Improvement of MCTS in Computer Go

Shi-Jim Yen[1], Cheng-Wei Chou[2], Shun-Chin Hsu[3] , Jr-Chang Chen[4] ,Tai-Ning Yang[5]

[1]Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. sjyen@mail.ndhu.edu.tw

[2]Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. d9721002@ems.ndhu.edu.tw

[3]Department of Information Management, Chang Jung Christian University, Tainan, Taiwan. schsu@mail.cju.edu.tw

[4]Department of Applied Mathematics, Chung Yuan Christian University, Chung Li, Taiwan. jcchen@cycu.edu.tw

[5]Department of Computer Science, Chinese Culture University, Taipei, Taiwan. tnyang@faculty.pccu.edu.tw

1. Introduction

This paper proposes a method to judge quality of the simulation of Monte Carlo Tree Search (MCTS) by using statistical result of the game's simulation. In the field of Computer Go, the simulation of MCTS is used to judge whether the situation of board is fine or not. The more accurately it judges, the more precise condition it reflects. Then, the program can search for the right direction. Unfortunately , so far, except for having programs to play against each other to calculate the rate of winning, there is not a better way to judge whether the simulation is correct or not [1]. By using statistical result of the game's simulation, we try to figure out whether simulation is accurate enough in reflecting the condition of the game to make sure the accuracy of its simulation. In this way, the inaccurate part of simulation can be discovered, and then it can also be better fixed to improve the accuracy of simulation. Therefore, the program can search for the correct direction and make it more strong.

2. Our Method : Evaluation Value

For a situation, we do simulation several times, and record the amount that every point is occupied by which color at the end of the simulation. If the point is empty, it should belong to the color beside it. For example, if the empty point is circled by black stones, it should belong to black. If it is circled by both white and black stones, the possibility for both colors to occupy this point is 0.5. In this way, we can figure out more precisely whether the simulation can accurately reflect the condition of the game. Figure1 is an example of this statistical.

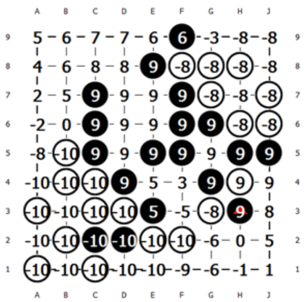

Figure 1 : an example of statistical

We normalize the result of this statistical into 10 to -10. The bigger the number is, the more possible the point will be occupied by black; on the other hand, smaller numbers mean that the point will be more possibly occupied by white. The values of white stones at the left bottom of Figure1 is -10, means white stones are definitely life. The region between the two colors shows that the result which is nearly 0 manifests that both colors may occupy this place. To normalize the statistical values , we can use the following formula to represent it:

$$\frac{\text{The amount of which the point is occupied by black} \times 20}{\text{The amount of simulation}} - 10$$

The result of the statistics is called "evaluation value." This value can be compared with human consideration of the board situation to judge whether the simulation is proper or not. After figuring out where the inaccuracy of this simulation is, we can aim at the improvement of the inaccurate part to develop the accuracy of the simulation and moreover to better the program’s strength. This method has already been used successfully in the Go program, and has great effect.

3. How To Decide Simulation Quality

"Evaluation value" provides an easy way to judge whether the board simulation’s evaluation is accurate. According to "evaluation value", we can know the possibility for the string to survive or some empty cell to be black or white. For example, for the black string, if "evaluation value" is bigger than 0, it means the simulation believes that string has more chance to survive. The bigger the evaluation value is, the more the survivability is. The way to judge the white string is just opposite to the black one. Besides, if the evaluation value of some string is 0, it means the side plays first can win this string, like figure 2.

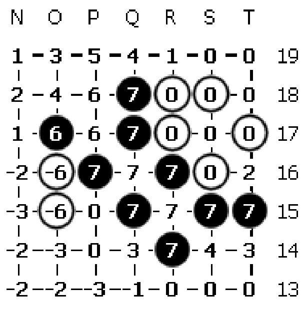

Figure 2 : an example when evaluation is 0

In Figure 2, the evaluation value of the white group on the right upper side is 0 which means if the black side plays first, it will kill this group; if the white plays first, it can save this group. By this kind of judgment and the consideration of amateur players with high dan or professional players to examine the losing record, we can find out the mistakes in the simulation and then correct them.

4. Use Evaluation Value to Fix Incorrect Simulation

Figure 3 is an example of incorrect simulation. The final move of black is on G2. After calculating the evaluation value of this situation, it is found that the simulation thinks white string E2 is dead and black stones occupy the territory near the string. This is unreasonable.

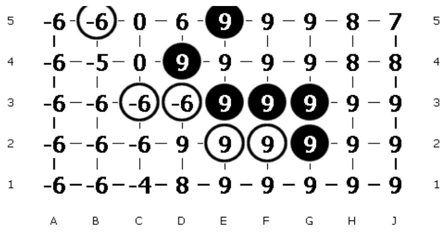

Figure 3: an example of incorrect simulation

After we examine the process of this simulation, we find the process as figure 4. The first step is pattern, the second is Atari, and the third is problematic. The white side saves one stone, but will be Ataried by the black which means the string will be killed by black stones. Thus, in the third step, the white shouldn't save one stone because there are "too many places to connect".

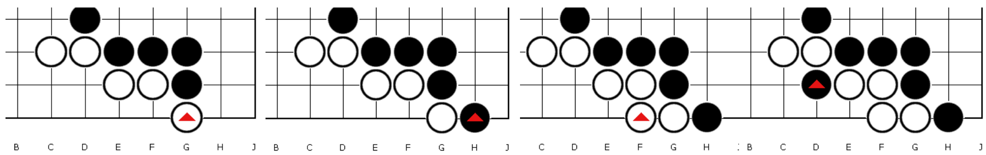

Figure 4: process of simulation

Please refer to Algorithm 1 for the way we judge the situation with "too many places to connect". In Figure 5, the result is more normal after the correction.

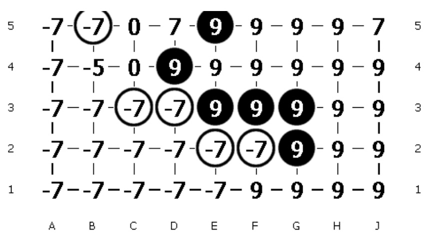

Figure 5: the result after correction

Algorithm 1 Algorithm for the prevention to play "too many places to connect "

if we are about to save a string with one liberty space **then**

if we place a stone on the liberty point of one string with one liberty space to make a string with two liberty spaces **then**

if the liberty points of the string with two liberty spaces are on the baseline or the upper line next to it **then**

if the two liberty points are neither on the hanging connection nor on the eye **then**

it is a situation with "too many places to connect", so no stone should be placed on the point

end if

end if

end if

end if

5. Experiment

After solving this problem, the program's strength is upgraded. The differences of the program's victory rate to win before and after the correction are as follows. The tested object is Fuego version 3.2; both sides are simulated for 3000 times, testing on 1000 games, and each side has the chance of 500 games to play first.

Without "too many places to connect"	36.9% (±1.5)
With "too many places to connect"	39.4% (±1.5)

Table 1: result of experiment

6. Conclusion

In conclusion, incorrect simulation will influence the judgment of the program and result in incorrect decision. We provide an easy and useful way to quickly find out the problems of the simulation. By solving these problems, we are able to improve the quality of simulation, and lead search to the correct direction to upgrade the program's strength. This method is employed on our Go program, and has great performance.

REFERENCE

[1] Chaslot G., Fiter C., Hoock J.-B., Rimmel A., Teytaud O, "Adding expert knowledge and exploration in Monte-Carlo Tree Search", Advances in Computer Games (2009)

[2] http://fuego.sourceforge.net/