A Storage Structure and Capture Judging Algorithm of Realizing the Computer Game Program of Surakarta Chess

Liqun Zhang, Lili Ding

Abstract— **Surakarta chess is one of Computer Olympiad chesses. Chessboard, chessman, and chess rule, the three key elements of Surakarta chess, are introduced. The logical structure of Surakarta chess is studied. A storage structure of realizing the computer game program of Surakarta chess is presented. Representation of chessboard and chessman are presented by using the storage structure. The capture circular queue concept is advanced, and the move judging algorithm and capture judging algorithm are implemented, and the chess rules are realized. The finding shows that the storage structure of realizing the computer game program of Surakarta chess is efficient and reliable. The move judging algorithm and capture judging algorithm run accurately and fast. This storage structure and capture judging algorithm have referential value to accomplish the computer game programs of other sorts of chesses.**

Key words: Computer game, Storage structure, Circular queue, Capture judging algorithm

I. The Three Key Elements of Surakarta Chess

Surakarta chess is one of Computer Olympiad chesses. The three key elements, chessboard, chessman, and chess rule, of Surakarta chess are defined as following.

A. Chessboard and chessman

The chessboard and chessmen of Surakarta chess are showed in Fig.1.

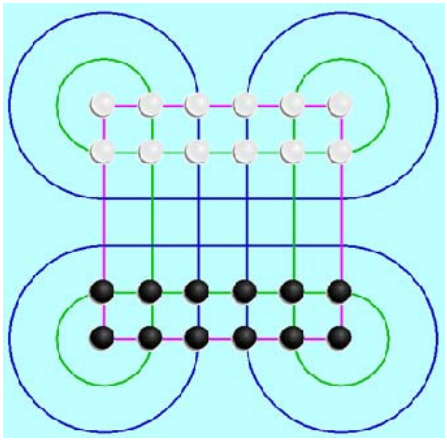

Fig. 1. Surakarta chess

Manuscript received October 18, 2010.

Liqun Zhang is with the Vocational and Technical College, Liaoning Shihua University, Fushun, 113001, P. R. China (0413-6863306; fax: 0413-6863900; e-mail: zzllqun@ sohu.com).

Lili Ding is with the Vocational and Technical College, Liaoning Shihua University, Fushun, 113001, P. R. China (e-mail: s111678@hotmail.com).

The square chessboard has six horizontal lines and six vertical lines. Thirty-six crossing points of horizontal lines and vertical lines are chess piece positions. All these horizontal and vertical lines are linked by eight arc lines. Each side to play the game has twelve chess pieces. To distinguish from each other, two different color chess pieces are used. In the beginning the chess pieces of each side are placed on the first two rows closest to the players, as showed in Fig.1.

B. Rules of Surakarta chess

1) Moves and captures

a) *The two players make moves alternately, and can move one piece a time.*

b) *Ordinarily pieces move one square vertically or diagonally to a position that is not occupied by another piece beside captures.*

c) *Your piece must pass through at least a whole arc line when an opposing piece is to be captured.*

2) Winning

a) *The player wins the game when all the opposing pieces are captured.*

b) *The player with more remainders of the pieces wins within the fixed time.*

II. The Storage Structure of Chessboard, Chessman, and Board Position

A. The storage structure of chessboard

It is obvious that the chessboard can be stored by a two-dimension array [1]. The following definitions and expressions are written by C Language. The chessboard is stored as follows:

```
int b[6][6];
```

The chessboard can be also stored by a one-dimension array as follows:

```
int a[36];
```

The one-dimension array a and two-dimension array b can be transformed by the following formulas:

```
a[i]=b[i/6][i % 6]
b[i][j]=a[i*6+j]
```

Surakarta chess computer game program is realized by a one-dimension array here. This storage structure can only

express the positions of chess pieces. It can't give the logical structure of the full chessboard, such as arc lines. An assisted data structure is employed to denote it, namely circular queue.

B. The storage structure of chessman

The chessmen of Surakarta chess have two colors, each contains twelve chess pieces which are not divided into arms of services any more. While storing the chess pieces, one color pieces are denoted by 1 and the other by 2. In real Surakarta chess game program, every 1 stands for a black chess piece while every 2 for a white chess piece (In order to judge whether there are any chess pieces on the routine while circular queue is used to solve capture problem, 1 or -1 are not used here.).

C. The Storage Structure of board position

Fig. 2 shows a hypothetical board position. The board position can be stored by a one-dimension array a. The values of array a are:

2,2,2,2,2,2,2,2,0,2,0,2,0,0,2,0,2,0,0,0,0,0,1,0,1,1,1,1,1,0,1 ,1,1,1,1,1

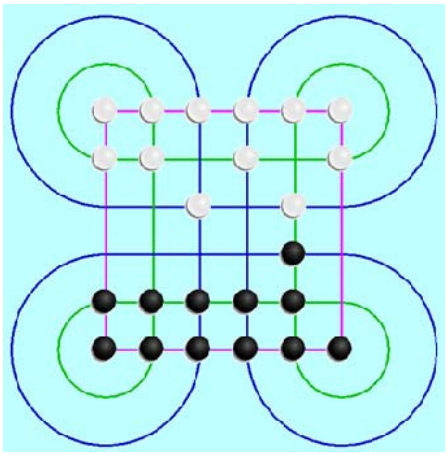

Fig. 2. A board position

III. Search Algorithm and Evaluation Function

During Surakarta chess computer game program running, there must be good Search algorithms and evaluation functions to support computer to generate good moves and captures.

A. Search algorithm

In the course of designing Surakarta chess computer game program, the following three search strategies could be considered.

1) *Exhaustive search*
2) *Selective search*
3) *Goal oriented search*

The game tree of Surakarta chess is large, because each side has twelve chess pieces, and every chess piece can move toward eight directions. Considering chess piece positions of chessboard, suppose there are forty-eight methods to move a piece, and there are seventy moves in a match ,there will be 48^{70} board positions, about 10^{117} board positions. As a result, it is impossible to use exhaustive search.

In certain given hardware conditions, selective search is an effective method. The following search methods may be adopted in Surakarta chess computer game program.

1) *Alpha-Beta search*
2) *Alpha-Beta window search*
3) *Iterative deepening search*

B. Evaluation function

Evaluation function is used to calculate the weighted value of a certain move position by computer. After every weighted value of the possible move position is calculated by evaluation function, the move position of maximum weighted value is selected to be real move position [2]. It is one of the key techniques to define evaluation function of Surakarta chess. It will take long time to practice and update to get the best evaluation function. In this way the level of computer game can reach a higher standard.

Because of the particularity of Surakarta chess, its evaluation function is expressed as follows.

$$p^e = \sum_{i=1}^{4} e_i$$

p^e is the sum of evaluation values.

e_1 is the position evaluation value. The positions of crossing points by blue lines and green lines are the positions with the greatest aggressivity. The four corner positions are the safest position.

e_2 is the capture evaluation value. An offensive move is very important, because two sides can capture opposing pieces each other at a certain time.

e_3 is the chessmen cooperating evaluation value. For example, on the circular routine there are more connect pieces of one side, when a piece is captured by the opposing side, the piece of opposing side can be captured, too. In this way the player can have the predominance.

e_4 is the captured evaluation value.

To calculate evaluation value, Surakarta chess computer game program must first define static rules. When the remaining pieces of two sides reach a certain amount, the static rules are not appropriate. Some static rules are needed to renovate, and evaluation function must be changed with it. At the moment the evaluation function is called dynamic evaluation function.

IV. Capture Judging Algorithm

Capture judging algorithm is one of key techniques of implementing Surakarta chess computer game program. It is

very complicated to implement the algorithm. Different storage structure has its corresponding capture judging algorithm.

A. Circular queue for capture judging

Queue is a liner list with fist-in fist-out (FIFO) characteristic. Circular queue is a sequence queue fabricated by ring storage space [3], [4]. Circular queue is used to solve phony overflow problems.

In rules of Surakarta chess, it must pass through at least a whole arc line when opposing piece can be captured. But when the chessboard is stored, the arc line is not stored at the same time. How to judge if it passes through an arc line? Based on the above storage structure, two orbits are expressed by two circular queues here. At capture, two circular queues are used to judge if the capture passes through a whole arc line, and if there are other pieces on the orbits, namely capture validity.

Two circular queues are used here. Circular particularity of circular queue is utilized instead of insert-operation and delete-operation, and find-operation of liner list is widely used.

Two circular queues are defined as follows:

```
int  q1(24);
int  q2(24);
```

Circular queue q1 represents the green orbit in Fig.1, and it is named q1 ring. The values of Circular queue q1 are as follows:

```
q1[0] = 14   q1[1] = 20   q1[2] = 26   q1[3] = 32
q1[4] = 25   q1[5] = 26   q1[6] = 27   q1[7] = 28
q1[8] = 29   q1[9] = 30   q1[10] = 35  q1[11] = 29
q1[12] = 23  q1[13] = 17  q1[14] = 11  q1[15] = 5
q1[16] = 12  q1[17] = 11  q1[18] = 10  q1[19] = 9
q1[20] = 8   q1[21] = 7   q1[22] = 2   q1[23] = 8
```

Circular queue q2 represents the blue orbit in Fig.1, and it is named q2 ring. The values of Circular queue q2 are as follows:

```
q2[0] = 15   q2[1] = 21   q2[2] = 27   q2[3] = 33
q2[4] = 19   q2[5] = 20   q2[6] = 21   q2[7] = 22
q2[8] = 23   q2[9] = 24   q2[10] = 34  q2[11] = 28
q2[12] = 22  q2[13] = 16  q2[14] = 10  q2[15] = 4
q2[16] = 18  q2[17] = 17  q2[18] = 16  q2[19] = 15
q2[20] = 14  q2[21] = 13  q2[22] = 3   q2[23] = 9
```

B. The realizing of capture judging algorithm

At capture, it needs to judge three cases as follows:

Whether the start point position and the target point position are in q1 ring or q2 ring.

Whether it passes through an arc line.

Whether there are chess pieces on the routine.

1) The definitions and meanings of variables in the algorithm

```
int  frompoint, index;
 /* start point position and the target point position */
int  h1next1, h1next2, h1next3, h1next4;
 /*h1next1 and h1next2 are two directions queue subscripts of next point for start point on q1 ring. Crossing points need h1next3 and h1next4*/
int  h1frompoint, h1index;
 /*The flag of start point position and the target point position in q1 ring*/
int  h1, h2;
 /* The flag of start point position and the target point position in q1 ring or q2 ring */
int  s11,s12,s21,s22;
 /*The sum of value on routine*/
int  h11[23],c11[23];
 /*To store two-dimension subscripts of position on routine*/
int  hb11,cb11, hb12,cb12, hb21,cb21, hb22,cb22;
 /*Eight flag variables*/
```

2) Judging algorithm of the start point and the target point on the same orbit

The judging algorithm of the start point and the target point on q1 orbit is given by the function qlsq(). The values of variable h1frompoint and variable h1index and variable h1 are used in the latter game program. Similarly, this can be applied to the judging algorithm of the start point and the target point on q2 orbit.

```
int qlsq(int frompoint,int index,int q1[])
{
   int i;
   h1frompoint=0;
   for(i=0;i<24;i++)
      if (frompoint==q1[i])
         h1frompoint=1;
   h1index=0;
   for(i=0;i<24;i++)
      if (index==q1[i])
         h1index=1;
   if((h1frompoint==1) &&( h1index==1))
      h1=1;
   else
      h1=0;
   return h1;
}
```

3) Judging algorithm of passing arc lines

The judging algorithm of passing arc lines is given by the function jghp(). The function judges passing arc lines in one direction, and the opposite direction is similar to this. It is needed to judging from four directions to the crossing points of green orbit and blue orbit. Suppose h1=1, in the event of passing arc line of q1 ring, then hb11=1 and cb11=1.The function also gives the sum of expressing value of chess pieces on the routine. The sum is presented by variable s11.

```
int jghp(int frompoint, int index, int h1next1, int q1[])
  {
     int i,j,k,m;
     h11[0]=frompoint/6;
     c11[0]=frompoint%6;
     s11=0;
     j=1;
```

```
    i=h1next1;
    while(q1[i]!=index)
      {
       h11[j]=q1[i]/6;
       c11[j]=q1[i]%6;
       if(q1[i]!=frompoint)
            s11=s11+a[q1[i]];
       i=(i+1)%24;
       j=j+1;
      }
    h11[j]=q1[i]/6;
    c11[j]=q1[i]%6;
    k=1;
    hb11=0;
    cb11=0;
    while((k<=j)&&!(hb11&&cb11))
      {
       if(h11[k]!=h11[0])
       hb11=1;
       if(c11[k]!=c11[0])
       cb11=1;
      }
    m=hb11*cb11;
    return m;
  }
```

4) Capture judging algorithm

Based on the above two algorithms, the capture judging algorithm can be realized. The algorithm is given by the function czpd().

```
int czpd()
{
  if (h1&&(s11==0)&&hb11&cb11)
      chiz();
  else
      if (h1&&(s12==0)&&hb12&&cb12)
          chiz();
      else
          if (h2&&(s21==0)&&hb21&&cb21)
              chiz();
          else
              if (h2&&(s22==0)&&hb22&&cb22)
                  chiz();
              else
                  printf("Illegal move!");
}
```

V. MOVE ALGORITHM

There are move rules and capture rules in Surakarta chess rules. Suppose start point position variable frompoint and target point position variable index are expressed with two-dimension as follows:

a[frompoint]=b[m1][n1]

a[index]=b[m2][n2]

Then while the target point position is not occupied by any piece, move rules will be equivalent to one of three conditions as follows:

m2=m1 ； n2=n1+1 or n2=n1-1

m2=m1+1 or m2=m1-1； n2=n1

m2=m1+1 or m2=m1-1； n2=n1+1 or n2=n1-1

These above-mentioned conditions are named M conditions.

Capture can be regarded as move in general. The generalized move algorithm can be given by the function xzsf() with the capture judging algorithm.

```
void  xzsf(int frompoint, int index)
{
  if(a[index]!=0)
     if(a[frompoint]!=a[index])
         czpd();
     else
         printf("Illegal move!");
  else
     if(M)
         move(frompoint,index);
     else
         printf("Illegal move!");
}
```

VI. CONCLUSION

Surakarta chessboard has arc lines. At capture, the chess piece is asked to pass through at least a whole arc line. These particularities give us many choices of alternative storage structures when realizing Surakarta chess computer game program. This paper has done a research on the logical structure and the storage structure of Surakarta chess as well as showing how to store Surakarta chessboard and chessmen. Based on this kind of storage structure, circular queue of capture judging is involved. It solved the problem of judging whether it passed a whole arc line well. In the end, it gives the algorithms of capture judging and generalized move.

REFERENCES

[1] Xinhe Xu and Jiao Wang, “Key technologies analysis of Chinese chess computer game,” *Journal of Chinese Computer Systems*, vol. 27, pp. 961-969, June 2006.

[2] Liqun Zhang, “Design and realization of Wudao chess computer game program,” *Computer Engineering*, vol. 36, pp. 221-225, May 2010.

[3] Shance Tang and Liusheng Huang, *Data Structure*. Hefei: University of Science and Technology of China Press, 1992, pp. 63-76.

[4] Weimin Yan and Weimin Wu, *Data Structure*. Beijing: Tsinghua University Press, 2002, pp. 63-65.