



**HAL**  
open science

## Generating the Expression of the Move of Go by Classifier Learning

Natsumi Mori, Takehito Utsuro

► **To cite this version:**

Natsumi Mori, Takehito Utsuro. Generating the Expression of the Move of Go by Classifier Learning. 16th International Conference on Entertainment Computing (ICEC), Sep 2017, Tsukuba City, Japan. pp.299-309, 10.1007/978-3-319-66715-7\_33 . hal-01771268

**HAL Id: hal-01771268**

**<https://inria.hal.science/hal-01771268>**

Submitted on 19 Apr 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Generating the Expression of the Move of Go by Classifier Learning

Natsumi Mori<sup>1</sup> and Takehito Utsuro<sup>2\*</sup>

<sup>1</sup> Grad. School of Systems and Information Engineering, University of Tsukuba,  
Tsukuba, Japan

<sup>2</sup> Faculty of Engineering, Information and Systems, University of Tsukuba,  
Tsukuba, Japan  
`utsuro@iit.tsukuba.ac.jp`

**Abstract.** The ancient Chinese board game, Go, with its simple rules yet highly complex strategies, requires players to encircle more territory than their opponent. However, owing to the rise in the capabilities of Go-playing software and a lack of Go instructors in Japan, there is a need for a software that actively assists human players in learning the high-level strategies required to win the game. This study focuses on generating a review for each consecutive player’s move. This paper studies how to generate an expression for each move based on the distribution of the stones on the board. To this task of generating the expression for each move in a game of Go, we apply a classifier learning technique.

**Keywords:** Entertainment Computing · Go · Classifier Learning · Move · Expression

## 1 Introduction

With the recent, well-known success of the computer Go program AlphaGo [4] that beat a high-level professional without handicaps, computer-based Go programs now represent a serious challenge to professional human players. Therefore, from an entertainment and educational perspective, developing software particularly designed to teach human players high-level techniques is an important task. This is an especially pressing task because, at present, Japan has only a small number of professional Go players who are capable of teaching amateurs. Furthermore, there is a need to develop high-level software that can play so-called “customer” Go, where the software analyzes a human player’s habits and predicts their next moves, allowing them to win on purpose while concealing the fact that they won with the aid of the software.

Based on those arguments, this paper aims at developing a software which assist human players to learn high level techniques on the move of Go. Within various functions of such a software of assisting human players to learn the techniques of the game of Go, this paper focuses on the task of generating a review for each move in a game of Go<sup>1</sup>. It is strongly required to develop such a

<sup>1</sup> Yoshikawa et al. [5] studied the protocols of the game of Go with respect to the relations between the player’s skill and the use of Go terms.

software of generating a review for each move in a game of Go mainly because it can assist amateur human players to learn high level techniques on the move of Go. Then, according to such a requirement on developing a software of generating a review for each move in a game of Go, this paper more specifically proposes to generate the expression for each move in a game of Go from the pattern of the stones on the Go board. Furthermore, to this task of generating the expression for each move in a game of Go, we apply a classifier learning technique.

In related research, Ikeda et al. [2] propose generating expressions of the shape of a particular move based on the pattern of the stones on the board by applying a classifier-learning technique. Out of the overall types of expressions of a move, Ikeda et al. [2] concentrated on generating those based on the shape of a move but failed to analyze strategic moves. Thus, one of the major contributions of the present study compared to Ikeda et al.'s [2] research is that it investigates expressions based on the shapes of moves while also taking strategic moves into account. Furthermore, in terms of the features of the classifiers used to generate an expression for each move, in addition to lower-level features, most of which have been studied by Ikeda et al [2] (see section 3.1), the current study illustrates the effects of higher-level features (examined in detail in section 3.2). Finally, in section 4, we show the optimal combination of features for each expression required to achieve the best-possible performance in a game, an aspect that was overlooked by Ikeda et al. [2].

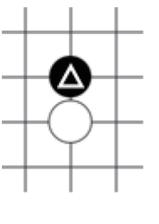
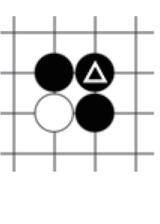
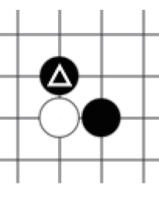
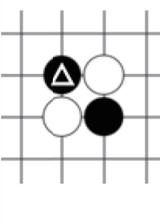
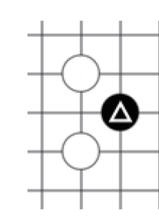
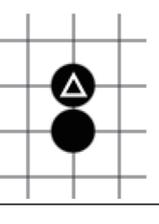
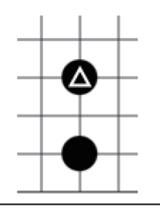
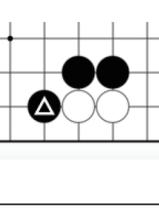
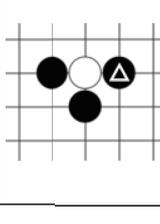
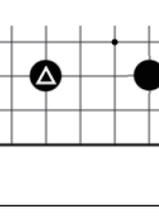
## 2 Expressions of Go Moves

### 2.1 Expressions Based on the Shape of the Move

Expressions based on the shape of a particular move are typically identified by considering the static local pattern of the stones on the board. As illustrated in section 2.3, we focus on the 10 most frequent expressions for the shapes of moves listed in Table 2. For each of these 10 expressions, Fig. 1 shows a typical example of a move; the black stone with a white triangle within it represents the last move with the expression of its shape.

### 2.2 Expression for the Strategic Move

As we mentioned in the previous section, the expressions for the shape of a move are typically identified by considering the static local pattern of the stones on the board. On the other hand, expressions for strategic moves are typically identified by considering a much broader view of the patterns of the stones. Each of these expressions of strategic moves represents fairly characteristic situations encountered throughout the whole-state transition within a game. Moreover, from the perspective of the overall time-series state transition of the moves within a game, a move's time-series sequences can be approximately segmented by strategic moves. Therefore, each strategic move is typically played separately from the previous moves.

1. attachment	2. connection	3. hane	4. cut	5. peep
				
6. extend	7. jump	8. block	9. ate	10. extension
				

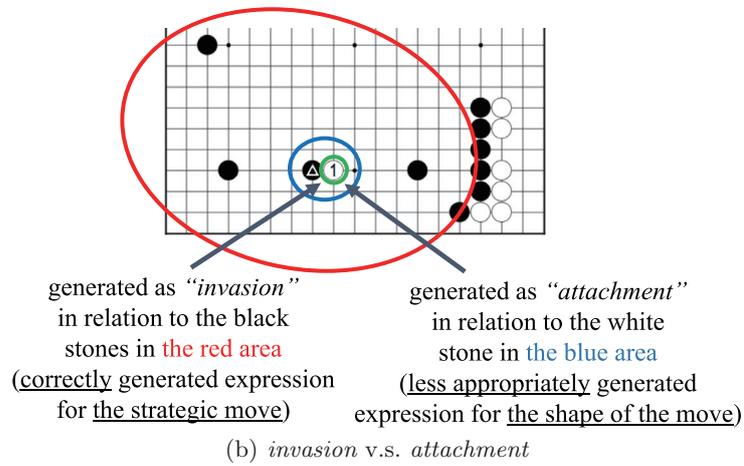
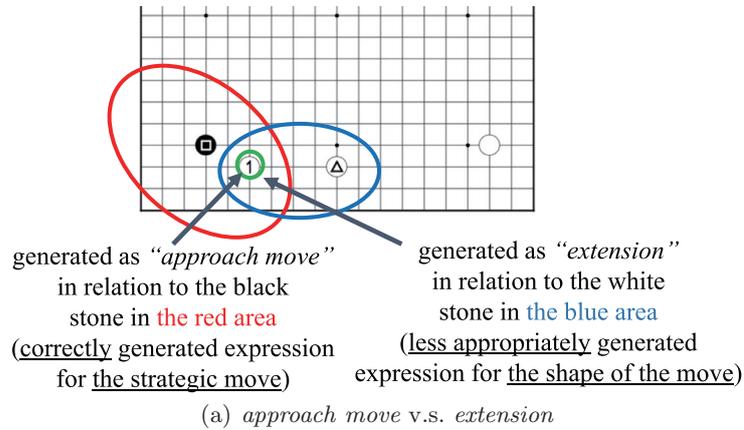
 the last move with the expression of its shape

**Fig. 1.** The most frequent 10 expressions for the shape of a move (expression for each last move)

As mentioned in section 2.3, we focus on the three most frequent expressions for the strategic moves listed in Table 2. A detailed description of each of these three expressions is given below.

- *Protect*—represents that the move by the current player is intended to protect one’s own stones from the opponent using various strategies.
- *Approach move*—represents that the move by the current player is intended to start approaching the opponent’s stone.
- *Invasion*—represents that the move by the current player is intended to invade an area occupied by the opponent.

Fig.2 illustrates examples of two of these three expressions for strategic moves as well as typical situations in which ambiguities against those for the shape of the move are present. In short, a particular move and the pattern of the stones on the board can be interpreted as strategic when considering it more seriously and trying to react to it as much as possible; conversely, it can also be interpreted as simply representing the shape of a particular move when it is considered less seriously, for example, by ignoring the opponent’s move and attempting not to react to it. However, in both of the cases shown in Fig.2 considering the pattern of the stones on the board more seriously and interpreting these moves as strategic is the correct approach.



**Fig. 2.** Examples of the ambiguity of the expressions for *shape* and *strategic moves*

### 2.3 The Dataset

The dataset used to evaluate the classifier for generating the expressions of the moves from the pattern of the stones was constructed from the SGF-move coordinates file as well as the game review text. We first collected these data from reviews of Go games available in newspaper articles over a period of 5 years. This included 60 game reviews played on a  $19 \times 19$  size board and comprising 9,209 moves in total. Then, as shown in Table 1, from each SGF-move coordinate file and the game review text for each move, its order, coordinates  $(x, y)$ , as well as expression were manually extracted. However, from the total number of moves, the expression for each move is available to a certain extent. This amounts to 1,207 moves out of a total of 9,209 moves. Of these 1,207 moves, 238 types of expressions were manually extracted and within this set, only 34 types were

**Table 1.** Generating the training/test dataset of the expression of the Go move from the SGF-move coordinates file and the game review text

Game review text (English translation)	Order of the move	Coordinate ( $x, y$ ) of the move	Expression of the move
...	1	(16,4)	...
...	2	(4,4)	...
...	3	(16,17)	...
...	4	(4,16)	...
...	5	(6,17)	...
In the lower right, only with the single <i>approach move</i> of the 6th white stone,	6	(16,15)	approach move
$\phi$	7	(16,12)	$\phi$
it is unusual to change into the <i>pincer</i> move of the 8th white stone.	8	(8,17)	pincer
Yamada’s move of the 9th black stone is also a careful <i>one-space jump</i> .	9	(14,17)	one-space jump
...	...	...	...

observed more than or equal to 10 times. Then, these 34 types of expressions (with 667 moves in total) were employed as the overall resource for evaluating the classifier for generating the expression of particular moves.

These 34 expressions can be further classified into 21 expressions based on the shape of the move at 478 moves (71.7% of the 667 moves) and the remaining 13 expressions for strategic moves at 189 moves (28.3% of the 667 moves). Then, as shown in Table 2, the 10 most frequent expressions for the shape of the move as well as the 3 most frequent expressions for the strategic move were selected (422 in total) as the positive/negative samples of the training/test dataset.

### 3 Features

The features examined in this study comprise lower-level features aimed at identifying the shape of a particular move as well as higher-level features that seek to identify strategic moves specifically.

#### 3.1 Lower Level Features of a Particular Move’s Shape

Most of the lower-level features examined here have been studied in previous work [2] on generating expressions for the shape of a move.

##### **$FL_1$ : Coordinates of a Move**

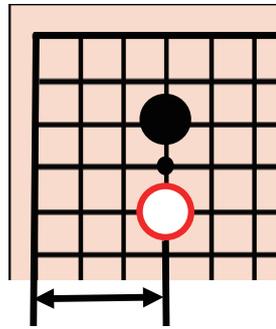
The  $x$  and  $y$  coordinates of a move are used as two features, which are denoted as  $FL_1$ .

**Table 2.** Numbers of positive samples in the dataset for each expression of the moves

Type of move and expression	ID	Expression	# of positive samples
Expression for the shape of the move	1	attachment	60
	2	connection	57
	3	hane	42
	4	cut	37
	5	peep	30
	6	extend	29
	7	jump	25
	8	block	25
	9	ate	21
	10	extension	19
Expression for a strategic move	11	protect	29
	12	approach move	25
	13	invasion	23
total			422

**$FL_2$ : Number of Stones**

The number of stones on the board when a move is made is used as a feature that represents the order of the move from the beginning of the game. This feature is denoted as  $FL_2$ .



distance to the  
closest board edge

**Fig. 3.**  $FL_3$ : distance to the closest board edge

**$FL_3$ : Distance to the Closest Board Edge**

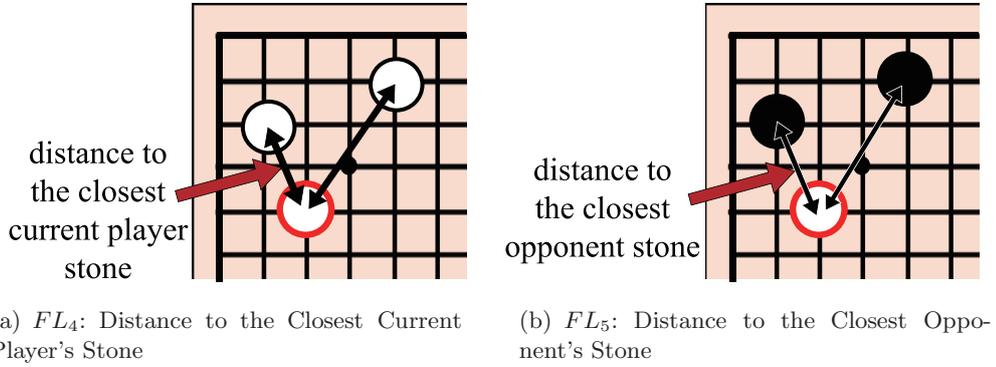
As shown in Fig. 3, the distance from a stone used in a particular move to the closest board edge is used as a feature and is denoted as  $FL_3$ .

#### **$FL_4/FL_5$ : Distance to the Closest Current Player's/Opponent's Stone**

As shown in Fig. 4, the distance from the stone resulting from a move to the closest current player's or an opponent's stone is used as a feature and these are denoted as  $FL_4$  or  $FL_5$  respectively. In terms of the distance measurement used for these features, that used by previous research [1,2] is employed. Thus  $\delta x$  and  $\delta y$  denote the differences of the  $x$  and  $y$  coordinates of the two stones, respectively:

$$\text{distance}(\delta x, \delta y) = |\delta x| + |\delta y| + \max(|\delta x|, |\delta y|) \quad (1)$$

In our implementation of these features, feature values over six are ignored.



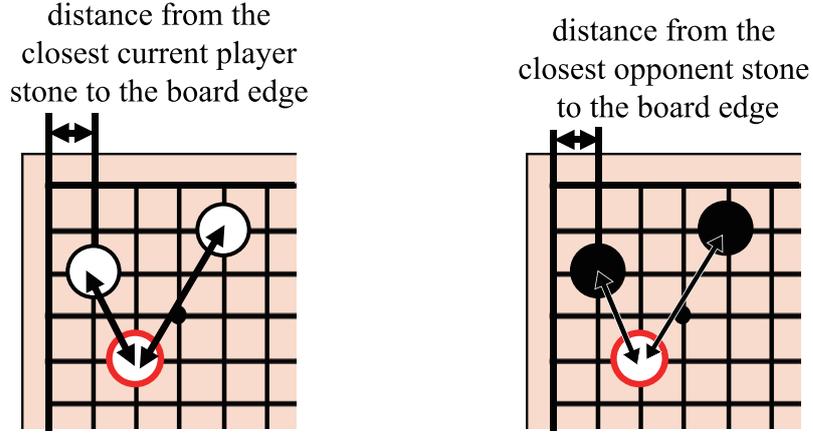
**Fig. 4.**  $FL_4/FL_5$ : distance to the closest current player's/opponent's stone

#### **$FL_6/FL_7$ : Distance From the Closest Current Player's/Opponent's Stone to the Board Edge**

As shown in Fig. 5, the distance from the closest current player's or opponent's stone to the board edge is used as a feature. This denoted as  $FL_6$  or  $FL_7$ , respectively.

### **3.2 Higher-Level Features for Strategic Move**

In addition to employing the features of lower levels to identify the shape of a particular move, this section presents features of higher levels that specifically aim to identify strategic moves. This study uses the following type of higher-level features.



(a)  $FL_6$ : Distance from the Closest Current Player's Stone to the Board Edge

(b)  $FL_7$ : Distance from the Closest Opponent's Stone to the Board Edge

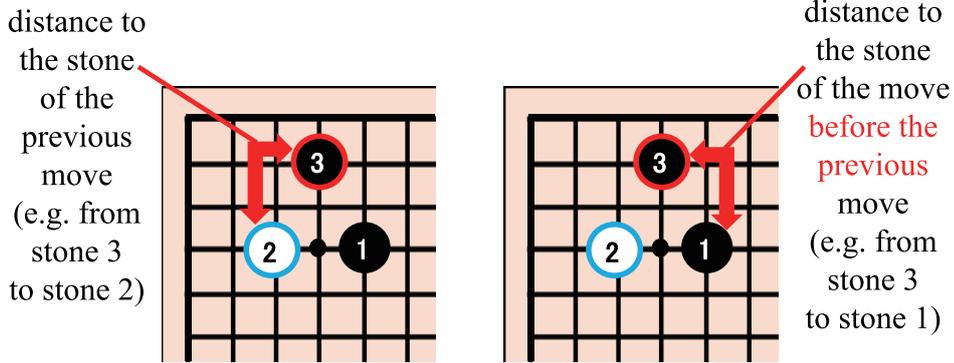
**Fig. 5.**  $FL_6$  /  $FL_7$ : distance from the closest current player's/opponent's stone to the board edge

**$FH_1/FH_2$ : Distance to the Stone from the Previous/Move Before the Previous Move**

As shown in Fig. 6, the higher-level features is the distance to the stone of the previous move or the move before the previous move, which is denoted as  $FH_1$  or  $FH_2$ , respectively. Strategic moves are typically characterized in terms of the time-series sequences of the current player's moves. As illustrated in Fig. 2, the overall sequence of time-series state transitions of moves within a game can be approximately segmented by strategic moves, each of which is typically played apart from the previous move as well as the move before the previous move. Therefore, these features of the distance between the current and previous moves are typically intended to express such time-series factors within a game. Furthermore, in our implementation of these features, instead of the distance measurement presented by equation (1) in the previous section, the following definition of the distance is employed, where  $\delta x$  and  $\delta y$  denote the differences between the  $x$  and  $y$  coordinates of the two stones, respectively:

$$\text{distance}(\delta x, \delta y) = |\delta x| + |\delta y| \quad (2)$$

This is because these features are required to be sensitive to a much broader area on the board compared to those of  $FL_4$  and  $FL_5$ , aiming at identifying the shape of a move, but not a strategic move.



(a)  $FH_1$ : Distance to the Stone of the Previous Move

(b)  $FH_2$ : Distance to the Stone of the Move before the Previous Move

Fig. 6.  $FH_1/FH_2$ : distance to the stone of the previous/move before the previous move

## 4 Evaluation

### 4.1 The Procedure

In the evaluation of the classifier for generating the expression of a particular move from the pattern of stones on a Go board, for each of the 13 expressions listed in Table 1, we train a binary classifier with all of the samples of the expression (as listed in Table 2) as positive samples. An equal number of negative samples are randomly selected from the pool of negative samples that is constructed by removing all the positive samples from the 422 samples listed in Table 2. The classifier for each binary classification is implemented using the decision tree's<sup>2</sup> module of the scikit-learn toolkit [3]. The parameters of the lower bound of the number of training samples within each node of the decision tree (in the range from 1 to 7) as well as the upper bound of the depth of the decision tree (in the range from 2 to 8) are tuned through a preliminary round of a 10-fold cross-validation with a random split into the training and test sets. This is also implemented through a module of the scikit-learn toolkit. The final evaluation results are obtained through another round of 10-fold cross-validation with the two parameters optimally tuned as above.

### 4.2 Evaluation Results

Table 3 shows the optimal combination of features as well as the optimal performance achieved by the combination of these particular features for each expression. Overall, for most expressions, including both those of the shape of a

<sup>2</sup> In this study, we applied decision trees as the classifier for identifying and generating the expression of moves as it was employed successfully by Ikeda et al. [2].

**Table 3.** Evaluation results (%)

Type of the move and the expression	ID	Binary classification	Evaluation results	Optimal features
Expression for the shape of a move	1	attachment/rest	95.0	$FL_4, FL_6$
	2	connection/rest	86.1	$FL_4, FL_5, FL_6, FH_2$
	3	hane/rest	89.5	$FL_4$
	4	cut/rest	84.2	$FL_2, FL_4, FL_6$
	5	peep/rest	81.7	$FL_2, FL_6, FH_1/FH_2$
	6	extend/rest	95.0	$FL_6, FL_7$
	7	jump/rest	91.7	$FL_4$
	8	block/rest	82.5	$FH_1$
	9	ate/rest	83.3	$FL_7, FH_2$
	10	extension/rest	87.5	$FL_6$
	Average		87.6	—
Expression for a strategic move	11	protect/rest	71.7	$FL_1, FL_2$
	12	approach move/rest	95.8	$FL_4$
	13	invasion/rest	94.2	$FL_6$
		Average		87.2
Average			87.5	—

move as well as the strategic moves, optimal combinations of features tend to include  $FL_4/FL_5$  (the distance to the closest current player’s/opponent’s stone),  $FL_6$  (the distance from the closest current player’s stone to the board edge), and  $FH_1/FH_2$  (the distance to the stone from the previous/before the previous move). These features typically represent the characteristics of the patterns of stones on a Go board. Other features such as  $FL_1$  (the coordinates of a move),  $FL_2$  (the number of stones on the board), and  $FL_3$  (the distance to the closest board edge) typically contribute little to achieving the optimal performance. This is mainly because such features represent the physical layout of the stones on the board that are not critically sensitive to detecting the expression of a move.

Notably, the higher-level features are more effective in the identification of the shapes of moves rather than those of strategic moves. This is because even in the binary classification task of a single expression for the shape of the move and the others, negative samples comprise not only other expressions for the shape of a move but also all the expressions for a particular strategic move. These results provide important clues for devising new types of features in future work.

## 5 Conclusion

This study aims to develop a software that assists human players in learning high-level Go strategies by focusing on generating a review for each individual move in a game of Go. In particular, this study proposes generating an expression

for each move based on the pattern of stones on the board. Furthermore, to this task of generating the expression for each move in a game of Go, we applied a classifier learning technique. In terms of future work, we aim to scale up the dataset to allow more accurate evaluation of the results and evaluate other types of classifiers such as support vector machines.

## References

1. R. Coulom. Computing Elo ratings of move patterns in the game of Go. In *Proc. Computer Games Workshop*, 2007.
2. K. Ikeda, T. Shishido, and S. Viennot. Machine-learning of shape names for the game of Go. In *Proc. ACG*, pages 247–259, 2015.
3. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
4. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–503, 2006.
5. A. Yoshikawa, T. Kojima, and Y. Saito. Relations between skill and the use of terms — an analysis of protocols of the game of Go —. In *Proc. 1st CG*, pages 282–299, 1998.