# Dynamic Nonlinear Search Technologies Based on Military Strength Model in Surakarta

QIU Hongkun[1], GAO Fuxiang[2], WANG Yajie[1], LIU Cheng[1], GU Leitao[1]

1. Shenyang Aerospace University, Shenyang 110136, China
E-mail: qiuhongkun_cn@sina.com

2. Institute of Information Science and Engineering, Northeastern University, Shenyang 110819, China
E-mail: gaofuxiang@ise.neu.edu.cn

**Abstract:** In the traditional Surakarta game, the single objective search technology which has to degrade the search depth to meet the single time constraint is often used .So dynamic nonlinear search technology, based on the "military strength" fuzzy set, is used to find the best answer to meet with the search time and the multi-objective search depth. The fuzzy model of the "military strength" is established by following a large number of experiments' data. And according to the changes of the game, global evaluation value is adjusted. And then, combined with the experiments we get the empirical equation. So the rule of multilevel search depth is established after multiple linear regression statistics. Multilevel search depth can also be adjusted to get the optimal solution of dynamic search under the multiple constraint conditions. The experiment proves that the overall strength is distinctly improved in the computer game of Surakarta by using the technology. And it is verified by the experiment.

**Key Words:** Surakarta, Computer game, military strength, dynamic search, fuzzy set

## 1 Introduction

Computer game is considered to be one of the most challenging research directions in the field of artificial intelligence. The basic idea of the algorithm is developing rapidly and a variety of search algorithm is also endlessly emerging. Generally, the typical algorithm adapted in a procedure includes Min-Max Search algorithm, Alpha-Beta pruning algorithm, PV algorithm and so on. But the algorithm applied to some kinds of chess, cannot be generalized to the others [1].

Surakarta, one of many popular computer games, is a complete information extended game, that is to say, the participant's strategies only rely on the game situation in the game [2]. Because of the complexity of calculating the step algorithm of Surakarta, it's difficult to pursue the optimal solution by enumerative technique in current condition of calculating power. In the early computer game algorithm, the single objective search technology is mostly adapted. In order to meet the single time constrain, the search depth usually has to be degraded. But some fine step algorithms are often omitted due to the reduction of search time. So in relate to the respective characteristic of Surakarta, balancing between advantages and disadvantages of various algorithms, we focus on how to establish a more reasonable, perfect and efficient evaluation function and searching algorithms in Surakarta in the next research.

## 2 Surakarta Search Technologies

### 2.1 Surakarta Basic Rules & Game tree

Surakarta rules are simple. Its board is composed of a square network and corners of the eight circular arcs. Each of the two sides, which are different colors, has 12 pieces, which are arranged in two rows. The goal of the game is to eat the opponent's pieces. At the beginning of the game, players toss a coin to decide who starts firstly, and each can only move one piece by turns. When a piece is going to eat the opponent's pieces, there must be a pass route, and the route must contain a (or several) arc. As long as no other pieces in the way, a pawn can go any distance to eat an opponent's piece, and the pieces being eaten should be removed from the board [3].

To realize the auto counterwork, as well as reduce front developing workload, based on the display interface developed by Beijing Institute of Technology (BIT) and its interface protocol, search engine for step algorithm is to be developed. As shown in Fig. 1:

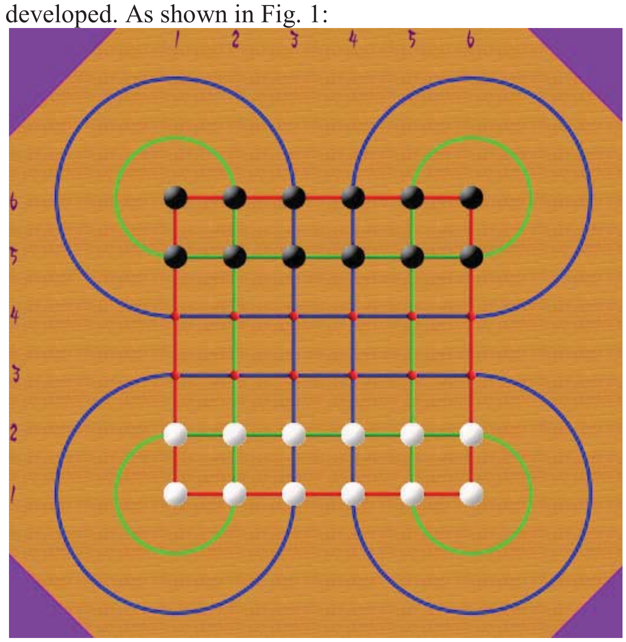


Fig. 1: Surakarta chessboard

For the computer game, move generation process is also the game tree expansion process. As shown in Fig. 2:

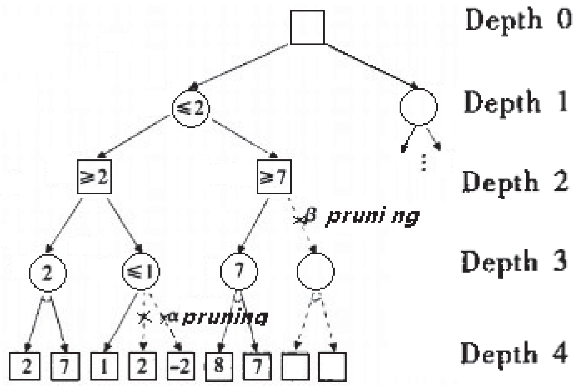

Fig. 2: Four degree game tree

For Surakarta, we can compose a game tree. In the four layers expanding game tree diagram, the node stands for the state, and the branch stands for the move. The root node is the current position. The leaf node is the end position expanded, and it corresponds to a node plies is currently the search depth. The end nodes correspond to the maximum search depth layers.

### 2.2 Traditional Alpha-Beta pruning algorithm

Search algorithm for game tree graph model is to solve the basic method, most of the current game program search algorithms are created on the basis that "Alpha-Beta heuristic search with the replacement table ", which is based on Alpha-Beta pruning search. Alpha-Beta pruning is based on the min-max search algorithm, although it does not travel all nodes of the certain sub-tree, it's still thoroughly completed search. In the ideal case, the number of nodes is described as following mathematical formula (1) and formula (2).

$$ND=2*B*D/2-1 \quad \text{(D is even integer)} \quad (1)$$

$$ND=B*(D+1)/2+ B*(D-1)/2 \quad \text{(D is odd integer)} \quad (2)$$

In the formula, we suppose that B is the branching factor and D is the search depth.

There are many problems in the traditional Alpha-Beta algorithm. First of all, in order to ensure that the node evaluation value falls in the estimated range, it applies the full window search strategy, which slows the speed of convergence in the intervals. Secondly, the algorithm is quite sensitive to the node order. If the node order is good, the efficiency of the algorithm can be significantly better than that of the worse order. Besides, the Alpha-Beta algorithm is completed on the base of the recursive model, and the simple recursive model leads to repeated calculation regarding to the same node, which reduces the efficiency of the algorithm. Finally, the game itself has certain characteristics, i.e., at different stage, the complexity of situation varies obviously. However the traditional Alpha-Beta algorithm, without considering above factors, can not vary with the dynamical situation and adjust the search depth, so that it cannot make full use of the hardware resources [4].

In the ideal case, the bigger the search depth is, the better the efficiency is. However, in the computer games, limited by the computer operation ability and the play time, the big search depth is not distinctly allowed sufficient time and memory space to guarantee the realization of the objectives. In addition, at the opening and middle-game stages, the larger search space and search depth mean the need to consume a large quantity of time. Although establishing reasonable opening database and local database helps to solve the problem, it may cause many optimal or near-optimal solutions to be pruned. At the final stage, where there is seldom pieces, the larger search depth not only wastes more machine time and memory space, but also does harm to finding the optimal solution. So varying with the situation of the game, how to select the appropriate search depth to form a perfect dynamic search is also especially important in the computer games [5].

### 2.3 Dynamic Nonlinear Search technologies based on military strength model

Surakarta game is generally divided into three phases: opening, middle game and end game.

Opening stage is the start of the game and lasts till the moment when and after any side has a first piece to be eaten or the first entering stalemate. This stage focuses on the whole layout. The middle game stage covers the accomplishment of layout and ends in each side occupying respective line or one side occupying one absolute line. The stage put emphasis on how to effectively attack and defense. The end game stage is from the moment when each side occupies one line or one side absolutely occupies the line to the end of the chess game.

For a long time, to create a high level game program, the most time-consuming work is the evaluation function design, realization and manual adjustment. Unlike some of the other computer games, such as the Amazons, Connect6, etc., the chess number of Surakarta game gradually decreases as the game is going on.

At different stages, the number of pieces and the function of each piece to the situation are not the same, so the game assessment search also should make corresponding adjustments. For example, if set single search depth is D = 5, at the game layout stage, as the number of the pieces is large, although the search node number is more, due to the limited mobile space at this stage, it is apparently not necessary to waste a lot of time to make too much search. On the contrary, at the stage of the middle-game, in order to increase the possibility of getting optimal solution and capture the opponent in search tactics of the loopholes, we should search as many leaf nodes as possible with the permission of time, whereas less search depth may means missing a lot of global optimal solution. Along with the reduction in the number of game pieces, the situation changes greatly. It also makes it difficult to grasp the best search depth. At the end-game stage, because the number of pieces has reduced to a certain amount, the game becomes gradually clear. Thus enhancing the search depth can only get a local optimal solution, instead of the global optimal solution.

Therefore, how to adjust the search time varying from minute to minute situation, and make the proper depth search, is a crucial problem affecting the game result. The dynamic search technologies based on fuzzy set model of "military strength" will be helpful to solve the problem.

In the assessment of the game situation, the quantity of pieces is vital. Followed is the location of pieces. A good location may be the key to turning the tide. And the pieces number and the position evaluation form the "military strength". The "military strength" is closely related to the assessment of the situation. The dynamic search based on fuzzy set model of "military strength" assesses the "military strength" under the condition of the constraint time. Based on the formula (3) we establish the "military strength" evaluate function, and dynamic adjust the search depth.

$$MS=num*a1+bnum*a2+wnum*a3 + a0 \quad (3)$$

Among them, the variable bnum is the quantity of black side pieces; the variable Wnum is the quantity of white side pieces; the variable num is the total of both side pieces; the variable a0 is position impact factor; the variable a1, a2 and a3 are called the military strength impact factor. Their initial value can be manually supplied and adjusted appropriately according to the evaluation function. Usually when their pieces in quantity is less than the other side, that game on adverse, should automatically increase search leaf node number, so its initial value is negative. While the a0 evaluates the position evaluation based on the advantages and disadvantages of the situation, and then adjust the search of the leaf node. At different stages of the game, the value of the a0 is also not the same. With the position of pieces changing, the game situation is dynamic.

All kinds of data which cannot be presented as precise numbers or cannot be precisely classified are called nonprecise or fuzzy. Also, precision measurement results of continuous variables are not precise numbers but always more or less fuzzy. So depending on the situation," military strength" evaluation function, based on the experience formula for determining the strength coefficient of the situation, set the corresponding power fuzzy set F ( Fuzzy Set ). It is defined a set U by the following formula (4) [6].

$$U=\{x(1),x(2),x(3),\ldots\ldots,x(n)\} \quad (4)$$

Then for a finite element fuzzy set, available Zadeh method is expressed as the formula (5).

$$A= \{A(x(1))/u1+ A(x(2))/u2....+ A(x(n))/un\} \quad (5)$$

Among them, A is defined as a set x arrives [0, 1] in a mapping, A: X→[0,1], it represents military strength of the board game piece under a certain kind of condition. A(x(n)) is the degree of x(n) subjection the fuzzy set A.

The fuzzy model of the "military strength" is established by following a large number of experiments' data. And according to the changes of the game, global evaluation value is adjusted. And then, combined with the experiments we get the empirical equation. A multiple linear regression model can be defined as following:

Suppose that a multiple linear regression model of the form

$$y_i= \beta_0+\beta_1*X_{i1} + \beta_2* X_{i2} + \ldots + \beta_k*X_{ik}+\varepsilon i \quad (6)$$

is to be fit to data, which we denote as

$\varepsilon_i \sim N(0,\sigma^2)$, $i \square \infty [1-n]$ [7, 8]

So the rule of multilevel search depth is established after multiple linear regression statistics. Multilevel search depth can also be adjusted to get the optimal solution of dynamic search under the multiple constraint conditions.

According to the fuzzy matrix of correlation coefficient, combined with a large number of tests, we fit out the curves of dynamic search depth, which determine the dynamic function of the search depth. Then we can judge whether the evaluation function value meets the defined value, so as to determine the search depth at different stages of the game.

The part pseudo-codes of the Dynamic Nonlinear Search program are given below:

```
Search (Parameters)
{
 Initialize     // variable initialization
 Repeat
 if (chessboard  != empty)
 Increase       // increase the number of the quantity of
                pieces
endif
endrepeat
SetValue       //set  search depth to an initialization value
if (ms <  given value) //judge the military strength
Adjust         //According to some rules to adjust the
               search depth
endif
Initialize     //Hash table initialization
SetValue       //set   evaluation   parameters   to   an
               initialization value
Search
Adjust         //modify the score of evaluation parameter
Release        // release hash table
Return
}
```

Based on the fuzzy set model of "military strength", dynamic adjustment of the search depth process is an iterative process. Along with the game proceeding, change both "military strength" value and the dynamic search depth, and then form a perfect game situation assessment and dynamic search [9].

## 3 Experiments

We performed some different experiments and relative statistical analysis, and tried to verify the performance of the program using different search techniques. To compare the procedure that applies dynamic search technologies based on fuzzy set model of "military strength" with other search technologies, we had two groups of tests. And each group did 20 times independent experiments. Game interface is shown in Fig. 3.

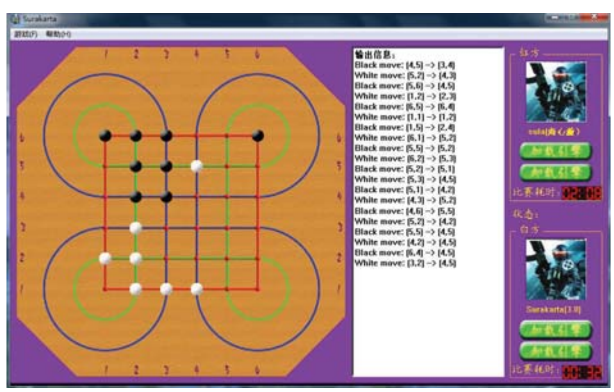

Fig. 3: Contrastive experiment

In order to ensure the impartiality and objectivity of contrast experiments, we executed several different search procedures in the same software condition, which based on the same hardware platform. Then we selected a laptop as the experiment's hardware platform. And the information of hardware and software used for the experiment is shown as follows:

The type of the CPU is Intel Atom N450, which master frequency is 1.67 GHz. The computer has memory capacity of 2GB, and it has hard disk capacity of 250GB. We installed operating system of Windows 7 Home Premium Edition with 32bit in the computer.

### 3.1 Dynamic Nonlinear Depth Search VS Single Depth Search

In the contrast between Dynamic Nonlinear Depth Search based on the fuzzy set model of "military strength" and Single Depth Search (Depth = 5), we wrote programs to do some experiments. And the following is a partial listing of the pseudo-code of the Single Depth Search program.

```
1:Search (Parameters)
2:{
3: Initialize     // variable initialization
4: Repeat
5: if (chessboard  != empty)
6: Increase       // increase the number of the quantity of
                  pieces
7:endif
8:endrepeat
9:SetValue        //set  search depth to an initialization value
10: Initialize    //Hash table initialization
11: SetValue      //set evaluation parameters to an
                  initialization value
12: Search
13: Adjust        //modify the score of evaluation parameter
14: Release       // release hash table
15: Return
16: }
```

After summarized the statistics data, we obtained the experimental results that as shown in Table 1 and Table 2:

Table 1: Results of experiments (Dynamic Nonlinear Depth Search starts firstly)

| Contrastive Items | Dynamic Nonlinear Depth Search | Single Depth Search(Depth = 5) |
|---|---|---|
| Winning Number | 10 | 0 |
| Average Step | 22 | 21 |
| Average Time of Each Game(sec) | 535 | 563 |
| Average Time of Each Step(sec) | 24.3 | 26.8 |

Table 2: Results of experiments (Single Depth Search starts firstly)

| Contrastive Items | Dynamic Nonlinear Depth Search | Single Depth Search(Depth = 5) |
|---|---|---|
| Winning Number | 10 | 0 |
| Average Step | 26 | 26 |
| Average Time of Each Game(sec) | 624 | 605 |
| Average Time of Each Step(sec) | 24 | 23.3 |

In the contrast experiment with Single Depth Search (D = 5), through collecting and analyzing the data, the experimental results are shown in beneath (See detail information in Fig. 4), and from it we can see that: the Dynamic Nonlinear Depth Search based on the "military strength" fuzzy set makes the strength obvious rise, and assesses the game situation more accurately; moreover, it is similar in the used time to experimental contrast object, so its performance is more effectively.

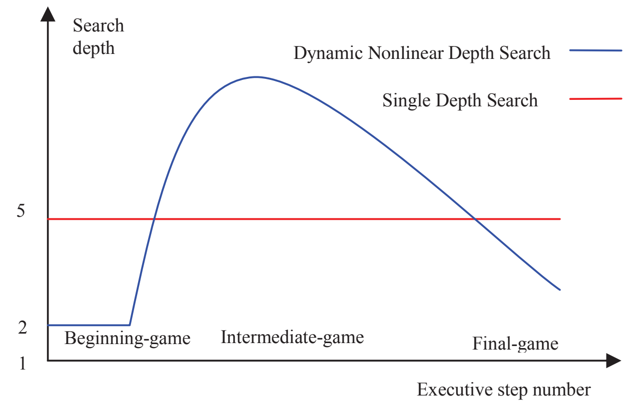


Fig. 4: The correlation curve of search depth

### 3.2 Dynamic Nonlinear Depth Search VS Dynamic Linear Depth Search

In the contrast between the Dynamic Nonlinear Depth Search based on the fuzzy set model of "military strength" and Dynamic Linear Depth Search based on steps, we wrote programs to do some experiments. And the following is a partial listing of the pseudo-code of the Linear Depth Search program.

```
1:Search (Parameters)
2:{
3: Initialize     // variable initialization
4: Repeat
5: if (chessboard  != empty)
6: Increase       // increase the number of the quantity of
                  pieces
7:endif
8:endrepeat
9:SetValue        //set  search depth to an initialization value
10: if (ms <  given value)//judge the military strength
11: Increase      // Linearly increase the search depth
12:endif
13:Initialize     //Hash table initialization
14:SetValue       //set evaluation parameters to an
                  initialization value
15:Search
16:Adjust         //modify the score of evaluation parameter
17:Release        // release hash table
18:Return
19:}
```

After summarized the statistics data, we obtain the experimental results that as shown in Tables 3and 4:

Table 3: Results of experiments (Dynamic Nonlinear Depth Search starts firstly)

| Contrastive Items | Dynamic Nonlinear Depth Search | Single Depth Search(Depth = 5) |
|---|---|---|
| Winning Number | 10 | 0 |
| Average Step | 35 | 34 |
| Average Time of Each Game(sec) | 854 | 1510 |
| Average Time of Each Step(sec) | 24.4 | 44.4 |

Table 4: Results of experiments (Dynamic Linear Depth Search starts firstly)

| Contrastive Items | Dynamic Nonlinear Depth Search | Dynamic Linear Depth Search |
|---|---|---|
| Winning Number | 10 | 0 |
| Average Step | 41 | 41 |
| Average Time of Each Game(sec) | 981 | 1805 |
| Average Time of Each Step(sec) | 23.9 | 44 |

In the contrast experiment with the Dynamic Linear Depth Search based on steps, through collecting and analyzing the data, the experimental results are shown in beneath (See detail information in Fig. 5), and from this we can see that: the Dynamic Nonlinear Depth Search based on the "military strength" fuzzy set compared to the Dynamic Linear Depth Search based on steps, the time is saved nearly 46%, so it controls the search time more reasonably and effectively; at the same time, it assesses the game situation more accurately.

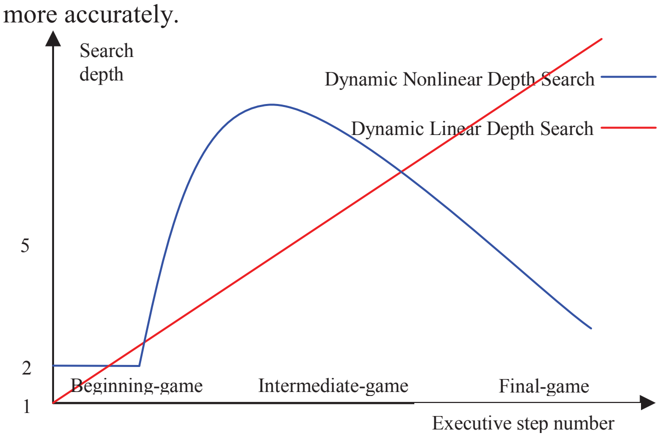


Fig. 5: The correlation curve of search depth

## 4 Conclusion

To sum up, in the Surakarta games, the Dynamic Nonlinear Depth Search based on the fuzzy set model of "military strength" uses the search time more scientific and reasonably, and the game's situation is evaluated more accurately. According to the different situation of the computer game, the software can adjust different search depth dynamically, so that the computer's behaviors are looked as more intelligently.

In the National College Student Computer Games Tournament, better results were obtained by using this method, and they also gave affirmation to the Dynamic Nonlinear Depth Search technique based on the fuzzy set model of "military strength".

However, in the usual tests and actual combats, many problems has been exposed many problems, such as the game situation assessment is not accurate; for the evaluation of "military strength" and the establishment of the correlation coefficient of fuzzy set, which based on a large amount of test samples, need to fit the empirical formula; whether search depth curve can reflect the real situation need further validation; how to design more reasonable experimental to get the test samples, by which we can establish the truth degree of fitting higher regression model.

How to establish the evaluation function more scientifically and accurately, reasonably adjusting the search depth, still remains to be perfected further, and this is our goal.

## References


[1] Thomas.H.Cormen,Ronald.L.Rivest,Charles.E.Leiserson,Clifford.Stein, *Introduction to Algorithms*,2009, chapter 1.

[2] XU Changming, MA Zongmin, XU Xinhe, LI Xinxing, *Study of Temporal Difference Learning in Computer Games*, Computer Science, V01. 37 No. 8:219-223, Aug 2010

[3] M. J. Osborne (Canada), A. Rubinstein (USA), *Computer algorithm of game theory*, China Social Sciences press,2000

[4] Wang Qi,Sun, Jianling, *the Game Tree Search Algorithm Based on Optimized Iterative Deepening*, Computer Applications and Software, V01. 25 No. 2:228-230,Feb. 2008

[5] XU Changming, NAN Xiaofei, WANG Jiao, XU Xinhe, *Adaptive time allocation strategy in computer game of Chinese Chess*, CAAI Transactions on Intelligent Systems, V01. 1 No. 2:39-42,Oct. 2006

[6] Reinhard. Viertl, *Statistical Methods for Fuzzy Data*, Wiley, 2011, chapter 1.

[7] Hogg, Craig, *Introduction to Mathematical Statistics*, 5ed, Pearson 1995, Publishing House of electronics industry, 2003

[8] Moore, McCabe, Craig, *Introduction to the Practice of Statistics*, 6ed, Freeman, 2009

[9] M.H. Alsuwaiyel, *Algorithms Design Techniques and Analysis*, World Scientific, 1999