Universiteit Leiden
The Netherlands

Opleiding Informatica

Solving Rullo,

A Mathematical Puzzle

Daniel van den Boogaard

Supervisors:
Walter Kosters & Luc Edixhoven

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

18/08/2023

Abstract

This thesis covers RULLO puzzles, which are mathematical puzzles where the player has to include or exclude numbers in a grid to make the sum of every line equal to the given line-sum. We decided on a way to determine the difficulty of RULLO puzzles which we use to examine puzzles. We focus mainly on simple puzzles where guessing is not required to solve them. But we also touch upon common "deadly patterns" found only in non-simple puzzles.

Contents

1 Introduction

RULLO is a mathematical puzzle where the goal is to eliminate numbers in the grid such that the numbers on the side of the playing fields equal the sum of all active number in that specific row or column. A more extensive explanation of the rules and an example game will be given in Section 1.1.

RULLO was developed in 2017 as an mobile app, it was created by Cresentyr [Cre]. They later republished the game under the name Meganum.

	10	26	16	10	20
23	8	9	8	5	6
13	7	8	3	5	9
17	8	1	8	5	3
8	2	8	5	2	6
21	2	8	4	5	8

	10	26	16	10	20
23	8	9	8	5	6
13	7	8	3	5	9
17	8	1	8	5	3
8	2	8	5	2	6
21	2	8	4	5	8

Figure 1: Example 5×5 puzzle with its solution.

1.1 The rules of Rullo

RULLO is played on a grid of height h and width w, where each square is filled with a number between 1 and 9. All numbers are given at the start of the puzzle, so a player does not have to fill in any numbers, unlike a puzzle like SUDOKU. On the boundary of the $m \times n$ grid there are numbers. These numbers represent sum of the corresponding rows or columns. However, at the start the sums do not necessarily correspond correctly with the numbers in their rows or columns. Going forward, we will call these sums the *target-sums* of a given line (i.e. row or column). As a player one has to eliminate numbers in the field such that the sum of each row and column correctly corresponds with their target-sum.

In this paper we colour code the squares to denote the state they are in. If the background of a square is white it means that the state is unknown or not yet checked. At the start of a game no line has been checked so all squares are white. If we know that a square should be excluded because there is no combination using this element possible to get to the target-sum, we mark this square as black. Finally, if we are certain a number should be included to get to the desired target-sum we mark it as green. If we know every square of a certain line we will also make the target-sum green to denote that the line is completed. See Figure 1 for an example.

1.2 Thesis overview

We explain the formal definitions and explain the rules through an example puzzle in Section 2. In Section 4 we cover some difficulties found within RULLO. Section 5 lays out the implementation used. Section 3 presents a discussion of related work. Section 6 provides a detailed explanation of the conducted experiments and illustrates the respective outcomes they have produced. Section 7 discusses some aspects of RULLO which make puzzles difficult. Section 8 presents the conclusions drawn from the experiments, and discusses future work. This bachelor thesis was supervised by Walter Kosters and Luc Edixhoven.

2 Playing the game

In Section 1.1 we informally have explained the rules of RULLO puzzles, in this section we give some formal definitions. We also go through an example game, where we show some tactics and thought processes which help when trying to solve RULLO puzzles manually.

2.1 Definitions

As previously touched upon, a grid in a game of RULLO is of size $h \times w$, where h is the height of the grid and also the number of rows. And w is the width of the grid and also the number of columns. Often it is the case that $h = w$, but this is not necessary.
Because rows and columns have no unique characteristic other than their orientations we will be using the the term *line* to denote them. Also, the sum of a row or column will be called the target-sum.
Furthermore we use a set for which numbers can be put in the grid. We will call this the *range* R. R is often a set of consecutive numbers, but this does not have to be the case. A good example would be the first 10 prime numbers, which will result in interesting puzzles.
If we know the range we can also give some limitations to the target-sum, which satisfies $0 \leq$ target-sum $\leq \max(R) \times \ell$. Here ℓ denotes the length of a line.

2.2 An example game

To give a better understanding of how to achieve the goal, an example game will be played from start to finish. We use the puzzle discussed in Section 1.1 which can be seen in Figure 2.

	10	26	16	10	20
23	8	9	8	5	6
13	7	8	3	5	9
17	8	1	8	5	3
8	2	8	5	2	6
21	2	8	4	5	8

Figure 2: Example 5×5 puzzle.

If we take a look at the first column we can see that the line-sum should be equal to 10. With the numbers $\{2, 2, 7, 8, 8\}$ we can only ever get to 10 by combining a 2 and 8 in any combination. This leaves the 7 with which it is impossible to make a combination that sums up to 10. Therefore we can cross out the 7 (depicted with black). For the columns 3 and 4 the thought process is the same to get to respectively 16 and 10. For column 2 and 5 one would have to approach it slightly differently, because if we take a closer look at the second column, we find that there is only one unique combination possible to create 26, which is $\{1, 8, 8, 9\}$. Since there are three 8s in this column we do not know which we need to pick. But the other two numbers 1 and 9 are required to get to 26, thus we can mark them as included (depicted with green). Looking at the fifth column we can see a small pattern, because if we want to get the line-sum 20 with any combination from the multiset $\{3, 6, 6, 8, 9\}$ we have two possibilities: we can either take $\{3, 8, 9\}$ or $\{6, 6, 8\}$. We can see that the 8 is present in both combinations, which means we can mark it as included. But we can also say something about the other four numbers: if we know one of the four we know the others as well.

	10	26	16	10	20
23	8	9	8	5	6
13	7	8	3	5	9
17	8	1	8	5	3
8	2	8	5	2	6
21	2	8	4	5	8

Figure 3: Board of the example puzzle after the first vertical pass.

After applying the steps to all columns we get the board state as shown in Figure 3. We can now continue with the rows and this should be easier because some squares are already known. For example, if we take a look at the first row we see that the 9 is already marked as included. This means that we are looking for combinations of the multiset $\{5, 6, 8, 8\}$ that sum to $23 - 9 = 14$. The only way to do this is $6 + 8$. Since there is only one 6 we can mark that as included. We cannot conclude anything about the two 8s. But we do know that 5 should be excluded. If we look at the second row we also have a square already marked, but here we know the 7 should be excluded. This means we need to find a combination in $\{3, 5, 8, 9\}$ that sums up to 13. There is only one way to do this, which is $5 + 8$. Thus we can exclude the 3 and the 9, which completes this particular row. In the third row we already have the 1 marked as included, so this means we need a combination of the multiset $\{3, 5, 8, 8\}$ which sums to $17 - 1 = 16$. There are multiple ways to reach this, so we cannot mark anything, what we can say with certainty is that if the 5 is required so is the 3, this works both ways. We also know that at least one of the 8s is required. The fourth row has another useful peculiarity, because the line-sum is even and there is only one odd number in the line, we can confidently say that the odd number, in this row the 5, should be excluded. Unfortunately in this row we still do not know which numbers to include since we can get to the line-sum in two different ways: either 8 or $2 + 6$. The last row has the exact opposite situation compared to the previous row. Because the target-sum is odd and there is only one odd number in the line we can mark it as included. This leaves us with a multiset $\{2, 8\}$ which needs to sum to $21 - 8 - 5 = 8$. The only option is to include the 8 and exclude the 2.

	10	26	16	10	20
23	8	9	8	5	6
13	7	8	3	5	9
17	8	1	8	5	3
8	2	8	5	2	6
21	2	8	4	5	8

Figure 4: Board of the example game after the first horizontal pass.

Now that we have done a vertical and horizontal pass we get to the game state as seen in Figure 4 and we can continue with another vertical pass. We have previously deduced that for the first column we need a 2 and 8 in any combination to get to the line-sum of 10. Since there is only one 2 left we can mark it as included, we cannot deduce anything about the 8s. In the second column we already have four numbers marked as included. These sum up to $1 + 8 + 8 + 9 = 26$, which is equal to the line-sum. Therefore the last unknown number can be marked as excluded. The situation in the third column is inverted compared to the second column. We already know that three numbers should be excluded, which leaves us with two 8s which add up to the line-sum. So we can mark both as included. In the fourth column we have two numbers marked as included and two marked as excluded. The sum of the two included numbers is $5 + 5 = 10$ which is equal to the line-sum 10. Thus the last unknown number should be excluded as well. Recall from the first pass that if we know one of the numbers $\{3, 6, 6, 9\}$ in the fifth column we can deduct the others. We know that the 6 is included so the other 6 should be included as well. We also know that the 9 is excluded so we should also exclude the 3.

	10	26	16	10	20
23	8	9	8	5	6
13	7	8	3	5	9
17	8	1	8	5	3
8	2	8	5	2	6
21	2	8	4	5	8

Figure 5: Board of the example game after the second vertical pass.

After doing this vertical pass we end up with the game state depicted in Figure 5 and we are left with only two unknown numbers. These can be solved by doing a final horizontal pass. For the first row the numbers marked as included add up to $6 + 8 + 9 = 23$ which equals the target-sum, thus we can mark the 8 as excluded. The third row currently sums to $1 + 8 = 9$ which means we need the other 8 to get to the line-sum of 17. Which means we mark it as included. After finishing up these two rows we get to the final solution which can be seen in Figure 6.

	10	26	16	10	20
23	8	9	8	5	6
13	7	8	3	5	9
17	8	1	8	5	3
8	2	8	5	2	6
21	2	8	4	5	8

Figure 6: Example 5×5 puzzle solved.

3 Related work

While there has not been any research into Rullo, as far as we could tell, there are many papers on puzzles that share similarities to Rullo. One of these puzzles are Nonograms, and they share a similar structure where one has to mark the field based on the numbers on the outside. As [BK12] describes,"Nonograms are a popular type of logic puzzles, where a pixel grid has to be filled with black and white pixels, based on a description that indicates the lengths of the consecutive black segments for each row and column". If one makes a tomography (for each line adding the given lengths) of a Nonogram one gets a Rullo puzzle if the range is $\{1\}$. But while the Nonogram might be considered simple and with a unique solution the Rullo puzzle is not necessarily. There are Nonograms which can be converted to Rullo puzzles with a unique solution, but this is the exception, not the rule. Some examples are Nonograms which are either completely filled or completely empty, as they will be exactly the same as Rullo puzzles. The implementation used to calculate difficulty in [BK12] is a similar approach to the one used for determining the difficulty for Rullo puzzles.

Another puzzle which is related to Rullo is Kakuro, also known as Cross Sum. Kakuro is a mathematical puzzle, designed by Nikoli [Nik], where, similar to Rullo, the sums are given outside of the playing field. However, there are two major differences between Kakuro and Rullo puzzles. Firstly in Kakuro none of the numbers in the playing field are given, the player has to fill in these numbers themselves. To make this possible another rule is used, because in Kakuro numbers are not allowed to appear more than once in a single column or row, similar to Sudoku. There exists a puzzle which, when solved, adheres to both the rules of Rullo and Kakuro. One can achieve this on any $n \times n$ grid, where the field is a Latin square and the sums equal to $\frac{n(n+1)}{2}$. This puzzle has a unique solution when adhering to Rullo rules, however when adhering to Kakuro rules the solution is no longer unique. See Figure 7 for a 3×3 example.

	6	6	6
6	1	2	3
6	2	3	1
6	3	1	2

Figure 7: Solved puzzle which is both a Rullo and a Kakuro

An approach to solving similar pencil-puzzles to Rullo is to describe them as a *Constraint Satisfactions Problem* (CSP [RN20, Chapter 6]). If we do that we can use generic CSP solvers to tackle Rullo puzzles. Some others have already done this for previously mentioned puzzles. For Nonogram puzzles there have been multiple approaches. Yu, Lee and Chen [YLC11] used a chronological backtracking algorithm for solving the CSP, while Batenburg and Kosters [BK09] used combining relaxations into a 2-Satisfiability problem instead. Kakuro has also been converted to a CSP [Sim08]. Simonis used a "generalized arc consistent (GAC) version of the *alldifferent-sum* constraint" to solve Kakuros and compared the results to models using MILP (mixed integer linear programming) and a PseudoBoolean mapped to a SAT solver.

4 Difficulties

When solving RULLO puzzles it is not always possible to solve a puzzle without guessing. In this paper our goal is to find puzzles with a unique solution. When one has to guess to solve a puzzle perhaps this puzzle does not have a unique solution. There are some common patterns which, when they are present in a puzzle, make the puzzle not unique. To differentiate between puzzles for which you have to make a guess and puzzles which do not require any guessing we use the terms non-simple and simple puzzles respectively. When we are talking about non-simple puzzles, this means that the player has to make a guess in order to get to a solution. A simple puzzle on the other hand can be solved step by step without any guessing.

4.1 Variants

There is almost an endless number of possible puzzles for a certain grid size, namely $|R|^{h*w} * 2^{h*w}$. It is also possible to think of many different variations. One such variation concerns the numbers that are filled in the grid. In the example above we used the *range* $R = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, denoted by 1–9, to make it easier to understand for people who are used to SUDOKUS, however one can also go for the numbers 1–19 or 2–4. It is even possible to limit the numbers to 1s and 2s. Another variant is to change the way the numbers on the side are reached. One could multiply each number in the row instead of adding them. But it is also possible to subtract or divide. When subtracting or dividing there are different numbers on the top side versus the bottom side. In that case the number on top minus all numbers in the playing field should equal the number on the bottom. It is also possible to combine the previous two types of variants, which would result in a playing board where each field not only contains a number but also contains a sign.

4.2 Deadly patterns

Not every puzzle is of the same difficulty, some are easy, others are hard. But there is another distinction between puzzles, which is whether they have a unique solution or not. The ideal situation would be that each puzzle would have one and only one solution, but that is often not the case. One of the reasons a puzzle does not have a unique solution is a so-called "deadly patterns", a term which is used in SUDOKUS [Sud] to denote groups of squares which can be filled in in multiple combinations and still have a correct solution. For RULLO it means a group of squares with usually the same numbers which can be selected in multiple ways.

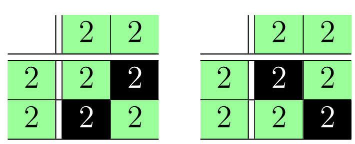

Figure 8: A simple example of a "deadly pattern" in RULLO.

Figure 8 shows a trivial example of a "deadly pattern". There are also more complex situations where not all numbers are the same but there are multiple numbers which together cause multiple "deadly patterns", see Figure 9 for an example. In this example we have set all numbers which are not part of the "deadly pattern" to 0. On the right we have guessed the top left number to be included. This reveals that there are actually two separate “deadly patterns” in this puzzle. One is simple and consists of the four 6’s in the puzzles. The other one is more complex consisting of both 4’s and 2’s, spanning over multiple columns and rows. But it still only contains two different solutions.

	4	10	4	6
2	2	2	0	0
4	0	4	4	0
6	0	6	0	6
8	2	6	2	6
4	2	2	2	0

	4	10	4	6
2	2	2	0	0
4	0	4	4	0
6	0	6	0	6
8	2	6	2	6
4	2	2	2	0

Figure 9: A more complex example of a “deadly pattern” in RULLO.

5 Methodology

In Section 1.1 it was explained that fields in a solved board can have one of two states, either they are included or they are excluded. But while we are solving the puzzle we have a third state, namely unknown. At the start of the puzzle every field is in the unknown state and it indicates that we are not certain in what state the field needs to be. At the start this is the case because we have not yet checked any lines/fields. The addition of this extra state will help us with solving puzzles.

5.1 Solving a line

To solve a single line we need to have an array of numbers in that line and the line-sum. Before we will try to solve the line we need to check if we already know the solution to the line. There are two scenarios which need to be considered. The first one is straightforward and this is a row that is already completely known and correct. These lines do not need any further processing. The other scenario exists because of the added unknown state. It is possible that all numbers marked as included already add to the line-sum even though there are still unknown numbers. This means that all unknown numbers left need to be marked as excluded. The easiest example of this is a line with a line-sum of 0, because all numbers need to be marked as excluded.
After all completed lines have been filtered out we get to the main loop for solving a single line. We have chosen to go through every possible combination within a line. But before check these combination we filter out all combinations which have a number which is bigger then target-sum, since these can never be included. If we have a combination, we first check if the combination corresponds with what we currently already know. This means that if we know a number is excluded we should filter out all combinations that do not have that number excluded. The same thing needs to be done for all included numbers.
After filtering the combinations down to all combinations that comply with what we already know, we calculate the line-sum that combination will result in. If the calculated line-sum equals the target-sum, we know that that specific combination is a possible combination. We mark all numbers used in that combination. If we do this for all possible combinations we get a list of all numbers in the line and how often they appear in combinations. If the number does not appear in any combination with a correct sum we can mark that number as excluded. Conversely, if a number appears just as often as there are valid combinations we can mark that number as included.
In Figure 10 we can see an example of a line. This line has been ordered from small to large, but the same idea applies if the order is different. As we can see the target-sum is 14 which means that we instantly exclude 15 since it is too big. Then since 14 is an even number and we have only one odd number left we can also exclude the 3. Furthermore we can see that to get to 14 with the remaining numbers $\{2, 4, 8, 8\}$ we need to have one 2, one 4 and one 8. Since there is only one 2 and also one 4 we can include both. But since we have two eights and need only one we cannot say anything about them. We need to get a hint from one of the columns to be able to make a decision.

14	2	3	4	8	8	15

14	2	3	4	8	8	15

Figure 10: Example of a line.

5.2 Solving puzzles

In this section it is explained how our algorithm generates and solves puzzles, as well as which problems can be encountered when solving Rullo puzzles.

5.2.1 Generating puzzles

Before we can solve a puzzle we first need to generate a new puzzle. To generate a random puzzle we will start by filling every field on the board with a random number in range R. After every field is filled we generate a solution mask. To do this, we will randomly select a 1 or 0 for a field. This mask will decide if a field is going to be included (1) or excluded (0) in the final solution. So now we have a solution mask and a board filled with numbers, so we can now calculate which line-sums should be used. This is done by looking at each line, and adding the numbers marked as included to each other, the resulting sum is the line-sum.
This implementation has a pretty big flaw, which is that it is unknown whether a puzzle has a unique solution. This means that while after we have generated the puzzle we do not know if the puzzle contains a "deadly pattern" or not. This would be useful information before we start solving the puzzle, but right now we will have to figure out whether the puzzle contains a "deadly pattern" or not during the solving of the puzzle.

5.2.2 Solving the puzzle

We start by solving for the columns, we do this by going over each column and solve it as described in Section 5.1. We then check if the result is the correct solution, if it is not than we will also solve for every row in the same manner. After one iteration of solving the columns and rows we check if we actually made any changes. Because if we did not that means that we have reached a "deadly pattern" which is not solvable by just looking at a single line. This will require us to use a different approach as explained in Section 5.2.3.

5.2.3 Exhaustive search

The exhaustive search should only be used when a "deadly pattern" has been found since it is significantly slower than the solution described in Section 5.1. However, there is one problem which is that it is hard to detect which fields are part of the "deadly pattern". Therefore we decided to exhaustively search through all unknown fields at the moment this method is used. For each unknown field we mark it both included and excluded and in both cases we use the method described above (Section 5.2.2) to solve the new puzzle. For some puzzle it is necessary that you have to guess again, this means that Exhaustive search will be used again. After we have gone through all unknown fields and solved them we compare all found solutions and see how many different solutions there are. This is useful information for the experiments (see Section 6).

5.2.4 Difficulty measure

Now that we know the approach for solving Rullo puzzles, we can think about how we should classify the difficulty. There are different approaches to do this, but we chose a simple approach.

We look at how often we have to check the rows and columns. But since there are puzzles where it matters whether one starts with rows or columns we use the following formula:

$$\frac{D_{row} + D_{col}}{2}$$

So we take the difficulty achieved when starting with rows (D_{row}) and add that to the difficulty achieved when starting with columns (D_{col}) and we take the average. This means that we have to do the steps described in Section 5.2.2 twice, firstly starting with checking the columns and then starting with checking the rows. We have tried to make it as straight forward as possible for the non-simple puzzles. What we have done is exhaustively search through all possible guesses and take the difficulty from that point forward and add this to the already achieved difficulty.

5.3 Inverted puzzles

Something worth noting is that when a puzzle gets inverted, the order of steps towards solving the puzzle stays the same. Inverting a puzzle means that we invert the solution and recalculate the sums. So if a square is included in one puzzle, it should be excluded it its inverse.

26	9	8	1	8	8

8	9	8	1	8	8

Figure 11: Example of an inverted line.

In Figure 11 we see the second column from the example puzzle. And as we found in Section 2.2 we know in the original line that the 9 and 1 are required to get to 26. In the inverted line the opposite happens, because here it is impossible to make 8 with either the 9 or the 1 included, and therefore both are excluded. This logic can be extended to the complete example puzzle, which results in Figure 12. Here we can see the original puzzle on the left and the inverted puzzle on the right. Trying to solve this puzzle with the exact same steps as used in Section 2.2, we get to the solution on the right. The only difference is that when you originally could include a number, we can now exclude it.
Knowing that inverting a puzzle keeps the difficulty the same and also the approach to solving the puzzle we will only look at half of the puzzles going forward. To be more specific, we will look at the puzzles with fewer than or equal to half of the fields required. We did this because we found it easier to understand puzzles with fewer required fields compared to puzzles with more required fields when manually solving puzzles.

	10	26	16	10	20
23	8	9	8	5	6
13	7	8	3	5	9
17	8	1	8	5	3
8	2	8	5	2	6
21	2	8	4	5	8

	17	8	14	12	12
13	8	9	8	5	6
19	7	8	3	5	9
8	8	1	8	5	3
15	2	8	5	2	6
6	2	8	4	5	8

Figure 12: The example puzzle and its inverted counterpart.

6 Experiments

We ran multiple experiments searching for the most difficult puzzle. Because the number of puzzles increases exponentially as the size of the puzzles scales, we had to find a balance between size and thoroughness. This means that for the smaller puzzles we look at all possible puzzles but as the size increases we only look at a random subset of puzzles.

6.1 Small-sized puzzles

Starting with the smallest puzzles and working our way up, we begin with experiments on a 3×3 grid. The goal of this first experiment is to go through all possible puzzles and see if there is anything extraordinary. To be able to solve all possible puzzles we need to limit R. Because if we use same R as the example puzzle where $R = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ this will result in too many puzzles, which is not feasible to solve in a reasonable amount of time. To put a concrete number on this, if we use a R of 1–9 we would have $9^9 = 387,420,489$ possible ways to fill the grid with numbers, and each possibility has $2^9 = 512$ different solutions. This means that there are $387,420,489 \times 512 = 198,359,290,368$ different puzzles and trying to solve all of them will take too much time. Therefore we have decided to limit the range to 2–4. This means there are $3^9 = 19,683$ different ways to fill the grid and thus $19,683 \times 512 = 10,077,696$ different puzzles, this does include symmetrical puzzles (see Section 5.3, which is still a significant number of puzzles but it is solvable in a reasonable amount of time. Run time is about 5 minutes.

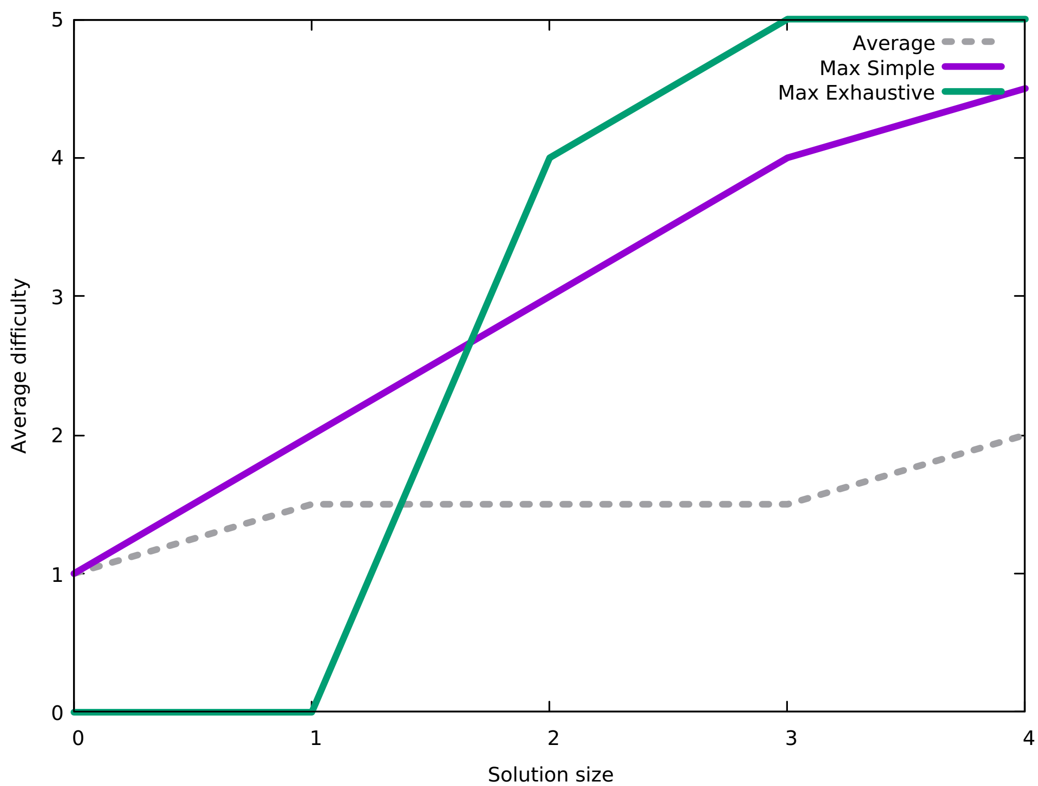

Figure 13: Average difficulty based on number of fields required in solution for 3×3 puzzles.

In Figure 13 we provide the average difficulty compared to the number of required fields in a solution. With the number of required fields in a solution we mean that given the solution how many squares should be marked as included. The average difficulty does not go above 2, this means that in an average puzzle the player only needs to check the rows once as well as the columns once. And it does not matter which direction one starts the puzzle.

Difficulty	Simple	Exhaustive
1	166,371	0
1.5	1,395,258	0
2	2,640,060	0
2.5	511,848	0
3	106,596	87
3.5	13,284	0
4	1,368	183,690
4.5	108	216
5	0	19,962

Required fields	Simple	Exhaustive
0	19,683	0
1	177,147	0
2	695,466	13,122
3	1,589,394	63,978
4	2,353,203	126,855

Table 1: Distribution between normal and exhaustive search for a given difficulty (left) or number of required fields (right) on 3×3 puzzles.

Table 1 shows a table with the exact number of puzzles that require exhaustive search, grouped by difficulty and number of required fields in the solution. As we can see from the tables there are no non-simple puzzles when there 0 or 1 required fields, this is because if there are zero the puzzles is already solved and if there is 1 required field there is only one unknown number, and this is always in the same position. But as we increase the number of required fields the number of puzzles which include a "deadly pattern" also increases. There are a total of 4,834,893 simple puzzles and 203,955 non-simple puzzles, this means that 95.95% of the puzzles are simple and can be solved without guessing. For the other 4.05% one has to make at least one guess to be able to solve them.

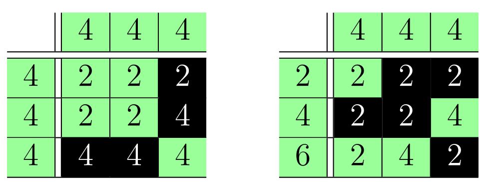

Figure 14: The most difficult simple puzzles on a 3×3 grid. These puzzles have a difficulty score of 4.5.

Figure 14 shows the most difficult puzzles that were found. This might give us some insight in how difficult puzzles can be made and we might see some returning patterns in experiments on bigger grid sizes. If we also count the inversions of all puzzles, then for 3×3 puzzles there are a total of 216 simple puzzles with the highest difficulty found, which is 4.5. But since rearranging the columns and rows does not change the puzzle in a meaningful way we can always sort the grid in the same way, with the smallest number in the top left and the biggest in the bottom right. This leaves us with two unique puzzles with a difficulty of 4.5. These difficult puzzles share some common characteristics, for starters the first square one can mark can only be marked when checking either the rows or columns, the other direction does not yield any progress. For the first puzzle one has to start with checking the columns and the first known field is the 2 in row 1, column 3. For the second puzzle we have to start with the rows, here the first known field is the 4 in row 3, column 2.

	4	5	6
4	2	2	4
5	2	3	2
6	2	2	4

Figure 15: Most difficult 3×3 puzzle which uses all number in R. Difficulty score of 4.

As we have seen before the most difficult 3×3 puzzles do not use all numbers in the range. They only require two different numbers, but ignore the existence of the 3. If we require all numbers to be included at least once, the most difficult puzzle we get can be seen in Figure 15. Contrary to the most difficulty 3×3 puzzle, for this puzzle it does not matter whether one starts checking the rows or columns first.

6.2 Medium-sized puzzles

We decided on a more random approach for the 5×5 puzzles, since going over all possible puzzles is simply not possible. For any 5×5 grid, disregarding the range of the puzzle, there are a total of $2^{25} = 33,554,432$ possible solutions. Even if we were to limit the other variables to extremes one would still end up with too many puzzles. If we for example take a range $R = \{2, 3, 4\}$, which we used for the 3×3 puzzles, we would still end up with $3^{25} \times 33,553,543 \approx 2.843 \times 10^{19}$ different puzzles which we need to solve. Not only is this more puzzles than we evaluated on a 3×3 grid, each puzzle will also take more time to solve since there are more possibilities to check in each line. Thus we came to a compromise: we will create a grid filled with random numbers, and then we will try to solve every possible solution on that grid. We did this on 200 different grids. This results in a total of $200 \times 33,554,432 = 6,710,886,400$ puzzles being evaluated.
Figure 16 shows the maximum difficulty for simple and non-simple puzzles as well as the average difficulty based on the required fields in the solution. As we can see the average barely increases between a solution size of 4 with a difficulty of 2 and 12 with a difficulty of 2.5. We can also see that the peak of the highest difficulty for simple puzzles is higher than the peak of the non-simple puzzles. This has to do with the fact that we fill a grid completely random. While there are some "deadly patterns", none of these 200 boards has more then a few "deadly patterns".
As indicated in Table 2, the higher difficulty simple puzzles defy initial expectations, showcasing an unexpected and intriguing pattern. Note that difficulty 9 has no simple puzzles at all but difficulty 9.5 has 1,192 simple puzzles. This phenomenon can be explained by the fact that if one gets the same difficulty no matter the starting direction the puzzle is inherently easier, since both directions have a starting point, while if only one puzzle has a starting point and the other direction gives no information the puzzle will always be more difficult. For the exhaustive puzzles this somewhat inverted where difficulty 8.5 has no non-simple puzzles and difficulty 9 has 7,664 non-simple puzzles. This is most likely for a similar reason as for the simple puzzles. Since a "deadly pattern" does not care about which direction one is checking it from, we suspect that some of the non-simple puzzles with a difficulty of 9 have a "deadly pattern" at the start. This means that the difficulty

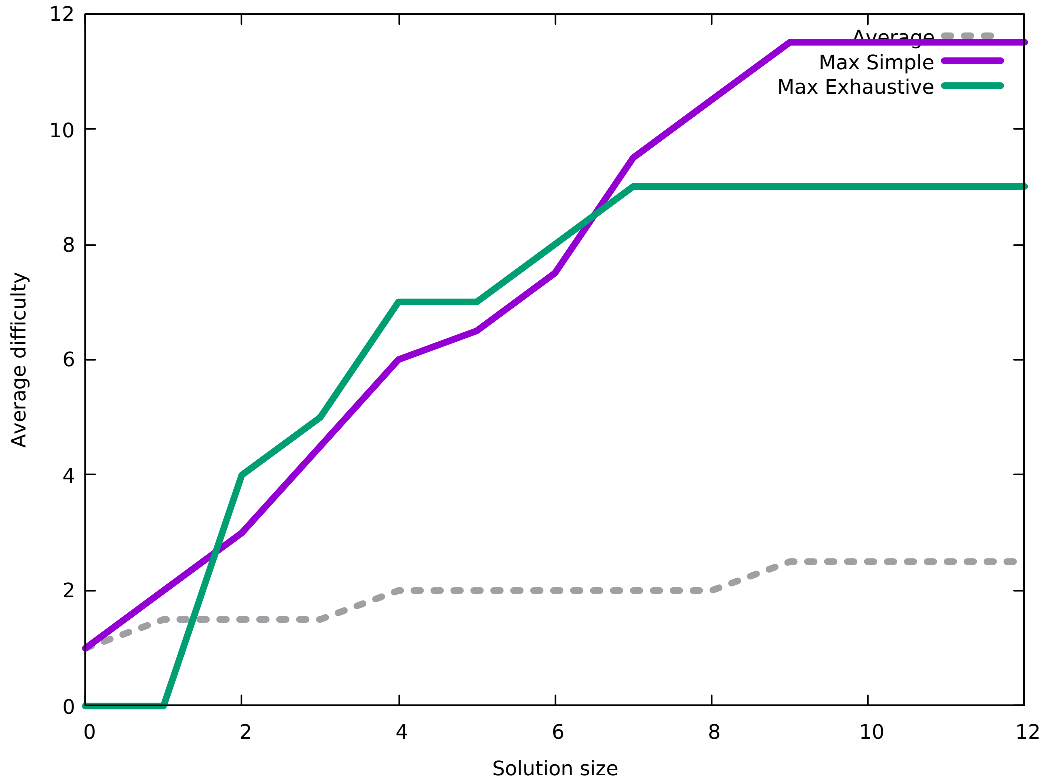

Figure 16: Average difficulty based on number of fields required in solution for 5×5 puzzles.

for starting with either direction is always the same since the first number we mark as included or excluded is always going to be a guess.
In Table 2 on the right we can see that the percentage of non-simple puzzles, compared to the total number of puzzles given a set number of required fields, increases as the number of required fields also increases. This goes from 0.14% for 2 required fields up to 3.14% for 12 required fields. As previously mentioned there are two correct solutions for a "deadly pattern". Combining this knowledge with the fact that there are 86 non-simple puzzles with 2 required fields, we can say that there are at least 43 different "deadly patterns" spread out over the 200 boards. This does however not mean that there are 43 boards with a "deadly pattern", since it is possible that one board has multiple unrelated "deadly patterns" present.

	12	13	8	8	13
11	4	3	7	1	8
9	3	4	1	5	8
15	9	6	4	5	7
7	2	4	3	7	1
12	3	7	4	3	5

Figure 17: One of the most difficult simple puzzles on a 5×5 grid we found. The difficulty of this puzzle is 11.5.

In Figure 17 we can see the most difficult 5×5 puzzle which we have found. This puzzles has some

Difficulty	Simple	Exhaustive
1	84,072	0
1.5	37,493,040	0
2	2,259,840,718	0
2.5	2,308,181,300	0
3	1,301,983,058	0
3.5	463,709,884	0
4	94,626,372	87,766,456
4.5	42,225,718	218,396
5	4,406,920	97,400,764
5.5	3,685,510	371,644
6	214,946	6,088,704
6.5	392,180	27,668
7	4,946	1,835,672
7.5	53,140	3,216
8	142	252,400
8.5	10,274	0
9	0	7,664
9.5	1,192	0
10	0	0
10.5	268	0
11	0	0
11.5	136	0

Required fields	Simple	Exhaustive
0	200	0
1	5,000	0
2	59,914	86
3	458,163	1,837
4	2,511,371	18,629
5	10,506,662	119,338
6	34,878,006	541,994
7	94,283,622	1,856,378
8	211,336,054	4,978,946
9	397,878,896	10,716,104
10	634,939,154	18,812,846
11	864,249,640	27,230,360
12	1,007,350,226	32,709,774

Table 2: Distribution between normal and exhaustive search for a given difficulty (left) or number of required fields (right) on a 5×5 puzzles.

interesting characteristics about it. For example, even though the difficulty differs depending on in which direction one starts, one can in both directions mark the 6 as included in the second column and third row, as well as the 5 in the fifth column and row. But because both the 6 and 5 do not have any further implications one still gets a lower difficulty when starting with the rows. Another thing worth noting is that in the first few steps one can progress with 2 new squares. After a few switches one starts to be able to only go further one square at a time. To get to the most difficulty puzzle, this streak of only being able to progress one square for each step would need to be as long as possible. Going back to this 5×5 puzzle, once one is past the series of steps were one is able to progress one square at a time, the puzzles starts ramping up in pace and one can fill in more squares with each step. Until eventually the puzzle is completely solved. We have counted how many squares of progress one can make with each step, given one starts checking the columns first. This results in the following list: $\{2-1-1-1-2-3-2-2-3-3-3-2\}$. So it starts of great with only being able to progress one or two squares each step, but near the end one can progress three squares each step for multiple steps in a row, which is less than ideal.
As we saw with the most difficult 3×3 puzzles, the most difficult puzzle does not necessarily contain all numbers in range R. However, this puzzle does contain all numbers in R, but this does not say anything about the difficulty of the puzzle.

6.3 Large-sized puzzles

For our final experiment we decided to look at even bigger puzzles, namely 7×7 puzzles. As explained previously we cannot compute all possible 5×5 puzzles, let alone 7×7 puzzles. And since bigger puzzles require more time to solve we also cannot use the same approach as we used for the 5×5 puzzles. So we have decided to make the puzzles completely random. We generate both a completely random board as well as a random solution and we try to solve this. All puzzles are generated independently and we also do not have a check to see if the puzzle has already been solved, so it is theoretically possible that there are one or more duplicate puzzles in the examples. But there are a total of $9^{49} \approx 5.73 \times 10^{46}$ possible ways to fill a 7×7 grid with the numbers in $R = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Add to that the fact that there are $2^{49} \approx 5.63 \times 10^{14}$ possible solutions on one grid. This means that the chances that we have included a duplicate or an inversion of a puzzle are extremely small.

We have evaluated $100,000,000$ puzzles and while this number seems quite small compared to the number of 5×5 puzzles we have evaluated it still took more time to finish them all. The time needed to solve all of the 5×5 puzzles was approximately 4.5 hours. Solving all the 7×7 puzzles that we generated took 15 hours. This has a few reasons, firstly because we randomly generate the solution and we only look at one solution at a time there is a small chance that we get a solution with very few required fields. In addition the grid size nearly doubled, which means that there are more combinations to check in each line. Another issue is the existence of non-simple puzzles, because if we hit a "deadly pattern" too soon we will exhaustively solve a puzzle with too many unknowns and this takes most of the time. We have tried to prevent this as much as possible by setting a requirement for exhaustive searching. We will only try to solve non-simple puzzles if we have already 20 certain squares before we start guessing. Without this limitation it can take hours before we can solve a single puzzle if we are unlucky.

In Figure 18 we can see the result of this limitation. Because we set this limitation on non-simple puzzles they do not reach the same difficulty as the simple puzzles. A result of the randomness of this experiment is the fact that the difficulty peak for simple puzzles is at 23 required fields instead of 24 as we might expect. We believe that we got "lucky" with a puzzle of size 23 and not with any puzzle with 24 required fields. The same principle applies to the maximum difficulty of non-simple puzzles. This fluctuates quite drastically and we believe this is caused by the randomness of the puzzles in combination with the limitation we set on guessing.

Based on Table 3 we can deduct that the percentage of non-simple puzzles increases as the number of required fields increases. Even though the puzzles are completely random. There is one exception to this which is 8 required fields. Here 5.56% of the puzzles are non-simple and for 9 required fields it goes back down to 1.88%. This outlier is likely caused by the random nature in combination with the low sample size of 54 puzzles with 8 required fields. In the 5×5 puzzles we saw that at most 3.14% of the puzzles were non-simple. For the 7×7 puzzles this goes up to 10.47%. While this is only a comparison between two experiments, we can observe that augmenting either the randomness or the grid size leads to a higher percentage of non-simple puzzles.

On the left in Table 3 we can again see that the highest difficulty for non-simple puzzles does not go above 16.5, because we have limited when the solver is allowed to switch to guessing. Something that is interesting to note is the fact that the highest number of games of non-simple puzzle is more difficult than the peak of the simple puzzles. There are 8 million non-simple puzzles with a difficulty of 6 up to 6.5, which is the most common for non-simple puzzles. For simple puzzles

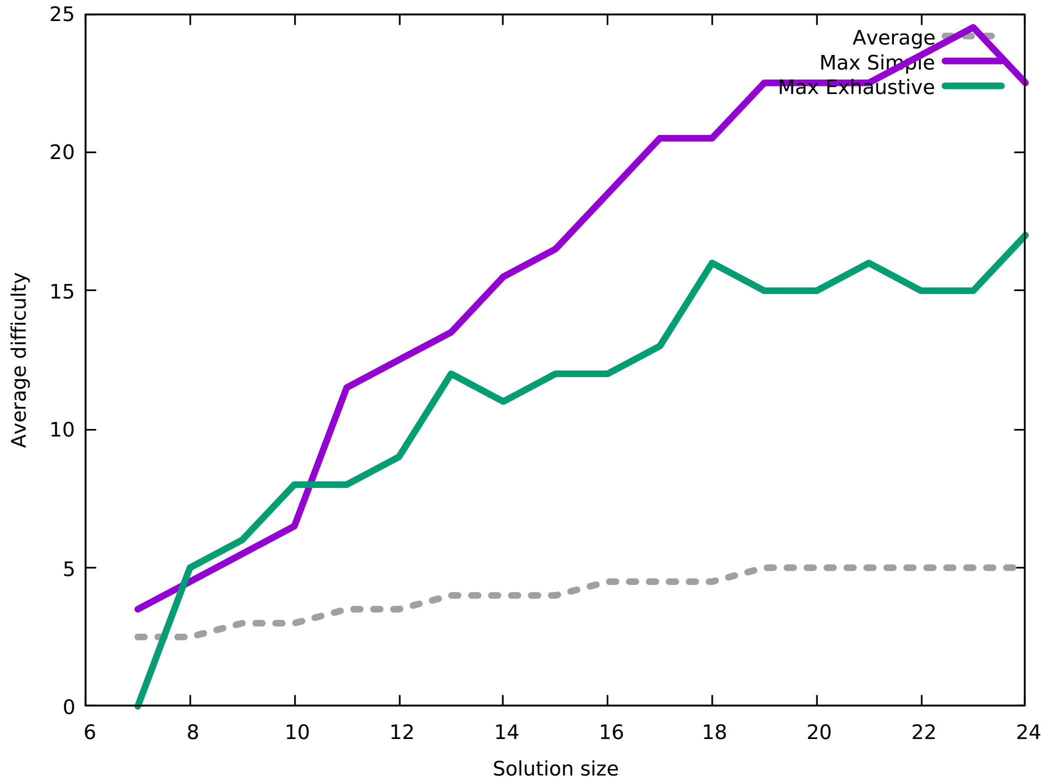

Figure 18: Average difficulty based on number of fields required in solution for 7×7 puzzles.

the most common difficulty is between 4 and 4.5, with 61.5 million puzzles. While the number of puzzles for each difficulty is always greater for simple puzzles compared to the non-simple puzzles, the peak differs. This is because it is unlikely to have a puzzle with both very low difficulty and which includes one or more "deadly patterns".

	8	14	16	26	13	12	18
16	3	6	2	6	5	2	1
22	7	8	7	8	1	3	6
12	1	2	4	8	1	7	1
26	8	1	3	9	2	3	9
10	6	7	4	6	3	5	3
11	5	5	1	3	1	7	3
10	1	1	2	1	9	6	9

Figure 19: The most difficult simple puzzles on a 7×7 grid we found. This puzzles has a difficulty score of 23.5.

The most difficult simple puzzle we found can be seen in Figure 19. When trying to solve this puzzle one might find out that starting with trying the columns does not yield any progress whatsoever. But if one starts with the rows one will be able to fill in two different fields. However these two fields are both in the sixth column, which means that in the next step one can only mark fields in

Difficulty	Simple	Exhaustive
2.0/2.5	566,524	0
3.0/3.5	24,553,506	0
4.0/4.5	61,564,060	18,092
5.0/5.5	46,004,624	4,701,448
6.0/6.5	22,678,964	8,144,660
7.0/7.5	10,079,992	4,013,450
8.0/8.5	4,660,222	1,668,608
9.0/9.5	2,249,700	615,720
10.0/10.5	1,131,952	192,964
11.0/11.5	564,944	52,356
12.0/12.5	277,816	12,232
13.0/13.5	131,206	2,528
14.0/14.5	59,642	450
15.0/15.5	25,464	64
16.0/16.5	10,672	14
17.0/17.5	4,270	0
18.0/18.5	1,618	0
19.0/19.5	630	0
20.0/20.5	216	0
21.0/21.5	70	0
22.0/22.5	22	0
23.0/23.5	6	0

Required fields	Simple	Exhaustive
7	5	0
8	47	2
9	311	7
10	1,543	49
11	6,616	305
12	24,347	1,340
13	75,884	4,585
14	207,045	14,300
15	498,512	38,141
16	1,080,104	91,415
17	2,108,990	192,769
18	3,731,251	364,210
19	6,032,168	619,096
20	8,940,764	960,087
21	12,185,841	1,354,013
22	15,320,772	1,749,462
23	17,832,131	2,071,155
24	19,236,729	2,250,357

Table 3: Distribution between normal and exhaustive search for a given difficulty (left) or number of required fields (right) on 7×7 puzzles.

one column. But one is able to almost solve the complete column, the only numbers uncertain are the 7's, since one is needed but at this moment it is not known which should be included and which excluded. If one would start with checking the rows the number of found fields goes as follows: $2-3-3-3-1-1-2-2-1-1-3-2-3-2-2-1-2-3-1-1-1-2-4-3$. As one can see it fluctuates between being able to solve 1, 2 or 3 fields each step. To get to the most difficult puzzle, one would need the streak of being a progress one field at a time to be as long as possible. In this puzzle we only have a streak of 3 consecutive steps each only being able to progress one field.

Figure 20 shows one of the most difficult non-simple puzzle we found. We found multiple non-simple puzzle with a difficulty of 16. This puzzle contains multiple intertwined "deadly patterns", so to be able to solve this puzzle we have to guess multiple times. In the left puzzle we have marked each "deadly pattern" in a different color. This puzzle also contains a complex "deadly pattern" as explained in Section 4.2.

The right puzzle shows how far one can get in the puzzles before one is forced to make a guess to solve the puzzle. As we can see, we can get quite far without having to make a single guess, but to know the last 8 squares we have to make a few guesses. There are two intertwined "deadly patterns", one in row 1 and 4, and column 2 and 4 consisting of only ones; the other "deadly patern" can be found in row 2 and 5, and column 4 and 6 consisting of threes. Since these two

	21	14	10	12	10	12	21
5	4	1	5	1	8	2	4
15	4	8	6	3	2	3	8
16	6	5	2	6	1	9	2
12	9	1	3	1	3	9	2
11	9	4	1	3	3	3	4
22	8	3	7	3	1	1	3
19	9	4	1	5	6	2	8

	21	14	10	12	10	12	21
5	4	1	5	1	8	2	4
15	4	8	6	3	2	3	8
16	6	5	2	6	1	9	2
12	9	1	3	1	3	9	2
11	9	4	1	3	3	3	4
22	8	3	7	3	1	1	3
19	9	4	1	5	6	2	8

Figure 20: One of the most difficult non-simple puzzles we have found on a 7×7 grid. This puzzle has a difficulty score of 16.

"deadly patterns" have no overlap, it is required to make two different guesses to be able to get to a solution.

7 Difficult puzzles

We have made an attempt at creating a generic puzzle which has the maximum difficulty. Unfortunately we did not manage to create such a puzzle. In this section we will go over some approaches we tried which unfortunately did not yield any success.

15	8	8	4	4	2	2	1	1

15	8	8	4	4	2	2	1	1

Figure 21: Example consisting of powers of 2.

Looking at just a single line we can create pairs, where if the first number of a pair needs to be included the second one should be the opposite. We can add multiple of these pairs to create a longer line. An example can be seen in Figure 21. We choose these pairs such that they do not interact with each other. If we use numbers which are a power of two we get the example row. The sum required should be $2^{\frac{\ell}{2}} - 1$. Here ℓ is the length of the line. This is assuming we start with 1 as the lowest number, and only use consecutive powers of two.
The right line in Figure 21 shows a potential solution of this particular line. But the interesting fact about this line is that we can switch the inclusion and exclusion of any number without affecting the end result. This is a helpful characteristic when trying to create more difficult puzzles. Unfortunately it is not possible to create difficult puzzles consisting entirely of lines which are similar, because one runs into a few problems. Firstly, as we noticed before with difficult puzzles, we want that the starting orientation matters, however with a line like this it is impossible to create a starting point that results in a simple puzzle and which does not yield all information when starting with either the rows or columns.

$a+b$	$2(a+b)$
b	$a+b$
b	b
a	a
a	$a+b$

Figure 22: Example pattern across two columns.

Previously we have looked at a single line, but if we try to combine multiple lines together we have also come up with a potential pattern for difficult puzzles. In this pattern, seen in Figure 22, we have tried to create a zigzag pattern, where if one knows one field, one knows all other fields, but one only progresses one square each step.
In Figure 22 we used a and b, which are numbers in range R, but they should be different from each other. This pattern only works if the a in the bottom left is known first, and it is included. If we know that we know that the first a in row 3 is not included. Because we use the same principle as before we can also say something about the a on the third row and second column.
There are still a few unanswered questions with this pattern, since we need to have a starting point, which is ideally the a in the bottom left. We can do this by adding one or more columns which force the a in the bottom left to be included. These additional columns should not be able to give the player any further hints, which is difficult to create. We have not yet been able to create some padding for this pattern to be able to create a simple puzzle with a high difficulty. While we do not

have an answer for the padding, we can extend the pattern to the right multiple times without any limit.

However, extending the pattern vertically is much more difficult. We would need to introduce a third variable. We have added an attempt in Figure 23. The first column can still be kept easy by using the first pattern from this section. For the second column we need to make sure that we do not give any hints too early. We have added an example for this pattern with $a = 1$, $b = 2$ and $c = 4$. This gives a good indication on how it works and what the potential problems are. Because while this works for two columns one still runs into some issues when looking at the rows. Take for example row 3, where we want to go from a known b to an unknown $a + b$. This is only possible because of the padding of other rows. An even more complicated situation is the second row, because here one wants to go from a $a + b$ which is known to a c. For this to work one either needs to make sure that $c = a + b$ or have some values in the padding which make this step possible. Trying to expand it even further than 6 rows will create even more of these problems.

$a+b+c$	$3(a+b)+c$
c	$a+b+c$
c	$a+b$
b	$a+b$
b	b
a	a
a	$a+b+c$

7	13
4	7
4	3
2	3
2	2
1	1
1	7

Figure 23: Example pattern across two columns.

8 Conclusions and further research

In this thesis RULLO is studied. RULLO is a mathematical puzzle game where the player has to include or exclude numbers in a grid to make the sums of each line equal to their respective line-sum which is shown on the outside of the grid. We differentiate between non-simple and simple puzzles: in a non-simple puzzle the player has to guess at least one field to be able to solve the puzzle. Puzzles which do not require any guessing to get the solution are simple puzzles, and they are the main focus.
There are some common patterns which, if they are present, require guessing to solve the puzzle. These patterns are "deadly patterns". They appear in a multitude of ways, from simple "deadly patterns" which only include one number which is present on four squares, to more complex "deadly pattenns" which exists of more numbers across multiple rows and columns.
We have also found another pattern which helps tremendously when trying to solve all possible puzzles, which is invertibility. We have found that all puzzle are invertible, which means that we only need to calculate the difficulty of half of the puzzles. Invertibility means that if we have a puzzle, inverting the solution and recalculating the line-sums the puzzle requires the exact same steps to solve as the original puzzle. And since the steps to solve the puzzles are the same the difficulty of both is also equal.
We have ran experiments on three different grid sizes. For each grid size the approach used is slightly different. While for the small-sized puzzles we can go through all of the possible puzzles, for the medium-sized and large-sized puzzles this was not possible. Even though the experiments are slightly different and can not be compared easily, we still found some interesting patterns. Because to get the most difficult simple puzzle, one requirement is that it matters whether one starts with checking the rows or columns.

8.1 Further research

There is still many aspects of RULLO puzzles undiscovered. But some things we found interesting and warrant additional research. Firstly, we suspect that the question of whether a RULLO puzzle is solvable or not, is NP-complete, although we have not proved it. Similar puzzles are already proven to be NP-complete, e.g., NONOGRAMS [UN96], KAKURO [Set02, RH10], SUDOKU [YS03], and more. Therefore we suspect that RULLO puzzles are as well.
We have also tried to create difficult puzzles ourselves, but we did not manage to create a generic approach to create them. This aspect of RULLO puzzles can be further expanded upon. Another possibility related to creating the most difficult puzzle, is calculating the theoretical maximum difficulty for any $m \times n$ sized puzzle.

References

[BK09] Kees Joost Batenburg and Walter A. Kosters. Solving Nonograms by combining relaxations. *Pattern Recognition*, 42(8):1672–1683, 2009.

[BK12] K.J. Batenburg and W.A. Kosters. On the difficulty of Nonograms. *ICGA Journal*, 35(4):195–205, 2012.

[Cre] Cresentyr home page. http://crescentyr.com. Last accessed on 15 July, 2023.

[Nik] Nikoli. https://www.nikoli.co.jp/en. Last accessed on 10 Augustus, 2023.

[RH10] Oliver Ruepp and Markus Holzer. The computational complexity of the Kakuro puzzle, revisited. In *International Conference on Fun with Algorithms*, pages 319–330. Springer, 2010.

[RN20] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 4th edition, 2020.

[Set02] Takahiro Seta. The complexities of puzzles, cross sum and their another solution problems (ASP). *Senior Thesis, Department of Infomation Science, the Faculty of Science, University of Tokyo*, 2002.

[Sim08] Helmut Simonis. Kakuro as a constraint problem. *Proceedings Seventh Internation Workshop on Constraint Modelling and Reformulation*, 2008.

[Sud] Sudopedia. http://sudopedia.enjoysudoku.com/Deadly_Pattern.html. Last accessed on 30 July 2023.

[UN96] Nobuhisa Ueda and Tadaaki Nagao. NP-completeness results for nonogram via parsimonious reductions. *preprint*, 1996.

[YLC11] Chiung-Hsueh Yu, Hui-Lung Lee, and Ling-Hwei Chen. An efficient algorithm for solving nonograms. *Applied Intelligence*, 35:18–31, 2011.

[YS03] Takayuki Yato and Takahiro Seta. Complexity and completeness of finding another solution and its application to puzzles. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 86(5):1052–1060, 2003.