Masyu Solver

Shi-Jim Yen
National Dong Hwa University
Dept. of CSIE
Taiwan, R.O.C
sjyen@mail.ndhu.edu.tw

Shih-Yuan Chiu
National Dong Hwa University
Dept. of CSIE
Taiwan, R.O.C
d9621005@ems.ndhu.edu.tw

Cheng-Wei Chou
National Dong Hwa University
Dept. of CSIE
Taiwan, R.O.C
d9721002@ems.ndhu.edu.tw

Jr-Chang Chen
Chung Yuan Christian University
Dept. of AM
Taiwan, R.O.C
jcchen@cycu.edu.tw

***Abstract*—This paper proposes an efficient method to solve Masyu, a logical-type puzzle game with N by M grid. By using two methods, intersection method and elimination search, we can solve Masyu quickly and efficiency. The solver is authenticated by solving problems taken from Internet.**

Keywords- Masyu, elimination search, puzzle game

I. INTRODUCTION

Recently, many puzzle games become more and more popular. Rules of these puzzle games are usually very simple, but solving them needs excellent logic concept. With contemplation, it is possible to finish these games, such as Sudoku, Nurikabe, Number Link, and so on.

Masyu is a logical-type puzzle game with NxM grid. About how to solve puzzle game, there are several proposed methods. For example, previous researchers tried to use pattern matching and DFS to solve Nonogram puzzle game[3][4], and had breakthrough in the speed of solving them. However, DFS may step into a wrong track, stuck in it, and cause great deal of resource waste. In addition, GA had been used to solve Nonogram, but the genetic algorithm often got stuck in local optima[8].

To avoid the situation, this research tries to use intersection method and elimination search. By making use of the feature of elimination search which won't get stuck into a wrong track, Masyu solver in this research is able to solve questions of Masyu.

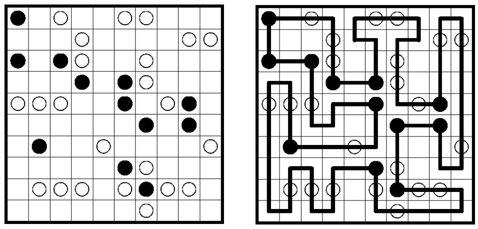
In the second section, we will introduce rules of Masyu in details and the major method used in this research in the third section, including intersection method and elimination search. Experimental results will be presented in the fourth section, and the fifth section is conclusion.

II. GAME RULES

Figure 1 (a) is an initial board of Masyu. There black dots and white dots in the figure. The purpose of the game is to connect all the black and white dots with lines. Correct answers of the game should follow the rules below:

1. All the black and white dots can reach to any dots through lines.
2. Lines passing through black dots have to take a turn, but cannot immediately take another turn.
3. Lines have to pass through white dots immediately, and when passing through a white dot, a line must take a turn on either side of the white dot.
4. Lines should not have branches.

Figure 1 (b) shows the way the game is finished.



(a) (b)

Figure 1 : (a) is an initial borad of Masyu, and (b) is the way it is finished

III. OUR METHOD

A. Overall Process

Ways we use to solve Masyu are intersection method and elimination search. Before using the two ways, we have to do some basic tests, and the whole process is in Fig. 2.

The three sections below illustrate the ways we use. Section III.B explain basic tests. Section III.C illustrates intersection method, and section III.D illustrates elimination search method

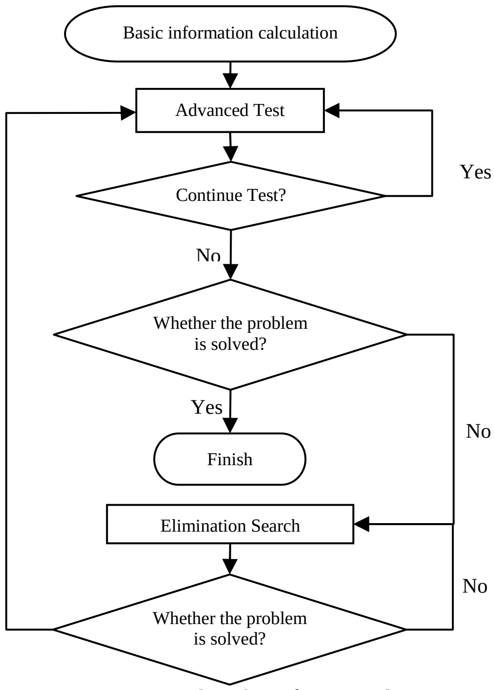
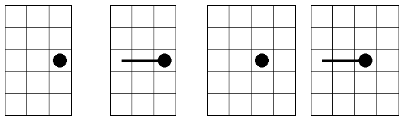


Figure 2 : Flow chart of Masyu Solver

B. Basic information calculation

Before using intersection method and elimination search, we have to do some preparation, which means basic test. According to heuristics, we find out the way lines passing through black and white dots in some situation. Here, we use five kinds of heuristic described below.

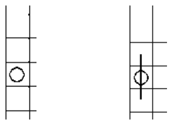
First, if a black dot is located near the edge of the question, like Fig. 3(a) and (c), there must be a line stretching from the black dot, like Fig. 3(b) and (d).



(a) (b) (c) (d)

Figure 3 : Heuristic 1

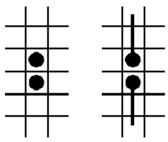
Secondly, if a white dot is located at the edge of the question, like Fig. 4(a), there must be a line passing through the white dot horizontally, like figure Fig. 4 (b).



(a) (b)

Figure 4 : Heuristic 2.

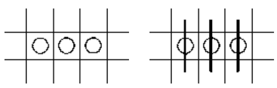
Third, if there are two black dots next to each other like Fig. 5(a), there must be two lines with opposite directions, like Fig. 5(b).



(a) (b)

Figure 5 : Heuristic 3.

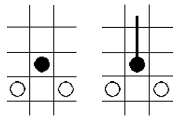
Fourth, if there n white dots next to each other, n>=3, like Fig. 6(a), there will be n lines passing through the n white dots horizontally like Fig. 6(b).



(a) (b)

Figure 6 : Heuristic 4.

At last, if there are two dots next to black dots in an oblique angle like Fig. 7(a), there will be a line stretching out in the opposite direct of the two white dots.



(a) (b)

Figure 7 : Heuristic 5.

C. Intersection method

The major concept of intersection method is to find out the common part of all the possibilities in various possibilities to get information. For example, there four possible patterns in every black dot, like Fig. 8; every white dot has 8 possible patterns, like Fig. 9. When intersecting, we will exclude impossible situations and find out absolute possibilities by making use of the rest of possibilities.

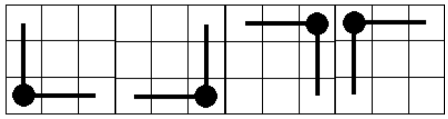


Figure 8 : All the possible patterns of black dots

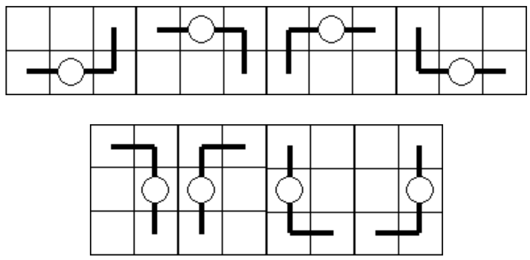
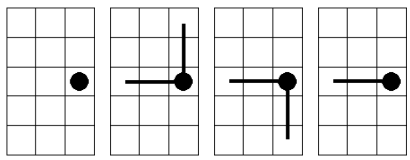


Figure 9 : All the possible patterns of white dots

For example, Fig. 10(a) is a black dot located at the edge, whose possible extending lines are like figure Fig. 10(b) and (c). Because both of the two ways have lines stretching to the center, according to the concept of intersection, we can suppose that there must be a line stretching to the center like figure Fig. 10 (d).



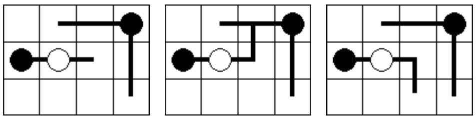
(a) (b) (c) (d)

Figure 10 : An example of intersection

There are four impossible situations below:

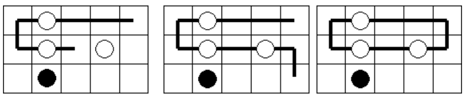
1. Lines with branches: a line should not have branches. Thus, the situation causing branches is impossible situation, like Fig. 11(a). Meanwhile, there are two possible ways of stretching for the white dot in the center, like Fig. 11(b) and (c). At the same time, if the white dot stretches in the same way like Fig. 11(b), there will be branches, which is against rule 4.

Therefore, the white dot can only stretch like the way in figure Fig. 11(c).



(a) (b) (c)
Figure 11 : Dots with branches

2. Line loops: According to rule 1, all dots can reach to another through lines. Thus, if there are line loops without passing through all dots, it is against rule 1. In Fig. 12(a), white dots without lines passing through have two ways of stretching lines like Fig. 12(b) and (c). However, if white dots stretch like Fig. 12(c), there will be line loops without passing through all the black and white dots, which is against rule 1. Therefore, it is an impossible situation.

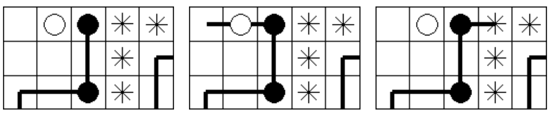


(a) (b) (c)

Figure 12 : Possible situations causing line loops

3. Dead lines: For an empty spot P in the game, on which there will be more than three directions unable to be passed in the four directions of the spot horizontally and vertically, this spot is called dead spot. Situations of being unable to be passed are as follows: 1. the direction causing branches; 2. the spot is on the edge; 3. the spot is a dead spot.

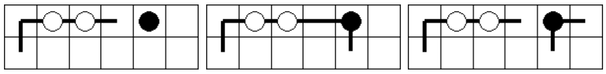
Once lines reach dead spots, because lines cannot stretch to any directions legally, this will make a game unable to be solved, they are dead lines. Like Fig. 13(a), black dots have two possible stretching ways like figure Fig. 13(b) and (c), and sign * means dead spots. If black dots choose the way of stretching like figure Fig. 13(c), there will be dead lines. Thus, black dots can only choose the way of stretching in figure Fig. 13(b).



(a) (b) (c)

Figure 13 : An example of dead lines

4. Causing errors for other dots: like Fig. 14(a), the black dot has two ways of stretching, like Fig. 14(b) and (c). However, if the black dot chooses the way of stretching in Fig. 14(b), it will cause white dots against rule 3, in which either side of a white dot must has lines with an immediate turn. This will cause other dots errors, and becomes an impossible situation.



(a) (b) (c)
Figure 14 : An example of causing other dots errors

D. Elimination search

Like Fig. 15, when there is no more new information by using intersection method, elimination search must be used by testing all kinds of possibility, deleting wrong connection match to increase possibilities of new information of intersection method. The process of elimination search is divided into three steps: they are assumption, extending ratiocination, and error testing.

Assumption: assumption means to assume some possibility is true for some island, and keep doing intersection to change the board according to the assumption.

Extending ratiocination: using intersection method to suppose more information

Error testing: judging if there are errors for the board after extending ratiocination

We use an example to illustrate the three steps below. We suppose the board is like Fig. 15. At present, if intersection method can no longer solve any information, we will select an uncertain spot and assume some way for lines to pass is true. For example, in Fig. 15, for the white dots in the black circle in the center of the top, we do assumption for the way they stretch outward. Then the board will become Fig. 16, and the dotted lines are the ways of connection in this assumption.

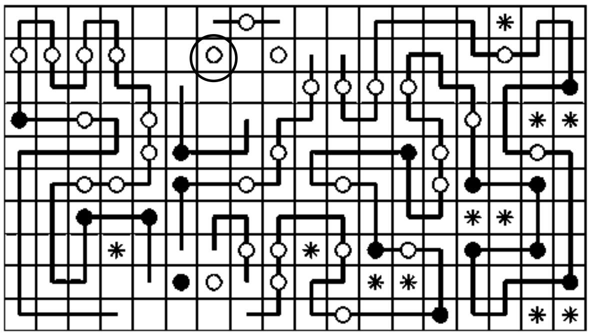


Figure 15 : The board which must be dealt with by elimination search

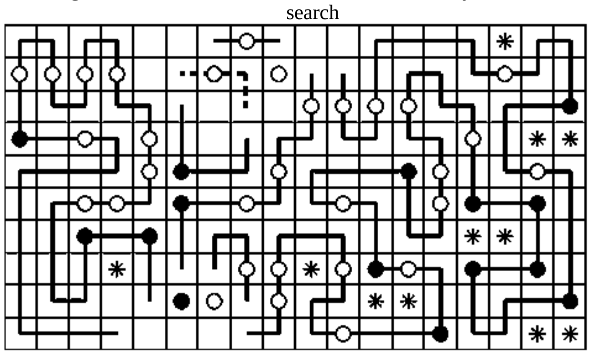


Figure 16 : An example of step “assumption” in elimination search

Then, use extending ratiocination according to the new board, and do error testing to the board after using extending ratiocination to make sure whether it is against rules. Figure 17 is the board after using extending ratiocination for Fig. 16. The dotted lines are the results of extending ratiocination. After doing error testing, we can find that lines in the black circle will have branches no matter how they stretch, leftward, rightward, or downward. Thus, the lines cannot keep stretching out, and cannot connect to other lines, causing errors of dead lines.

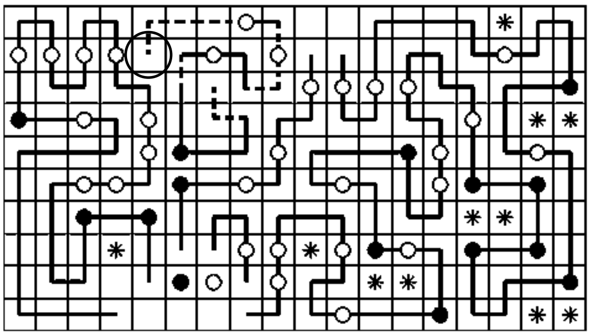


Figure 17 : The board after using intersection

If it is against rules, that means the assumption is not true. On the other hand, we can exclude a possibility and get new information. Because elimination search is slower than intersection method, as long as elimination search solves new information, we will use intersection method to test if it can solve more new information. If we do error test and find nothing against rules, we keep conducting elimination search.

IV. EXPERIMENT RESULTS

We use the method of section three to actually solve Masyu problems and list the results here. We use MATLAB 2009 to practice Masyu Solver, and the objects are the questions in Masyu text book[5][6], 182 questions as the total number. Our experimental system is composed of MD E8400 CPU, 8G RAM, and the operation system is VISTA64. Experimental results are listed in table 4.1.

From the experimental results, we can see that intersection method is 10~1000 times faster than elimination search. Their speed difference gets bigger while the question gets bigger. This improves that it is correct for us to immediate use intersection method right after using elimination search to get new information.

TABLE I. COMPARISON BETWEEN INTERSECTION METHOD AND ELIMINATION SEARCH

size of question	average time of solving a question	The rate of using eliminatio n search	The average time taken by elimination search to solve every 1% information	The average time taken by heuristic and testing every 1% information
10X10 (36 questions)	0.0376	3.53%	28*10e-4	2.88*10e-4
18X10 (56 questions)	1.1329	22.56%	473*10e-4	8.48*10e-4
24X14 (62 questions)	18.8176	29.74%	6269*10e-4	25*10e-4
36X20 (28 questions)	268.0739	32.75%	81719*10e-4	66*10e-4

V. CONCLUSION

This research studies how to quickly solve Masyu. This research tries to use intersection method and elimination search to solve Masyu, and proves that this method can quickly and effectively solve it. According to the experiment results, we can find that at most 32.75% fields need to be solved by using elimination search. For every question, 67%~96% information is found by intersection method in average. With the larger board, more and more fields have to be solved by using elimination search, and spent more time. Reducing the number of using elimination search will be the possible direction of improving the Masyu solver.

REFERENCES

[1] Brandon McPhail, “Light Up is NP-Complete,” Feb. 2005. URL: http://www.reed.edu/~mcphailb/lightup.pdf.

[2] Graham Kendall, Andrew Parkes and Kristian Spoerer, “A Survey Of NP-Complete Puzzles, ” ICGA Journal, vol. 31, no. 1, pp. 13-34, March 2008.

[3] Min-Quan Jing, Chiung-Hsueh Yu, Hui-Lung Lee and Ling-Hwei Chen, “Solving Japanese Puzzles with Logical Rules and Depth First Search Algorithm, ” Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009.

[4] Chiung-Hsueh Yu, Hui-Lung Lee and Ling-Hwei Chen, “An Efficient Algorithm for Solving Nonograms, ” Applied Intelligence , 13 November 2009.

[5] Database of Japanese puzzles : Nicoli.2009. Masyu1 7thed,Japan: Nikoli Co., Ltd..

[6] Database of Japanese puzzles : Nicoli.2009. Masyu3 3thed,Japan: Nikoli Co., Ltd..

[7] Gary McGuire, “Sudoku checker and the minimum number of clues problem, ” http://www.math.ie/checker.html.

[8] Wiggers, W.: “A comparison of a genetic algorithm and a depth first search algorithm applied to Japanese nonograms, ” Proceedings of the 1st Twente Student Conference on IT, pp. 1–6, 2004.