Artificial Intelligence Algorithm for the Game of NoGo based on Value Evaluation

Qianyu Guo
Computing Center of School of Computer and Software
East China Normal University
Shanghai, China
51174507006@stu.ecnu.edu.cn

Youguang Chen
Computing Center of School of Computer and Software
East China Normal University
Shanghai, China
ygchen@cc.ecnu.edu.cn

***Abstract*—The game of NoGo is one of the chesses for machine game, and is a variation of go-chess. NoGo and its related rules are firstly introduced with demonstration on recent researches on artificial intelligence for the game. Then a model function of value evaluation for arbitrary chessboard is proposed through analysis of characteristics of NoGo, and one or more candidate points with the highest evaluation value shall be given. Hence, an optimal solution can be given if the candidate point is sole, otherwise, the optimal solution can be given via scattering rule or the algorithm of Monte Carlo tree search (MCTS) according to different cases. Finally, the game result between the proposed algorithm by this paper and the famous open source software (OSS)——OASE-NOGO is obtained, and the proposed algorithm gives 95% of win rate, which, comparing with previous researches, proves to be feasible and efficient.**

Keywords—Artificial Intelligence; Machine Game; Value Evaluation; Monte Carlo

I. INTRODUCTION

Machine game has been a hot topic in the field of artificial intelligence. Particularly in 2016 AlphGo developed by Google defeated the human world champion Li Shishi[1], artificial intelligence for game is getting more and more attention. NoGo initiated several years ago, and is similar with go-chess that they both use the same stones and have semblable concepts such as Liberty, Eye, etc. On the other hand, go-chess determines victory or defeat according the number of crunodes suppressed by each other while NoGo determines one party defeated if it suppresses the other party's chess pieces or it gives up the chess positioning. In this way, NoGo has a quite different thinking mode for the competition strategy, and artificial intelligence of NoGo also differs from that of go-chess.

With the mentioned problems above, this paper has proposed an innovative function of value evaluation for computing all the points in arbitrary chess board, and MCTS as well as scattering rule is integrated for artificial intelligence of NoGo. Part one introduces NoGo, NoGo's rules and recent researches on artificial intelligence of NoGo. Part two offers some consideration for NoGo and gives the model function of value evaluation. Part three introduces how the optimal solution is obtained via the scattering rule and MCTS. Part four gives results of the game and the game graph with OSS——OASE-NOGO, and a case is given to demonstrate advantages of the proposed algorithm compared with the traditional one. Part five is conclusion.

II. RULES OF NOGO AND RESEARCH STATUS

A. NoGo and Its Rules

NoGo uses 9×9 chessboard, black stone goes first and then white and black stone position alternately. No suicide is allowed during the game, and one party shall lose the game if it defeats his opponent or chooses PASS. The game rule is the same with that of go-chess, that is, one stone or a series of stone of one color is on the chessboard in a straight line, and cross points beside the line is called Liberty. The stone shall be defeated if all Liberty is occupied by only one color.

NoGo has been added into the international computer Olympic race since 2011, and it was added into competitive events in China machine game race held by Chinese Artificial Intelligence Association. Since then, artificial intelligence for NoGo is getting attention for research.

B. Research Status of Artificial Intelligence for NoGo

Computer gaming is to enable the computer learn and play games like human. In brief, a computer can be enabled to have exact thinking, judgment and reasoning abilities like human. A research group of chess called Deep Blue comprising of several international chess masters and computer experts developed Deeper Blue in 1997, and it defeated the world champion Garry Kasparov with 3.5:2.5[2]. It was not until 2016 that chess is solved by Google via the combination of deep learning and tree search, during which the UCT (Upper Confidence Bound Apply to Tree) algorithm based on Monte Carlo idea has dominated chess in field of artificial intelligence for almost ten years, and References [3] [4] and [5] all exploit to optimize UCT algorithm and improve searching speed of game tree.

Research on artificial intelligence of NoGo which derives from chess is mostly solved by MCTS similar with that of chess. The earliest research on artificial intelligence of NoGo began in 2011 which found that by comparing with chess, methods of fast evaluation, Monte Carlo tree search also apply to NoGo[6]. Similar with References [7] and [8], they all use MCTS to solve the problem, among which [7] applied the

978-1-5386-5738-6/18/$31.00 ©2018 IEEE

Changchun, China
July 6-8, 2018

method of pattern matching similar with that of chess when choosing the points, and thus optimizing the searching space caused by the application of MCTS to some extent, and [8] simulated the chessboard with higher score preferentially when initiating MCTS algorithm, which can reduce simulation times. Besides, Reference [9] proposed genetic learning mechanism like evolutionary computation and fuzzy logic to solve the problems of NoGo.

NoGo is a new pattern of chess and related researches about it are few, most of which apply Monte Carlo idea. The difficulty is how to provide candidate points for MCTS and how to prune. Methods of evolutionary computation and pattern matching increase the complexity and have higher requirements for time and hardware level when for computing, so they are not suitable for software whose hardware level is low or running speed is slow.

III. GAMING THINKING OF NOGO AND EVALUATION FUNCTION

To solve the above problems, this paper proposed a model function of value evaluation which can calculate the value of every cross point in arbitrary chessboard. This method mainly derives from the gaming thinking of NoGo. During the game, to avoid losing, that is, not to defeat the opponent's stone, there would be two ideas. One is to make less Liberty than the opponent, the other is to make more eye than the opponent.

With the above two points, this paper defines the number of stone atari by the opponent and the number of eyes on one's own side as the number of right, and then a model and a function of value evaluation for NoGo are made. Apparently, both parties try to increase their own right number and the party who has bigger right number shall win. Therefore, the purpose of playing NoGo is to make more right for one's own side or break the opponent's right. The production and break of right is defined as right rule in this paper, and approaches for this rule are to make and configurate chessboard such as atari, eye, etc. But in actual cases, the chessboard is fickle which is far from being listed one by one. And the application of pattern base cannot avoid the problem of much occupied space. Therefore, we consider from the viewpoint of Liberty, and formation of right value is to form one or more cross points upon which the opponent cannot position among the points. That is, this cross point is one single eye position on one's own side or the last Liberty in that position, shown in Figure 1 and 2. In this way, the cross point become one's own right.

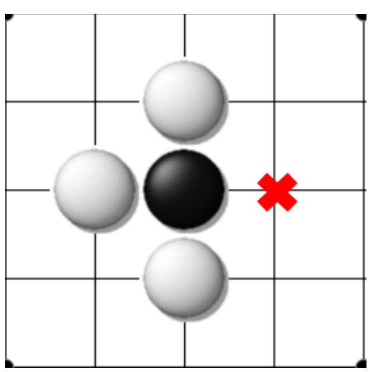

Fig.1 right schematic diagram1

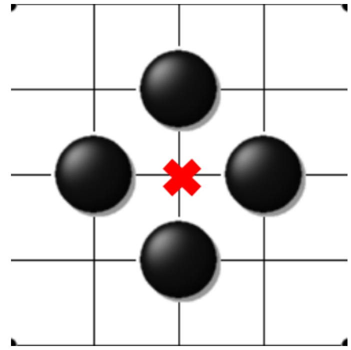

Fig.2 right schematic diagram 2

Thus, in a 9×9 chessboard, coordinate points are marked from (1,1) to (9,9) and the coordinate of the jth cross point (1≤j≤81) is (m, n) (1 m 9, 1 n 9). Assuming that stone for the ith (1 i 81) step is black, and the white party cannot position at (m, n), that is, (m, n) is eye position for the black party or the last "Liberty" of the black stone or of the stone series, which is marked as NW (non-position marked as 1, position marked as 0). In this way, the right value of black Bi can be evaluated as Equation (1).

$$B_i = \sum_{j=1}^{81} N_w^j \tag{1}$$

Similarly, assuming that stone for the ith step is white, and the black cannot position which is marked as Nb (non-position marked as 1, position marked as 0). In this way, the right value of white Wi can be evaluated as Equation (2).

$$W_i = \sum_{j=1}^{81} N_b^j \tag{2}$$

For the model construction for value evaluation for specific chessboard, when it proceeds to the ith step, total right value of the chessboard is marked as Pi, then

$$P_i = B_i + W_i \tag{3}$$

Color difference will not affect the value of the current chessboard during the game, that is, superiority and inferiority of one point will not vary with the stone color. Therefore, in spite of white or black, just choose the maximum Pi to position when for the ith step (1 i 81). Hence, the evaluation function for value V of one point (p, q) is Equation (4).

$$V(p,q) = \sum_{j=1}^{81} N_w^j + \sum_{j=1}^{81} N_b^j \tag{4}$$

With Equation (4), we can calculate the value of all points which can be positioned on the chessboard, and its pseudo code of the evaluation function is:

Value point (choose point, the ith step)

{

Simulate this point as black stone;

While (81points are all included)

{

Examine all positions upon which the white stone cannot position, marked as a;

Then, right value of black stone Pb ← a-i;

}

Simulate this point as white stone;

While(81points are all included)

{

Examine all positions upon which the black stone cannot position, marked as b;

Then, right value of white stone Pw ← b-i;

}

Evaluation value of this point is P= Pb + Pw;

Return P;

}

IV. SCATTERING RULE AND MCTS

The model function of value evaluation in the above section can calculate all points which can be positioned in arbitrary chessboard. In most cases, results can be outputted when the point whose value is highest in the current chessboard is sole. For the case in which the highest value is not sole, value of all cross points might be 0 (e.g. opening) and the highest value is not 0 and there might exist several points with the same value. For the above two situations, scattering rule is used when evaluation values are all 0; Monte Carlo is used when there exist several points with the same value which is not 0.

A. Scattering Rule

Case that evaluation values are all 0 for all the points which can be positioned usually can be seen in the opening, and the value of all the points have no difference in terms of superiority and inferiority, and scattering rule is applied. When the chessboard is empty, the algorithm will choose the center, that is, TianYuan point as opening.

Manhattan distance (5) is chosen to carry out scattering, that is, the sum of absolute axle spread of two points on the standard coordinate system

$$d(i,j) = |X1 - X2| + |Y1 - Y2| \tag{5}$$

Procedures for the function of scattering are detailed.

Step 1: choose all points which can be positioned and eliminate the points which are cross points of the given stone and which cannot play on one's own side, such as the opponent's eye position or the position where the opponent's stone can be defeated;

Step 2: calculate the minimum value of all points which can be positioned and the Manhattan distance of given stone;

Step 3: find the maximum value among the values in step 2 and mark the corresponding point. If it is sole, this point is the optimal solution for scattering rule. If it is not sole, random selection is made.

B. MCTS

For case when several candidate points have the highest values and not to be 0, MCTS method is applied for the optimal solution.

Monte Carlo planning is a widely used method to solve the problem of Markov decision [10] (stochastic dynamic model based on Markovian process theory) [11]. It is a planning method based on ideas of Monte Carlo method in which possible process of state transition in Markov tree search can be shown via status tree, which differs from normal construction of search tree. By repeatedly giving the sampling events from the original state, Monte Carlo planning gradually extends every node on the tree, and every event shall conform to triad sequence of "state-behavior-return".

MCTS has four procedures.

(1) Search: search downward from root node (outcome state) of the game tree until the existing leaf node (current chessboard):

(2) Extend: extend the leaf node of the current game tree;

(3) Simulation: carry out probabilistic simulation of Monte Carlo from the current leaf node of the game tree and give a probability statistics value of Monte Carlo;

(4) Renew: renew the result of Monte Carlo simulation to every node on the game tree in the searching process.

In this paper, the candidate points with the highest value evaluated by the value function are regarded separately as root nodes of Monte Carlo search tree, then find out Monte Carlo value of these candidate points and choose the highest one to position.

V. EXPERIMENTAL RESULT

A. Function Display

This paper proposed artificial intelligence software for NoGo, which can operate on a laptop with normal configuration (4GB memory, dual-core) within 2 seconds of response time. Figure 3 and 4 are the operation results among which Figure 3 shows the gaming result between the proposed software and the OSS——OASE-NOGO.

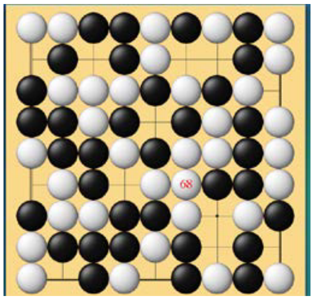

Fig.3 Result Legend

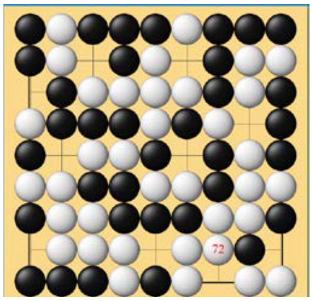

Fig.4 Gaming with OASE-NOGO of advanced edition

Besides, the program for this algorithm can be operated on Android cellphone. The tested cellphone is of 3GB memory with 8-core, and the response time is within 2 seconds. The result is given in Figure 5.

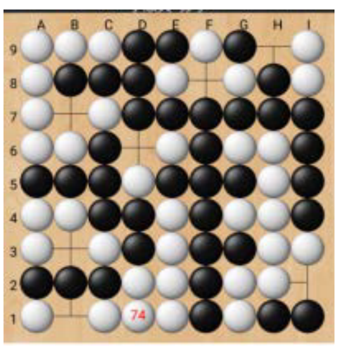

Fig.5 Operation schematic diagram on cellphone

B. Effect Comparison

Table 1 shows gaming results between the proposed algorithm and OASE-NOGO as well as statistics of win rate. The win rate of the proposed algorithm is above 95%. For the common chessboard in figure 6, compared with complexity of traditional Monte Carlo algorithm, the proposed algorithm proves to be feasible and efficient.

TABLE I. Statistics of Results

Testing system	***Opponent***	***Testing times***	***Win times***	***Win rate***
System in this test	OASE-NOGO V1.1	200	193	96.5%
System in this test	OASE-NOGO V1.1 advanced edition	200	190	95%

Win rate of the program via pattern matching and UCT algorithm of Monte Carlo idea in Reference [8] against OASE-NOGO V1.1 is 90%, less than the win rate of the proposed program in this paper.

VI. Conclusion

For the high complexity of the current artificial intelligence researches on NoGo, this paper proposed a new solution based on the function of value evaluation with consideration of the gaming characteristics of NoGo. Construction of the model and calculation process of the value function are detailed introduced, and scattering rule and Monte Carlo idea are applied to choose the optimal solution to corresponding situations in order to solve the problem that several candidate points may appear in the value evaluation. The software with the proposed algorithm as the core has achieved good results, which proves to be feasible and efficient.

References

[1] David Silver, Aja Huang, et al, Mastering the Game of Go with Deep Neural Networks and Tree Search[J],Nature, 2016:484-489.

[2] Dulie Qi, The contest between computer and chess world champion [J]. Electronic Products China, 1996, 04: 120-121.

[3] Rong Gu, Research on Some Problems of Computer Go Game System [D], Beijing, Tsinghua University, 2003.

[4] Ying Peng, Fang Wang, Ping Luo. Go Situation Judgment Algorithm Based on Mathematical Morphology [N]. Natural Science Journal of Xiangtan University, 2011,33(1):110-112.

[5] Jing Huang. Application of UCT Algorithm in Computer Go Game and Its Improvement [D]. Beijing University of Posts and Telecommunications.

[6] C.W.Chou,O.Teytaud,and S.J.Yen.Revising MonteCarlo Tree search on a normal from game:NOGO[C].The main European events on Evolutionary Computation,Torino,Italy,2011, Apr.27-29.

[7] Yuxia Sun,Yajie Wang,Fei Li,Pattern Matching and Monte-carlo Simulation Mechanism for the Game of NoGo[C],Proceedings of IEEE CCIS2012.

[8] Liang Guojun, Realization of UCT Algorithm in Go Game [N], Journal of Shaoguan University.2015, 8: 17-21

[9] LEE C S, WANG M H, CHEN Y J, et al.Genetic fuzzy markup language for game of NoGo [J] . Knowledge-Based Systems, 2012(34):64-80.

[10] Wei Shao. The Monte Carlo method is used in the model statistics [D]. Shan Dong University.2012.

[11] Aijun Bai. Research on Uncertainty Planning and Perception Based on Markov Theory [D]. University of Science and Technology of China.2014