Universiteit Leiden

Master Computer Science

Solving, Generating and Classifying HITORI

Name: Roos Wensveen

Date: August 7, 2024

Specialisation: Computer Science

1st supervisor: dr. Walter A. Kosters
2nd supervisor: dr. Hendrik Jan Hoogeboom

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

HITORI is a single-player puzzle, where the goal is to eliminate all duplicates on a given line, i.e., horizontally or vertically, for an $m \times n$ grid filled with characters. Doing so, one must preserve connectivity between those cells that have not been eliminated, and neighboring cells cannot be eliminated. In this thesis, we will provide a difficulty measure for HITORI and consider the generation and solving of the puzzle. Introducing rules based on a single line, multiple lines, connectivity and probing, these puzzles have been classified accordingly. Besides this rule-based classification, a satisfiability-based classification will be considered as well. Based on the principle of Bell numbers and pruning, HITORI will be generated. An analysis of some board characteristics will be provided as well.

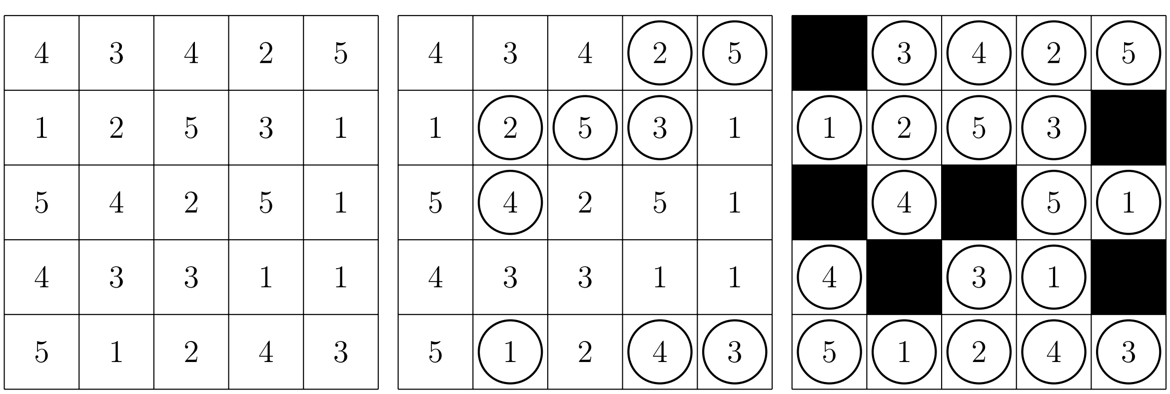
Contents

1 Introduction

Hitori, which can literally be translated to "alone", is a Japanese logic puzzle invented by Nikoli [10]. Originally, Hitori considered a sequence of numbers placed in an $n \times n$ grid. The goal of the puzzle is to get a grid that contains no duplicate numbers in every row and column by eliminating some of the given numbers. Even though the original Hitori solely consists of a sequence of numbers, a set of characters, symbols, or a combination of the two can be used as well. When solving a Hitori, we can use a circle to indicate that a cell should be left as is, and is thus left *white*. If we want to eliminate a cell, we will cross out the cell by making it *black*. There are three rules to take into account:

1. No duplicate characters may occur in a row or column.
2. Black cells cannot be adjacent to one another.
3. All white cells must be connected via their direct, i.e., horizontal and vertical, neighbors. This excludes the diagonal neighbors of a cell.

Figure 1 shows a Hitori puzzle, an intermediate step, and the final solution. If a character occurs only once in its corresponding row and column, there is no need to cross out this cell. Therefore, we will mark the cell as white. Checking this for all cells in Figure 1a, we will get the board shown in Figure 1b. To get to a solution, we will reason about the remaining cells, which we will call *grey* cells. These grey cells are yet unknown. The last column contains three cells with '1'. If we cross out the middle '1', then we make all neighboring cells white. The top and bottom cells containing '1' would then be white. However, no duplicates were allowed, meaning that crossing out the middle '1' cannot occur. So, the middle '1' must be white, and since duplicates are not allowed, the top and bottom '1' should both be black. Now, the neighbors of black cells can be made white. Next, consider the top left cells '$4 - 3 - 4$'. At least one of the cells containing '4' has to be crossed out, as we cannot have duplicates. It does not matter whether we cross out the first or second '4', or even both. In all cases, the cell contain '3' will always be a neighbor of at least one black cell, making this cell white. By repeating these steps and the provided rules, we can obtain the solution, see Figure 1c.



(a) A Hitori puzzle. (b) An intermediate step. (c) The solution.

Figure 1: A 5×5 Hitori, an intermediate step and the solution.

The focus of this research lies on generating and solving HITORI puzzles, as well as determining a difficulty measure for HITORI puzzles. Therefore, we will consider the research questions:

- "How can we generate and solve HITORI puzzles in reasonable time?"

- "How can we determine the difficulty of a HITORI puzzle?"

To generate all HITORI of a given size, we will consider an extension of Bell numbers [4] to capture all semantically different puzzles. The unique solvable HITORI will have different layouts in terms of black and white cells that adhere the connectivity and no black neighbor rules of the board. Considering the different layouts, the number of black cells on for a valid $m \times n$ HITORI thus varies, for which an upper and lower bound will be provided.

The difficulty measure for the HITORI puzzle will be based on the different techniques used to solve such a puzzle. The complexity of these techniques determines the way the puzzles are categorized.

In this thesis, we will go over the essentials of the HITORI puzzle, for which we will consider relevant related work in Section 2. Section 3 introduces the definitions, characteristics and rules of HITORI. The rule set can then be used to solve HITORI puzzles. Classifying uniquely solvable HITORI will be discussed in Section 5. Besides the rule set classification, the principle of satisfiability will be explored as well, which is considered in Section 6. Section 7 considers ways to generate HITORI, which combined with the data described in Section 4, are used in performed experiments in Section 8. Finally, Section 9 will discuss future work and provide a conclusion.

This research is part of a master thesis for the Computer Science study program at the Leiden Institute of Advanced Computer Science (LIACS), supervised by Walter Kosters and Hendrik Jan Hoogeboom.

2 Related Work

Different from the very popular Sudoku, which has been extensively researched, not much research has been done on HITORI. While these Japanese puzzles are very similar, HITORI takes connectivity into account as well, which makes the puzzle and its research more complex. In *Enhancing GNNs: An Exploration of Iterative Solving and Augmentation Techniques, Chapter 9, Solving HITORI Puzzles* [16], the author used the Menneske dataset, provided by Vegard Hanssen [2], to test and train a convolutional neural network (CNN) and graph neural network (GNN). The author used the same difficulty distributions as provided in the Menneske dataset. However, only the difficulties easy, medium and hard were used to train and test the two networks. While the GNNs performed quite well on a cell level, they were not able to solve the majority of puzzles. The performance of the CNN is only measured on a cell level, which showed improvement of the performance. However, when solving HITORI, puzzle level performance is essential to measure the overall solving performance. This makes it hard to conclude on how well the CNN actually performs. The main pitfalls were ambiguous solutions and the complexity of the difficulty levels.

Van de Knijff [17] considered Satisfiability Modulo Theories (SMTs) for, among others, the HITORI puzzle. In *Solving and generating puzzles with a connectivity constraint*, an SMT syntax is given. With this syntax, they were able to get a solution which describes which cells should be black and white. However, with no dataset or experiments mentioned, it is hard to obtain the performance of the SMT. After concluding that randomly generating a grid of numbers almost never led to a solvable HITORI, it was left for future research.

In *Card-Based ZKP for Connectivity: Applications to Nurikabe, HITORI, and Heyawake* by Robert et al. [11], the authors have used Zero-Knowledge Proofs (ZKPs) for several Nikoli puzzles, including HITORI. They have provided a generic approach, which they use to set up the ZKP protocol for HITORI puzzles. During the verification phase of the protocol, it is checked that each row and column contains only unique cells. Another check is done to verify that the black cells are separated from each other. Completeness and soundness were also considered.

Suzuki et al. [14] considered the so-called *Hitori number* in their paper *Hitori Number*, which is the minimum number of different integers used for a uniquely solvable n-HITORI, which consists of an $n \times n$ grid. The minimum is taken over all the instances of n-HITORI. The authors have provided an upper and lower bound for the Hitori number. Besides looking at the upper and lower bound of the minimum number of different integers, they have also looked at the upper and lower bound of the largest number of consecutive integers.

In *Games, Puzzles & Computation*, Hearn and Demaine have proven NP-completeness for HITORI by constructing a planar constraint graph consisting of so-called ANDs and ORs gadgets [3]. Finding a legal configuration of the planar graph results in a solvable HITORI, and vice versa, finding a solution to a HITORI means there exists a legal planar graph.

Similar to HITORI, *Heyawake* is a grid-based puzzle, where the goal is to properly fill the grid with black and white cells according to a set of rules. A *Heyawake* starts with several rooms of varying sizes, where, if present, a number in the room indicates the number of black cells. For the by Nikoli invented puzzle, the black cells may not be adjacent to one another, the white cells must form a connected path, and an orthogonal line of white cells may not cover more than two rooms. For $n \times n$ sized *Heyawake* puzzles, the maximum number of black cells is known as *Heyawake numbers* (see sequence A239231 of the OEIS [5]).

3 Solving

HITORI consists of an $m \times n$ grid with a set N of possible characters. Here N is a finite set, usually a subset of the natural numbers. Initially, the grid will solely be filled with these $|N|$ possible characters. One can select a cell, making the cell *white*, or deselect, making the cell *black*. The goal of the puzzle is to obtain a grid that satisfies the following rules:

1. No duplicate characters may occur in a row or column.
2. Black cells cannot be adjacent to one another.
3. All white cells must be connected via their direct, i.e., horizontal and vertical, neighbors. This excludes the diagonals of a cell.

One can solve a HITORI by using these three principles. However, solving more complex HITORI will be easier when considering a set of rules. In this section, we will consider different techniques to solve a HITORI puzzle, introducing those as rules. These techniques are divided into five categories; trivial, simple, connectivity, advanced and probing. These rules will be used to ultimately determine the difficulty of a HITORI.

3.1 Rules

There are many techniques that can be used to solve a HITORI. Depending on size and complexity, some rules might be more effective than others in terms of solving. HITORI that solely use the fact that the white cells must be unique in their row and column and form a connect path as well as no black cells are neighboring are considered to be *trivial*. Rules that solely consider one row or column fall into the *simple* category. Rules that consider multiple rows and do not rely on connectivity are considered to be *advanced*. Since all cells need to be connected, we can use this characteristic to gain knowledge about whether cells should be crossed out or left as is. The category *connectivity* will consider all rules that use this principle. When these principles do not provide new information, we can move to the last category, which is *probing*. We can try to make a cell black or white and see what happens to the board. This, however, is the most time-consuming technique.

Trivial

UC ("Unique Cell") All cells that are unique in their row and column need to be white. Unique cells already fulfil the rule of no duplicates in their corresponding row or column, so there is no need to cross them out. An example is shown in Figure 2. This rule also applies when, after crossing a cell out, another cell becomes unique in its row and column.

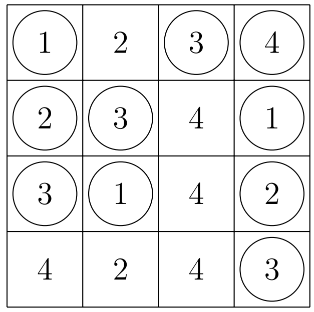


Figure 2: All values that are unique in both their row and column should be white, denoted by a circle.

NoB ("Neighbors of Black cells") Direct, i.e., horizontal and vertical, neighbors of a black cell should always be white.
Since black cells cannot be adjacent, the horizontal and vertical neighbors of a black cell must be white. The diagonal neighbors are excluded. No matter the content of the cells, this always holds. Figure 3 shows the direct neighbors of a black cell that are white, denoted by a circle.

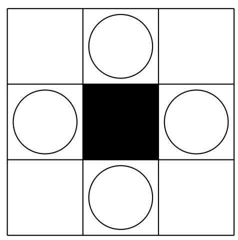


Figure 3: Horizontal and vertical neighbors of a black cell are always white, denoted by a circle.

Simple

DoW ("Duplicates of White") When a cell is marked white, all duplicates in the corresponding row and column must be black.
By marking a cell white, their content should be the only occurrence in the corresponding row and column. Hence, other cells with the same content should be made black. Figure 4 shows an example of a partially solved HITORI, where DoW is applicable. The first row already has a 2 that is white, as one may apply Sim-U3 (which will be discussed later). Thus, the remaining 2s in the corresponding row should be black.

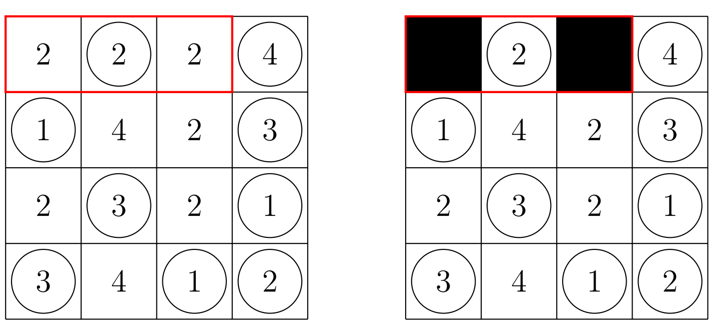


Figure 4: A partially solved HITORI, where the remaining 2s in the first row will be black.

Sim-M3 ("Simple sequence of Multiple characters of length 3") If consecutive cells of type aba occur in a row or column, where $a, b \in N$ and $a \neq b$, then b should be white.

Suppose the middle cell is black, both neighboring as will then be white, which cannot be the case. When aba occurs in a row or column, at least one a needs to be black. Regardless of which a will be made black, b will always be white, as this cell will neighbor a black cell in both cases. An example of this rule can be seen in Figure 5. Note that this rule applies to the whole board, not just the borders or corners.

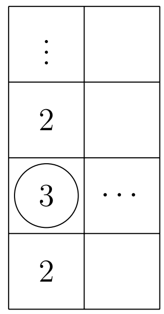


Figure 5: Regardless of the position of the aba sequence in the HITORI, b will always be white. Here, $a = 2$ and $b = 3$.

Sim-U3 ("Simple sequence of Uniform characters of length 3") If three adjacent cells all contain a, i.e., aaa occurs on a row or column, where $a \in N$, then the middle a should be white, while the outer as should be black.

Since no duplicates may occur on a line, having an occurrence of aaa means that at least two as need to be crossed out. Black cells may not be neighbors. Therefore, the only viable solution is making the outer as black and the middle a white. Figure 6 shows an example when to use this rule. Again, this rule applies to the whole board, not just the borders or corners.

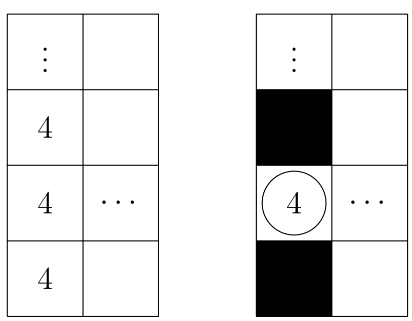


Figure 6: Consider $a = 4$. Regardless of the position of the aaa sequence in the HITORI, the middle a will always be white.

Sim-U* ("Simple sequence of Uniform characters separated by some (*) cells") If cells of type $aa \ldots a$ or $a \ldots aa$ occur on a line, i.e., row or column, then the singleton a should be black.

The content of the cells that are between a and aa is not relevant, as long as the directly neighboring cells of a and aa do not consider a fourth a. The occurrence of a and aa should be separated by at least one b. A row or column containing $aa \dots a$ or $a \dots aa$ should have two as that will be crossed out. Suppose the singleton a becomes white, then the adjacent as both should become black. However, this violates the rules of HITORI, as no black cells should be adjacent. Therefore, the singleton a needs to be black. Regardless of the position of the occurring sequence, this rule will always hold. An example of the occurrence of Sim-U* is shown in Figure 7, where the first cell of the bottom row can be made black. Because of NoB, we will know that the second cell of this row should be white.

4	1	3	4
3	4	1	2
1	3	2	4
3	1	3	3

4	1	3	4
3	4	1	2
1	3	2	4
	1	3	3

Figure 7: Consider the bottom row with $a = 3$. Regardless of the position of the $a \dots aa$ sequence in the HITORI, the singleton a will always be white.

Sim-M4 ("Simple sequence of Multiple characters of length 4") If consecutive cells of type $abab$, where $a, b \in N$ and $a \neq b$, occur in a row or column, then the outer a and b should be black.
Making the inner a or b black, would lead to either two as or two bs to be white, which cannot occur. Regardless of other cells containing a and b in corresponding rows and columns, this rule will hold. Figure 8 shows an example of such a case. Note that when $a = b$, there is no solution to the line, and thus the HITORI is unsolvable.

3	3	2	4
4	3	4	3
1	2	4	3
2	4	3	1

3	3	2	4
	3	4	
1	2	4	3
2	4	3	1

Figure 8: Regardless of the position of the $abab$ sequence in the HITORI, the outer a and b will always be white. Here, $a = 4$ and $b = 3$.

Sim-M* ("Simple sequence of Multiple characters separated by some (*) cells") If a sequence of cells occurs as $abba \dots a/b$ on a line, where $a, b \in N$ and $a \neq b$, then the singleton a or b should be black.
Suppose the singleton a becomes white. Then the other as should be black, meaning that the enclosed bs are both white. This violates the no duplicate rule of HITORI,

and thus the singleton a should be white. Figure 9 shows a HITORI where all unique cells have been made white. The red marked cells show an occurrence of $abba \dots a$, where $a = 1$ and $b = 2$.

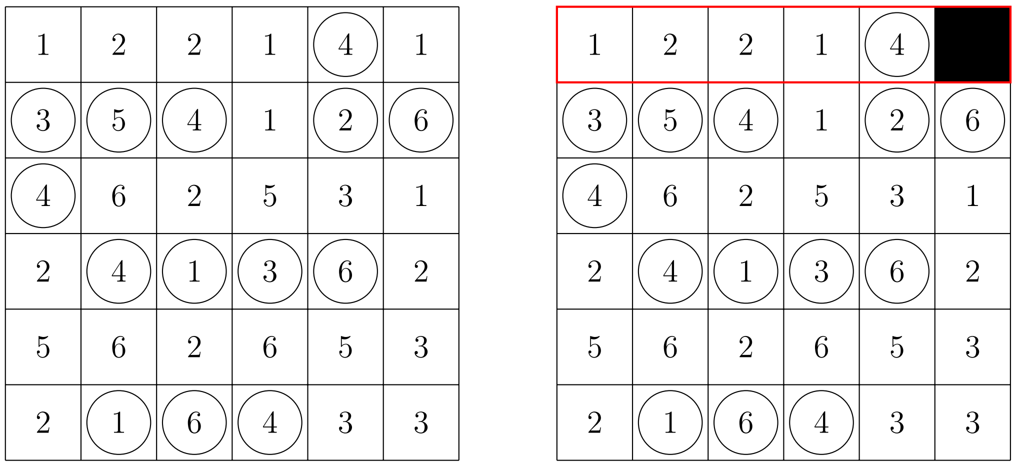


Figure 9: A partially solved HITORI, with an occurrence of Sim-M*. The singleton a will always be black. Here, $a = 1$ and $b = 2$.

Advanced

DP ("Double Pair") Suppose we have a pair a–b for which in their corresponding row or column there is another occurrence of a and b, but diagonally. Then the other diagonal of this second occurrence should be white.
The pair a–b both have a duplicate. For each duplicate, one must become black. However, both of the pair cannot be black as they are adjacent. Then at least one of their duplicates must become black. If the duplicate of a becomes black, then the neighbors a become white, which includes the other diagonal of the duplicates. If the duplicate of b becomes black, then the neighbors b become white, which includes the other diagonal of the duplicates as well. So for both options, the other diagonal of the duplicates will always be white. For DP to occur, one needs to have at least two rows and three columns, or vice versa. When $a = b$, the outcome is not effected. Consider Figure 10, where a pair 1–2 occurs in the first row. In their corresponding columns, they have a duplicate, which are diagonal to one another. Only one of the pair at the top can be black. Then one of the duplicates needs to be black as well. In that case, its neighbors will be white. This leads to the result shown in Figure 10.

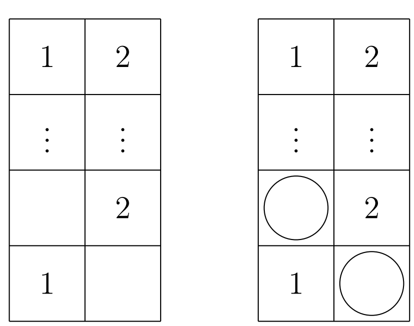


Figure 10: A double pair setup.

SbP ("Single between Pairs") If there are two pairs of a–b and another occurrence of a or b within that corresponding row or column, then that cell should be black.

If there are two occurrences of the pair a–b, then one of the as and one of the bs must be made black. Making one black causes its neighbor to be white, meaning that if we make a of a pair black, then b must be white. Since b has become white, the other occurrence of b must be black, making the a next to it white. If there is any other a or b in the corresponding row or column, it must be black, since we already have a white cell with this content. Similar to DP, one needs at least two rows and three columns for SbP to occur, and $a = b$ may occur as well. In Figure 11, a singleton 1 occurs between two pairs of 1–2. For both of these pairs, only one can be made black, meaning that if we make one of the 2s black, the other one will be white. The 1 next to the blacked out 2 will be white, making the remaining 1s black. Thus, the singleton 1 will always be black.

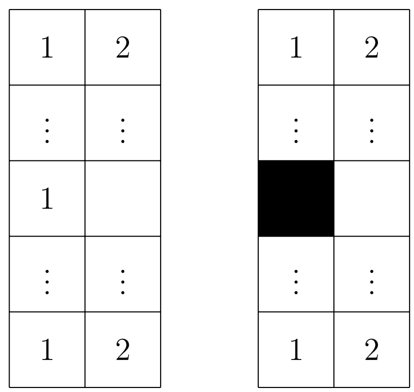


Figure 11: A singleton occurrence between two pairs should be black.

Connectivity

NCO ("No Cut-Off") All white cells should be connected, and thus form a connected graph. Therefore, black cells should never cause a connected subgraph.

Diagonally connecting the black cells should not form any enclosed shape, such as those seen in Figure 12. A *cut-off* occurs when some white cells are enclosed by the black cells, and thus causes the white cells to no longer form a connected path. No such form should be created when including the borders as well. In all cases, cut-off may occur.

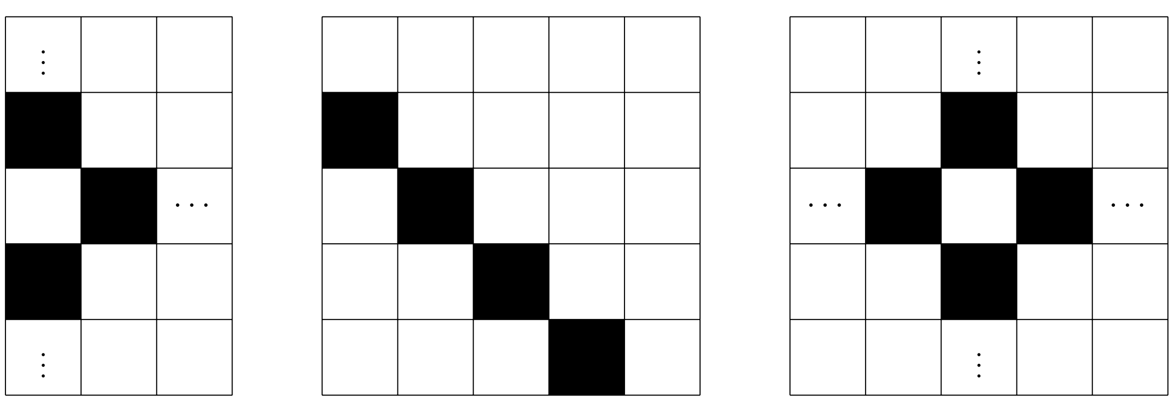


Figure 12: Three occurrences where a cut-off occurs.

NCO-B ("No Cut-Off at the Border") Suppose two black cells are at the border and separated by one white border cell, then the remaining neighbor of the white cell should be white as well.
The white cell is surrounded by a border and two black cells. If the remaining neighbor becomes black as well, the white cell will be enclosed and is no longer connected to the other white cells of the board. This violates the connectivity rule of HITORI, meaning that the remaining neighbor of the white cell must be white. Figure 13 shows two instances for which a triangle occurs when crossing out the marked cell. Therefore, the red marked cell should always be white.

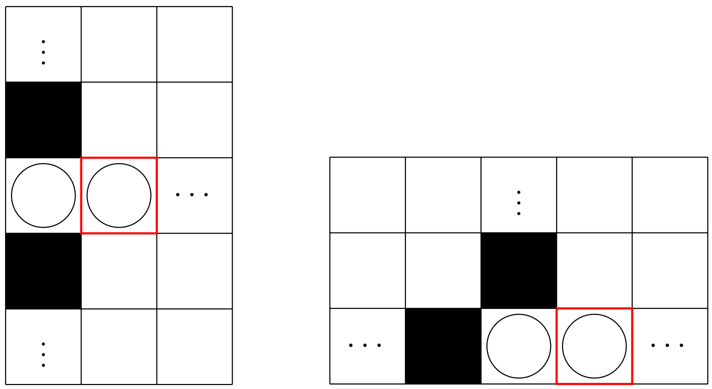


Figure 13: Two different occurrences where no cut-off should occur. The marked cell must be white.

NCO-C ("No Cut-Off in the Corner") If one of the neighbors of the 2×2 corner cell is black, then the other neighbor should be white.
Due to connectivity, the corner cell can only have one black neighbor. To access the white corner cell, the other neighboring cell must be white, as can be seen in Figure 14.

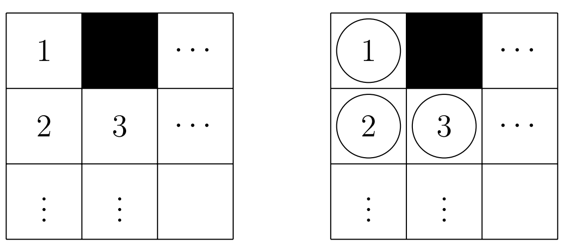


Figure 14: The cell containing 2 should be white, otherwise a cut-off occurs.

CP ("Corner Pairs") If two pairs occur at the 2×2 corner, then the corner cell should be black, as well as its diagonal.
Two pairs in a corner may occur horizontally or vertically oriented. Regardless of the orientation of the pairs, the corner cell will always be black, as well as its diagonally adjacent cell. The other two cells will be white. Suppose the corner cell was made white, then the adjacent cells would be black, cutting off the corner cell. The white cells would then no longer be connected, violating the connectivity rule of HITORI. An example of two pairs at a corner is shown in Figure 15. This rule will also apply if the content of the two pairs are the same.

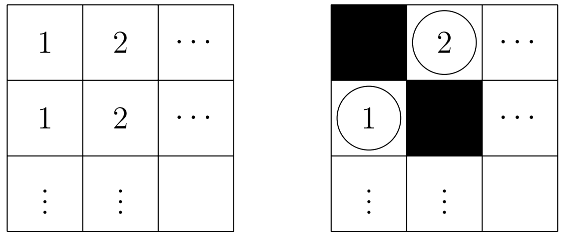


Figure 15: Two pairs occurring at the corner always leads to the corner cell and its diagonally cell to be black. The remaining two cells will be white.

DoT ("Diagonal of a Triple") Suppose we have a 2×2 corner for which a corner cell and its neighbors contain the character same a or the neighbors and diagonal of the corner cell all contain a. Then either the corner cell or its diagonal becomes black, respectively. The remaining cell contains a character b, where b may be identical to a.

Consider a corner cell and its neighbors to have the same content. Then, if the corner cell would be made white, the neighbors should both become black as they all have the same content. With this, the corner cell is no longer connected to the other white cells on the board. This cannot occur, and thus the corner cell should always be black. A 2×2 corner for which the diagonal and the neighbors of the corner cell are the same, will result in the diagonal of the corner cell to be black. Making the diagonal white, would make the neighboring cell black. The corner cell is then no longer connected to the other white cells on the board. So, in both cases, the diagonal of the differing cell should always be black. Figure 16 shows two examples of how such a corner case would be solved.

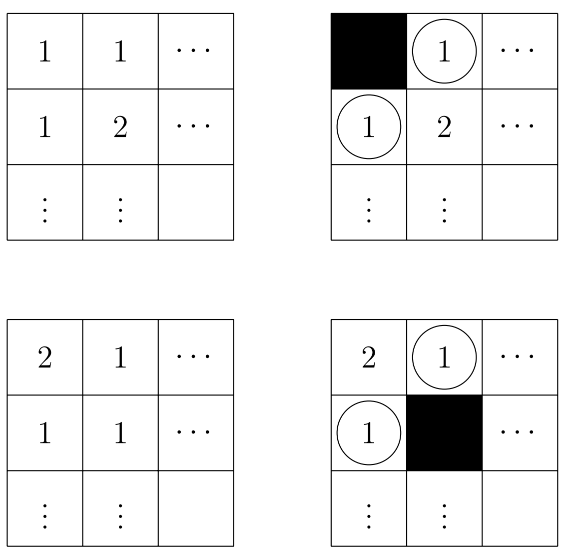


Figure 16: Two situations where we can make a cell with two neighbors with the same contents black and the neighboring cells white.

This is not guaranteed to work for other orientations. However, we can guarantee that if we have a 2×2 corner where the corner cell, its diagonal and one neighbor all have the same content, the remaining cell should always be white. If the remaining cell is black, its neighbors should be white. Then the last cell of the triple will be black. This will enclose the corner cell and thus cannot occur. While we can have different possible ways to fill the triple, the remaining cell should always be white. Figure 17 provides an example of the different ways to fill the triple, while the remaining cell will be white in both cases.

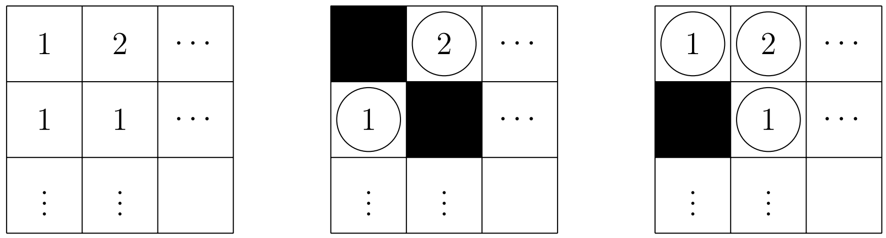


Figure 17: An other oriented triple where we can make the cell containing b white.

BoP (“Border of Pairs”) There are multiple pairs at the border. For each of the pairs, one must be made black. The other cell of the pair will then be white. The same holds for the other pairs. To ensure that all white cells remain connected, the cells that enclose the pairs must be white.

When starting with the pair at the border, one of the cells must be black, as there cannot be any duplicates in a row or column. Moving on to the adjacent pair, it is already determined which cell will be black and which one will be white. The black cell of a border pair causes the adjacent cells to be white. The remaining cell will then be black. This continues until all pair has been filled in. There will be a checkered pattern for these two rows or columns, which means that the white cells, except for the last pair, will be surrounded by either three black neighbors or a border and two black neighbors. Their remaining neighbor should thus be white, in order to preserve connectivity. Figure 18 shows a Hitori where multiple adjacent pairs occur at the border. There are two possible ways to solve the pairs, depending on the remainder of the board. For both, the cells surrounding the pairs are always white. When starting at the border, for each pair, except the last one, there will be three black neighbors surrounding the white cells. This leads to the outer cells to be white. Generally, if we have n adjacent pairs, then $2 \cdot (n-1)$ surrounding cells, starting at the border and going down, will be white.

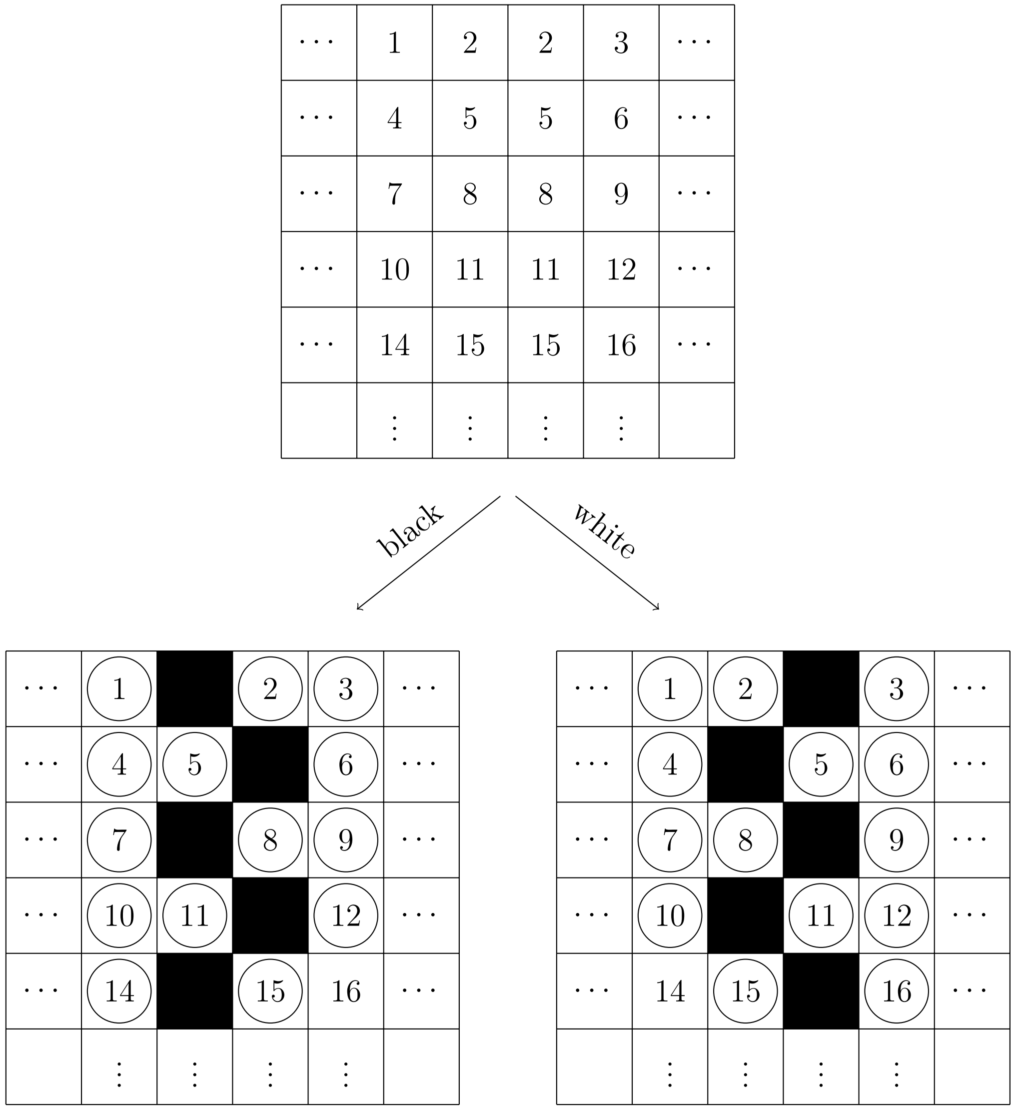


Figure 18: No matter the number of adjacent pairs, the cells that enclose the pairs and are parallel to the border, will be white.

BoP-D ("Border of Double Pairs") Suppose there are any two adjacent pairs occurring at the border. The two border cells that enclose the pairs should be white.
Figure 19 shows two pairs at the border. It does not matter if the left or right cell of a pair becomes black. Both cause the enclosing border cells to be white. Note that it does not matter whether the two pairs are horizontally oriented, as show in Figure 19, or vertically.

⋯	1	2	2	7	⋯
⋯	3	5	5	6	⋯
	⋮	⋮	⋮	⋮	

⋯	1	2	2	7	⋯
⋯	3	5	5	6	⋯
	⋮	⋮	⋮	⋮	

Figure 19: A double pair setup at the border. One pair consisting of 2s and one of 5s.

BoP-T ("Border of Triple Pairs") If three pairs at the border occur, the four cells that enclose the pairs should be white.
Similar to BoP-D, the enclosing cells of a triple pair occurrence at the border will always be white, see Figure 20. An addition to that are the enclosing cells of the second pair. Note that orientation matters here, the pairs should be aligned parallel to the border.

1	2	2	7
3	5	5	6
⋮	4	4	⋮
	⋮	⋮	

1	2	2	7
3	5	5	6
⋮	4	4	⋮
	⋮	⋮	

Figure 20: A triple pair setup at the border.

EP ("Enclosing Pair") Suppose that we have a corner cell, a border cell with one black neighbor or a cell that has two black neighbors. There are two other neighbors, one that is part of a pair. The remaining neighbor will be white.
The cell is enclosed on two sides by either two borders, by one border and a black cell or by two black cells. So either the cell that is part of the pair or the other cell has to be white, to ensure that all white cells remain connected.

EP-B ("Enclosing Pair at the Border") A cell is surrounded by a pair, black cells and some border. Since the cell itself is a neighbor of some black cell, it will be white. The remaining neighbor of this cell will always be white.
Figure 21 shows a red marked cell surrounded by a pair, black cell and border. Suppose the left 1 becomes black, then the right 1 will be white. Now, three of the neighbors of the red marked cell are black, meaning that the remaining neighbor has to be white. If the right 1 becomes black, then the left 1 is white, as well as the remaining neighbor of the red marked cell. In both cases, the remaining neighbor of the red marked cell, will be white.

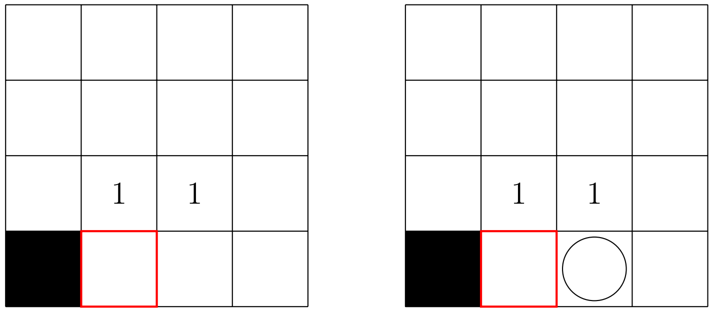


Figure 21: A pair, black cell and a border enclose the red marked cell. The remaining neighbor of this cell should always be white.

EP-M (“Enclosing Pair in the Middle”) Suppose a cell is surrounded by a pair and black cells. The one remaining neighbor will always be white.
An occurrence of a cell surrounded by such a combination is shown in Figure 22. Note that orientation does not matter for this rule.

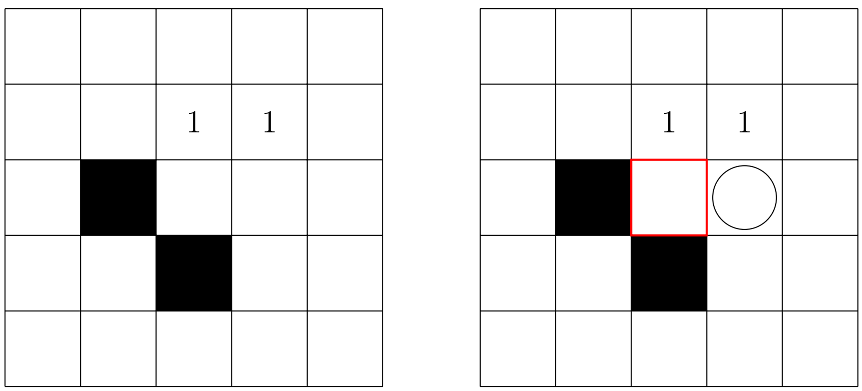


Figure 22: A pair and black cells enclose the red marked cell. The remaining neighbor of this cell should always be white.

Probing

When the rules cannot be applied, one can use the method of *probing* to try and solve a Hitori. The described rules each have a specified gain, whereas probing does not guarantee any progress. To determine if any conclusion can be drawn from using the method of probing, one must reuse the introduced rules to determine whether the probe leads to progress.

Prob (“Probing”) An unknown cell can be probed, where the cell may be black or white. First, color the cell black and see whether this results in a contradiction (BP). If a contradiction occurs, one knows with certainty that the cell must be white. Second, color the original cell white and see whether this results in a contradiction (WP). A contradiction caused by a white probe indicates that the probed cell must be black. However, if both a white and black probe cause a contradiction, the Hitori is unsolvable. Combined, BP and WP will be one probe (Prob). An example of when to apply a probe is shown in Figure 23. None of the rules can be applied. However, after performing a black and white probe, we can continue solving the Hitori. A white probe causes the ‘1’ in the blue marked area to be disconnected from the other white cells, which violates the rules of Hitori. Therefore, the red marked cell should be black. Performing a black probe for the red marked cell leads to a fully solved board.

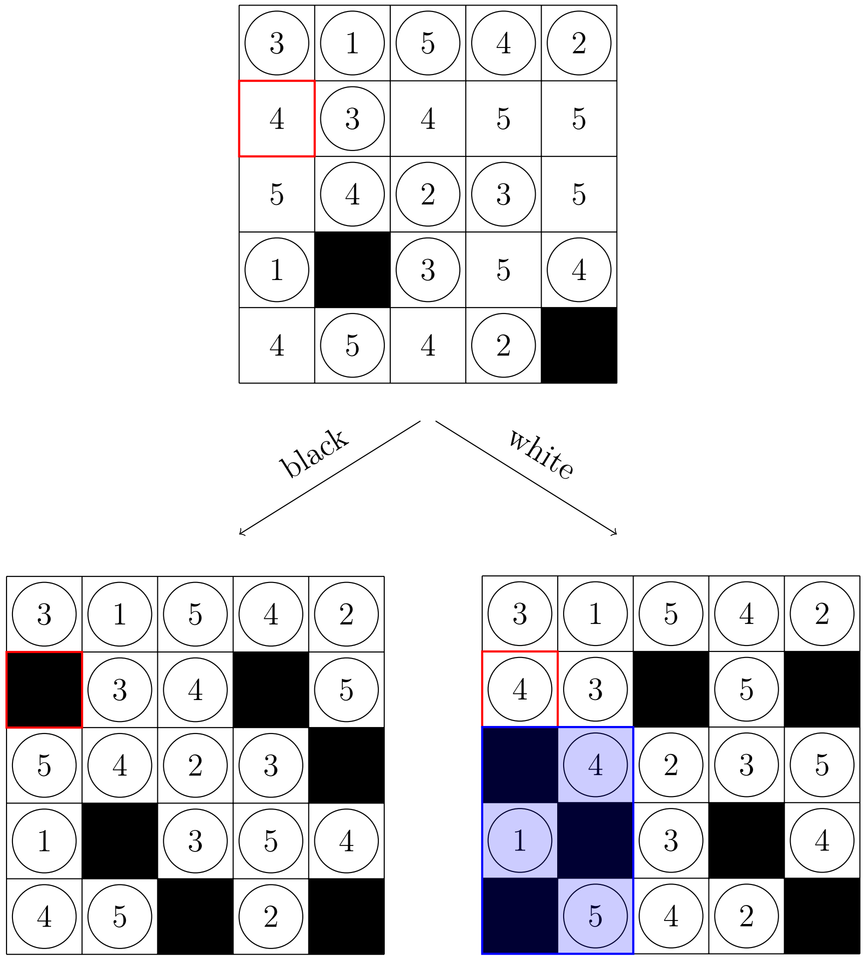


Figure 23: Probing, as no other techniques can be applied at this stage. We try the red marked cell, and color the ‘4’ black and white. Coloring this cell white leads to a contradiction, meaning that the cell should be black. This example is automatically solved by probing once.

Table 1 provides an overview of the categorisation of the rule set, including the method of probing.

Category	Rules
Trivial	UC, NoB
Simple	DoW, Sim-U3, Sim-M3, Sim-M4, Sim-U*, Sim-M*
Advanced	DP, SbP
Connectivity	NCO, CP, DoT, BoP, EP
Probing	Prob

Table 1: An overview of the rule set.

4 Menneske dataset

As mentioned in Section 2, Hanssen [2] has provided the Menneske dataset. While directly translated to “man”, the Menneske dataset solely considers puzzles, numbers and sequences. This includes numerous HITORI of varying sizes, for which a comprehensive overview will be provided in this section.

The Menneske dataset [2] consists of 376,211 HITORI of different sizes. Table 2 shows the distribution over the different sizes. The majority of the puzzles are smaller-sized, and generally easier to solve.

Puzzle size	Number of puzzles
5×5	31,610
6×6	29,902
8×8	35,972
9×9	241,948
12×12	31,131
15×15	3,807
17×17	1,836
20×20	5
	376,211

Table 2: The size distribution of the Menneske dataset [2].

The complexity of a HITORI potentially correlates with the number of duplicated that need to be eliminated. For the Menneske dataset, the maximum number of black cells of the different sizes is as shown in Table 3. As one might observe, this varies around one-third of the total number of cells for a given puzzle size.

Puzzle size	Maximum number black cells	Percentage black cells
5×5	9	36.00%
6×6	12	33.33%
8×8	21	32.81%
9×9	27	33.33%
12×12	46	31.94%
15×15	71	31.56%
17×17	90	31.56%
20×20	121	30.25%

Table 3: The maximum number of black cells according to the Menneske dataset [2].

Hanssen has classified numerous HITORI into the categories; *Super Easy*, *Pretty Easy*, *Easy*, *Medium*, *Hard*, *Very Hard*, *Super Hard* and *Impossible* [2]. However, no explanation on how these classes were determined has been provided. Hanssen introduced 22 rules [2], which are all used across different categories, and thus provide no clear distinction regarding classifying HITORI. Table 4 shows the weighted average of the number of methods used for each given category and size. Based on the weighted average of the number of different methods used to solve a HITORI, there appears to be some correlation between the number of different methods and the class it belongs to, see Table 4. Lower classified categories have a lower average of the number of used methods to solve a puzzle. However, this cannot be guaranteed due to the small number of puzzles occurring in the very hard, super hard and impossible categories as well as the small number of puzzles of 12×12 and larger.

Difficulty	5×5	6×6	8×8	9×9	12×12	15×15	17×17	20×20	Average
Super easy	5.253	5.747	6.380	6.606	6.847	6.870	6.934	7.000	6.430
Pretty easy	6.299	6.708	7.179	7.352	7.445	7.414	7.445	8.000	7.201
Easy	7.023	7.321	7.665	7.733	7.644	7.416	7.367	0	7.667
Medium	7.435	8.200	9.000	9.002	8.959	8.552	8.565	0	8.774
Hard	7.708	8.498	9.501	9.641	9.398	9.400	11.000	0	8.821
Very Hard	7.769	8.745	10.083	10.230	11.615	0	11.000	0	10.1549
Super Hard	0	0	10.033	9.891	10.814	11.429	12.000	0	9.945
Impossible	0	0	0	0	0	8.030	0	0	8.030
Average	6.019	6.573	7.1347	7.260	7.221	7.203	7.189	7.4	

Table 4: The average of the number of different methods used by Hanssen for each difficulty and size of the Menneske dataset.

5 Classification of Hitori

The complexity of Hitori greatly varies. Some puzzles might only require basic logic, while others need more complex techniques to make any progress during solving. In this section, we will discuss the difficulty measure based on the provided rule set, and another based on satisfiability. The difficulty measure combines the applied techniques and the frequency of usage. The frequency will only consider the techniques of the highest so-far used class. When entering a higher class, the techniques of the lower classes become “free”.

5.1 Rule set classification

Classifying Hitori will be done based on the complexity of solving. For this, consider categories based on the rules described in Section 3 and a numeric measure for the needed steps of the provided class rules.

Trivial rules are considered to be “free”, as these are fundamental when solving a Hitori correctly. Considering a completely unsolved Hitori, one can only apply UC, as there are no black cells for which neighbors can be made white, making it impossible to use NoB. These rules will always be checked after applying other categorized rules. To determine whether an $m \times n$ Hitori is solvable by solely using trivial rules, all cells need to be checked. This requires comparing cells on their respective row and column, which can be done in $2m + 2n$ steps when keeping track of unique and duplicate values. For $n \times n$ Hitori, this class contains all Latin squares of that given size. A Hitori solvable by solely using trivial rules has a difficulty classified as class A. Since we know the number of evaluations needed to determine whether the puzzle is completely solvable using trivial rules beforehand, there is no need to keep track of any number of steps. This class will therefore only consist of an A.

When a Hitori cannot be solved with solely using trivial rules, we introduce the usage of Simple rules and DoW, and move to a higher difficulty class B. Simple rules only consider one row or column, and can be applied simultaneously to multiple rows or columns. Therefore, we introduce a horizontal *H-sweep* and vertical *V-sweep*, where the simple rules are applied on either all rows or columns, respectively. These rules are independent of the solving stage, as they deal with the contents of cells rather than connectivity or the already existing occurrence of black and white cells. This difficulty class includes the usage of DoW. Combined with NoB, these rules ensure that the basics of Hitori, i.e., black cells cannot be adjacent to one another and no duplicate values may occur in a row or column, are met. When making a cell black using NoB, only the direct neighbors of a cell need to be checked. The rule DoW can be applied in the same sweep-like manner as for the simple rules, and is thus included in this class. When a Hitori can be solved solely using techniques of class B, or lower, the number of sweeps is taken into account as well. Depending on whether the solving process starts with a H- or V-sweep, the number of sweeps may vary by 1. To provide a consistent classification, a Hitori will first be solved starting with a H-sweep, followed by solving starting with a V-sweep. The number of sweeps needed to solve a Hitori is then averaged. As a consequence, we can end up with .5 values.

While simple rules may consider a single row or column to make some progress, the advanced rules need to check both rows and columns in order to make progress. Two adjacent lines, i.e., rows or columns, and their orthogonal lines are considered. Just like the simple rules, these rules are not dependent on the solving stage and can thus be "sweeped". When a HITORI can be solved using advanced rules, it falls in difficulty class C.

The rule BoP considers multiple pairs adjacent to a border. The rule, based on connectivity, will provide some gain about cells that enclose those pairs. We consider at least four adjacent lines, and there are at least two adjacent value-pairs. The number of value-pairs is variable, and thus the gain is variable as well. Since BoP solely considers rows and columns and is solving-phase-independent, it will be categorized as a class D rule. This class considers solving-phase-independent rules for which at least four lines are needed to obtain the pattern used in BoP.

HITORI that can be solved using classes A, B, C and D are mostly independent of the solving phase, with the exception of using NoB and DoW. However, a HITORI needs to maintain a contiguous area consisting of all white cells. HITORI for which connectivity rules are needed to completely solve the puzzle, will be of a higher class, i.e., class E or F. The connectivity rules are distinguished by their complexity. Connectivity rules that are independent of the solving phase will be classified as class E rules. Both CP and DoT are independent, and can be checked in constant time, as only the 2×2 corners need to be considered. Instead of using a number of sweeps for this class, consider the number of cells solved when applying these rules to all corners. In all cases, a total of 16 cells needs to be evaluated, for $m \times n$ boards with $m > 2$ and $n > 2$.

To obtain any progress in solving a puzzle, the connectivity rules NCO and EP can only be applied if there exists at least one cell that is black. Different from the previous classes, class F is fully solving phase dependent. These rules can be applied at the corner, border and the remainder of the board. Since it is phase dependent, sweeps may not be a proper measure. Considering NCO, making a cell black may lead to the cut-off of a half grid. These connectivity based rules could consider the sum of potentially cut-off white cells as a proper measure, as this provides some information on how much one needs to take connectivity into account when solving the puzzle.

When these approaches are not able to solve a HITORI puzzle, consider the last class G, which is based on probing. We will try an unknown cell and see how the remaining board is handled. Probing a black cell, BP, will provide more knowledge about its direct neighbors. When performing a black probe, one needs to check that no white cells are enclosed by black cells and possibly borders. If such an enclosure occurs, there is no need to continue solving this board. It is certain that the probed cell should be white. If no enclosure problems arise, we can still check if this black probe leads to an incorrect board. If the board is incorrect, we will know for certain that the probed cell should be white. When the board is partially solved, we cannot be certain that this cell should be black or white. When performing a black probe on a cell of a HITORI leads to a contradiction, one knows that the probed cell must be white, and vice versa. A white probe, WP, will provide knowledge about duplicates of the probed cell. If present, these will be black. Similar to the black probe, if a white probe leads to an incorrect board or a connected subgraph, the cell should be black. When performing a probe, one can iteratively probe other cells until a fully determined board or a contradiction is found. This depth-first search approach considers probing all other white cells. The number is determined by probing every cell and then considering the highest class methods that are used after that to solve the Hitori. Considering the weighted average of the highest classes for each probe then determines the step size.

Table 5 provides an overview of how the rules and methods are divided over the difficulty classes. The determination of the step size differs for each class. Classes D, E and F consider solving phase-independent rules, while class G considers solving methods rather than rules.

Difficulty	Steps	Methods
A (trivial)	-	Trivial
B (pretty easy)	Average of H- and V-sweeps	Simple
C (easy)	Average of H- and V-sweeps	Advanced
D (medium)	Number of solved corner cells	CP, DoT
E (pretty hard)	Average of H- and V-sweeps	BoP
F (hard)	Sum of potentially cut-off white cells	NCO, EP
G (super hard)	Average of highest class of solving each probe	Prob

Table 5: A classification of the difficulty based on different methods.

A Hitori may have multiple viable solutions, as long as the original rules of the puzzle are not violated. Consider Figure 1c, where making the '4' in the top row black leads to a valid solution as well. A minimal solution is one that satisfies the original rules of the puzzle while using the fewest possible number of black cells. When classifying Hitori puzzles as described in the previous section, the number of steps is essential to determine the classification of a Hitori. By minimizing the number of black cells, the classification is minimal, as no additional steps need to be taken. This allows a consistent way to categorize Hitori.

5.2 Expanding the rule set

Compared to the *Simple* rules mentioned in Section 3.1, a SAT-based classification might capture more complex patterns earlier on. However, one might expand the rule set to emulate a similar classification.

Consider a line of a Hitori, i.e., a row or a column, which may be partially solved, or not. Much like the conditions of the single line used in Section 6.1, elements that have already been determined, i.e., made black or white, will not be considered. Uniquely occurring elements will be, again, temporarily selected as one cannot guarantee its uniqueness in the orthogonal line. Assuming the line does not contain any unique elements, every value occurs at least twice. Duplicates may occur adjacent or separated. In a way, these duplicates form "blocks" of yet to be determined elements.

To maximally solve a given line, a set of rules can be applied, as seen in Section 3.1. This set considers several instances of sequences:

- $bac \ldots bc$, including sequences that contain a mirrored cab or cb.
- $aa \ldots a$, including the sequence aaa.
- aba, which is included in sequences such as $abab$.
- $ab \ldots ab \ldots a$, including sequences that contain ba or b. This includes $abba \ldots a$ as well.

These patterns cover at most three distinct values; a, b and c. However, cases starting with $abc \ldots$ potentially followed by $def \ldots$ may not be covered by these rules. Selected, i.e., white, cells are denoted by a dot under the symbol, and unselected, i.e., black, cells by a line under the symbol. If no guarantee can be provided on the value of an element, it will not have any sign, i.e., z, where z represents the value of the element.

Consider a sequence $A = a_1 a_2 \ldots a_{n-1} a_n$ with different elements. Then A' represents any permutation of A and can be denoted as $A' = a'_1 a'_2 \ldots a'_{n-1} a'_n$. A sequence $A = a_1 a_2 \ldots a_{m-1} a_m$ which we can reorder in a way that the elements on even positions remain on even positions and the odd on odd positions, is denoted as $A^* = a^*_1 a^*_2 \ldots a^*_{m-1} b^*_m$. Similarly to A^*, a sequence $A = a_1 a_2 \ldots a_{p-1} a_p$, where we can reorder A in a way that the elements on even positions are now on odd positions and elements on odd positions are on even positions, is denoted as $A^{\sim} = a^{\sim}_1 a^{\sim}_2 \ldots a^{\sim}_{p-1} a^{\sim}_p$. Note that if A is of odd length, $A^{\sim}$ cannot occur, as we cannot place all elements that were originally on even position on all odd positions and vice versa. However, A^* can occur, as the number of elements on odd positions equals the number of odd positions, and the number of elements on even positions equals the number of even positions, which is not the case when $A^{\sim}$ occurs.

5.2.1 Instances of A, A^* and $A^{\sim}$

When a line cannot be solved using the rules shown in Section 3.1, one may consider blocks of duplicates. Suppose a line contains a permutation A' of another subsequence A. If an element in A is white, its counterpart in A' must be black. If present, adjacent elements of this given counterpart in A' must then be white. In turn, those equal in A will be black. One may fully determine A and A', once a single element is determined.

Lemma 5.1. *Sequences A and A^* together have the same number of black and white elements.*

Proof. A and A^* contain the same n elements, resulting in $2n$ elements in total. For each value-pair, at least one of the elements needs to be black, as duplicates cannot occur. Consider the n elements of A, with each a duplicate in A^*, we know that, in total, at least n elements must be black.

A sequence of adjacent elements of length n can alternate between black and white, i.e., $ZWZW \ldots$ or $WZWZ \ldots$. For an even n, this results in at most $n/2$ black cells, as two black cells may not be adjacent. If all duplicates in A and A^* need to be covered, there need to be at least n black elements, and we can cover at most $n/2$ black cells in both A and A^*. Therefore, there must be exactly $n/2$ black elements in A as well as in A^*. The remaining $n/2$ elements for both A and A^* are white, meaning that A and A^* combined have the same number of black and white elements.

Alternating elements for an odd n results in at most $\lceil n/2 \rceil$ black cells. Making $\lceil n/2 \rceil$ elements black in one of the sequences, results in making $\lceil n/2 \rceil$ elements white in the other sequence. This sequence can then have at most $\lfloor n/2 \rfloor$ black cells. If one of the sequences covers at most $\lceil n/2 \rceil$ and the other at most $\lfloor n/2 \rfloor$ black cells, and there are at least n duplicated elements that need to be black, then A and A^* jointly have exactly n black elements. Since the remainder of elements cannot be black, they must be white, meaning that A and A^* jointly have exactly n white elements as well. Therefore, A and A^* together have the same number of black and white elements. □

Lemma 5.2. *Sequences A and $A^\sim$ together have the same number of black and white elements.*

Proof. Similar to A and A^*, A and $A^\sim$ have the same alternation between even and odd elements. This allows that both A and $A^\sim$ can have an alternating pattern of $ZWZW\dots$ or $WZWZ\dots$, where no neighboring black elements occur. The remainder of the proof is identical to Lemma 5.1. □

5.2.2 Enclosing and alternating permutations

Knowing that a sequence A and A^*, as well as A and $A^\sim$, have the same number of black and white elements, one may obtain a partial or full solution solely based on the pattern of black and white, rather than the actual content of the sequences. However, there exist permutations A' of A, for which one could obtain a partial or full solution as well. We will consider the following patterns:

ABB′A* Consider the sequence $s = ABB'A^*$, where the sequences A and B do not contain any duplicates, i.e., consisting of different elements, and B does not contain any element of A. When $ABB'A^*$ occurs, the sequence A must be of even length, B can be of even or odd length, and $|AB| \geq 3$. The sequence A may not be empty, but B can be, which means that AA^* is accepted as well. The second occurrence of B can be fully reordered, and is thus denoted as B'. The second occurrence of A may be reordered as well. However, A can only be reordered in a way that the elements on even positions remain on even positions and elements on odd positions on odd positions, and is denoted as $A^* = a_1^* a_2^* \dots a_{n-1}^* a_n^*$. The sequence $ABB'A^*$ includes $ABB^{-1}A$ as well, where B^{-1} denotes the inverse of B.

If $ABB'A^*$ occurs, the elements of A and A^* enclosing BB' will be white. Making the enclosing element of A or A^* black, would immediately determine the cells of the neighboring B, as the cell of B neighboring the instance of A will be white. In turn, this determines the cells of the other instance of B, as the last cell of the first instance will be black, and thus the adjacent cell of the second instance will be white. The other end of the second instance will then be a black cell, neighboring a cell of either A or A^*. Since there is an enclosing element of A or A^* that is black, the other instance of A must have an enclosing element that is black as well, due to the ordering of A and A^*. However, based on the instances of B, if the first instance of B starts with a white cell, then the second instance ends with a black cell. We can make their duplicates black. Repeating this, will eventually result in a pattern of $\underline{a_1}\underset{\cdot}{a_2}\underline{a_3}\underset{\cdot}{a_4}\dots\underline{a_{|A|-1}}\underset{\cdot}{a_{|A|}}$ for the first sequence A. For the second sequence A, the

selection of white and black will of course be inverted, i.e., $\underset{\cdot}{a_1}\underline{a_2}\underset{\cdot}{a_3}\underline{a_4}\dots\underset{\cdot}{a_{|A|-1}}\underline{a_{|A|}}$. Ultimately, both A and A^* are fully determined.
As an example, suppose we have the sequence *abccab*, see Figure 24. Then using $A = ab$ and $B = c$, this would result in $\underline{a}\underset{\cdot}{b}cc\underset{\cdot}{a}\underline{b}$.

Splitting the sequence. The sequence $ABB'A^*$ may be non-adjacent. If $|B|$ is odd, then the sequence $s = ABB'A^*$ can be separated as $A \dots BB'A^*$, $ABB' \dots A^*$ or $AB \dots B'A^*$, with the exception of sequences where BB^* occurs. If BB^* occurs, one can alternate the entire BB^*, starting with either black or white. Since there are two alternations possible for BB^*, there is no guarantee for the cells of an adjacent A. When $|B|$ is even, s can be separated if $BB^\sim$ does not occur. The sequence $BB^\sim$ will fully alternate, starting with either black or white, and will not enforce any cell of A, which causes no guaranteed determination of given cells. If both A and B are of even length and occur as $AB \dots B^*A^*$ or $AB \dots B^\sim A^\sim$, there will be at least two manners in which these sequences may alternate. For sequences where these As and Bs occur separated, there is thus no deductible gain. When $|A|$ is even and $ABB'A^*$, the B-enclosing elements will be white. Ultimately, this results in fully determining A and A^*. For odd $|A|$ and even $|B|$, where the given line contains $AB \dots B^*A^*$, will have one possible way to fill B and B^*. One of the sequences A and A^* contains the maximum number of black cells, meaning this A will begin and end with a black cell. Since B and B^* are enclosed by the As, and their patterns are mirrored to one another, the A-neighboring cells of B and B^* must be white. Similarly, if A and B are of even length, an occurrence of $AB \dots B^*A^\sim$, has white A-neighboring cells for both B and B^*.

Example. Suppose there is a line for which the sequences $A = abcd$, $B = efghi$, $B' = (B^* =)$ *ehgfi*, and $A^* = cdab$ occur as an instance of $AB \dots B^*A^*$. The sequence is separated by "…", which represents several irrelevant, potentially already solved, cells. Either B or B^* contains the maximum number of black cells, and thus start and ends with black cells. If the B-neighboring cell of A is black, the adjacent B will not be maximally filled with black cells, implying that B^* must be maximally filled. However, A^* will mirror the B-neighboring cell of A, as an element on an even position, as white. Sequence A will then have a pattern in which the elements on odd positions are white, and A^* will have white elements on the even positions. This results in black B-neighboring cells for A and A^*, meaning that none of the Bs may contain the maximum number of black cells. Therefore, the B-neighboring element of A must be white. This implies its counterpart in A^* must be black, and existing neighbors must be white. Repeating this ultimately results in a solved A and A^*: $\underline{a}\underset{\cdot}{b}\underline{c}\underset{\cdot}{d}efghi \dots ehgfi\underset{\cdot}{c}\underline{d}\underset{\cdot}{a}\underline{b}$, and it is thus a partially solved sequence.

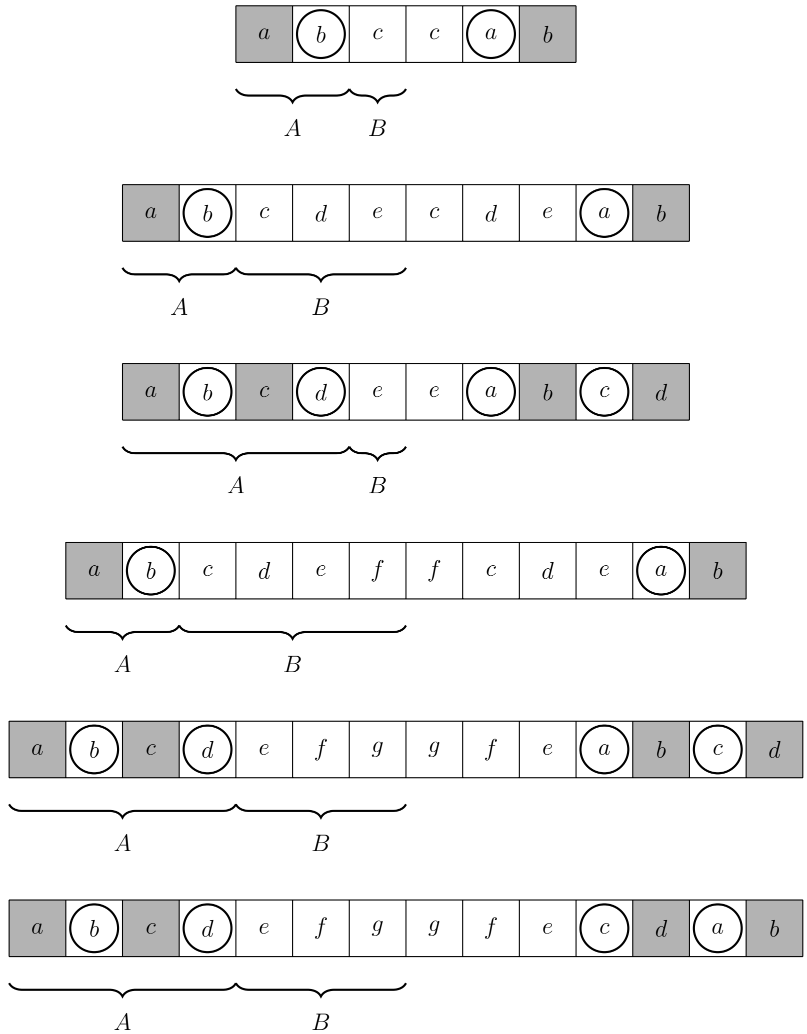


Figure 24: Lines solvable using the pattern $ABB'A^*$.

ABA′B′ Consider the sequence $s = ABA'B'$. Again, the sequence A does not contain any duplicates. The sequence B does not contain any duplicates as well, and has no elements that belong to A. When a line has an A and B of even length, and the second instances of these sequences occur as $A^\sim$ and $B^\sim$, there will be two solutions and s is thus excluded. Suppose a line containing sequences A and B, where $|AB|$ is odd, i.e., $|A|$ is even and $|B|$ is odd or vice versa, and the second occurrences of these A and B occur as A^* and B^*, there will be two solutions and s is thus excluded. If a sequence s has A and B of odd length, all orders are permitted. Table 8 provides an overview of which orders are permitted for all possible even-odd combinations of sequences A and B. Due to the different order restrictions, we denote sequences that do not violate the restrictions as $s = ABA'B'$. Depending on the parity of A and B, A or B, and its counterpart, may be determined. For odd $|A|$ and $|B|$, the pattern ABA^* causes this B to be minimal, and thus determines both Bs. Similarly, the pattern BA^*B^* causes the A to be minimal, and thus determines both As. Considering an even $|A|$ and $|B|$, the gain of these patterns is vice versa, as the As enclosing B for ABA^* must both enclose B with a white element. For BA^*B^*, the A-neighboring elements of the Bs must be white, which allows determining both A and A^*. If $|AB|$ is odd, an instance of A and $A^\sim$ enclosing B, or vice versa, determines the cells of the enclosed sequence, B and A respectfully.

Figure 25 shows several examples of sequences where $ABA'B'$ occurs. Consider the line *abcdecbaed*, where $A = abc$ and $B = de$. The sequence *abcdecbaed* occurs as $ABA'B'$, where A and B have been reordered for their second occurrence.

Splitting the sequence. When $s = ABA'B'$ occurs non-adjacent on a given line, A or B may still be determined. Again, based on the parity of A and B, there are some exceptions regarding whether a sequence s can be non-adjacent. The sequence ABA^*B^* cannot be separated as $AB \dots A^*B^*$ for both and even $|A|$ and $|B|$ as well as an odd $|A|$ and $|B|$. Any other partition is allowed when considering A and B to both be of even length. If both are of even length, partitions $ABA^{\sim} \dots B^*$ and $A \dots BA^*B^{\sim}$ are not allowed as well. When $|A|$ is even and $|B|$ is odd, $A \dots BA^{\sim}B^*$ is not allowed. Likewise, if $|A|$ is odd and $|B|$ is even, $ABA^* \dots B^{\sim}$ is not allowed. One might note the following about the non-adjacent sequence $ABA'B'$:

- If an odd sequence B is enclosed by a sequence A and A^* (ABA^*), then we will know something about the enclosed B. This works for an enclosed sequence A as well.
- If a sequence B is enclosed by an even sequence A and A^* (ABA^*), then we will know something about the enclosing sequence A. This works for an enclosed sequence B as well.
- If $ABA'B'$ is evenly partitioned, where either A or B is of even length, and $A^{\sim}$ or $B^{\sim}$ occurs, then we will know something about A if $A^{\sim}$ occurs, or B if $B^{\sim}$.
- If $|B|$ is even, then the second occurrence of B starts with a white cell. If $|B|$ is odd, then the second occurrence of B starts with a black cell. If we know something about $|A|$, then the first sequence A will start with a black cell.

Example. Consider a line containing a sequence $abcde \dots fghiabcdeghif$ with $A = abcde$ and $B = fghi$, and thus a non-adjacent instance of $ABA'B'$. More specifically, the sequence can be written as $A \dots BA^*B^{\sim}$. Since $|A|$ is odd, either A or A^* will be maximally filled with black cells. Suppose A^* contains the maximum of black cells, meaning it starts and ends with a black element neighboring the two Bs. These Bs are equally positioned, meaning that they will mirror each other's patterns, which cannot occur when the A-neighboring elements of the B become white due to the black elements of A^*. Therefore, these B-neighboring cells of the As must be white. This results in the partially solved line $\underline{a}\underset{\cdot}{b}\underline{c}\underset{\cdot}{d}\underline{e} \dots fghi\underset{\cdot}{a}\underline{b}\underset{\cdot}{c}\underline{d}\underset{\cdot}{e}ghif$.

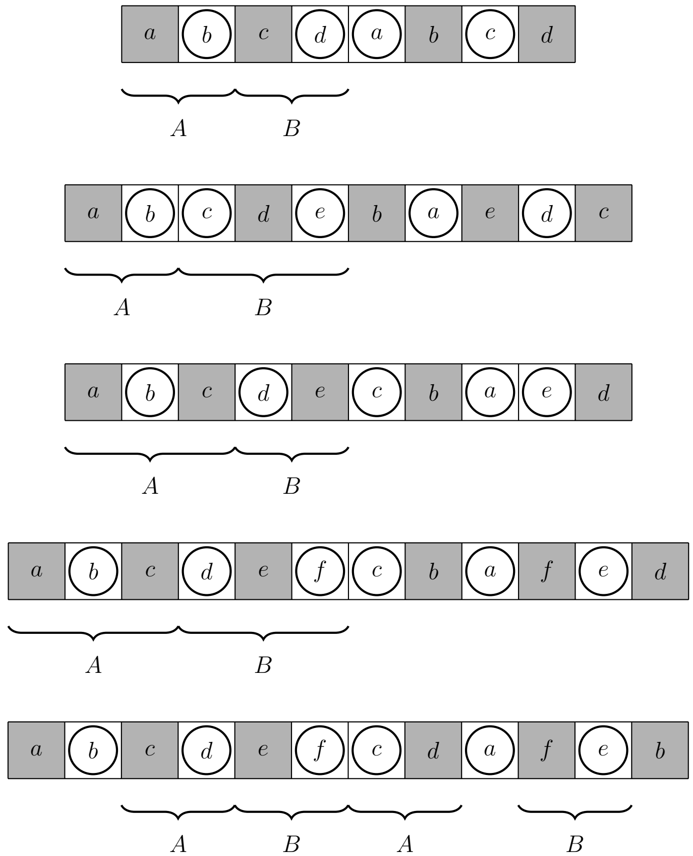


Figure 25: Lines solvable using the rule of alternating permutations.

Tables 6, 7, 8 and 9 show the properties of both patterns. Note that for both patterns, $|AB| \geq 3$ and $B \geq 0$. If only A occurs, the minimal length of $|A| \geq 2$. Reordering is allowed for the sequences of both patterns, as long as the patterns shown in Tables 6, 7, 8 and 9 are adhered. For an $ABB^{\sim}A^{*}$, the sequence instance of A will always be A^{*}. For non-adjacent instances of the pattern, there is one exception, as $AB \ldots B^{*}A^{\sim}$ is allowed for even $|A|$ and $|B|$.

$\|A\|$	$\|B\|$	Acceptable orders
Even	Even	All
Even	Odd	All
Odd	Even	All but $ABB^{\sim}A^{*}$
Odd	Odd	All but $ABB^{*}A^{*}$

Table 6: Overview of the acceptable occurrences of $ABB'A^{*}$.

$\|A\|$	$\|B\|$	Acceptable partitions
Even	Even	All but $AB \ldots B^{*}A^{*}$, $AB \ldots B^{\sim}A^{\sim}$, $ABB^{\sim} \ldots A^{*}$, $A \ldots BB^{\sim}A^{*}$
Even	Odd	All but $AB \ldots B^{*}A^{*}$, $ABB^{*} \ldots A^{*}$, $A \ldots BB^{*}A^{*}$
Odd	Even	All but $AB \ldots B^{\sim}A^{*}$, $ABB^{\sim} \ldots A^{*}$, $A \ldots BB^{\sim}A^{*}$
Odd	Odd	All but $AB \ldots B^{*}A^{*}$, $ABB^{*} \ldots A^{*}$, $A \ldots BB^{*}A^{*}$

Table 7: Overview of the acceptable partitions of $ABB'A^{*}$.

$\lvert A\rvert$	$\lvert B\rvert$	Acceptable orders
Even	Even	All but $ABA^{\sim}B^{\sim}$ (, including $ABA^{-1}B^{-1}$)
Even	Odd	All but $ABA^{*}B^{*}$
Odd	Even	All but $ABA^{*}B^{*}$
Odd	Odd	All (, thus $ABA'B'$)

Table 8: Overview of the restrictions of $ABA'B'$.

$\lvert A\rvert$	$\lvert B\rvert$	Acceptable partitions
Even	Even	All but $AB\ldots A^{*}B^{*}$, $ABA^{\sim}\ldots B^{*}$, $A\ldots BA^{*}B^{\sim}$
Even	Odd	All but $A\ldots BA^{\sim}B^{*}$,
Odd	Even	All but $ABA^{*}\ldots B^{\sim}$,
Odd	Odd	All but $AB\ldots A^{*}B^{*}$

Table 9: Overview of the acceptable partitions of $ABA'B'$.

In some cases, a sequence $s = ABA'B'$ or $s = ABB'A^{*}$ can be separated even further. Consider a sequence *abcdefghcbad ... fghe* as an instance of $ABA'B'$ for which $A = abcd$, $B = efgh$. One may partition this even further by *abcdefghcbad ... fg ... he*, where B' is partitioned as well. Similar to *abcdefghcbad ... fghe*, one is still able to fully determine A and A^{*} for *abcdefghcbad ... fg ... he*. However, partitioning such sequences depends on the content of the elements, which causes the patterns $ABA'B'$ or $ABB'A^{*}$ to not cover every possible partition of an adhering sequence.

While these patterns apply to numerous permutations, to guarantee that such patterns may only be applied to solvable sequences, we consider the following *enclosing permutations* and *alternating permutations*. Sequences containing $ABB'A^{*}$, are considered to be *enclosing permutations*, as the sets of A enclose those of B. Consider a sequence containing two occurrences of both A and B. If either A or B is of odd length, then for sequences that contain $AB\ldots B^{*}A^{*}$, B and B^{*} are fully determinable. The A-neighboring cells of the two Bs will be white, which determines the remainder of the two Bs immediately as well. If $\lvert A\rvert$ is even, and sequences A and A^{*} enclose either B and B^{*} or $B^{\sim}$, then the B-neighboring cell of the two As are determined to be white, which determines the remainder of the As as well. Lastly, if A and $A^{\sim}$ enclose B and B^{*} of even length, then one can completely determine B, where the A-neighboring cells are both white. Again, this fully determines the remainder of B.

Sequences that contain $ABA'B'$ are alternating between an instance of a sequence A, B and their permutations, hence, *alternating permutations.* Again, for even $\lvert A\rvert$, an instance of AA^{*} will lead to determining the sequence as their two neighboring will both be made white. When A and A^{*}, where $\lvert A\rvert$ is even, enclose an even length instance of B, the B-neighboring cells of the two instances of A may be either both black or white. If both are black, then the sequence will not have a solution, as B will either begin or end with a black cell, which will violate the no-neighboring black cells property of HITORI. So, the B-neighboring cells of the two As must be white. Note that when B and B^{*}, with even $\lvert B\rvert$, enclose an instance of A, either A^{*} or $A^{\sim}$, this principle will hold as well, resulting in determining the elements of B. When both A and B are of odd length, the instances of ABA^{*} and $BA^{*}B^{*}$ work as well. However, for instances of ABA^{*}, one can determine

B, as it must start and end with a white cell. Having an instance of $AB \dots A^*B^\sim$, when $|B|$ is even, determines the elements of both As, where the B-neighboring cells of both As becomes white. Similarly, if $AB \dots A^\sim B^*$, when $|A|$ is even, then the A-neighboring cells of both Bs becomes white. Once, these cells are known, the remainder is determined as well. Suppose either A or B is of even length, and the other of odd length. If the even length instances enclose the odd length instance, then the elements of the odd length are determined, i.e. B is determined for $ABA^\sim$ for $|A|$ even and $|B|$ odd. Table 10 provides an overview of acceptable partitions and orders of the enclosing and alternating permutations. For more (partially) solvable sequences adhering the enclosing or alternating permutations, see Table 11.

	Enclosing (ABBA based)	Alternating (ABAB based)
Even $\|A\|$	AA^*	AA^*
Even $\|B\|$	ABB^*A^*	ABA^*
	$ABB^\sim A^*$	BA^*B^*
	$AB \dots B^*A^\sim$	$BA^\sim B^*$
	$AB \dots B^\sim A^*$	$AB \dots A^*B^\sim$
		$AB \dots A^\sim B^*$
Even $\|A\|$	AA^*	AA^*
Odd $\|B\|$	ABB^*A^*	$ABA^\sim$
	$AB \dots B^*A^*$	$AB \dots A^\sim B^*$
Odd $\|A\|$	$AB \dots B^*A^*$	$BA^*B^\sim$
Even $\|B\|$		$AB \dots A^*B^\sim$
Odd $\|A\|$	–	ABA^*
Odd $\|B\|$		BA^*B^*

Table 10: Instances of the rules enclosing and alternating permutations.

The rules *enclosing permutations* and *alternating permutations* may be added to the rule-based classification, as these rules solely consider a single line. This allows earlier detection of more complex patterns.

Length	Sequence
6	a b c c a b
8	a b c d a b c d
10	a b c d e b a e d c
10	a b c d e c b a e d
10	a b c d e c d e a b
10	a b c d e e a b c d
12	a b c d e f c d a b f e
12	a b c d e f c d a f e b
12	a b c d e f c d e b a f
12	a b c d e f c d e f a b
12	a b c d e f f b a e d c
12	a b c d e f f c b a e d
12	a b c d e f f c d e a b

Table 11: Sequences that can only be solved using enclosing and alternating permutations.

5.2.3 Proof

To guarantee that one may always determine some elements of a Hitori line that contains enclosing or alternating permutations, consider Theorem 5.3.

Theorem 5.3. *If a sequence contains adjacent enclosing or alternating permutations, there will always be some gain.*

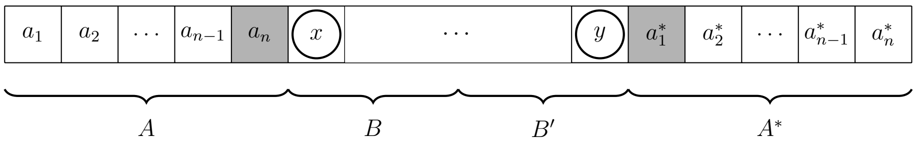
Proof. Suppose we have the following line, where we can apply $ABB'A^*$:

$$a_1a_2\dots a_{n-1}a_nb_1b_2\dots b_{m-1}b_mb'_1b'_2\dots b'_{m-1}b'_ma^*_1a^*_2\dots a^*_{n-1}a^*_n$$

The rule of enclosing permutations indicates that the cells of A that enclose BB' should always be white. Because the sequence A is of even length n, the sequence A will have $\frac{n}{2}$ black cells. For the even sequence, the pattern will either start with a black cell and end with a white cell, or vice versa. If one cell of the sequence A is known, the whole pattern is set immediately. Now, suppose a_n in the left A is black. Then, we know that for the left A, all even positions, and thus all elements on even positions, are black. For the right A, elements on the even positions will then all be white, causing the elements on odd positions to be black. The cell a^*_1 of the second A is thus black, enclosing BB' by two black cells. Consider b_1 of the left sequence B to be x, and b'_m of the right sequence to be y. The cells x and y must be white, as their neighbors of A and A^* are black. Two situations may occur:

- The cells x and y have the same value; $x = y$. Both cells are white, but, according to the rules of Hitori, only one of each value may be selected. Thus, we have a contradiction.

- The cells x and y have the differing values; $x \neq y$. We consider BB'. The sequence BB' consists of pairs of the same values, meaning that at least one of each value-pair should be black. In total, at least half of the cells of BB' will be black. However, x and y are white. Regardless of whether B is even or odd, BB' will be even, and thus m cells of BB' will be black. Since both x and y are white, and BB' is even, we will need to place m black cells into $2m-2$ remaining cells of BB'. This means that there will be black cells adjacent, which contradicts the rules of Hitori, where black cells may not be adjacent. Therefore, we have a contradiction.



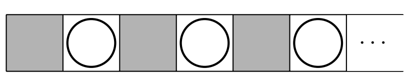
Therefore, if we have a line where we can apply $ABB'A^*$, we can say with certainty that the cells of A and A^* that enclose BB' should always be white. ☐

Proof. Suppose we have a line for which we can apply $ABA'B'$. Both A and B can be even or odd. If we can apply $ABA'B'$, then the a_1 of the first $A = a_1, \ldots a_n$ is always black and there exists at least one pair of two adjacent white cells.

If the sequence A is of even length n and occurs twice in a given line in any order, then for each A, $\frac{n}{2}$ cells should be black. To fill each A with $\frac{n}{2}$ black cells and that no black cells are adjacent, we know that there are only two ways to fill an even sequence:

This means that if one cell of A is determined, then all of A are determined. Note that, due to possible ordering, we cannot determine the pattern of the second occurrence A' solely based on the black-white pattern of the first occurring A.

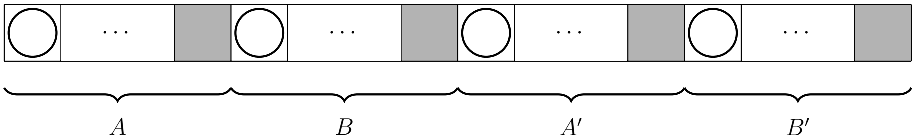
If we have an odd sequence A that occurs twice on a given line in any given order, then one of the occurring sequences A will have $\lfloor \frac{n}{2} \rfloor$ black cells and the other will have $\lceil \frac{n}{2} \rceil$ black cells. The sequence containing $\lceil \frac{n}{2} \rceil$ black cells will be considered to be the maximally filled sequence, and can only be filled in one way:



The minimally filled sequence will have two adjacent white cells.

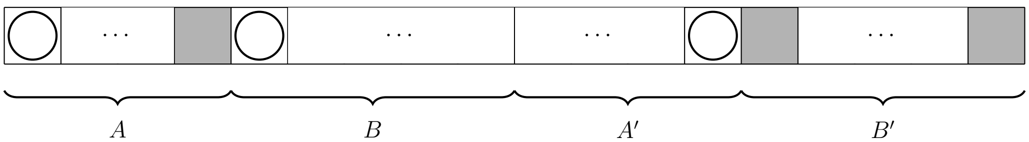
Now consider the different combinations for A and B.

(1) *A and B are both even.* Suppose a_1 of the first occurrence $A = a_1 \ldots a_n$ is white. If a_1 is white, then A_1 is completely determined, and will end with black. Since B_1 is neighboring A_1, the first cell of B_1 will then be white. B is even as well, so if the first cell of B_1 is white, the rest of B_1 will be set as well. The last cell of B_1 will then be black. A_1 and B_1 are now completely determined, and since the last cell of B_1 is black, A_2 will start with white and end with black. Then B_2 will start with white and end with black as well. This results in:

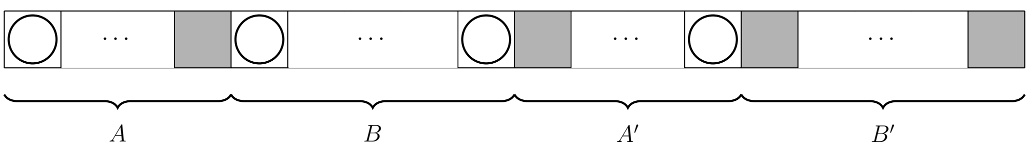


There is no room for two white neighbors to occur while filling half of the number of cells for each sequence.
So, for A is even, and B is even, a_1 of A_1 has to be black.

(2) *A is even, B is odd.* Again, suppose the first cell of A_1 is white. Then the last cell of A_1 will be black, causing the first of B_1 to be white. Since B is odd, and B_1 starts with white, B_1 cannot cover $\lceil \frac{m}{2} \rceil$ black cells. Thus, B_2 has to cover $\lceil \frac{m}{2} \rceil$ black cells. The progress on the line so far:



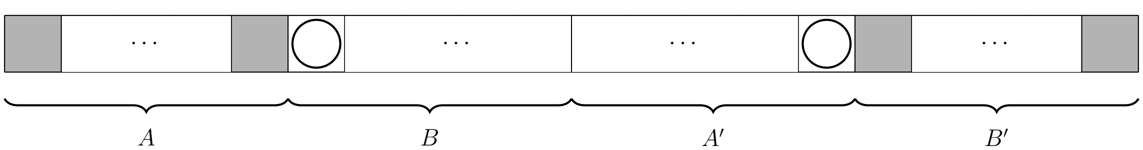
Since B_2 is maximally filled, the last cell of A_2 will be white. A is even, so A_2 is completely determined now. The first cell of A_2 is black, and thus the last cell of B_1 is white. Now B_1 starts and ends with a white cell. This results in the following line:



In order to fill B_1 with $\lfloor \frac{m}{2} \rfloor$ black cells, B_1 cannot have two white neighbors. Therefore, if A is even and B is odd, the first cell of A_1 has to be black.

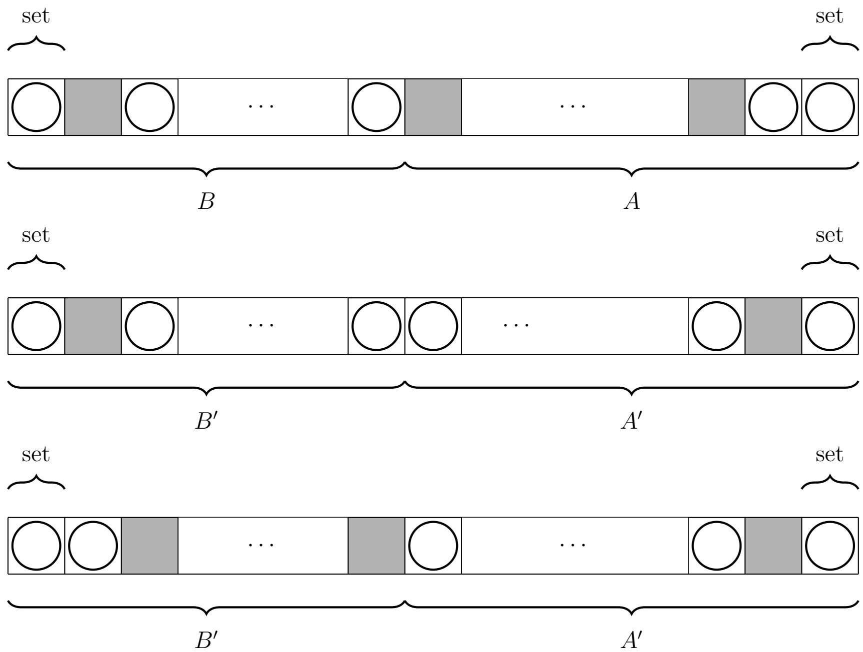
(3) *A is odd, B is even.* Considering the reasoning used for (2), the first cell of A_1 has to be black, if A is odd and B is even.

(4) *A and B are both odd.* The maximally filled sequence of A and B cannot be adjacent. So $A_{max}BAB_{max}$ is the only configurable line. The first cell of A_1 has to be black, otherwise it cannot be the maximally filled sequence. Since A_1 ends with a black cell, the first of B_1 will be white. The last cell of A_2 will be white as well, as B_2 will start with a black cell. We will have the following line:



So a_1 of the first A is black.

Now, zoom into B_1A_2. We know that the outer cells are white. For B_1, we still need to place $\lfloor \frac{m}{2} \rfloor$ black cells. To do so, we will need $2\lfloor \frac{m}{2} \rfloor$ cells. The first cell is already determined, we can then add $2\lfloor \frac{m}{2} \rfloor$ cells starting from the second cell. The remainder of B_1 is even, meaning that we can fill the remainder of B_1 as either ■◯■◯... or ◯■◯■.... The same applies to A_2. We can make the following combinations, which all contain two white neighbors:



So, for A and B are odd, we see that every possible combination contains an occurrence of two white neighbors.

The same reasoning regarding two white neighbors works when considering two even sequences A and B. Since an even sequence cannot have all elements, that were on even positions, on odd positions, as this would violate the restrictions of $ABA'B'$, we know that the pattern of $ABA'B'$ cannot fully consist of ■○■○ . . .. Therefore, there needs to be at least one occurrence of two white neighbors.

For A is even and B is odd, and A is odd and B is even, When having a combination of an even and odd length sequence, there is one sequence of odd length that is maximally filled with black cells, and thus has a first and last cell that are black. The adjacent cell of the adjacent even length sequence, will then be white. As mentioned earlier, if one cell of an even sequence is determined, the whole sequence is completely determined. The one end of the even length sequence is white, therefore the other end should be black. Now, considering the adjacent odd length sequence, the adjacent cell should be white. The remainder of this sequence is now even, and we can apply the same approach as described for two odd sequences.

So, every valid solution of $ABA'B'$ starts with a black cell and has at least one occurrence of two adjacent white cells. □

5.2.4 Finding enclosing and alternating permutations

For the smallest Hitori, $m, n \leq 5$, there is no need to check for the occurrence of $ABA'B'$ and $ABB'A^*$, as these only occur for $|AB| \geq 3$. To find a permutation p in a sequence of length ℓ, one can simply use a brute force approach with time complexity $\mathcal{O}(|p| \cdot \ell)$. However, scaling in quadratic time becomes troublesome for larger m and n. Instead of using a brute force approach, one can use a *suffix tree* to find larger occurrences of $ABA'B'$ and $ABB'A^*$. A line of the Hitori can be converted into a suffix tree, where the tree considers all possible suffixes of the line. Compressing the suffix tree results in a compact suffix tree, which saves space. If a leaf is reached when traversing through the suffix tree to find an instance of the permutation p, an occurrence of p is found. The time complexity of the construction of a suffix tree is $\mathcal{O}(\ell)$. Combined with checking whether the pattern is present in the sequence, the time complexity will be $\mathcal{O}(\ell + |p|)$. Using a suffix tree makes searching for a pattern more efficient. Checking all lines for $m, n \geq 10$ becomes rather exhaustive when using a brute force approach, making it more beneficial to use a suffix tree.

5.3 Permutations

Lines that adhere to the pattern $ABA'B'$ or $ABB'A^*$ essentially consist of two parts that are permutations of one another. These two parts, a sequence and its permutation, can be adjacent or non-adjacent. Depending on the permutation A' of A, the number of solutions will vary between none and more than two. Generally, when a line contains A and A', the line has

- 0 solutions; if every possible solution is contradicting.
- a partial solution; if there is at least one contradicting solution.
- 1 solution; all, except one, possible solutions lead to a contradiction.
- 2 solutions; only alternating solutions are permitted.
- 2+ solutions; if there are exactly two contradicting solutions.

Now, consider a line that contains A and A'. Table 12 provides an overview of the number of sequences for different lengths $|A|$ that are not adjacent to A', and the number of solutions that have either 0, partial, 1, 2 or more than 2 solutions. For an adjacent A and A', see Table 13.

$\|A\|$	0	partial	1	2	2+	Total
2	0	0	0	2	0	$2! = 2$
3	0	4	0	2	0	$3! = 6$
4	2	4	10	6	2	$4! = 24$
5	32	44	32	12	0	120
6	298	62	288	58	14	720
7	3,000	696	1,200	144	0	5,040
8	28,518	1,192	9,458	998	154	40,320
9	295,920	16,560	47,520	2,880	0	362,880
10	3,174,938	31,274	393,788	26,202	2,598	3,628,800
11	36,945,216	570,816	2,314,368	86,400	0	39,916,800
12	455,869,606	1,122,820	20,972,374	972,262	64,538	479,001,600

Table 12: The type and number of solutions for $|A| \leq 12$ where A and A' are not adjacent.

$\lvert A\rvert$	0	partial	1	2	2+	Total
2	0	0	1	1	0	2
3	1	1	2	2	0	6
4	9	1	10	4	0	24
5	66	6	36	12	0	120
6	512	8	164	36	0	720
7	4,088	56	752	144	0	5,040
8	35,800	88	3,856	576	0	40,320
9	337,800	840	21,360	2,880	0	362,880
10	3,486,240	1,440	126,720	14,400	0	3,628,800
11	38,985,840	19,440	825,120	86,400	0	39,916,800
12	472,815,504	34,704	5,632,992	518,400	0	479,001,600

Table 13: The type and number of solutions for $|A| \leq 12$ where A and A' are adjacent.

When determining the number of sequences that have exactly two solutions, we must observe the possible solution layouts. For a sequence A of length $|A|$, there are $\frac{|A|-2}{2}$ possible solution layouts that have two adjacent white elements. The total number of solution layouts is $\frac{|A|-2}{2} + 2$, as there are exactly two solution layouts that are fully alternating between black and white. For non-adjacent sequences A and A' of even length, there are exactly

$$2 \cdot \left(\left(\frac{|A|}{2} \right) ! \right)^2$$

sequences that have an alternating sequence, meaning that these sequences have at least two solutions. Alternating black and white for an odd $|A|$, directly determines the layout of an occurring A once another occurrence of A has been set. It does not matter whether A and A' are adjacent or not. Therefore, the number of sequences for an odd $|A|$ that have at least two solutions is determined by

$$\left(\left\lfloor \frac{|A|}{2} \right\rfloor \right) ! \cdot \left(\left\lceil \frac{|A|}{2} \right\rceil \right) !$$

Consider an even $|A|$ where sequences $A = a_1 \dots a_n$ and $A' = a'_1 \dots a'_n$ are adjacent. For sequences that consider an adjacent A and A', and have exactly two solutions, the sequence A' needs to alternate between black and white elements. Since A ends with a_n, a'_1 has to represent an odd indexed element of A. Once a'_1 is determined, the remainder of A' is determined as well. So, regardless of whether $|A|$ is even or odd, if A and A' are adjacent, the number of sequences with exactly two solutions are

$$\left(\left\lfloor \frac{|A|}{2} \right\rfloor \right) ! \cdot \left(\left\lceil \frac{|A|}{2} \right\rceil \right) !$$

Number of sequences with exactly two solutions. Considering sequences that have exactly two solutions, the alternating sequences need to be preserved and any other sequence needs to be contradicted. To create a contradiction, consider the white cells of a solution layout. An even sequence has exactly $\frac{|A|}{2}$ white cells, resulting in $\binom{\frac{|A|}{2}}{2}$ pairs that cause a contradiction for a given solution layout. Since there are $\frac{|A|-2}{2}$ possible solution layouts with each $\binom{\frac{|A|}{2}}{2}$ pairs that cause a contradiction, there are $\frac{|A|-2}{2} \cdot \binom{\frac{|A|}{2}}{2}$ possible pairs to evaluate. However, many of these pair consider elements that are both on an even or odd position in the first A sequence, resulting in a second sequence that cannot alternate if that pair is included. Removing such pairs provides the total number of pairs that do not violate the alternating property as

$$T_n = \frac{1}{6}\frac{|A|}{2} \cdot \left(\frac{|A|}{2} + 1\right) \cdot \left(\frac{|A|}{2} + 2\right)$$

where T_n is equivalent to the *Tetrahedral numbers* [6]. These pairs are represented by $\frac{1}{2}\frac{|A|-2}{2}(\frac{|A|-2}{2} + 1)$ unique pairs, which are equivalent to the *Triangular numbers* [7]. Considering $|A| \geq 4$, where A is of even length, there needs to be at least a pair containing a_2 present in the sequence for it to be able to contradict all possible solution layouts that contain white neighbors. There are $\frac{|A|}{2} - 1$ pairs that contain a_2 and contradict a possible solution layout. There will be $\frac{|A|}{2} - 2$ different pairs that contain a_4, $\frac{|A|}{2} - 3$ different pairs that contain a_6, which will be repeated until there is only one distinct pair for an element. Depending on how much solution layouts the pair containing a_2 covers, there are a number of additional pairs needed to cover the remaining layouts. Determining the number of sets that cover all solution layouts that contain white neighbors is an instance of the set coverage problem, and is therefore NP-complete. The number of minimal set covers, that contradict all solutions where two white elements are adjacent, can be determined by using

$$a(n) = 3a(n-1) - a(n-2)$$

where $n \geq 2$ and $a(0) = a(1) = 1$. Another way to determine the number of minimal sets covers is by summing the minimal set covers of length ℓ can be found by using $T(n, \ell) = \binom{n+\ell}{2\ell}$ (A085478);

$$\sum_{\ell=0}^{n} \binom{n+\ell}{2\ell}$$

where $n = \frac{|A|-2}{2}$. Once the number of minimal set covers is determined, one can use the inclusion-exclusion principle [12] to determine the cardinality of the minimal set covers as

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{k=1}^{n}(-1)^{k+1}\left(\sum_{1\leq i_1<\cdots<i_k\leq n} |A_{i_1} \cap \cdots \cap A_{i_k}|\right)$$

Doing so, results in the number of sequences of length $|A|$ that have exactly two solutions. This determines the number of sequences with more than two solutions as well.

Example. For $|A| = 8$, with $A = abcdefgh$, there are five possible solutions. For each solution, there are six pairs that cause a contradiction;

(1) $ZWZWZWWZ$ with bd, bf, bh, df, dh, fh

(2) $WZWZWZWZ$ with ac, ae, ag, ce, cg, eg

(3) $ZWWZWZWZ$ with bc, be, bg, ce, cg, eg

(4) $ZWZWWZWZ$ with bd, be, bg, de, dg, eg

(5) $ZWZWZWWZ$ with bd, bf, bg, df, dg, fg

To find the number of sequences that have exactly two solutions, pairs that consist of two elements on even positions in A, i.e., $a_2, a_4, \dots, a_{n-2}, a_n$, or odd positions, i.e., $a_1, a_3, \dots, a_{n-3}, a_{n-1}$, can be removed as these violate the alternating solutions. Consider solution layouts (3), (4) and (5) for which the same positioned pairs are removed;

(3) $ZWWZWZWZ$ with bc, be, bg

(4) $ZWZWWZWZ$ with be, bg, de, dg

(5) $ZWZWZWWZ$ with bg, dg, fg

Figure 26 shows how these pairs contradict different solution layouts.

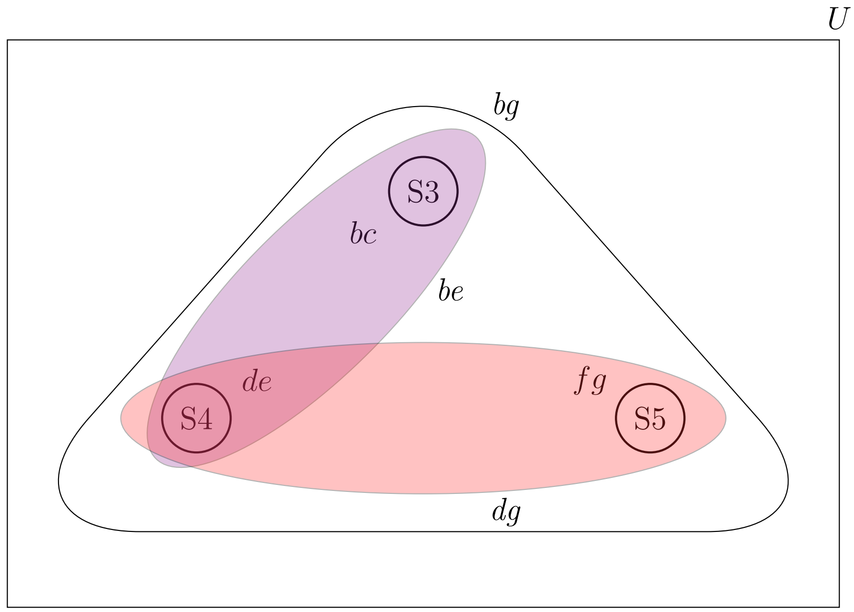


Figure 26: A visualization of how the contradicting pairs capture different solution layouts, where Sn, with $n = 3, 4, 5$ denotes the solution layout.

There are 5 minimal set covers that violate layouts (3), (4) and (5); $\{bg\}$, $\{be, dg\}$, $\{be, fg\}$, $\{bc, dg\}$, $\{be, de, fg\}$. Now, for each set cover, determine the number of sequences.

1. $\{bg\}$; The pair bg may occur mirrored as gb. There are $2 \cdot \left(\left\lceil \frac{|A|}{2} \right\rceil - 1\right)! \left(\left\lfloor \frac{|A|}{2} \right\rfloor - 1\right)! =$ 72 possible combinations where bg or gb occurs. The sequence has to be alternating, which limits the number of positions on which bg or gb can occur to seven positions. In total, there are 504 alternating sequences where bg or gb occurs.

2. $\{be, dg\}$; Placing the non-overlapping elements results in $\left(\left\lceil \frac{|A|}{2} \right\rceil - 2\right)! \left(\left\lfloor \frac{|A|}{2} \right\rfloor - 2\right)! =$ 4 possible ways to place the remaining elements. Again, be and dg can be mirrored. However, to preserve the alternation between even and odd elements, placing the first element of a pair on an odd or even position is considered separately. There are four positions to place the first pair, which leaves three positions for the second pair, when the first pair starts on an odd position. If the first pair starts on an even position, there are three positions where the first pair can occur. This leaves two positions for the second pair. There are $2 \cdot (4 \cdot 3 + 3 \cdot 2) \cdot 4 = 144$ alternating sequences

where be and dg or eb and gd occur. Taking alternation into account, the number of ways to position be and gd or eb and dg is limited to twelve. Thus, there are $2 \cdot (2 \cdot 2 \cdot 3) \cdot 4 = 96$ sequences where be and gd or eb and dg occur. Considering all positions, there are 240 sequences where be and gd, or mirrored, occur.

3. $\{be, fg\}$; There are 240 alternating sequences where $\{be, fg\}$, or mirrored, occur.

4. $\{bc, dg\}$; There are 240 alternating sequences where $\{bc, dg\}$, or mirrored, occur.

5. $\{bc, de, fg\}$; After placing bc, de and fg, the remaining two elements have one possible position. Again, the sequences can be mirrored. Starting at an odd position, there are four ways to place the first pair, three ways to place the second pair and two ways to place the third pair. Starting on an even position in the sequence results in three ways to place the first pair, two for the second pair and one for the third pair. This results in $1 \cdot 2 \cdot (4 \cdot 3 \cdot 2 + 3 \cdot 2 \cdot 1) = 60$ alternating sequences where bc, de, fg, or mirrored, occur. Considering $\{bc, de, gf\}, \{bc, ed, fg\}$ and $\{bc, de, gf\}$, there are ten possible positions for each of these covers, which can be mirrored as well. This results in $3 \cdot 10 \cdot 2 = 60$ alternating sequences that contain $\{bc, de, gf\}, \{bc, ed, fg\}$ or $\{bc, de, gf\}$. In total, there are 120 alternating sequences where bc, de, fg or any mirroring occurs.

To find the number of sequences of the union of these covers, the inclusion-exclusion principle must be used. As there exist five minimal set covers, the principle provides $|\{bg\} \cup \{be, dg\} \cup \{be, fg\} \cup \{bc, dg\} \cup \{be, de, fg\}|$, which results in $1344 - 390 + 50 - 6 + 0 = 998$ sequences for $|A| = 8$ with exactly two solutions. There are $2 \cdot (\frac{8}{2})!^2 - 998 = 154$ sequences with more than two solutions.

Theorem 5.4. *If A and A' are adjacent, there will always be at most 2 possible solutions.*

Proof. Suppose the sequence $A = a_1 a_2 \dots a_n$ and $A' = a'_1 a'_2 \dots a'_n$ are adjacent, where $|AA'| = 2n$. Consider a_n and a'_1, as they are adjacent. Deselecting both elements may not occur, as no black elements may be neighbors. Either a_n, a'_1 or both are selected, i.e. made white.

$|A|$ is even. When both a_n and a'_1 are white, the remaining elements are determined as well, as half of the elements of A must be black, see Lemma 5.1. If a_n becomes black, the elements a_{n-1} a'_1 will be white. The remainder of A and A', i.e. the sequences $a_1 \dots a_{n-1}$ and $a'_2 \dots a'_n$, become of odd length. To eliminate all duplicates, either $a_1 \dots a_{n-1}$ or $a'_2 \dots a'_n$ has to contain $\lfloor \frac{m}{2} \rfloor + 1$ black elements, i.e. the maximum number of black elements. The maximum number of black elements may only occur in an alternating manner, starting and ending with a black element. Since a_{n-1} has become white, the sequence $a_1 \dots a_{n-1}$ cannot contain the maximum. Therefore, $a'_2 \dots a'_n$ will have $\lfloor \frac{m}{2} \rfloor + 1$ black elements, and immediately determines $a_1 \dots a_{n-1}$ as well. Making a_n black results in a fully determined solution. Respectively, making a'_1 becomes black, will lead to a fully determined solution as well. Note, that it is possible that it is possible for a sequence to have exactly two possible solutions.

$|A|$ is odd. See Theorem 5.5. □

Theorem 5.5. *If $|A|$ is odd, regardless of A and A' being separate or adjacent, there will be at most 2 solutions.*

Proof. Suppose the sequence $A = a_1a_2 \dots a_n$ and $A' = a'_1a'_2 \dots a'_n$ occur on a given line, where they can be adjacent or separated.

Either A or A' contains the maximum number of black elements. As the maximum number of black elements may only occur in an alternating manner, starting and ending with a black element, the corresponding sequence will be fully determined. The remaining sequence will then mirror the elements in the already determined sequence. Suppose that both a black a_n and a black a'_1 lead to a valid sequence, then the sequence has exactly two solutions. When either a black a_n or a'_1 leads to a valid sequence, there will be at least some part of the solution that is known. If none of these lead to a valid sequence, there will be zero solutions. Thus, regardless of A and $|A'|$ being separate or adjacent, there will be at most 2 solutions when $|A|$ is odd. □

Number of sequences with zero solutions. Similar to the approach of determining the number of sequences that have exactly two solutions, consider all possible solution layouts. Here, the alternating solutions are included as well, meaning that there are $\frac{|A|-2}{2} + 2$ solutions to consider. As the two fully alternating solution layouts have no overlapping elements, there needs to be at least one covering pair for each alternating sequence to ensure that no solution is possible. Therefore, there needs to be an adjacent pair of even elements, and an adjacent pair of odd elements. If another pair is needed to cover all solutions, then such a pair will consist of an even and odd element. For each solution, there are $\binom{\frac{|A|}{2}}{2}$ pairs that cause a contradiction. This results in a total of $\left(\frac{|A|-2}{2} + 2\right)\binom{\frac{|A|}{2}}{2}$ possibly overlapping pairs that cause some contradiction. To provide an upper bound on the number of pair combinations that cause zero solutions, we may consider every pair combination that covers the two fully alternating solutions, meaning one needs to evaluate $\binom{\frac{|A|}{2}}{2} \cdot \binom{\frac{|A|}{2}}{2}$ pair combinations. After evaluating the two fully alternating solutions, one needs to consider the solutions that contain two adjacent squares. As described previously, those are equivalent to the Tetrahedral numbers. Since the number of minimal set covers of the solutions containing two adjacent white cells, the number of minimal set covers needed to obtain zero solutions has an upper bound of

$$\binom{\frac{|A|}{2}}{2}^2 \cdot \sum_{\ell=0}^{\frac{|A|-2}{2}} \binom{\frac{|A|-2}{2} + \ell}{2\ell}$$

The set of sequences that have zero solutions where A and A' are not adjacent, are a proper subset of the those where A and A' are adjacent. The remainder of the sequences with zero solutions are those that cause a contradiction because a_n and a'_1 are adjacent.

Theorem 5.6. *If a line containing a sequence A has an instance A' with adjacent pairs $\{a'_2, a'_3\}, \{a'_4, a'_5\}, \dots, \{a'_{n-2}, a'_{n-1}\}$ and $\exists x, y : \{a'_x, a'_{x+1}\} \wedge \{a'_y, a'_{y+1}\}$, where x and $x + 1$ are elements on even positions and y and $y + 1$ are elements on odd positions, then the line has zero solutions.*

Proof. Suppose there is a sequence of length $|A|$. There are $\frac{|A|-2}{2} + 2$ possible solution layouts.

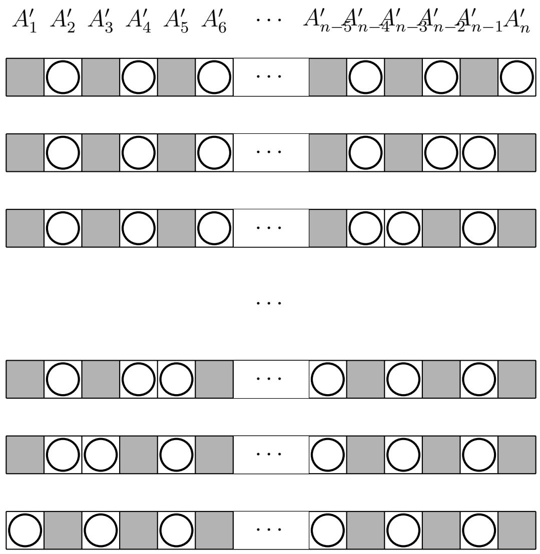


Figure 27: All solution layout for a sequence A', where $|A'|$ is even.

All solution layouts, except the first and the latter, have an adjacent white pair, see Figure 27. So, for each of these solution layouts, take the adjacent white pair. If done for all, $\frac{|A|-2}{2}$ solution layouts are contradicted. The remaining two solution layouts are fully alternating. To contradict those, a pair of elements on even positions needs to be adjacent, as well as a pair of elements on odd positions. All solution layouts are contradicted, and thus sequences that have pairs $\{a'_2, a'_3\}, \{a'_4, a'_5\}, \ldots, \{a'_{n-2}, a'_{n-1}\}$, one pair consisting of elements on even positions and one pair consisting of elements on odd positions have zero solutions. $\square$

Theorem 5.7. *Suppose a line contains a sequence A and A'. Any line that has zero solutions needs at least one adjacent pair in A' of two elements on even positions of A, and at least one adjacent pair in A' of two elements that are on odd positions of A.*

Proof. Consider Figure 27 once more. One pair can at most cover $\left\lceil \frac{1}{2}(\frac{|A|-2}{2} + 2) \right\rceil$ different solution layouts. The layouts all consist of alternating black and white, even those that have two adjacent white elements. Consider starting the alternation with black. The first layout fully alternates when starting on the left, the last layout fully alternates when starting on the right. The remaining layout start alternating from both left and right, and will meet at one point in the sequence. For the first layout, n elements are *left-alternating*, leaving zero *right-alternating* elements. The second layout has $n-2$ left-alternating elements and two right-alternating elements. This continues until all elements are right-alternating. The pair $\{a'_2, a'_4\}$ then is able to cover layouts that have at least four left-alternating elements, which are $\left\lceil \frac{1}{2}(\frac{|A|-2}{2} + 2) \right\rceil$ different layouts. Similarly, the pair $\{a'_{n-3}, a'_{n-1}\}$ covers layouts that have at least 4 right-alternating elements, which are $\left\lceil \frac{1}{2}(\frac{|A|-2}{2} + 2) \right\rceil$ different layouts. Since A' is of even length, there is exactly one layout that the same number of left- and right-alternating elements.

The adjacent pairs $\{a'_2, a'_4\}$ and $\{a'_{n-3}, a'_{n-1}\}$ each cover $\left\lceil \frac{1}{2}(\frac{|A|-2}{2} + 2) \right\rceil$ different solution layouts. When A' contains both pairs, all possible layouts are contradicted, meaning that one pair of elements on odd positions and one pair of elements on even positions are able to cover all possible layouts. $\square$

6 Satisfiability

While using a set of rules is quite straightforward, using the principle of satisfiability might allow for capturing more complex patterns earlier on in the solving process. In this section, a 2-satisfiability (2-SAT) approach will be introduced for solving lines of a Hitori. To capture the connectivity of the puzzle in terms of satisfiability, a distance matrix will be introduced. This matrix can then be combined with the approach described in Section 6.1, expressed in 3-SAT, to solve a Hitori.

6.1 2-SAT

Solving a single line, i.e., a row or column, of a Hitori is based on the constraints that no duplicates may occur, and no black cells may be adjacent. Assume that a line may be partially solved, meaning that some cells are already set to black or white. Uniquely occurring cells on a given line are temporarily selected, i.e. made white, albeit we do not know whether they are indeed fully unique in their corresponding orthogonal line. If such a cell is unique on the given line, but was not already selected, then there are other elements in its orthogonal line that prevent selection. One cannot determine whether this cell should be selected or not by solving the considered line, meaning that there is no use to include this cell when trying to find progress on the line. To ensure this cell does not affect the determination of the other cells, one can temporarily select the cell. Since every uniquely occurring cell is already determined, only cells that occur at least twice are considered. These cells may be separated by sequences of already solved cells, creating disjoint sequences of unsolved cells, which can be referred to as “blocks”. For each unsolved cell, we introduce a Boolean variable, where `true` represents the selection of the corresponding cell. These variables need to obey the rules of Hitori, meaning the following can be said about the Boolean variables of the unsolved cells:

- Consider neighboring cells x and y. If x is selected, y can be selected as well. However, both cannot be black, or deselected. So, their Boolean values must satisfy $\neg x \rightarrow y$, or $x \vee y$.
- A solved line may not contain any duplicates. If a cell x and y contain the same value a, both cannot be selected simultaneously. The corresponding Boolean values must satisfy $x \rightarrow \neg y \wedge y \rightarrow \neg x$, or $\neg x \vee \neg y$.

These rules, captured in *Conjunctive Normal Form* (CNF), can express an entire line in Boolean variables. However, there is no guarantee that a line will be fully solved. Based on the distribution of the blocks, a line might be partially solvable, at best, or completely unsolvable. While the latter results in an unsatisfiable Hitori, partial progress may lead to a fully solved line later on.

While satisfiability problems are NP-complete, solving such lines can be described using 2-SAT, and can thus be solved in linear time. Consider a variable x and its negation $\neg x$. If both x and $\neg x$ are unsatisfiable, the line has no solution. Using the negations of variables as literals, we can create an implication graph. Considering the *strongly connected components* (SCCs) of the implication graph, we can then check if a variable x and its negation $\neg x$ are part of the same SCC. If so, we can reach x from $\neg x$ and vice versa.

Selecting x or $\neg x$ will then always lead to a contradiction, which causes the entire HITORI to be unsatisfiable. To find the SCCs, we can use the linear time algorithm by Aspvall, Plass and Tarjan [1], or Kosaraju's algorithm [13]. When all x and $\neg x$ are evaluated and determined not to be in the same SCC, one needs to check whether selecting a variable y leads to $\neg y$ or vice versa, but not both. If y leads to $\neg y$, and thus a contradiction, the literal $\neg y$ needs to be selected. For $\neg y$ leading to y, the literal y needs to be selected. Checking this for all variables, leads to a selection of literals. Overlapping all possible outcomes shows which variables will always be selected or unselected.

6.1.1 Example

Consider the partially solved line shown in Figure 28 (top). Those numbers that are not solved occur more than once. We consider the Boolean variables $x_0, \ldots, x_7$, corresponding with these cells. Since neighboring cells cannot both be deselected, we get the following formula in conjunctive normal form (CNF):

$$(x_0 \vee x_1) \wedge (x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge (x_6 \vee x_7)$$

Based on the Boolean expression for two cells that have the same value, we get the following CNF formula for the given line:

$$(\neg x_0 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_4 \vee \neg x_7) \wedge (\neg x_5 \vee \neg x_6)$$

The graph in Figure 28 (middle) provides an implication graph of the used clauses. Black arrows indicate clauses that are based on neighboring cells, blue arrows indicate clauses that are based on those that have matching values. Keep in mind that deselecting a cell does not always lead to selecting a duplicate of this cell, as there might be another occurrence of this cell later on in the line.

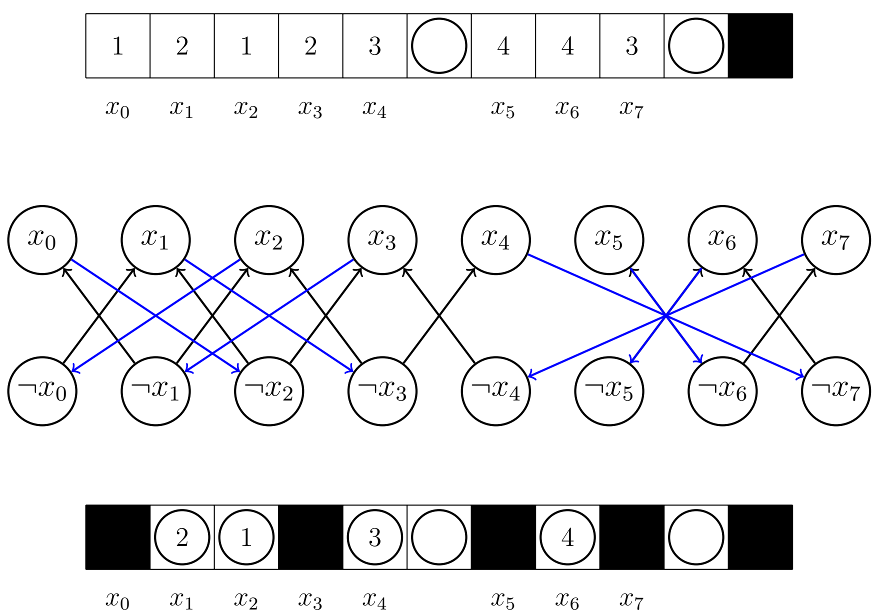


Figure 28: An example line, its implication graph and solution.

The line shown in Figure 28 (top) has several SCCs: $\{x_0, x_3, \neg x_1, \neg x_2\}$, $\{\neg x_5, x_6\}$, $\{\neg x_6, x_5\}$, $\{\neg x_7\}$, $\{\neg x_0\}$, $\{x_2\}$, $\{x_4\}$, $\{\neg x_3\}$, $\{x_1\}$, $\{\neg x_4\}$, $\{x_7\}$. Checking whether the line yields unsolvable, one can see that none of the SCCs contain a variable and its negation. The line is thus considered to be solvable. Following the implication graph shown in Figure 28 (middle), selecting x_0 will lead to a contradiction. Starting with x_0, we can reach $\neg x_2$. From here, we can reach x_1 and x_3, where the latter leads to $\neg x_1$. The variable $\neg x_1$ is thus reachable from x_1, which causes a contradiction. Therefore, we know that we cannot use x_0. Now, starting with $\neg x_0$, we can reach x_1, $\neg x_3$, x_4, x_2, $\neg x_7$, x_6 and $\neg x_5$, which results in a valid solution for the considered line. Selecting x_0 will lead to a contradiction, and selecting $\neg x_0$ will give a solution, meaning that we must use $\neg x_0$ to have a valid solution. For each variable and its negation, we can check whether they lead to a contradiction. Doing so will give a fully solved line, as shown in Figure 28 (bottom).

6.2 Capturing Connectivity

As stated by Tran [16], capturing the connectivity of HITORI is hard. In Section 6.1, the rules of having no duplicate white cells and no neighboring black cells were expressed using 2-CNF. One may simultaneously apply these clauses to all rows and columns. However, these clauses do not capture the connectivity of HITORI. To capture the connectivity, we introduce a distance matrix, also expressed in CNF.

Suppose we have a distance matrix d of size $\ell \times k_{max}$, where $\ell = m \times n$ and k_{max} denotes the maximum number of steps in a given path from the so-called *ground cell*. The Boolean variable $d_{i,k}$ is true when cell i is at distance k from the ground cell in the given path. The ground cell will have a distance zero, and thus $d_{i,0}$ will be true. For every other white cell i, where $1 \leq i \leq \ell$, $d_{i,0}$ will be false and there must exist a neighboring cell that is one step closer to the ground cell on the given path:

$$d_{i,k} \rightarrow (d_{i_\uparrow,k-1} \vee d_{i_\downarrow,k-1} \vee d_{i_\rightarrow,k-1} \vee d_{i_\leftarrow,k-1})$$

where $\{i_\uparrow, i_\downarrow, i_\leftarrow, i_\rightarrow\}$ denote the indices of the top, bottom, left and right neighbor, respectively. For the $m \times n$ HITORI, if such neighbors exist, these neighbors can be determined as:

- Top neighbor, where $\forall i \geq n+1 : i - n$
- Bottom neighbor, where $\forall i + n \leq m \cdot n : i + n$
- Left neighbor, where $\forall (i \bmod n \neq 1) : i - 1$
- Right neighbor, where $\forall (i \bmod n \neq n) : i + 1$

Note that the upper bound of k_{max} is the maximum number of white cells, and thus $m \times n$. Therefore, the distance matrix will be at most a grid of size $\ell \times \ell$. The lower bound of k_{max} is the total number of cells minus the maximum number of black cells, as will be described in Section 7.3.

The connectivity constraint applicable in HITORI, is present in other types of puzzles as well, such as Heyawake and Nurikabe. For grid-based puzzles, one can express the puzzle as a graph, where one can express connectivity. Similar to Theorem 1 introduced by van der Knijff [17] on connectivity of a graph, one may define the following Theorem.

Theorem 6.1. *A graph $\mathcal{G}(V, E)$ is connected if and only if there exists a mapping $f : V \to \mathbb{N}$ such that*

1. *there is a unique vertex $g \in V$ with $f(g) = 0$*
2. *for all $i \in V - \{g\}$, there exists $j \in V$ such that $(i, j) \in E$ and $f(j) + 1 = f(i)$.*

To capture the connectivity of an $m \times n$ Hitori, one may consider using Theorem 6.1 along with the collections $\mathcal{B}$ of black cells and $\mathcal{W}$ of white cells. This can be used to construct a CNF formula containing the following:

- There is exactly one unique white cell that is the ground cell, meaning that

$$\exists i \in \mathcal{W} : d_{i,0} \wedge \bigwedge_{1 \leq i,j \leq \ell} (\neg d_{i,0} \vee \neg d_{j,0})$$

- For every other white cell in the set $\mathcal{W}$, $d_{i,k}$ must have a neighbor for which the distance is one less than the provided k, and thus

$$\forall i \in \mathcal{W} : d_{i,k} \to (d_{i_\uparrow,k-1} \vee d_{i_\downarrow,k-1} \vee d_{i_\rightarrow,k-1} \vee d_{i_\leftarrow,k-1})$$

  This can be rewritten in CNF as

$$\forall i \in \mathcal{W} : \neg d_{i,k} \vee d_{i_\uparrow,k-1} \vee d_{i_\downarrow,k-1} \vee d_{i_\rightarrow,k-1} \vee d_{i_\leftarrow,k-1}$$

- All black cells in set $\mathcal{B}$ are not reachable and thus have "no" distance to the ground cell: $\neg x_i \to \neg d_{i,k}$. In CNF, this is written as

$$\forall i \in \mathcal{B}, \forall k : x_i \vee \neg d_{i,k}$$

Combining these with the CNF formula described in Section 6.1 will allow one to capture the connectivity of a Hitori.

6.2.1 Example

To check whether one can make progress on a partially solved Hitori, we consider the following approach. Figure 29 provides a partially solved Hitori, which until this point may have been solved solely using the 2-SAT approach described in Section 6.1. For each cell, the top-right positioned number provides the index of the corresponding Boolean variable.

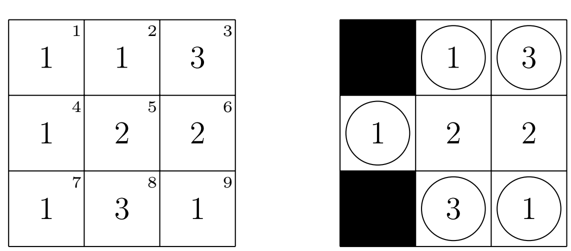


Figure 29: A partially solved Hitori.

As only two cells are undetermined, there are only four possible configurations. However, $\{x_5, x_6\}$ and $\{\neg x_5, \neg x_6\}$ both violate clauses discussed in Section 6.1, where two cells with the same value are selected and two adjacent black cells occur, respectively. This leaves two possible configurations $\{x_5, \neg x_6\}$ and $\{\neg x_5, x_6\}$.

First, consider $\{\neg x_5, x_6\}$. Considering that there must exist exactly one ground cell, one will get the clauses

$$(d_{2,0} \vee d_{d,0} \vee d_{4,0} \vee d_{6,0} \vee d_{8,0} \vee d_{9,0}) \wedge$$
$$(\neg d_{2,0} \vee \neg d_{3,0}) \wedge (\neg d_{2,0} \vee d_{4,0}) \wedge \ldots \wedge (\neg d_{2,0} \vee d_{9,0}) \wedge (\neg d_{3,0} \vee d_{4,0}) \wedge \ldots \wedge (\neg d_{8,0} \vee d_{9,0})$$

Black cells x_1, x_5 and x_7 cannot reach the ground cell, so

$$(x_1 \vee \neg d_{1,0}) \wedge (x_1 \vee \neg d_{1,1}) \wedge \ldots \wedge (x_1 \vee \neg d_{1,6}) \wedge$$
$$(x_5 \vee \neg d_{5,0}) \wedge (x_5 \vee \neg d_{5,1}) \wedge \ldots \wedge (x_5 \vee \neg d_{5,6}) \wedge$$
$$(x_7 \vee \neg d_{7,0}) \wedge (x_7 \vee \neg d_{7,1}) \wedge \ldots \wedge (x_7 \vee \neg d_{7,6})$$

For a valid configuration of d, where for all white cells, with the exception of the ground cell, it must hold that

$$\forall i, k \geq 1 : \neg d_{i,k} \vee (d_{i_\uparrow,k-1} \vee d_{i_\downarrow,k-1} \vee d_{i_\rightarrow,k-1} \vee d_{i_\leftarrow,k-1}) \tag{1}$$

Consider the white cell x_4. If x_4 is not the ground cell, then $d_{4,k}$ must have a neighbor with a distance $k-1$ from the ground cell. However, x_4 has no white neighbors, so Clause 1 cannot be satisfied. Now suppose x_4 is the ground cell, then every other cell must have a neighbor with the value $k-1$. However, this cannot occur as there is no way to fill the grid with values for k such that every cell has a neighbor with the value $k-1$. Thus, if x_4 is the ground cell, one cannot satisfy Clause 1. So regardless of whether x_4 is the ground cell or not, the configuration $\{\neg x_5, x_6\}$ is not satisfiable.

Figure 30 shows configurations of d, where each cell i contains the value k for which $d_{i,k}$ is true. The two figures show paths where x_4 is not connected to the other white cells, as it is isolated by its neighbors and a border. Figure 30a shows a path where x_2, and thus not x_4, is the ground cell. Regardless of the value of k for which the Boolean $d_{4,k}$ is true, it will not have a neighbor where $k-1$ occurs. If x_4 is the ground cell, see Figure 30b, then there will always be another white cell that has no neighboring cell with a k that is exactly one smaller. For Figure 30b, x_2 with $d_{2,1}$ has no neighbor with an occurrence of $k-1$.

(a) There is no path between the ground cell x_2 and x_4.

(b) There is no path between the ground cell x_4 and the other white cells.

Figure 30: Paths based on distance matrix d for $\{\neg x_5, x_6\}$ where the white cell x_4 is considered as the ground cell and not as ground cell. Both are unsatisfiable.

Now, consider $\{x_5, \neg x_6\}$. Figure 31 shows two possible paths derived from a configuration of the distance matrix d. If x_4 is considered to be the ground cell, a possible configuration for d could be created in such a way that satisfies the described formula, see Figure 31b. Figure 31a shows a satisfiable configuration where x_4 is not the ground cell. The two configurations in Figure 31 show paths where several cells have an identical distance towards the ground cell, and are considered to be *branches*. Branching such a path keeps the path connected and is dependent on which cell is chosen as ground cell. A branched path still satisfies the constraints and thus may occur. Since there exists a satisfiable configuration for d and all x_i, where $i \in \{\mathcal{W}, \mathcal{B}\}$, the partially solved HITORI is thus satisfiable, where one has found $\{x_5, \neg x_6\}$. Of course, one only needs a single satisfiable configuration of d to show that the HITORI is satisfiable.

(a) The ground cell x_2 is connected to all other white cells.

(b) The ground cell x_4 is connected to all other white cells.

Figure 31: Several paths based on distance matrix d for $\{x_5, \neg x_6\}$ are satisfiable, regardless of which cell is considered to be the ground cell. The path may consist of branches which have the same step distance from the ground cell.

6.3 MINISAT

Sörensson and Eén have created an open-sourced SAT solver, named MINISAT [9], which can be integrated with other frameworks [15]. The SAT solver can easily be optimized using the several parameters accompanying the solver. For the 2-SAT approach, MINISAT is used for solving lines to obtain either an unsolvable, partially solvable or fully solvable line. After that, one must check whether the uniquely occurring cells were indeed unique when incorporating the (partial) solution into the grid of the puzzle. One can do so, by checking the orthogonal line of the uniquely marked cells.

While MINISAT has been used for the 2-SAT based solver, combining it with the approach of capturing connectivity would solve a HITORI more efficiently as it only needs a CNF formula to fully solve the puzzle, rather than additional checks.

7 Generation

Unlike generating Sudoku, which are essentially Latin squares, generating HITORI has some additional difficulties due to the connectivity constraint of the puzzle. A HITORI may have several duplicates on a given line, as long as the solution will ultimately have at most one unshaded, i.e., white, occurrence of such a character. As repetition of characters is allowed, and not every character has to appear on each line, the number of possible HITORI increases exponentially. When duplicates occur, one needs to shade, i.e., make black, cells in order to get a proper solution. There exist HITORI for which no cells will be made black, i.e., boards representing Latin squares where all cells are unique in both their row and column. Therefore, one can say that the lower bound of the number of black cells $b(m, n)$ on an $m \times n$ HITORI is 0. In this section, we will go into an upper bound of $b(m, n)$, as well as how this contributes to the complexity of HITORI. First, we will consider generating uniquely solvable HITORI.

7.1 Generating HITORI

To generate all possible HITORI of a given size, we could exhaustively run all board combinations of a given set of characters. However, doing so is computationally hard. For example, a 4×4 board with the character set $N = \{1, 2, 3, 4\}$, would result in $4^{16} = 4,294,967,296$ different board combinations. Most of these boards will not be solvable due to connectivity issues, e.g., a board solely consisting of 1s. Computing all boards exhaustively becomes rather tedious, as the number of possible boards grows exponentially. In this section, we will explore ways to reduce the number of potential HITORI.

Suzuki et al. [14] introduced the *Hitori number* $h(n)$ as the minimum number of characters used on an $n \times n$ board. The provided lower bound is

$$h(n) \geq \left\lceil \frac{2n - 1}{3} \right\rceil$$

For $n = 4$, the minimal number of characters for a uniquely solvable HITORI equals 3, meaning that boards consisting of one or two characters should not be considered when searching for uniquely solvable HITORI. There exist 4 sequences that solely consist of one character. Consider $n = 4$ for 4×4 HITORI, then there are 2^{16} sequences that consist of 2 characters where we have $\binom{3}{2} = 6$ possible ways to select those 2 characters. In total, $4 + 2^{16} \cdot 6 = 393,220$ sequences can be eliminated prematurely. For 5×5, $h(n) = 3$, resulting in $5 + 2^{25} \cdot 10 = 335,544,325$ not uniquely solvable HITORI. However, pruning these boards does not significantly impact the number of potential HITORI.

A solvable HITORI should not have any duplicates in any given row or column, which is similar to the idea of a Latin square. Formally, a Latin square of size $n \times n$ is a square where all n characters occur exactly once in each row and column. Replacing black squares of a solved HITORI with other elements of the character set can lead to a Latin square. While numerous solutions of HITORI can be reduced to a Latin square of that size, this does not cover all possible HITORI. Figure 1 shows an example of a solvable HITORI for which the solution is not reducible to a Latin square. Using Latin squares will therefore not cover all possible HITORI.

Instead of generating all possibilities exhaustively, it might be more beneficial to consider the equivalence classes of HITORI, as these cover all possible HITORI. Only one of each equivalent class needs to be generated and solved in order to determine whether all puzzles in that equivalence class are solvable. *Bell numbers* provide the number of possible partitions of a given set of length n. To obtain the Bell sequences, consider that each element of a sequence is at most one larger than the until then found maximum. Suppose we have a set $\{1, 2, 3\}$, with a sequence length $n = 3$. Then the following sequences will cover all possible partitions: $(1\ 1\ 1), (1\ 1\ 2), (1\ 2\ 1), (1\ 2\ 2), (1\ 2\ 3)$. We can create similar sequences by substituting elements. These 5 sequences represent 5 equivalence classes that cover all $3^3 = 27$ different sequences. The principle of the Bell sequences can be used to generate all possible HITORI forms. For a 3×3 Hitori, the sequences will be of length $3^2 = 9$. The sequences 1 1 1 1 1 1 1 1 2 and 1 1 1 1 1 1 1 1 3 represent similar puzzles, and fall into the same equivalence class. If we can solve 1 1 1 1 1 1 1 1 2, then 1 1 1 1 1 1 1 1 3 can be solved as well, making it unnecessary to generate both. Therefore, only one of each equivalence class is generated and then solved.

Bell numbers work for sequences of length $|N|$ over a set of N elements. The principle of Bell numbers, however, works more generally. *Stirling numbers of the second kind* provide the number of possibilities for a sequence of length ℓ for a set of $|N|$ elements, where $\ell \leq |N|$. For $n \times n$ HITORI boards, we have a sequence of length $\ell = n^2$ that will use n elements, meaning that $\ell > n$. Stirling numbers of the second kind will be 0 in that case. Generally, the number of possible sequences of length ℓ with a set of $c = |N|$ elements can be obtained by using the following formula [8]:

$$a(c, \ell) = \frac{c^\ell}{c!} + \sum_{k=1}^{c-2} \frac{k^\ell}{k!} \sum_{j=2}^{c-k} \frac{(-1)^j}{j!}$$

For $m \times n$ HITORI boards, we have a sequence of length $\ell = m \cdot n$. These extended Bell numbers grow very rapidly. However, it will outweigh exhaustively computing all possible sequences for an $m \times n$ Hitori:

$$a(c, \ell) = \frac{c^\ell}{c!} + \sum_{k=1}^{c-2} \frac{k^\ell}{k!} \sum_{j=2}^{c-k} \frac{(-1)^j}{j!} < c^{mn}$$

Table 14 shows the difference between exhaustively generating $m \times n$ HITORI and using $a(c, \ell)$. Note that the set size of N for $a(c, \ell)$ is equivalent to m. Increasing the size, increases the difference between all possible boards and $a(c, \ell)$. For 3×3, $a(c, \ell)$ only considers 16.7% of all possible boards, which reduces to 0.14% for 6×6 boards. While $a(c, \ell)$ only considers a slight percentage of the total number of possible sequences, the number of sequences remains prominent.

Board size	Exhaustive	$a(c,\ell)$
3×3	$3^9 = 19,683$	$3,281$
4×4	$4^{16} \approx 4.295 \cdot 10^9$	$178,973,355$ $(\approx 1.79 \cdot 10^8)$
4×5	$4^{20} \approx 1.100 \cdot 10^{12}$	$45,813,246,635$ $(\approx 4.58 \cdot 10^{10})$
5×4	$5^{20} \approx 9.537 \cdot 10^{13}$	$795,019,337,135$ $(\approx 7.95 \cdot 10^{11})$
5×5	$5^{25} \approx 2.98 \cdot 10^{17}$	$2,483,597,478,617,802$ $(\approx 2.48 \cdot 10^{15})$
5×6	$5^{30} \approx 9.31 \cdot 10^{20}$	$7,761,038,612,902,285,822$ $(\approx 7.76 \cdot 10^{18})$
6×6	$6^{36} \approx 1.031 \cdot 10^{28}$	$14,325,688,388,877,185,110,915,943$ $(\approx 1.43 \cdot 10^{25})$

Table 14: The number of puzzles for different board sizes.

7.2 Pruning unsolvable HITORI

There are several single-line subsequences that are guaranteed to result in an unsolvable Hitori. Based on the rule Sim-U3, see Section 3.1, a HITORI containing an occurrence of 1 1 1 1 on a line is unsolvable. Other single-line subsequences that cause an unsolvable HITORI are {2 1 1 1 2}, {2 1 2 1 2}, {1 2 1 2 2}, {1 2 2 1 1}, {1 1 1 2 2 2}, {1 2 3 2 3 1}. Figure 32 shows general sequences, where $a, b, c, d \in N$, that have no valid solution and are thus cause unsolvable HITORI. These sequences are based on the rules in Section 3.1 as well. Note that a, b, c and d may be identical.

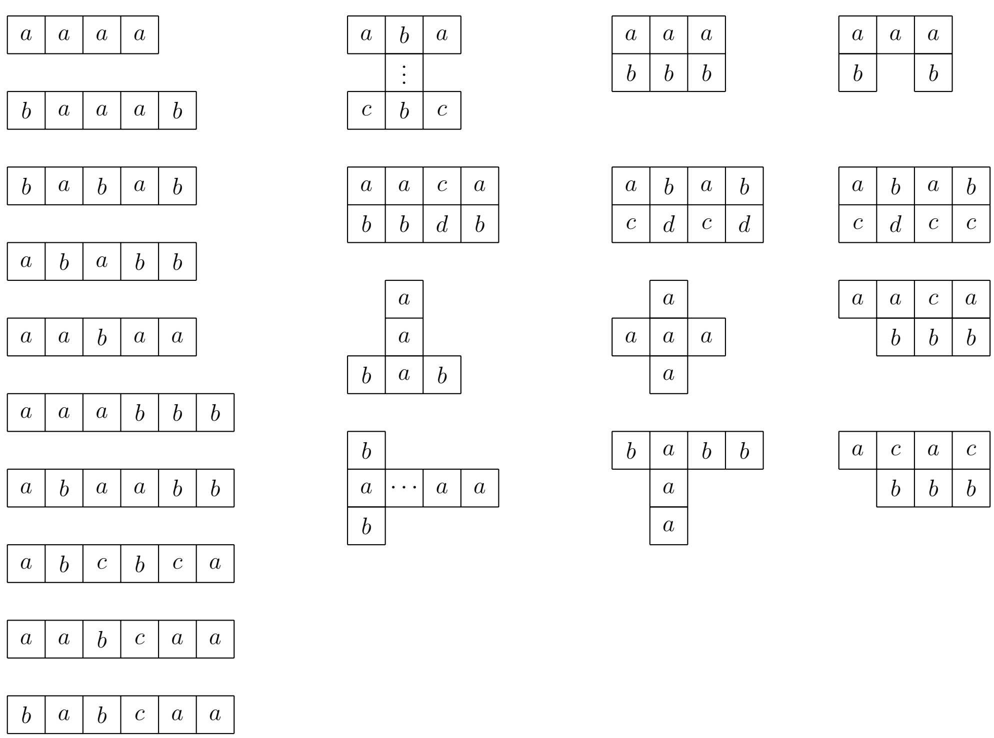


Figure 32: Patterns that cause unsolvable HITORI; $a, b, c, d \in N$.

When generating all $m \times n$ HITORI, one can eliminate potential HITORI that have any occurrence of these so-called *unsolvable subsequences*. Considering the sequences of $a(c, \ell)$, one can reduce the number of potential HITORI. Table 15 shows how the number of potential HITORI left to evaluate after pruning compared to an unpruned $a(c, \ell)$. The patterns used to prune are those shown in Figure 32. Note that this includes rotation and mirroring of the patterns.

$\|N\|$	$m \times n$	Pruned $a(c, \ell)$	$a(c, \ell)$	%
3	3×2	112	122	91.8
3	3×3	2,776	3,281	84.6
3	3×4	40,007	88,574	45.2
3	3×5	588,810	2,391,485	24.6
3	3×6	9,043,984	64,570,082	14.0
3	3×7	111,076,162	1,743,392,201	6.4
3	3×8	2,114,822,448	47,071,589,414	4.5
4	4×2	2,512	2,795	89.9
4	4×3	489,697	700,075	70.0
4	4×4	85,097,188	178,973,355	47.6
4	4×5	12,645,780,645	45,813,246,635	27.6
4	4×6	1,924,883,302,791	11,728,128,223,915	16.4
5	5×2	73,463	86,472	85.0
5	5×3	173,959,355	255,514,355	68.1
5	5×4	385,036,717,713	795,019,337,135	48.4

Table 15: Pruning sequences that contain unsolvable patterns reduces the number of potential HITORI.

Of course, other patterns may occur as well. One can include a border or corner to enclose some cells. Consider a 2×2 corner where each cell contains the same character. If any of the directly neighboring cells of this 2×2 corner, i.e., horizontal and vertical neighboring cells, contains the same character, there is no solution and thus not a uniquely solvable HITORI. Of course, these corner and border cases will not capture as many sequences as the patterns shown in Figure 32.

The remaining sequences can be evaluated to determine whether they are uniquely solvable HITORI. While these patterns do not eliminate all unsolvable sequences, the number of to be evaluated sequences reduces drastically when increasing the size of the board.

7.3 Finding the maximum number of black cells

While wanting to keep the number of black cells minimal, the upper bound on the number of black cells $b(m, n)$ provides us with some knowledge on the complexity of a HITORI. Considering an $m \times n$ grid, the aim is to find the maximum number of black cells so that the original rules of HITORI are not violated.

Figure 33 provides an example of a 6×4 board that has a maximum number of black cells. By shifting the bottom row by 1, we can create another board that has a maximum number of black cells while not violating the original rules of HITORI.

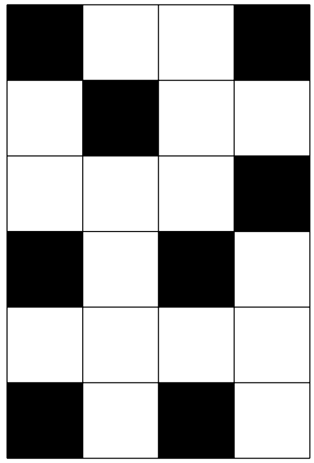


Figure 33: A 6×4 HITORI board with a maximum number of black cells.

One can see a pattern in constructing boards with a maximum number of black cells for $n = 8$, see Figure 34. The first three rows are constructed in such a manner that the left and right enclosed patterns are reachable. The pattern shown in red considers three rows, where the first and third row have $\frac{n}{2} - 1$ black cells, and the second has 2 black cells.

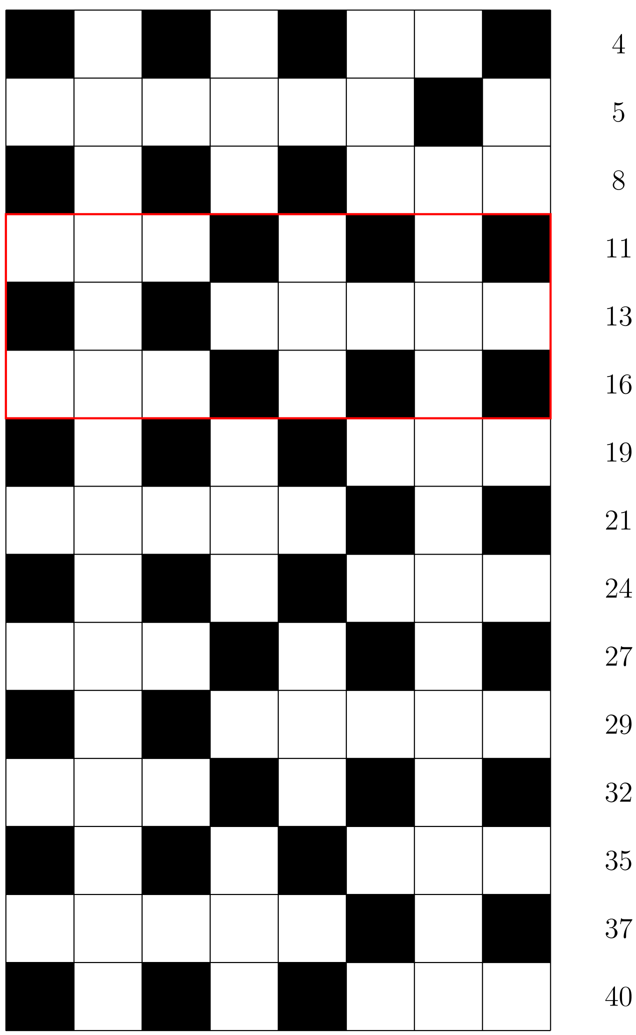


Figure 34: For $m \geq 6, n = 8$, this structure gives the maximum number of black cells.

Consider an $m \times n$ board, for which cells alternate between black and white. This chess-like board solely considers the constraint of no neighboring black cells, where the maximum number of black cells equals $\left\lceil \frac{1}{2}mn \right\rceil$. Once the connectivity constraint is included, the maximum number of black cells decreases. Table 16 shows the found maximum of black cells for an $m \times n$ board, with $m, n \in \{2, 3, \ldots, 13, 14\}$. The maxima were found using an exhaustive approach. For m or n divisible by 3, the number of black cells $b(m, n)$ will be exactly $\frac{1}{3}mn$. For other m and n, the number of black cells might deviate by one. A Hitori of size 7×7 will have a maximum of 17 black cells, while $\frac{1}{3} \cdot 7 \cdot 7 = 16\frac{1}{3}$. A line, i.e., row or column, of length 7 contains at most 4 black cells. In such cases, rounding up will provide the found maximum. Otherwise, one may round to the nearest integer, denoted as $[x]$ for the value x. Generally, for $m \times n$ Hitori, where $m \geq n \geq 6$, the number of black cells can be determined by the following formula:

$$b(m, n) = \begin{cases} \left[\frac{1}{3}mn\right], & \text{if } n \text{ is even} \\ \left\lceil \frac{1}{3}mn \right\rceil, & \text{otherwise} \end{cases}$$

Since every black cell must have two white neighbors on a given line, horizontally or vertically, one can repeat such a pattern at least per three lines. The upper bound for the maximum number of black cells $b(m, n)$ for an $m \times n$ Hitori is given by $b(m, n) \leq \frac{1}{3}mn + 1$.

One might observe that the found maximum for the square boards represent the Heyawake numbers for $n \times n$ boards [5]. As mentioned in Section 2, the Heyawake puzzle considers black and white cells, for which black cells may not be neighbors and white cells must be connected. Note that $b(m, n)$ can be applied to $m \times n$ sized Heyawake as well.

$m \cdot n$	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1												
3	2	4											
4	2	4	5										
5	3	6	7	9									
6	3	6	8	10	12								
7	4	8	9	12	14	17							
8	4	8	10	14	16	19	21						
9	5	10	12	15	18	21	24	27					
10	5	10	13	17	20	24	27	30	33				
11	6	12	14	19	22	26	29	33	37	41			
12	6	12	15	20	24	28	32	36	40	44	48		
13	7	14	17	22	26	31	35	39	43	48	52	57	
14	7	14	18	23	28	33	37	42	47	52	56	61	65

Table 16: Maximum number of black cells for $m \times n$ Hitori.

7.3.1 White squares

A solvable $m \times n$ Hitori needs to have a connected area of white cells, which makes determining the maximum number of black cells NP-complete. Since solving a Hitori is NP-complete, finding the maximum numbers of black cells is also NP-complete, as it inherits the properties of the puzzle. When exhaustively computing the maximum number of black cells, one can ignore any partial board solution that will not surpass the so-far found maximum. This reduces the number of possible board solutions in search for the maximum number of black cells. Boards that will not surpass the so-far found maximum include those with an equivalent number of black cells, meaning that there is no need to compute these boards. An example of such equivalent boards are those containing a 2×2 white square. When a board contains a 2×2 white square, one may always move an adjacent black cell into the 2×2 white square. This makes it useless to compute such boards when trying to obtain the maximum number of black cells, as they will not surpass the so-far found maximum.

Suppose we have a solution containing a 2×2 white square. We assume that this solution board satisfies that no black neighbors occur, as well as maintaining a continuous path consisting of all white cells. If a 2×2 white square occurs, we can always move an orthogonally or diagonally adjacent black cell into this square as long as it is not maximally enclosed by black cells. Such a white square is maximally enclosed if it is surrounded by six black cells, see Figure 35. Figure 35 shows that there exist a solution for which we cannot swap any adjacent black cell with a cell of the white square, as this would either lead to two adjacent black cells or a disconnected white area.

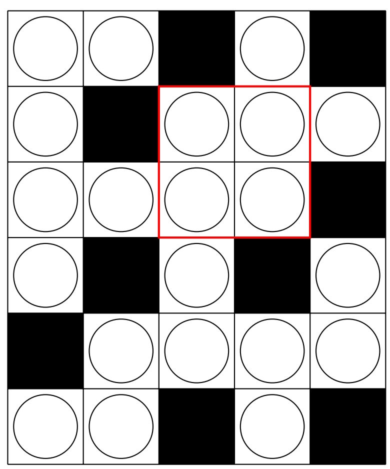


Figure 35: A maximally enclosed 2×2 square occurs on a Hitori solution board.

Instead of maximally enclosing the 2×2 white square, consider a solution that contains $b(m,n) - 1$ black cells. This gives us two possible structures, see Figure 36.

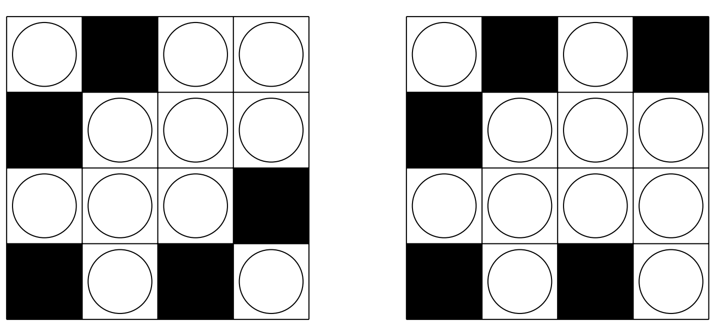


Figure 36: Segments of Hitori solution boards where a 2×2 square occurs.

Consider a graph $\mathcal{G} = (V, E)$ that represents the solution, where V consists of all white cells and E are undirected edges between direct, i.e., horizontally and vertically adjacent cells, neighbors. The boards shown in Figure 36 can then be constructed as a graph, see Figure 37. When a 2×2 white square occurs, it is possible to replace one of the cells in this given 2×2 white square by a black cell. Doing so, will remove the corresponding edges. However, there are some restrictions of when to swap those cells.

(1) If a white cell has two black neighbors, it is not possible to swap the given white cell with any black cells, as this violates that no black cell may have a black neighbor.

(2) If a white cell has three adjacent diagonals that are black, it is not possible to swap the given white cell with any black cells, as this potentially cuts off some white cells.

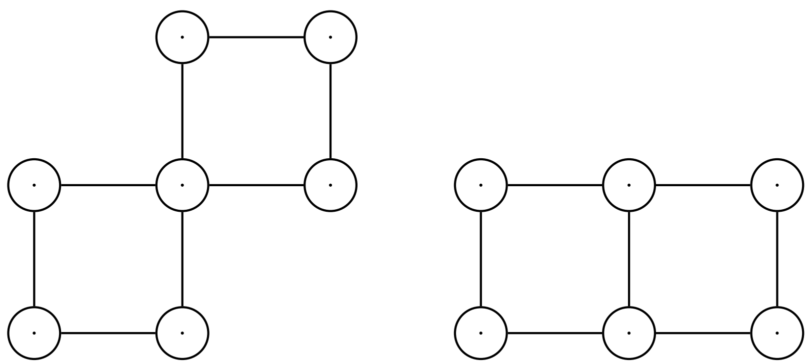


Figure 37: Patterns in the graphs of segments of HITORI.

When a single 2×2 white square is not maximally enclosed, we will have at least two overlapping 2×2 white squares. Surrounding the white squares with black cells, we can minimize the number of ways to access the white square. When the 2×2 white squares overlap with 2 cells, the maximum is 7 orthogonally or diagonally adjacent black squares. For 2×2 white squares overlap with 1 cell, there are 8 surrounding black squares at most.

For the pattern shown on the left in Figure 37, we can ensure that a diagonally adjacent black cell can always be swapped with the center white cell. This cell has four white neighbors, meaning that restriction (2) cannot occur. Since the bottom left cell is white, we also ensure that (1) cannot occur. Therefore, we can always swap a diagonally adjacent black cell with the center white cell.

For the pattern shown on the right, we can ensure that for the center cells (1) cannot occur, as the center cells have two diagonally adjacent white cells. Since both center cells have at most two diagonally adjacent black cells, as two diagonally adjacent cells are already set white, (2) cannot occur as well. Therefore, we can always swap an adjacent black cell with the center white cell. So, even if these situations are fully enclosed by black cells, we can always move a black cell into a 2×2 white square.

Therefore, if a found solution contains a 2×2 white square that is not maximally enclosed by black cells, we can always move an adjacent black cell into the white square. If there exists a solution with a white square that is surrounded by the maximum number of black cells, there is no need to compute boards for which we can move an adjacent black cell into the white square. Those instances can be pruned.

8 Experiments

In order to determine the number of Hitori that are uniquely solvable for a given size, we will consider the two solvers introduced in Sections 5.1 and 6.1. First, the performance of the solvers is measured against the Menneske dataset [2]. The Hitori in the dataset are assumed to be uniquely solvable. The classification for the dataset as well as for a given size will focus on the class rather than the needed steps.

8.1 Classifying Menneske

Considering the Menneske dataset, both the rule set and SAT solver are able to solve the dataset completely, with the exception of three Hitori present in the dataset that are not uniquely solvable. Both solvers are able to solve the part of the puzzles that are uniquely solvable. Of the $376,211$ puzzles present in the dataset, both solvers are able to solve $376,208$. The three not uniquely solvable Hitori can be found in Figures 38, 39 and 40. Here, the puzzles have been solved until the point no definitive progress can be made, the remaining cells are thus grey. The 8×8 Hitori shown in Figure 38 has three possible solutions. The 4s and 6s are dependent on one another and act as a switching component, resulting in two ways to solve the 4s and 6s. The 1s do not influence the connectivity of the puzzle and making either one of the two black may result in a possible solution. The Hitori shown in Figure 39 has three solutions. If the top 3 is white, the remaining 3 in the same column becomes black, and due to connectivity the last remaining 3 must be white as well. If the leftmost 3 becomes white, the remaining 3 in the same row must be black, the top 3 can then become either black or white. As the probing rules are based on contradictions, and no such contradiction occurs, all three solutions are considered viable. The last unsolved puzzle has four different solutions. When the top left 1 is black, the adjacent 1 becomes white. This, in turn, makes the bottom 1 black, and sets the two 2s as well. When the top left 1 is white and the adjacent 1 thus black, the bottom 1 may be black or white. If made black, the two 2s are determined as well. If made white, both 2s may be made white.

8	4	6	3	2	7	6	2
2	7	5	1	6	3	8	1
5	4	6	7	8	5	1	4
7	2	7	8	6	6	3	5
7	2	2	5	3	6	4	8
6	8	3	2	4	1	5	2
3	7	4	4	2	8	7	6
3	6	1	5	4	4	4	7

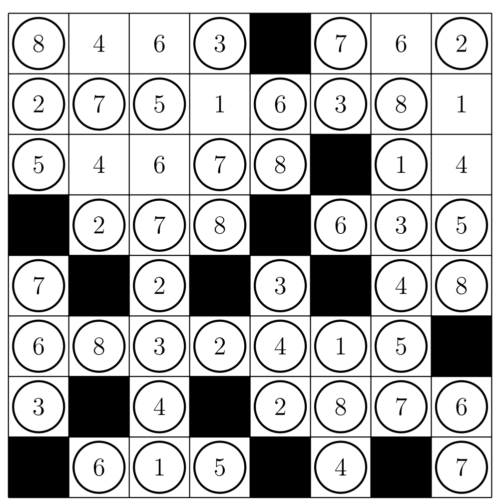


Figure 38: An 8×8 Hitori from the Menneske data set (22066).

1	2	4	5	6	9	8	1	4
2	3	5	1	9	7	1	8	3
8	7	9	4	8	2	6	8	5
7	9	6	3	1	8	7	4	2
3	2	8	9	2	2	7	5	1
6	4	3	8	4	7	2	6	1
4	1	5	6	3	5	4	2	4
5	3	1	3	7	7	8	9	4
6	8	2	1	6	6	4	2	9

	2	4	5	6	9		1	
2		5		9		1	8	3
8	7	9	4		2	6		5
7	9	6	3	1	8		4	2
3		8	9	2		7	5	
	4	3	8		7	2	6	1
4	1		6	3	5		2	
5	3	1	3	7		8	9	4
	8	2	1		6	4		9

Figure 39: An 9 × 9 Hitori from the Menneske data set (36545).

4	4	5	7	7	6	8	8	2
4	1	1	8	8	7	1	3	5
3	9	8	5	4	1	7	2	1
6	5	3	9	8	2	4	5	8
5	4	4	6	6	3	3	8	9
4	6	7	1	8	5	9	4	5
8	4	3	4	5	4	3	6	1
2	3	9	6	8	4	5	1	8
2	7	1	2	2	8	6	5	4

	4	5		7	6	8		2
4	1	1	8		7		3	5
3	9	8	5	4	1	7	2	
6	5		9		2	4		8
5		4		6	3		8	9
	6	7	1		5	9	4	
8		3		5	4		6	1
2	3	9	6	8		5	1	
	7	1	2	2	8	6	5	4

Figure 40: An 9 × 9 Hitori from the Menneske data set (169599).

Tables 17 and 18 show the classification of the Menneske dataset using both the rule-based and SAT-based solvers. The 2-SAT approach, which focuses on individual lines, similar to the rules in category B of the rule-based approach. When classifying Hitori one can treat them both as category B. The dataset does not contain any Latin squares, resulting in 0 puzzles classified as a category A puzzle. Overall, the solvers perform quite similarly. They are able to solve the same number of puzzles. However, there are some slight differences in classification. When using SAT, one might obtain a cell's color, i.e., black, white or grey, earlier on in the solving process. The rule-based solver will capture these in a later solving stage. Note that the rules of enclosing and alternating permutations will not help to capture these differences as $|AB| \geq 3$. Section 5.3 revealed no obvious patterns that could lead to a (partial) solution of a given line and showed that the number of sequences with such a solution grows almost exponentially. This makes it hard to create a rule set that is able to fully enclose the difference between the two solvers.

Category	5×5	6×6	8×8	9×9	12×12	15×15	17×17	20×20	
A	0	0	0	0	0	0	0	0	0
B	10,375	8,477	8,091	50,423	3,419	148	46	0	80,979
C	384	374	244	840	22	0	0	0	1,864
D	3,530	1,841	856	3,902	145	4	1	0	10,279
E	187	120	41	200	6	0	0	0	554
F	16,736	18,629	26,281	184,385	27,412	3,568	1,785	5	278,801
G	398	461	458	2,196	127	87	4	0	3,731
	31,610	29,902	35,971	241,946	31,131	3,807	1,836	5	376,208

Table 17: The classification of the Menneske dataset using the rule-based solver.

Category	5×5	6×6	8×8	9×9	12×12	15×15	17×17	20×20	
A	0	0	0	0	0	0	0	0	0
B	10,437	8,563	8,210	50,814	3,425	148	46	0	81,643
C	335	304	140	484	19	0	0	0	1,282
D	3,519	1,831	849	3,891	145	4	1	0	10,240
E	187	120	41	200	6	0	0	0	554
F	16,734	18,623	26,290	184,474	27,417	3,630	1,785	5	278,958
G	398	461	441	2,083	119	25	4	0	3,531
	31,610	29,902	35,971	241,946	31,131	3,807	1,836	5	376,208

Table 18: The classification of the Menneske dataset using the SAT-based solver.

8.2 Classifying Hitori

After generating and pruning the $a(c, \ell)$ sequences for $3 \times 3, 4 \times 4$, one can apply the two solvers to obtain the uniquely solvable Hitori. Considering the classification made in Section 5, one can classify these Hitori as seen in Table 19.

For 3×3 Hitori, two are classified as trivial. As mentioned in Section 5.1, those classified as trivial will only convey Latin squares. There are, in total, twelve 3×3 Latin squares. However, since $a(c, \ell)$ only considers sequences that start with a 1, eight of the Latin squares can be ignored. Of the four Latin squares that start with a 1, two sequences are isomorphic towards the remaining two, meaning that only two sequences should be classified as trivial. Another observation that can be made regarding 3×3, is that none of the puzzles are classified as difficulty C or E. Class C consists of the rules DP and SbP. For SbP an instance of 1 1 1 can occur. However, this is already captured by using a lower-class rule; Sim-U3. The rule DP makes two cells white. Recall Figure 18, where an instance of 1 3 1 captures one of the white cells using the Simple rule; Sim-M3. Due to connectivity, the 2 that is part of the switching component must be white. The remaining 2 will then be black, making the adjacent 1 white. The remaining 1 will then be black, making the second to be made white cell actually white. The rule of class E considers at least 4 different lines, which cannot occur on 3×3.

For 3×3, 4×4 and 4×3, the majority of the puzzles lie within the Hard category, which considers the connectivity rules. One of the difficulties faced by Tran [16] was that of connectivity on a puzzle-level. The connectivity rules capture the majority of the HITORI, which implies one needs an approach to capture the connectivity of the puzzle. Other classes that have a significant number of HITORI are class B and D. While one might expect simple HITORI to easily be solved by using *simple* rules, those solved by class D are those that use simply considering corner cases are quite significant. Of the pruned sequences, there is only a minor part that is in fact uniquely solvable. Of the equivalent classes, only 12.3% of the 3×3 HITORI are solvable, and only 1.6% of the 4×4 HITORI. For the 4×3 HITORI, where a set of four elements is used, a total of 8.4% of the equivalent classes is uniquely solvable.

Difficulty	3×3 (3)	4×4 (4)	4×3 (4)
A (trivial)	2	24	24
B (super easy)	92	637,927	13,989
C (pretty easy)	0	22,222	236
D (easy)	88	550,272	10,970
E (medium)	0	63,383	1,024
F (hard)	179	1,456,155	29,988
G (very hard)	43	114,148	2,711
Solvable	404	2,844,131	58,942
Total	3,281	178,973,355	700,075

Table 19: The classification of all equivalent classes of $m \times n$ HITORI for both rule-based and SAT-based solver.

For 5×3 HITORI, where one may consider a set of five elements, there is a difference in performance for both solvers, see Table 20. Although there are no significant differences, the SAT-based solver is able to classify more puzzles earlier on in the classification process. Both solvers consider 5.4% of the equivalent classes to be solvable. Consider the equivalent classes of 3×5 HITORI, where the set of characters consists of four elements. Again, there is a slight difference in classification, although there is no significant difference. Both solvers yield only 1.3% of the equivalent classes are uniquely solvable.

Difficulty	Rules 5×3 (5)	SAT 5×3 (5)	Rules 3×5 (4)	SAT 3×5 (4)
A (trivial)	552	552	0	0
B (super easy)	2,885,157	2,893,161	114,985	115,213
C (pretty easy)	91,034	88,808	4,098	4,006
D (easy)	1,612,287	1,605,431	63,105	62,873
E (medium)	83,721	83,395	4,037	4,038
F (hard)	8,271,443	8,278,295	361,484	361,502
G (very hard)	954,665	949,217	48,642	48,719
Solvable	13,898,859	13,898,859	596,351	596,351
Total	255,514,355	255,514,355	44,747,435	44,747,435

Table 20: The classification of all equivalent classes of $m \times n$ HITORI for the rule-based and SAT-based solver.

9 Conclusion and Future Research

The Japanese puzzle HITORI is a one-player game that considers an $m \times n$ grid with a set of N characters. One can shade and unshade cells, i.e., make cells black and white, respectively. The goal is to eliminate all duplicate characters by making them black in such a way that no black cells are horizontally or vertically adjacent and all white cells remain reachable from every other white cell.

Solving HITORI can be done by using a set of rules. Rules based on the basics of the puzzle are considered *trivial*. Single-line rules are part of the *simple* rules, while those that consider multiple lines are part of the *advanced* rules. Rules based on the connectivity of the puzzle fall under the *connectivity* rules. When a HITORI cannot fully be solved using rules from these categories, one can use *probing* to try and solve the puzzle entirely.

The two solvers, one fully rule-based and one 2-SAT-based, showed slight differences in classifying the Menneske dataset, but classified the generated HITORI of size 3×3, 4×4 and 4×3 identically. Other-sized boards, such as 5×3 with a set of five characters, were classified similarly. One can use pruning to eliminate unsolvable equivalent classes of HITORI prematurely. The numerous solvable HITORI have a varying number of black cells, for which we have seen an upper and lower bound. Different from the 2-SAT-based solver, using a distance matrix as described in Section 6.2 allows capturing the connectivity of the puzzle using CNF.

For future research, the provided rule-based solver showcases a gap in performance compared to the SAT-based solver. In an effort to close this gap, we considered the rules *enclosing permutations* and *alternating permutations*. One can examine whether using a suffix tree for these rules improves the rule-based solver on larger sized boards. Considering single lines of a HITORI containing a sequence A and a permutation A', an effort was made to derive the number of permutations that yield a specific number of solutions. While some observations were made about the number and type of permutations with a given number of solutions, no efficient manner of calculating the number of solutions could be provided. Further research could explore a more efficient approach. As the SAT-based solver considered satisfiability on individual lines of HITORI, one could explore a SAT-based approach that captures all lines and the connectivity constraint of a HITORI. The provided approach may be examined in an effort to explore its performance. To reduce the selection of potential HITORI, one could prune even more patterns. Additionally, different-shaped and other-sized boards could be evaluated and solved as well.

References

[1] Bengt Aspvall, Michael F. Plass, and Robert E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.

[2] Vegard Hanssen. Menneske.no. `https://menneske.no/hitori/eng/`. [Online; accessed 2023-10-30].

[3] Robert A. Hearn and Erik D. Demaine. *Games, puzzles, and computation*. CRC Press, 2009.

[4] OEIS Foundation Inc. Bell or exponential numbers. Entry A000110 in The On-Line Encyclopedia of Integer Sequences. `https://oeis.org/A000110`. [Online; accessed 2024-7-3].

[5] OEIS Foundation Inc. Heyawake numbers. Entry A239231 in The On-Line Encyclopedia of Integer Sequences. `https://oeis.org/A239231`. [Online; accessed 2024-3-4].

[6] OEIS Foundation Inc. Tetrahedral numbers. Entry A000292 in The On-Line Encyclopedia of Integer Sequences. `https://oeis.org/A000292`. [Online; accessed 2024-6-26].

[7] OEIS Foundation Inc. Triangular numbers. Entry A000217 in The On-Line Encyclopedia of Integer Sequences. `https://oeis.org/A000217`. [Online; accessed 2024-6-26].

[8] Nelma Moreira and Rogério Reis. On the density of languages representing finite set partitions. *Journal of Integer Sequences*, 8(05.2):8, 2005.

[9] Niklas Eén, Niklas Sörensson. MiniSAT. `http://minisat.se/`. Version 2.2.0, 2010-07-10.

[10] Nikoli. Hitori — Nikoli. `https://www.nikoli.co.jp/en/puzzles/hitori/`. [Online; accessed 2023-1-10].

[11] Léo Robert, Daiki Miyahara, Pascal Lafourcade, and Takaaki Mizuki. Card-based ZKP for connectivity: applications to Nurikabe, Hitori, and Heyawake. *New Generation Computing*, 40(1):149–171, 2022.

[12] Kenneth H. Rosen and Kamala Krithivasan. *Discrete mathematics and its applications*. Seventh edition, 2013.

[13] M. Sharir. A strong-connectivity algorithm and its applications in data flow analysis. *Computers & Mathematics with Applications*, 7(1):67–72, 1981.

[14] Akira Suzuki, Kei Uchizawa, and Takeaki Uno. Hitori number. In *Proceedings of Fun with Algorithms: 6th International Conference, FUN 2012*, pages 334–345. Springer, 2012.

[15] Niklas Sörensson and Niklas Eén. MiniSat v1.13 — A SAT solver with conflict-clause minimization. *SAT*, 2005(53):1–2, 2005.

[16] Andy Tran. Enhancing GNNs: An exploration of iterative solving and augmentation techniques. Bachelor Thesis. ETH Zürich. 2023.

[17] Gerhard van der Knijff. Solving and generating puzzles with a connectivity constraint. Bachelor Thesis. Radboud University. 2021.