Hashi 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

Tsan-Cheng Su
National Dong Hwa University
Dept. of CSIE
Taiwan, R.O.C
m9621008@ems.ndhu.edu.tw

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

Keywords- Hashi, 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.

Because of the popularity of puzzle games, many related studies become very popular. There are some directions for these studies. Some put focus on the generation of questions, trying to use the least implied numbers to generate a question with only one answer [6]; some try to demonstrate some puzzle games are NP-Complete problems [1][2]; besides, studying how to use programs to rapidly solve these puzzle games is an important direction. Previous researchers tried to use pattern matching and DFS to solve Nonogram puzzle games [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. To avoid this 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, Hashi solver in this research is able to solve questions of Hashi.

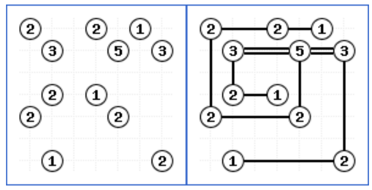
In the second section, we will introduce rules of Hashi in details and the major method used in this research in the third section, including intersection method and elimination search. Experimental results will be presented in the fourth section, and the fifth section is conclusion.

II. GAME RULES

The initial board of Hashi is shown in Fig. 1(a). In the figure, the circled numbers are "islands;" in addition, the lines connect islands are "bridges." The goal of the game is to connect all the islands with bridges. Correct solutions of the game must follow rules below:

1. Every bridge can only connect two islands vertically or horizontally, without any other islands between the two islands.
2. The numbers of all the bridges connecting islands must be equal to the numbers on the islands.
3. The number of every island will be between 1~8, and 2 bridges at most are allowed in every direction of an island, up, down, left, and right.
4. A bridge cannot cross another or has branches.
5. Every island can reach to any other through bridges.

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



(a) (b)

Figure 1 : An example of a Hashi problem (a) and its solution (b)

III. OUR METHOD

A. Overall Process

The way we use to solve Hashi is intersection method and elimination search. Before using these two methods, we need to do some basic information calculation. The whole process is shown in Fig. 2.

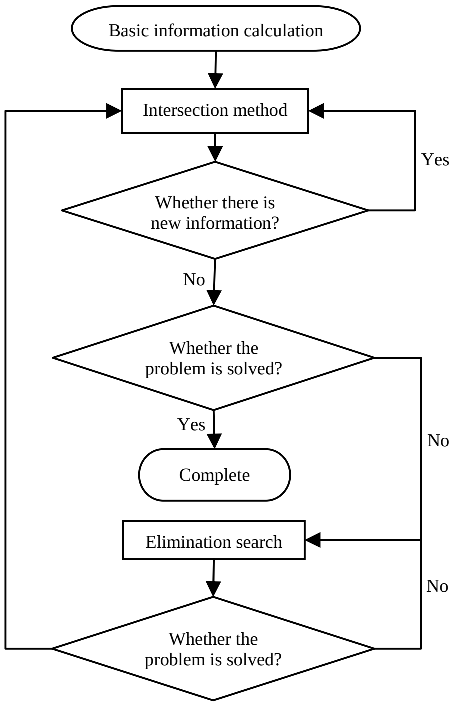
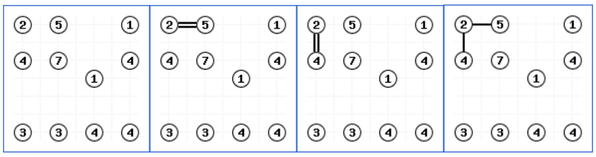


Figure 2 : Flow chart of Hashi Solver

The following three sections will illustrate the methods we use. Section III.B explains basic information calculation. Section III.C explains intersection method, and Section III.D illustrates elimination search method.

B. Basic information calculation

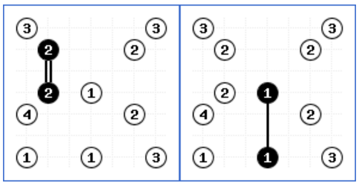
Before using intersection method and elimination search, some preparation has to be done, which is basic information calculation. This step mainly records possible ways for every island to connect. Taking Fig. 3(a) for example, the island with number 2 on the upper left corner has three ways to connect, listed in Fig. 3(b)(c)(d).



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

Figure 3 : Possible ways for a island to connect

In addition to the ways of connection, we do some simple judgments to decrease possible ways of connection, increasing the possibility of finding out solutions of intersection method, and decrease the use of elimination search. The way to judge is to delete impossible connection. For example, in Fig. 4(a), if two islands with number 2 connect each other with two bridges, they are impossible to connect other islands, which is against rule 5. Thus, in this step, this possible will be deleted; likewise, in Fig. 4(b), two islands with number 1 connecting together will be against rule 5.



(a) (b)

Figure 4 : Connections against rules will be deleted.

Table I lists how the possible connections of two highlighted islands with number 2 will change after the judgment above. Here, we use U(Up), D(Down), L(Left), and R(Right) to represent the connecting directions around a island, and add numbers after directions to show the numbers of bridges. For example, U2 means the island has two bridges connecting the island in the up direction.

TABLE I. HOW THE POSSIBLE CONNECTIONS OF THE TWO HIGHLIGHTED ISLANDS WITH NUMBER 2 IN FIG.4 WILL CHANGE

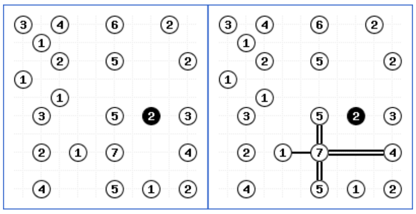
	original possible connections	**impossible connections**	**actual possible connections**
Up islands	R2 D1 R1 D2	R2 D2	D1 R1
Down islands	U1 R1 U2	U2	U1 R1

C. Intersection method

We can use an example to illustrate the main concept of intersection method. Assume an island only has two connecting ways, D1 L1 R2 and D2 L1 R1. Because both of the two possibilities include D1 L1 R1, we can believe that the three bridges, D1, L1, and R1, must exist. In other words, finding possible points in common among lots of possibilities to get information is the concept of intersection method.

Before using intersection method, according to present board, we can use more advanced way of sieving to decrease the possibility of matching so that the intersection method can probably get more information. We use three major ways of sieving here.

Because bridges cannot have branches or cross each other, if a and b are separated by bridges not connecting them, there will be no possibility for the two islands to connect. The first kind of sieving is used to exclude these impossible matches. For example, like Fig. 5(a), there are five possible connections for the highlighted island, but through some information calculation, the board situation will change into Fig. 5(b). The highlighted island's connection downward is blocked, so all the connections including Down are excluded for its impossibility. Table II shows the specific information.



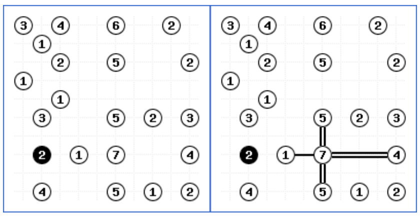
(a) (b)

Figure 5 : An example of first kind of sieving

TABLE II. HOW THE POSSIBLE CONNECTIONS OF THE HIGHLIGHTED ISLANDS WITH NUMBER 2 IN FIG.5 WILL CHANGE

original possible connections	**impossible connections**	**actual possible connections**
R2 L2 R1 L1 D1 L1 D1 R1	D1 L1 D1 R1	R2 L2 R1 L1

When some islands construct bridges, their ability of containing new bridges will decline; therefore, we can make use of this information to exclude some impossible connections. Like Fig. 6(a), there are five possible connections for the highlighted island, but, after some basic information calculation, the board changes into Fig. 6(b). The island with number 1 on the right of the highlighted island has one bridge, so its capability of containing new bridges is 0. Thus, all the possibility of connecting islands on its right side will disappear, and all the matches that include connections with islands on the right will be excluded. Specific information is in Table III.



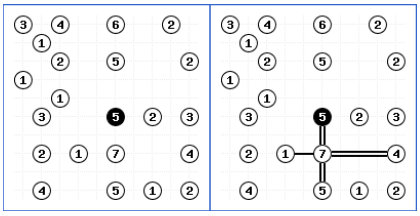
(a) (b)

Figure 6 : The second kind of sieving

TABLE III. HOW THE POSSIBLE CONNECTIONS OF THE HIGHLIGHTED ISLANDS WITH NUMBER 2 IN FIG.6 WILL CHANGE

original possible connections	**impossible connections**	**actual possible connections**
U2 D2 U1 D1 U1 R1 D1 R1	 U1 R1 D1 R1	U2 D2 U1 D1

At last, considering the original connections of an island itself can effectively exclude some impossibility. Like Fig. 7(b), because the highlighted island already has two bridges downward, all the connections excluding the two can be excluded. Specific information is in Table IV.



(a) (b)

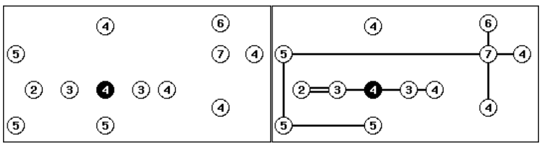
Figure 7 : The third kind of sieving

TABLE IV. HOW THE POSSIBLE CONNECTIONS OF THE HIGHLIGHTED ISLANDS WITH NUMBER 5 IN FIG.7 WILL CHANGE

original possible connections	**impossible connections**	**actual possible connections**
U1 R2 L2	U1 R2 L2	
U2 D2 L1		U2 D2 L1
U2 R1 L2	U2 R1 L2	
U2 D1 L2	U2 D1 L2	
U2 R2 L1	U2 R2 L1	
U1 D2 L2		U1 D2 L2
D2 R1 L2		D2 R1 L2
U1 D2 R2		U1 D2 R2
D1 R2 L2	D1 R2 L2	
U2 D1 R2	U2 D1 R2	
D2 R2 L1		D2 R2 L1
U2 D2 R1		U2 D2 R1

Before using intersection method to find information, using three kinds of sieving above can greatly decrease possible connections for the benefit of using intersection method. For example, if the present board is like Fig. 8(b), use the three kinds of sieving method to delete impossible connection matches. The result is listed in Table V. The red marks are impossible connections after using every kind of sieving. Once a connection match is considered to be impossible, it will be deleted as Invalid.

After sieving, there are two possibilities. After intersecting the two possibilities, we can get the conclusion that there is at least one bridge downward, and can condense original 19 possible connections into 2.



(a) (b)

Figure 8 : A board before using intersection method

TABLE V. THE RESULT OF USING THE THREE KINDS OF SIEVING METHOD TO DELETE IMPOSSIBLE CONNECTION MATCHES

original possible	**Sieving method 1**	**Sieving method 2**	**Sieving method 3**	**result**
U0 D0 L2 R2	Possible	*Impossible*	Possible	~~U0 D0 L2 R2~~
U2 D0 L1 R1	*Impossible*	Possible	Possible	~~U2 D0 L1 R1~~
U0 D2 L0 R2	Possible	Possible	*Impossible*	~~U0 D2 L0 R2~~
U1 D0 L2 R1	*Impossible*	*Impossible*	Possible	~~U1 D0 L2 R1~~
U0 D2 L2 R0	Possible	*Impossible*	*Impossible*	~~U0 D2 L2 R0~~
U1 D0 L1 R2	*Impossible*	Possible	Possible	~~U1 D0 L1 R2~~
U2 D0 L0 R2	*Impossible*	Possible	Possible	~~U2 D0 L0 R2~~
U2 D1 L0 R1	*Impossible*	Possible	Possible	~~U2 D1 L0 R1~~
U2 D0 L2 R0	*Impossible*	*Impossible*	*Impossible*	~~U2 D0 L2 R0~~
U1 D2 L0 R1	*Impossible*	Possible	Possible	~~U1 D2 L0 R1~~
U2 D2 L0 R0	*Impossible*	Possible	*Impossible*	~~U2 D2 L0 R0~~
U1 D1 L0 R2	*Impossible*	Possible	*Impossible*	~~U1 D1 L0 R2~~
U0 D2 L1 R1	Possible	Possible	Possible	**U0 D2 L1 R1**
U2 D1 L1 R0	*Impossible*	Possible	*Impossible*	~~U2 D1 L1 R0~~
U0 D1 L2 R1	Possible	*Impossible*	Possible	~~U0 D1 L2 R1~~
U1 D2 L1 R0	*Impossible*	Possible	*Impossible*	~~U1 D2 L1 R0~~
U0 D1 L1 R2	Possible	Possible	Possible	**U0 D1 L1 R2**
U1 D1 L2 R0	*Impossible*	*Impossible*	*Impossible*	~~U1 D1 L2 R0~~
U1 D1 L1 R1	*Impossible*	Possible	Possible	~~U1 D1 L1 R1~~

D. Elimination search

Like Fig. 9, when intersection method cannot find any new information anymore, elimination search should be used. By testing every kind of possibility to delete wrong connection match, we can increase the possibility of finding new information.

Elimination search's process is divided into three steps: assumption, extending ratiocination and mistake testing.

STEP1. Assumption: It means to assume some possibility is true for some island, and keep intersecting to change the board situation.

STEP2. Extending ratiocination: Use intersection method to ratiocinate to find more information.

STEP3. Mistake testing: Judge if there is mistake for the situation after extending ratiocination.

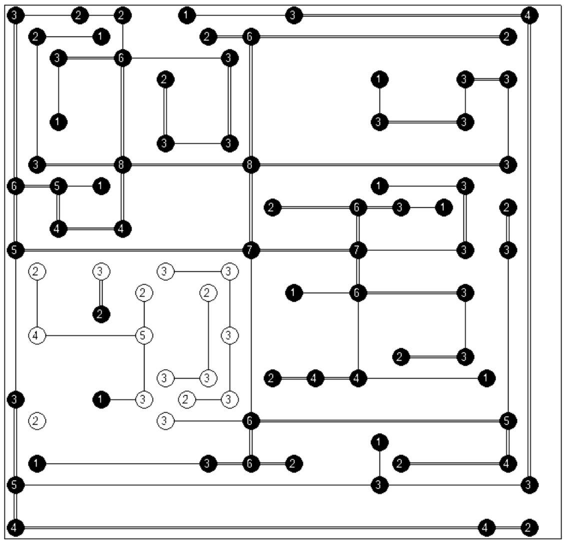


Figure 9 : A board that should use elimination search

We use an example below to illustrate the three steps. Assume the present board is like Fig. 9. Highlighted islands represent those solved, and other islands unsolved. Meanwhile, because intersection method cannot find any information, we will select an unsolved island, and assume some connection is true. For example, possible connections for the unsolved island with number 2 on the left button are R2, U2, and R1U1. Assume the island's connection is U2, and the board will become Fig. 10. The dotted lines are assumed connections.

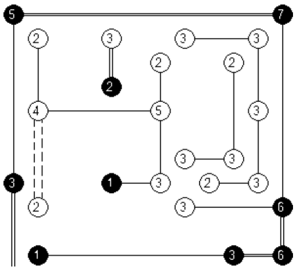


Figure 10 : "Assumption" step in a elimination search

Next, use extending ratiocination for the new board, and use mistake testing for the board with extending ratiocination to make sure whether it is against rules. Fig. 11 is the board after using extending ratiocination for Fig. 10. The dotted lines are results after using extending assumption. After using mistake testing, we find that because the nine islands in the circle on the left in Fig. 11 cannot connect any direction , they are against rule 5. Besides, the island with number 3 in the center on the up direction can only connect the right direction which is against rule 2 because the capability to contain new bridges of islands on its left, right, and down is 0.

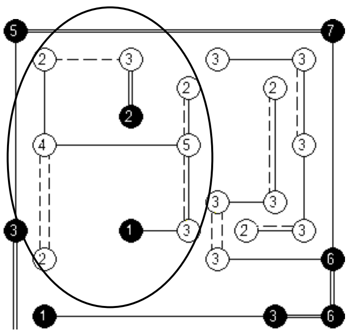


Figure 11 : The board after using intersection method

If it is against rules, it is proved that the assumption is not true. In other words, one possibility is excluded, which means we get new information. Because elimination search is slower than intersection method, as long as elimination search has new information, we will use intersection method to test if more new information can be found. If it is not against rules after mistake testing, keep using elimination search. In the example above, if the assumption is not true, possible connections for the island with number 2 in the center on the left are U1R1 and R2. After using intersection method, it is found that there must be a bridge connecting the island on the right.

The worst situation is that after using elimination search for all the islands' possibilities, the problem still cannot be solved. However, when we test questions of large size (25X25), we haven't found questions that intersection method and elimination search method cannot solve. Thus, we believe merely using intersection method and elimination search method is enough to deal with the need of solving Hashi.

IV. EXPERIMENT RESULTS

We use the method in the third section to actually solve questions of Hashi, and list the result here. We use MATLAB 2009 to implement Hashi Solver, and randomly select one thousand questions of 25X25 to the tested objectives from Internet[5]. Our experimental environment is formed by using AMD E8400 CPU, 8G RAM, and the operation system is VISTA64. Testing the one thousand questions, the average time took to solve every question is 0.0342 seconds. Among them, we need only intersection method to solve 76.6% questions without using elimination search, and these questions are those which can be solved without using the rule that "all the islands must connect together." About the comparison between intersection method and elimination search, we list it in Table VI. From the experimental results, intersection method is four times faster than elimination search, which proves that it is a correct way to use intersection method right after finding new information by using elimination search.

TABLE VI. THE COMPARISON BETWEEN INTERSECTION METHOD AND ELIMINATION SEARCH

method	average proportion of solving information for each question	average time it takes to solve every question	average time it takes to solve every 1% information
intersection	98.9%	0.0327s	3.3064 x 10^{-4}s
elimination search	1.1%	0.0015s	13.6364 x 10^{-4}s

V. CONCLUSION

This research studies how to quickly solve Hashi. This research tries to use intersection method and elimination search to solve Hashi, and proves that this method can quickly and effectively solve it. According to the experiment results, we can find that over three fourths questions need only intersection method to be completely solved, and the rest of questions can be solved by using elimination search. For every question, 99% information is found by intersection method in average, and only 1% information is provided by elimination search. This proves that the two ways have great performance in solving Hashi. At the same time, because the two methods do not need much domain knowledge basically, we can also try to apply them to other puzzle games and expect there will be great results.

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] http://www.puzzle-bridges.com/

[6] Gary McGuire, "Sudoku checker and the minimum number of clues problem," In http://www.math.ie/checker.html.