Hitori Solver

Shi-Jim Yen[1],Tsan-Cheng Su [2],Shih-Yuan Chiu[3]
[1]Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. sjyen@mail.ndhu.edu.tw
[2]Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. d9821009@ems.ndhu.edu.tw
[3]Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. d9621005@ems.ndhu.edu.tw

Abstract

Hitori, a logical puzzle game invented by Nikoli company in Japan, has its full name Hitori ni shite kure, meaning “let me alone”. The way to play Hitori, arranging every number to appear only one time in every row, has something similar to Sudoku. However, the difference is that a Hitori player has no need to fill in blanks with numbers; instead, he has to erase numbers appearing more than one time in a row.

Keywords: Hitori, pattern search, puzzle Game.

Introduction

This paper proposes a solver using pattern match and local search to quickly solve Hitori puzzle game. First, use the rule of thumb to deal with most part of the region in a very short time. Then, use the local search method to complement the part in which the rule of thumb is not able to solve. After authenticating our method by solving the games from Internet [1], the conclusion demonstrates that our approach is fast and efficient.

Detective algorithm

Hitori is a puzzle game of logical type. It is constructed by M*N grids, and every number in the grids would either be erased or kept in the end of the game. In order to have every number appear only for one time in every row, players need to erase some specific numbers. In addition, there shouldn’t be

two erased numbers next to each other. Besides, the numbers without being erased should connect to each other.

For precise rules and questions, please refer to website[1].The website is on http://www.menneske.no/hitori/17x17/eng/

Our solver is divided into three processes. First, use four patterns collected by the rule of thumb to deal with the whole region. The second ,search the region in which the pattern is unable to deal with by using basic rules. For example, the number, which appears only one time in every row, has no need to be erased. At last, use local search to solve the part both the former methods cannot deal with.

I. Basic search

The collected patterns are as follows.

In figure (a)：If the number M=N in the corner, Q must be kept.

In figure (b)：If there are the same numbers A and B next to each other in the same row, other same numbers C and D must be erased:

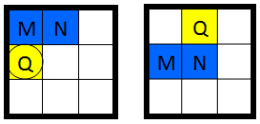
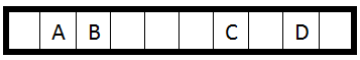


figure (a)

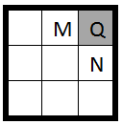


figure(b)

In figure (c)：The number in the middle of two same numbers should be kept:

In figure (d)：If M, N, and Q are the same numbers, Q must be erased.

figure(c)



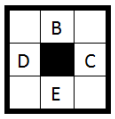
figure(d)

II. Advanced search

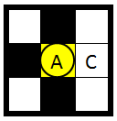
In figure (e)：It is based on the rule that any two erased numbers must not

close to each other, so when a number is erased, the numbers next to it in the position up, down, left, and right, should be kept.

In figure (f)：Because all the numbers without being erased should connect to each other, grid C, a number will be against the rule if erased, must be kept when grid A, a number surrounded by erased numbers, is sure to be kept.



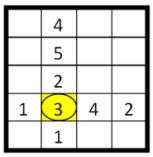
figure(e)



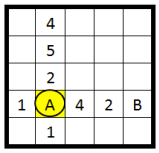
figure(f)

In figure (g)：the number “3” must be kept when there isn’t the same number in the horizontal row and vertical row of it.

In figure (h)：if A=B and A is sure to be kept, the same number B needs to be erased.



figure(g)



figure(h)

After the two searches, if there are numbers unsolved, like in figure (i), we will use local search to deal with the rest numbers.

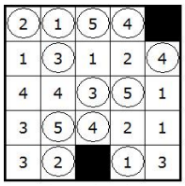
The following shows how local search works:

I. Do the following tests to every spot without being circled in figure (i):
   a. Suppose this spot should be kept, and use advanced search to do tests until it is unable to do any more tests.
   b. Check if there is any error in the board after process a, and resume the board to be the situation before process a.
   c. If there is any error in supposition process a, erase this supposed spot.

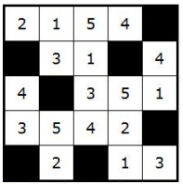
II. Do advanced search to the board again after being updated in process I.

III. Repeat I and II until all the questions are solved like figure (j).

Experimental



figure(i)



figure(j)

We find 140 games of 17*17 Hitori question from the website, http://www.menneske.no/hitori/17x17/eng/ , to testify the correctness of our approach.

size of the board	MiniSAT	Matthias Gander	Our Algorithm
17*17	3.262s	1.499s	0.0573s

We use matlab2009 to write the program with CPU Intel E8400, 8G ram, and Windows Vista 64bit.

Conclusion

The result of our experiment demonstrates that our solver can solve Hitori questions taking 0.0573(s) in average for each question. Comparing with Matthias Gander[2] and MiniSAT[3], our solver is very efficient. And our algorithm can be used in many game solvers and we can quickly solved the questions of games.

Reference

[1]Hitori puzzles，http://www.menneske.no/hitori/17x17/eng/

[2]Matthias Gander, Christian Hofer, Hitori Solver, 8th April, 2006

[3]http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/