansion which are often seen in the last stage of opening.

3 The PON based strategy

The possible ommission number (PON) was defined as follows [6].

[Definition] possible omission number (PON)

Consider a group G of color C.

(a) Group G is neutral (i.e. life and death depends on the next turn).
PON of G is 0.

(b) Group G is alive.
If G becomes neutral after n opponent moves in a row but G is still alive after $n-1$ consecutive opponent moves, PON of G is n.

(c) Group G is dead.
If G becomes neutral after C is allowed to make n consecutive moves but G is still dead after $n-1$ consecutive moves, PON of G is $-n$.

PON can be approximately calculated for open groups which appear mainly in the opening and middle games. For example, the next function f is a good approximation function.

$$f(S) = \lfloor 0.33S - 1.96 \rfloor \quad \text{where}$$

$$S = \sum_{i=1}^{4} w_i \sum_{k=\mathrm{e,t,u}} w_k d_{ik}$$

where, d_{ik} is the number of *dame* (liberties) of degree i, kind k (e: edge point, t: once *kosuri*[3] point, u: other kind of dame), where

$$w_1 = 1, w_2 = 0.5, w_3 = 0.35, w_4 = 0.25,$$
$$w_\mathrm{e} = 1.6, w_\mathrm{t} = 0.2, w_\mathrm{u} = 1.$$

[3] If the path from the group to the dame point touches an opponent stone once, the dame point is once kosuri. The number of kosuri is accumulated along the path.

The system SP evaluates a candidate move by evaluating the position caused by the candidate. A position is evaluated by the evaluation function which totalizes the estimated territories of both sides. Each estimated territory is the multiplication of the strength of the group calculated with its PON and the PONs of its surrounding groups and the size of the area dominated by the group. ("Group" is defined as the set of stones which cannot be disconnected by opponent. It is almost the same as the union of "group" and "chain" defined by Chen[2].) The evaluation function has been optimised using the first 50 problems of the 100 problems in the collection of choose-one-among-four problems [7] (See [4] for the details.).

4 Experiment

In the experiment, we took the following steps for each problem in the collection of choose-one-among-four problems [7]. Each problem has four choices. Each choice of the problems is assigned a score from 10 (the correct answer) to 3 (the most poor answer(s)). The number of hits and the total score can be obtained using the scores.

1. **Enumeration of all candidate moves**
   Generate all candidate moves by using the function of the candidate generation of SC, i.e. enumerate all candidates each of which is recommended by the corresponding case.

   In the experiment, 67 candidate moves were generated for a problem on the average. But each case generates candidates independently, so some candidates may be generated more than once. The number of candidates was 38 on the average when duplication is removed.

2. **Making the correspondence of the choices to the candidates**
   Check whether the same candidates as the four choices given by each problem are included by the set of the candidates

generated by SC or not (We call such a candidate *equal* candidate hereafter). If a choice is not included, then check whether a candidate, which can be regarded to have substantially the same effect, exists nearby or not (We call such a candidate *equivalent* candidate hereafter).

In the experiments, a problem had 2.4 equal candidates and 1.2 equivalent candidates on the average. In three problems, the generated set had no choices. The total was 3.6 on the average, which is about 90% of the choices.

3. **Making them solve the problems**
Make SC and SP solve the problems with equal candidates and equivalent candidates. Correct answers cannot be given by any one of the limited choices in some problems, since some of the four choices may not be generated.

In the experiment, SC generated the equal candidate for 69 problems and an equivalent candidate for 25 problems, i.e. 94 problems had the chance to be solved correctly.

Table 1 shows the scores by SC and by SP for each problem. The unparenthesised values show the scores when only the equal candidates are used to answer, and the parenthesised values show the scores when the equivalent candidates are also used. The cases where none of the four choices are generated are denoted by "-".

When the equivalent candidates are also used, the score usually increases but sometimes decreases. For instance, the scores both by SC and by SP increase in Q1, the both scores do not change (The correct answere is given.) in Q2, no equal candidates but some equivalent candidates are generated in Q3, and the score by SP decreases in Q4 when a equivalent candidate is also used.

Table 2 and Table 3 show the number of hits and the total score for each strategy, respectively. The meanings of parenthesised values and unparenthesised values are the same as in Table 1. If none of the four choices are included in the set of candidates, the score for the problem is set to be 0.

The following can be observed.

- **Advantage of the strategy based on PON**
It is clear that the performance of SP is better than that of SC in both cases, the case of choosing among equal candidates and the case of choosing among equal or equivalent candidates. When comparing the result of the second half of the collection of problems, the numbers of hit are 19 (by SC) vs. 28 (by SP), the total scores are 346 (by SC) vs. 392 (by SP). The differences increase when the equivalent candidates are also used. The correspondence table, which interprets the total scores as the stages (kyu or dan) of Go, attached in the collection of problems shows that SP is on a level of 2 dan.

- **Generation of the choices**
As seen before, the rate that the four choices, which are given in the collection of problems, are included in the set of candidates is about 60%, and it is 90% even when the equivalent candidates are also counted. The rate is not very good. The rate that the correct answer is included in the set of candidates in the first half of the collection of problems is different from that in the second half. The number of such problems is only 29 among 50 in the first half, while it is 40 among 50 in the second half. The reason is as follows: The number of stones on the board is rather small in the problems of the first half of the collection, while it is rather large in those of the second half. Therefore, the candidates recommended by the cases increase in the second half.

- **The use of equivalent candidates**
It is natural that the result depends on whether equivalent candidates are also

Table 1. The scores: by SC vs. by SP

shown as "only equal candidates (equal or equivalent candidates)"

problem No.	1	2	3	4	5	6	7	8	9	10
SC	5(10)	10(10)	-(10)	10(10)	6(6)	3(3)	5(5)	8(8)	7(7)	5(5)
SP	7(10)	10(10)	- (7)	10(7)	5(3)	7(7)	10(10)	10(10)	5(5)	10(10)
problem No.	11	12	13	14	15	16	17	18	19	20
SC	10(10)	10(8)	10(3)	5(5)	7(7)	5(5)	7(7)	7(7)	4(10)	5(5)
SP	10(10)	6(7)	10(10)	10(10)	7(5)	10(10)	7(10)	5(10)	6(10)	5(10)
problem No.	21	22	23	24	25	26	27	28	29	30
SC	10(10)	5(3)	5(10)	5(5)	6(10)	7(7)	5(5)	3(3)	5(5)	5(5)
SP	6(6)	10(3)	6(6)	10(10)	6(6)	10(10)	4(4)	6(6)	10(10)	10(10)
problem No.	31	32	33	34	35	36	37	38	39	40
SC	7(10)	4(4)	10(10)	10(10)	6(6)	10(10)	10(10)	4(4)	10(10)	- (7)
SP	7(10)	4(10)	5(7)	5(7)	6(6)	10(4)	10(10)	4(4)	10(6)	- (10)
problem No.	41	42	43	44	45	46	47	48	49	50
SC	5(5)	10(10)	5(5)	10(10)	7(10)	5(5)	6(6)	7(7)	3(10)	3(3)
SP	5(5)	7(7)	5(6)	10(10)	5(10)	5(4)	10(10)	7(7)	7(7)	10(10)
problem No.	51	52	53	54	55	56	57	58	59	60
SC	5(10)	10(7)	10(10)	10(10)	10(10)	5(7)	4(4)	10(10)	5(7)	7(5)
SP	5(10)	10(10)	7(7)	10(10)	5(5)	5(5)	10(10)	5(5)	10(10)	10(10)
problem No.	61	62	63	64	65	66	67	68	69	70
SC	10(3)	5(4)	7(7)	10(5)	6(7)	5(4)	- (5)	4(4)	4(4)	5(5)
SP	10(10)	10(10)	7(7)	7(7)	6(10)	5(5)	- (10)	4(5)	10(5)	10(10)
problem No.	71	72	73	74	75	76	77	78	79	80
SC	10(10)	10(10)	10(5)	4(4)	10(6)	10(10)	5(5)	4(4)	10(10)	6(6)
SP	10(10)	6(7)	10(10)	7(7)	10(6)	10(10)	5(5)	10(10)	10(5)	10(10)
problem No.	81	82	83	84	85	86	87	88	89	90
SC	5(5)	10(6)	10(10)	7(7)	4(5)	5(10)	10(10)	3(3)	10(10)	7(7)
SP	5(5)	4(4)	10(10)	3(3)	10(10)	8(10)	10(10)	3(3)	10(10)	10(10)
problem No.	91	92	93	94	95	96	97	98	99	100
SC	5(10)	10(10)	3(5)	7(7)	6(6)	6(6)	5(5)	7(7)	10(10)	5(5)
SP	5(5)	5(5)	10(10)	10(10)	10(10)	10(10)	10(10)	10(10)	10(10)	5(5)

Table 2. The number of hits: by SC vs. by SP

	the no. of hits in Q1-Q50	the no. of hits in Q51-Q100	total number of hits
SC	13 (19)	19 (16)	32 (35)
SP	20 (25)	28 (29)	48 (54)

Table 3. Total scores: by SC vs. by SP

	sum of the scores of Q1-Q50	sum of the scores of Q51-Q100	total of the scores
SC	317 (356)	346 (342)	663 (698)
SP	360 (392)	392 (401)	752 (793)

used to solve the problems or not. If the evaluation function is accurate, the score becomes better when equivalent candidates are also used, since the rate that the sets of candidates include the correct answers increases. If the evaluation function is inaccurate, however, it becomes worse when equivalent candidates are also used, since the number of choices becomes greater. The following can be observed.

- SC
  In the first half of the collection of problems, the score employing also equivalent candidates becomes better, while it becomes worse in the second half.
- SP
  The score employing equivalent candidates improves in both of the first half and the second half of the collection. The improvement is remarkable especially in the first half.

The difference can be explained by the characteristic difference of the problems between the first half and the second half.

– **The difference between the first half and the second half**
It is necessary to pay attention that the function parameters of SP has been optimised using the 50 problems of the first half. Therefore, it is natural that SP made a good result for the first half problems. However, in spite of the fact that SP has no advantage for the second half problems, the number of the hits and the total score for the second half problems is better than that of the first half problems. SC has also the same phenomenon. The major reason is that, as described above, the rate that the correct answer is included in the set of candidates is low in the first half problems.

5 Discussions

The result of SP is better than that of SC. However, it does not mean that SP always makes the correct answer to the problems to which SC makes the correct answer. SP makes a wrong answer to some problems to which SC makes the correct answer. In this section, we observe the characteristics of each strategies in the examples where both strategies make different answers. All positions of the problems are black to move.

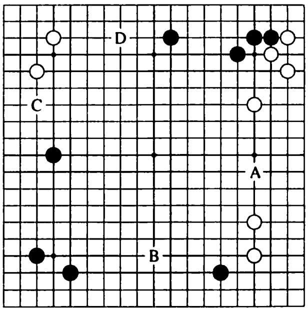

Fig. 1. Q53

The correct answer of Q53 (Fig.1) is B. SC gives the correct answer, but SP chooses D. The candidate B can form a large area named *kakuyoku no jin* or position of crane wings. However, SP is not good at the recognition of large *moyo*.

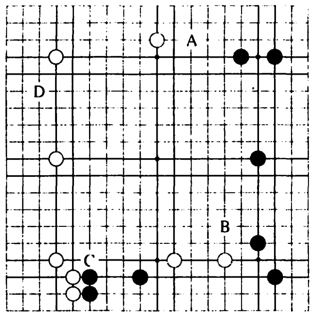

Fig. 2. Q58

The correct answer of Q58 (Fig.2) is C, which is a move to the contact point of a moyo and weak stones. SC gives the correct answer, but SP chooses B. The candidate C reinforces the lower black group and prevents the left white to grow into a moyo, while the candidate B controls the lower white group and expands the right black moyo. SP makes a mistake in the calculation to decide which the better candidate is.

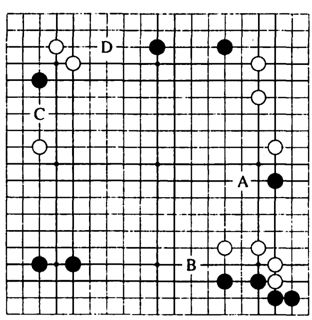

Fig. 3. Q80

The correct answer of Q80 (Fig.3) is A, which is the move to the contact point of weak stones. SP gives the correct answer, but SC chooses B. The candidate A helps the right weak black stone and controls the white four stones below. SP is good at the calculation of the balance between the groups of both sides, while SC is not.

The correct answer of Q92 (Fig.4) is D. SC gives the correct answer, but SP chooses B. The candidate D reinforces the upper left group and prevents the linkage between the left white group and the upper white group. However, it needs accurate recognition of life and death and reinforcement of groups by the linkage of stones. SP cannot cover such problems.

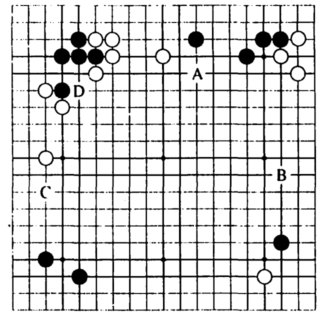

Fig. 4. Q92

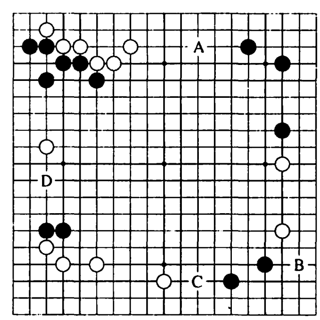

Fig. 5. Q95

The correct answers of Q95 (Fig.5) and Q96 (Fig.6) are D, a move to the contact point of weak groups, and A, a move to the contact point of moyo, respectively. SP gives the correct answer in either problem, but SC chooses A and B, respectively. In either problem, a good move with good balance of both attack and defense should be found. SP calculates it correctly, but the problem is difficult for SC which is fundamentally based on the local judgment.

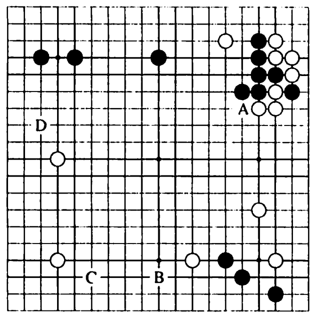

Fig. 6. Q96

6 Conclusion and Future Work

We compared the performance of the strategy based on PON with that based on case by the evaluation using the collection of choose-one-among-four problems. We made each strategy answer the best move for each problem among the choices limited to the set of candidate moves recommended by SC, and we confirmed that, provided that the same set of candidates is given, the performance of the strategy based on PON is better than that based on case. The result showed that the system is on a level of 2 dan. We had confirmed the effectiveness of the method in [4]. So, the result in this paper reconfirmed the effectiveness of the method based on PON.

That SC has less performance than SP means that the evaluation of SC is essentially local and that the compensation considering the surrounding situations has limitations. SP can well evaluate the relative values of candidates because it is free from such limitations, but it still lacks the accurate evaluation of candidates under complicated situations.

Attention should be given to the fact that the enumeration of the candidates has been done by the same way in either strategy, i.e. SP assumes that SC would recommend candidates on the first stage of the calculation. In short, SP depends on the preprocessing by SC. The whole calculation could be done by SP without the preprocessing, i.e. SP could evaluate all possible moves, but it is inefficient and impractical. And its effectiveness on the whole board has not been assured yet. Therefore, the combination of the function, the candidate recommendation by SC, and the strategy of SP seems promising to realize a useful strategy of the opening.

The followings are left for the future works:

- To improve the algorithm of the recommendation of candidates so that it can recommend necessary and sufficient choices,
- To improve the method based on PON so that it can find the best move among the choices at a better rate,
- To study the application of the method to the phases other than the opening phase, since the essential idea of the method is available for all the phases.

References

1. M. Müller. Computer Go: a research agenda, First International Conference on Computer and Games '98, in *Lecture Notes in Computer Science*, 1558, H.J. van den Herik and Iida H. (eds), 252-264, Springer, 1998.
2. K. Chen. Knowledge and search in computer Go, *Proceedings of GPW2001*, 94-101, 2001.
3. P. Ricaud. A model of strategy for the game of go using abstraction mechanisms, *Proceedings of IJCAI'97*, 678-683, 1997.
4. Tajima, M. and Sanechika, N.. An improvement of the method on the strategic placing of stones based on the possible omission number, *IPSJ SIG Notes*, Vol.2000, No. 98, 85-94, 2000.
5. Sanechika N. et al.. The specifications of "Go Generation", *Proceedings of the Game Playing System Workshop*, 73-155, Tokyo, 1991.
6. Tajima M. and Sanechika N.. Estimating the possible omission number for groups in Go by the number of *n*-th dame, First International Conference on Computer and Games '98, in *Lecture Notes in Computer Science*, 1558, H.J. van den Herik and Iida H. (eds), 265-281, Springer, 1998.
7. Nihon Kiin, *New Bessatsu Igo Club*, 31, 1993. (in Japanese)