ISSN: 2664-9640

Research and Improvement of 2048 Game Tree Search Algorithm Based on Data Analysis

Bohan Cui [1,*], Jing Zhu [2] and Sirong Liang [3]

[1] International College of Beijing University of Posts and Telecommunications, Beijing, 10000, China;

[2] School of Computer Engineering, Suzhou Vocational University, Suzhou 215100, China.

[3] School of Communication and Information Engineering, Xi'an University of Posts & Telecommunications, Xi 'an 710100, China.

* Corresponding Author

Abstract

This article first proposes the Extreme Value Extension algorithm to solve the 2048 problem. This algorithm obtains the optimal solution through a static evaluation function. However, in the process of implementation, there is a phenomenon of redundancy. When there are too many steps to move the grid, it is difficult to find a good evaluation function, which reduces the efficiency of the Extreme Value Extension algorithm. Subsequently, this paper adopts the Alpha-Beta algorithm, which is an improvement of the former. In the case of the same number of search nodes, the search depth can be doubled. The above model is improved by Monte Carlo evaluation. It conducts a large number of simulations on each optional point in the current situation to obtain the corresponding statistical characteristics of victory and defeat. In simple cases, the point with a higher winning rate can be considered as a better point for selection. Since the UCT algorithm can continuously adjust the strategy according to the previous results, it chooses which one can be evaluated first. Therefore, the UCT algorithm is used to improve the convergence speed on the basis of Monte Carlo. The probability can be obtained as 100%.

Keywords

Data analysis, Alpha-beta, Search algorithm, Monte Carlo simulation.

1. Introduction

2048 is a very popular puzzle game recently, many netizens claim that "once you play it, you can't stop".The rules of 2048 game are very simple: control all the squares to move in the same direction every time, two squares of the same number bump together and merge into their sum, after each operation will randomly generate a 2 or 4 in the blank square, finally get a "2048" square is the victory.If all 16 squares are filled and none of the adjacent squares are identical and cannot be moved, the game is over.

This paper first implements 2048 of artificial intelligence (AI) based on Extreme Value Extension and alpha-beta pruning algorithm.This paper argues that the 2048 game can be viewed as a game between a human and a computer, where the human controls all the squares to move in the same direction and merge, while the computer places a random "2" or "4" square in the blank.However, in the AI environment, both sides of the game are computers and neither of them are rational. Therefore, in the AI environment, it is more appropriate to choose the more conservative Extreme Value Extension game strategy than the Mini-Max-Tree.If the current pattern is taken as the parent node of the game tree, and the pattern caused by all possible moves in the next step is taken as a child node of the tree, if the Extreme Value Extension algorithm continues to be used, the efficiency of this algorithm is not ideal and many unnecessary steps will be caused.Since each subsection is followed by other subsections, there are many possible scenarios that loop back and forth until 2048, but not all nodes have to be searched. Some nodes are not necessary.In order to solve this problem, alpha-beta pruning algorithm can be adopted in this paper.

For the realization of 2048, Monte Carlo evaluation is a good solution. It can get the corresponding statistical characteristics of victory and defeat through a large number of simulation of each optional point in the current situation. In a simple case, the point with higher winning rate can be considered as the better point to choose. On the basis of Monte Carlo evaluation algorithm, uct greatly improves the convergence speed. Uct can continuously adjust the strategy according to the previous results, and select the priority evaluation point.

This paper will establish the corresponding model and mainly make the following contributions:

How to achieve 2048, give a general model, and use the number of moves required to complete the game and the probability of success two indicators to verify the effectiveness of the model.

2. Principle description

2.1. Extreme Value Extension algorithm

Extreme Value Extension is a change from Minimax algorithm. Only one condition is added, under the condition that both parties are irrational, the essence is still Minimax.

The definition of Minimax:

$$I(\mathrm{x})=\left\{i \middle| f_i(x)=F(x)\right\}$$

$$\min_{x} F(x)=\max_{i\leq i\leq m} f_i(x), x\in R^n,$$

Suppose that in the process of game, the other party always chooses the move with the smallest game value, and our party chooses the move with the largest game value. The other party calls it Min, and our party calls it Max. Both of these are irrational.Since the two sides of the game move alternately, the node of the game tree and its parent nodes belong to one of the two sides, and their types belong to Max and Min respectively.Each node in the game tree corresponds to a depth, and the depth of the leaves is zero.Therefore, at any node node, the optimal game value for both sides of the game is:

$$M\mathrm{inMax(node)}=\begin{cases} Evaluate(note) \\ MAX_{seSuccessors(\mathrm{node})}(\mathrm{MinMax(s)}) \\ MIN_{seSuccessors(\mathrm{node})}(\mathrm{MinMax(s)}) \end{cases}$$

Thus, it is natural to derive the Extreme Value Extension algorithm, which is used to find the game Value of the two-person zero-sum game problem satisfying some conditions.

2.2. Alpha-beta pruning algorithm

The Alpha-beta algorithm is an optimization of the Minimax algorithm. It runs more efficiently than Minimax and naturally faster than Extreme Value Extension. The Extreme Value Extension algorithm checks the entire game tree. The efficiency is very low. When the number of steps increases, each time you search for a deeper level, the size of the tree increases exponentially. With two obvious redundancy phenomena, the search efficiency is naturally reduced. The first phenomenon is maximum redundancy.

Now we know that the value of node B is greater than the value of node D.Since the value of node C should be the minimum among the values of its children, this minimum value must be less than or equal to the value of node D, and therefore must be less than the value of node B, which indicates that, continue to search for the other children of node C, E, F...They don't make sense anymore, they can't contribute anything, so they cut all the subtrees that have node C as their root.This optimization is called Alpha pruning.

The value of node A shall be the lesser of the values of node B and node C.Now we know that the value of node B is less than the value of node D.Because of its node C value should be the value of all the nodes of the great, the maximum value must be greater than or equal to the value of the node D which is greater than the value of the node B, this suggests that continue to C all other search node has no sense, and can put the node C as the root of the subtree cut off entirely, this optimization is called Beta pruning.

The alpha-beta algorithm is to choose the best possible route out of many routes.It is important to search as deeply as possible by examining the first few layers of the search tree and using heuristic evaluations at leaf nodes.Let's compare the Extreme Value Extension algorithm to the Alpha-Beta pruning algorithm.

For a Min node, if the upper limit Beta of its backward value can be estimated, and this Beta value is not greater than the lower limit Alpha of the estimated backward value of Min's parent node (Max node), that is, Alpha Beta, then There is no need to expand the remaining child nodes of the Min node, because the evaluation of these nodes has no effect on the backward value of the Min parent node. This process is called Alpha pruning.

For a Max node, if you can estimate the value of backward the infimum of Alpha, and the parent node of the Alpha value not less than the Max (Min) estimates of the backward values the supremum of Beta, Alpha Beta, is not to need to extend the Max again the rest of the child nodes of the node, because the valuations of Max parent node has no any impact on the back.This process is called Beta pruning.

C. The Alpha value of a Max node is equal to the current maximum final reverse value of its successor node, and the Beta value of a Min node is equal to the current minimum final reverse value of its successor node.

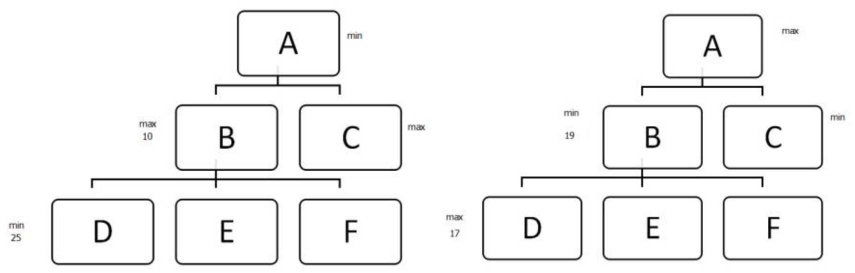


Figure 1:Schematic diagram of Alpha-beta algorithm

With Alpha-beta pruning, the search depth of Extreme Value Extension can be increased in the same time, so better results can be obtained.

2.3. Mathematical induction

I'm going to prove it by mathematical induction.It has previously been shown that, under ideal conditions, the maximum value in a 16 cell of 4X4 is 217, and when there is only one block, the maximum value is only 4.

It can be proved that f(1) Max =4 when n is equal to 4, f(4) Max =131072 Assuming that n=k (k≥n0, k is a natural number), the statement is true.Prove that the statement is also true for n=k+1.

When n=4, D= $\mathbf{f(x)} = \mathbf{4^n} + \sum_{n=1}^{16}(\mathbf{4^{n-1}})$

So when n=k, D= f(x) $= 4^n + \sum_{n=1}^{k^2}(4^{k^2} - n)$

When n=k is established, D= f$(k+1) = 4^n + \sum_{n=1}^{k^2+2k+1}(4^{k^2+2k+1-n})$

ISSN: 2664-9640

Then, when n=k+1, D= f(k) $= 4^n + \sum_{n=1}^{k^2+2k+1}(4^{k^2+2k+1-n}) - 4^n + \sum_{n=1}^{k^2}(4^{k^2-n})$

$$=\sum_{n=1}^{k^2+2k+1-n}(4^{k^2+2k+1-n}) - \sum_{n=1}^{k^2}(4^{k^2-n})$$

$$=4^{k^2+2k+1-n} - 4^{k^2-n}$$

3. Algorithm model establishment

3.1. Monte Carlo algorithm

Monte Carlo method is also called stochastic simulation method, which is an optimal and limited search method. Its basic idea is that in order to solve problems in mathematics, physics, engineering technology, and production management, first establish a probability model or stochastic process so that its parameters are equal to the solution of the problem: then through observation or sampling of the model or process Experiment to calculate the statistical characteristics of the required parameters, and finally give the approximate value of the solution.

In the process of applying Monte Carlo method to solve 2048 problem, there are several contents as follows in general:

Establish a simple and convenient probability statistical model for solving the problem, so that the solution is exactly the probability distribution or mathematical expectation of the established model.

According to the characteristics of the probabilistic statistical model and the needs of calculation practice, the model should be improved as far as possible in order to reduce the variance and cost and improve the calculation efficiency.

To establish a sampling method for random variables, including a method to generate pseudo-random numbers and a random sampling method to generate random variables for the distribution encountered.

Give the method to obtain the statistical estimate to be solved and its variance or standard error.

3.2. UCT algorithm (UCB for TreeSearch)

UCT, also known as UCB for Tree Search, is an application of Upper Confidence Bound (UCB) in Tree Search.UCB strategy is the best strategy to solve the factors that are independent of each other and have different rates of return, so as to obtain the maximum return.Roughly speaking, for each operation, UCB will add an additional parameter based on the current average profit value of each factor to obtain the UCB value of the factor in this operation. Then, according to this value, the factor with the largest UCB value is selected as the factor to be selected in this operation.Among them, the so-called extra parameters will be relatively reduced with the increase of the number of times each factor is selected. Its purpose is to make the selection of factors not too strictly adhere to the old performance, but to explore other factors appropriately.

The general form of the UCB formula:

$$Score = Exploitation + Exploration$$

In practice, a UCB formula with a better verification effect is expressed as follows:

$$\overline{X}_j + \sqrt{\frac{2\log n}{T_j(\mathrm{n})}}$$

When a Tree Search begins, UCT creates a Tree.

1) starting from the root node

2) each child node in the use of UCB formula of UCB value, choose UCB the child nodes of the highest value

3) if the child is not a leaf node (never visited node), then the node, repeat (2)

4) until the leaf node, then calculate profit value of leaf nodes, and update the root node to the node in this path all the profit on the

5) by (1) began to repeat, until the end of the time, or at a preset number

6) all child nodes from the root node, select the highest number of average profit, as a best nodes,This node is the result of UCT.

3.3. Improved model

When the UCT algorithm is used to determine which drop-off point will become the final drop-off point, the drop-off point with the maximum number of visits will always be selected.

Therefore, when using the absolute pruning condition, it is absolutely impossible for the node visited the most times to meet the pruning condition, which ensures that the final decision result based on the number of visits will be consistent with the original UCT method after using the absolute pruning condition.

If there is a drop-down point that is significantly better than other nodes, the total number of visits may have exceeded half of the expected number of visits without reaching our expected total number of visits.Therefore, when absolute pruning condition is added, the simulation can be terminated in advance to save time.

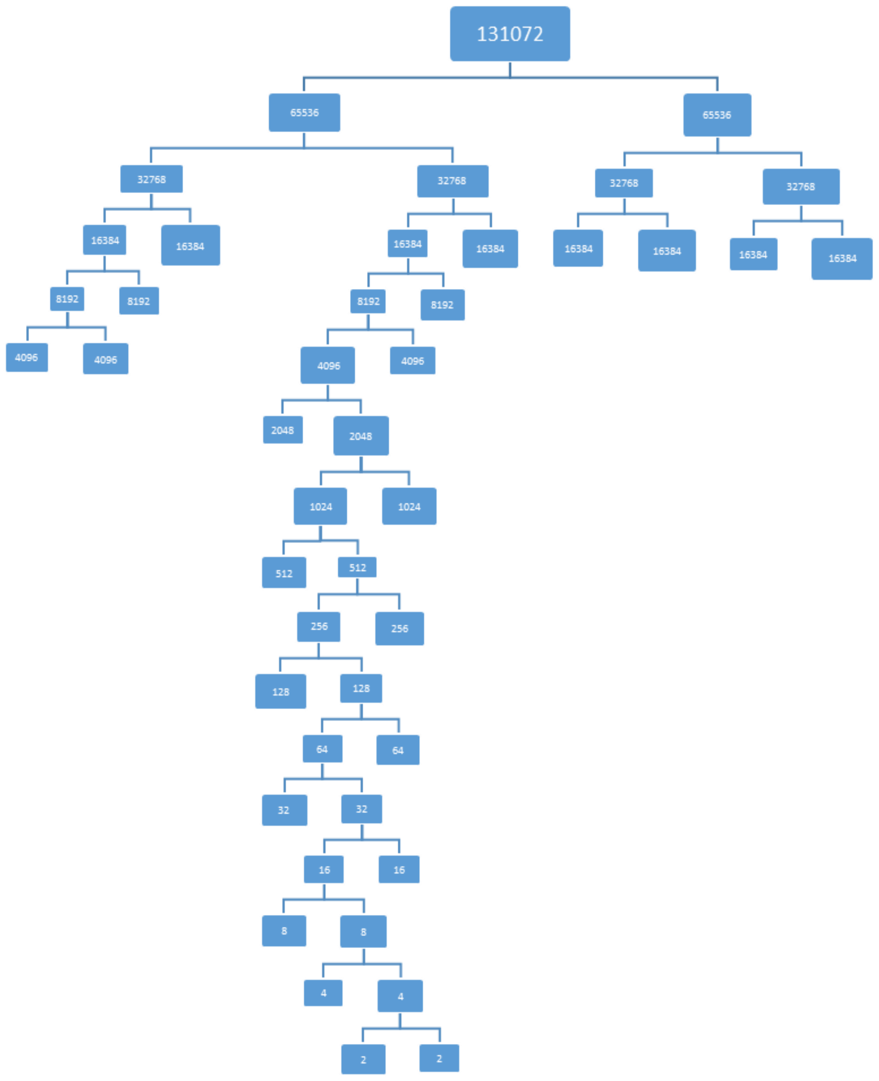
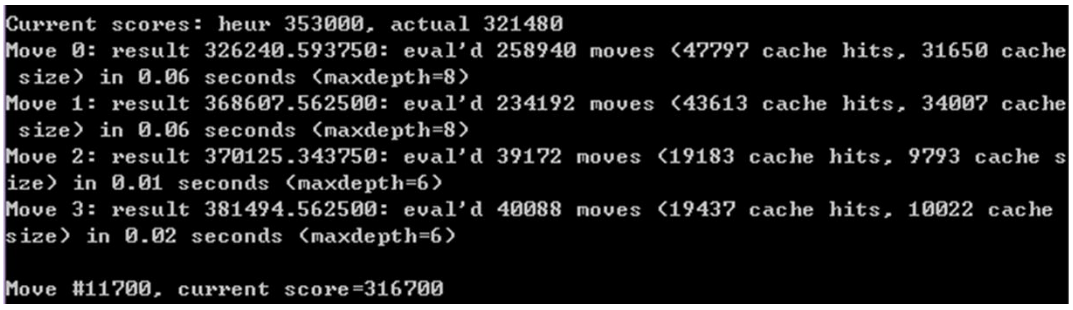


Figure 2:Schematic diagram of the improved model

3.4. Verification of the model

Analysis:

The number of moves and the probability of success are two indicators to verify the effectiveness of the model. It is necessary to separately propose the number of moves and the probability of success for analysis. The number of moves and the probability of success can be practiced through the program. The program contains the statistics of the number of movements and the set goals. Since the movement value reaches 2048, the automatic movement to 2048 uses AI, and the practice is automatically carried out. AI can automatically and reasonably avoid the possibility of not being able to move to 2048, making it possible to reach 2048.



```
Current scores: heur 353000, actual 321480
Move 0: result 326240.593750: eval'd 258940 moves (47797 cache hits, 31650 cache
 size) in 0.06 seconds (maxdepth=8)
Move 1: result 368607.562500: eval'd 234192 moves (43613 cache hits, 34007 cache
 size) in 0.06 seconds (maxdepth=8)
Move 2: result 370125.343750: eval'd 39172 moves (19183 cache hits, 9793 cache s
ize) in 0.01 seconds (maxdepth=6)
Move 3: result 381494.562500: eval'd 40088 moves (19437 cache hits, 10022 cache
size) in 0.02 seconds (maxdepth=6)

Move #11700, current score=316700
```

Figure 3: Running results of the algorithm model

As can be seen from the result in the figure above, when the player reaches 2048, there are 11,700 moves in total, and the score is 316,700. However, this result is not unique, because the starting value is different every time, so the AI will give different actions for different situations, so this is also the reason why the answer is not unique.

4. Conclusion

Through several simulation summaries, it is found that the number of movements is discrete, and the scattered distribution is between 10,000 and 15,000, which can be analyzed by the way of function division.The AI can run 100% of the time and successfully reach 2048, so the success rate is 100%.

Acknowledgements

I am very grateful to Chen Junke for his emotional support in writing his thesis.

References

[1] Mingming Zhou, Application and Improvement of UCT Algorithm in Computer Go 2012 (Supplement): 330-06
[2] Liu Yu, Application of Monte_carlo Method in Computer Go 47 (12)
[3] Yue Jinpeng, Research and Improvement of Chinese Chess Alpha_Beta Search Algorithm 2009-0445(2)
[4] Wang Hui, semiparametric regression model parameter estimation based on high-order difference method and its minimax properties [D]; Central China Normal University; 2012
[5] Zhang Jiajia-Incomplete Information Multiplayer Military Standard Game System Based on UCT Algorithm Harbin Institute of Technology December 2008
[6] Yan Bing-Research on C_S algorithm based on MonteCarlo simulation data in IMRT Hefei University of Technology
[7] Wu zhe - on the Roesser discrete state space model, Journal of Northeastern University (Natural Science Edition), 1982.
[8] Zhiming Lu-Solving Discrete State Space Equations Based on Symbolic Object Functions Journal of Electrical & Electronic Education.