Universiteit Leiden
The Netherlands

Opleiding Informatica

Using Trains for

Ice Block Puzzles

Tristan van den Akker

Supervisors:
Walter Kosters & Jeannette de Graaf

BACHELOR THESIS

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

May 21, 2024

Abstract

Ice block puzzles consist of a two-dimensional board, filled with normal floors, slippery ice floors, ice blocks and walls. The player is allowed to push ice blocks, that will then continue sliding over ice floors. The goal of these puzzles is to get an ice block on a specific target location on the board. In this thesis we will take a look at interesting movement patterns with ice blocks found in quickest solutions, as well as image generation with ice blocks.

Contents

1 Introduction

Many video games include puzzles as a way to challenge the player's problem solving skills. A specific type of puzzle is the ice sliding puzzle. Ice sliding puzzles come in many variations, such as an example from a game called RuneScape that can be seen in Figure 1.

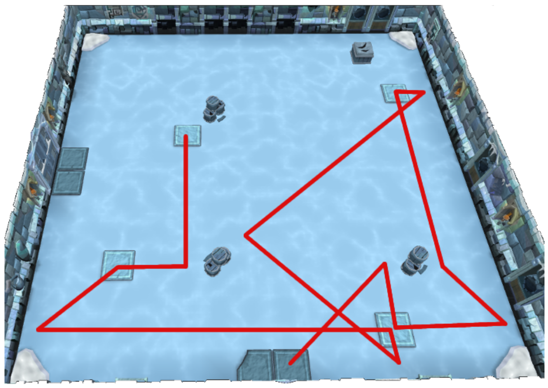

Figure 1: Runescape ice floor puzzle; source: https://www.tip.it/runescape/pages/view/dungeoneering_puzzles.htm.

A different example from the Zelda games can be seen in Figure 2.

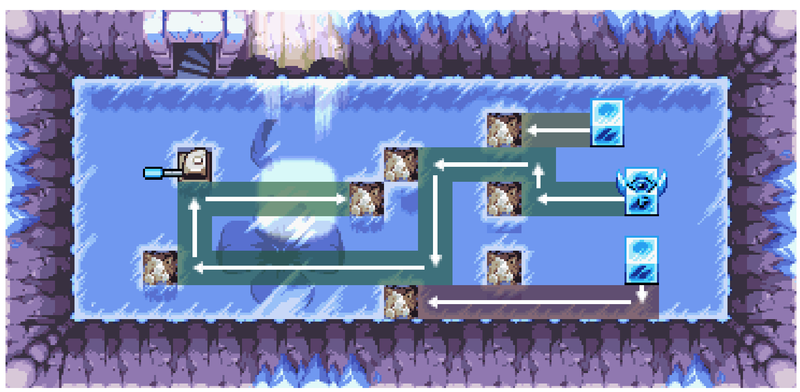

Figure 2: Zelda ice block puzzle; source: https://strategywiki.org/wiki/The_Legend_of_Zelda:_The_Minish_Cap/Temple_of_Droplets.

The specific rules and goals of these puzzles vary. In Figure 1 the player slides over the floor, and the goal is to have stood on all pressure plates on the floor. In Figure 2 the ice blocks slide over the floor, and the goal is to have the ice block with a key reach a certain position. While these puzzles are different, they do share a common theme: objects, or the player itself, slide over an icy floor.

The ruleset we use in this thesis is very similar to the example in Figure 2. The objective is to have ice blocks standing still on specific target locations on the board. The player is able to push ice blocks, which keep sliding until they get blocked.

We will look at different movement patterns used to solve ice block puzzles, such as the “trains”, and “zigzag”. We show the number of moves needed to reach every single tile on a number of different sizes of square boards, and find that using an increasing number of ice blocks leads to some interesting new methods of reaching a fastest solution. We will research boards of height two, and height three, and come up with formulas to calculate the number of moves needed to reach the hardest tile, using the width as a parameter. Lastly we discuss using ice block puzzles to create black and white pixel images. We discover that not every image is creatable, and prove that any image containing specific configurations is impossible to create in an ice block puzzle.

1.1 Thesis overview

Section 2 describes the rules of ice block puzzles, and how we illustrate them; Section 3 discusses related work; Section 4 describes the research done on different board sizes and the results; Section 5 contains attempts at image creation in ice block puzzles; Section 6 concludes this paper, and suggests some possibilities for future work.

This thesis is based on research conducted for a bachelor project at LIACS, Leiden University, and was supervised by Walter Kosters and Jeannette de Graaf.

2 The puzzle

Ice sliding puzzles usually consist of a two-dimensional grid. This grid contains fields that can be any of the following types:

1. Ice block: A block able to be pushed by the player.
2. Ice floor: A slippery tile that allows ice blocks to slide.
3. Normal floor: A non-slippery tile. Ice blocks can not be on top of these tiles.
4. Wall: An obstacle that can not be moved.
5. Targets: Special tiles that need to be reached by an ice block.

We will use illustrations to clarify. In these pictures the ice blocks are blue, the ice floors are white, the normal floors are beige, the walls are black and the targets are an ice floor with a black cross on them. Red lines may be used to show a move, or a series of moves for an ice block. For the sake of simplicity all puzzles depicted are surrounded by normal floors that are not included in the pictures themselves, unless otherwise specified. In Figure 3 the different tiles are shown, and Figure 4 shows an example of a complete puzzle, along with its solution shown in red.

Figure 3: From left to right: Ice block, Ice floor, Normal floor, Wall, Target.

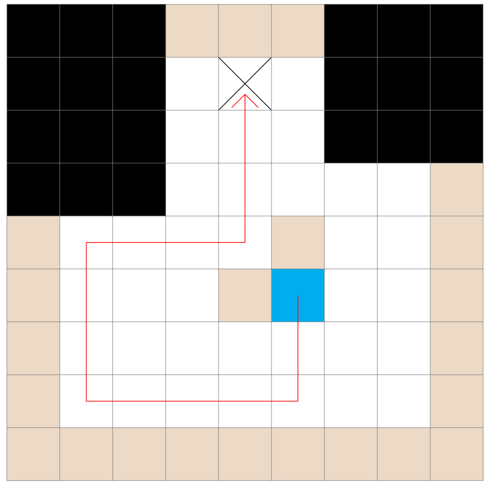

Figure 4: Example puzzle, with the fastest solution, consisting of 5 moves.

We will focus on ice sliding puzzles with the following ruleset.

The player can move horizontally or vertically. The player can move normally over ice tiles, so they will not keep sliding until reaching a blockade such as a wall or a normal floor tile. The player is not allowed to walk on wall tiles or ice block tiles. Other than ice tiles, normal floor tiles and the target are also tiles the player can walk on.

In order for the puzzles to be solved, ice blocks need to be pushed by the player. The player can push an ice block by standing directly (horizontally or vertically) next to it, and then moving in the direction of the ice block. This causes the ice block to start sliding in the same direction, unless the tile opposite the ice block was occupied by either a normal floor, a wall, or another ice block. In that case the ice block would not be able to move in that direction. If the ice block is pushed successfully, it will keep moving in the same direction over ice floors. The ice block stops sliding if the next tile is a normal floor tile, a wall, or an ice block. Figure 5 shows four attempts at ice block pushes. Only the leftmost position allows for the ice block to be moved, because it is not blocked by a normal floor tile, a wall, or another ice block.

Figure 5: The leftmost image shows a legal move, the other moves get blocked.

The goal of the puzzle is to place an ice block on every target on the grid. To place an ice block on the target means that the ice block has to be standing still. These target tiles are also part of the ice floor, which means an ice block is able to slide over it. Sliding an ice block across a target tile does not hit the target, which means this move will not solve the puzzle yet. Figure 6 shows an example of a move that will hit a target on the right, and a move that will not hit the target on the left.

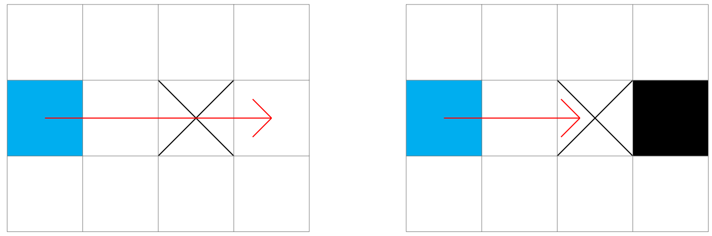

Figure 6: The left move slides over target, the right move slides onto target.

The number of targets can vary for every puzzle. The number of ice blocks can also vary for every puzzle. Some puzzles are not solvable due to the number of targets and ice blocks. For example, a puzzle with two targets and only one ice block is never solvable. In general, for a puzzle to be solvable there need to be at least as many ice blocks as there are targets.

3 Related work

One paper about ice sliding games has been published [DDF+18]. In this paper a type of ice sliding games where a robot has to slide across the ice to reach a target is discussed. The robot does not need to stand still on a tile to have reached it, contrary to our ice block puzzles, where the ice block needs to be at a full stop on a tile. The paper goes on to discuss the minimum number of walls that need to placed on the board, so that the robot can reach every tile.

PushPush is a game similar to ice block puzzles. In this game the blocks behave the same way as in our ice block puzzle, where they keep sliding until they are stopped. The goal in PushPush is different however, because the player itself needs to reach a certain target, rather than the blocks needing to be placed on targets. PushPush has been proven to be NP-hard in 2D [DDO00]. In this same paper Push-1, a version of PushPush where a block is only moved one tile at a time instead of sliding until stopped, is also proven to be NP-hard in 2D.

Yet another similar game, Sokoban, has been proven to be PSPACE-complete [Cul97]. In this game the player pushes boxes in a storage, one tile at a time. The boxes do not slide across the floor like in our ice block puzzles. The goal of Sokoban is the same however, because the boxes need to be pushed into specific positions on the board. In [Edi16] Luc Edixhoven translates a Sokoban puzzle into an instance of the SAT problem, so SAT-solvers can be used to solve the puzzle.

Sem Kluiver has done research on ice block puzzles very similar to ours [Klu20]. In his thesis he discusses different algorithms to find solutions to ice block puzzles. Kluiver discusses solvability and reversibility of ice block puzzles and investigates the search space, even coming up with formulas for the number of states certain puzzles can be in. Lastly Kluiver attempts to grade the difficulty of ice block puzzles. He uses a combination of the number of moves necessary to complete a puzzle, and the number of times the block currently being pushed changes, to determine the difficulty of any given ice block puzzle.

4 Movement patterns and board experiments

In ice block puzzles, the starting position of the ice blocks can matter. As mentioned before, the puzzles are surrounded by normal floor tiles. Depending on the initial position, some results may vary, such as the movement pattern used to reach a solution, as well as the number of steps required for that solution. In some cases, the initial position can even cause the puzzle to be unsolvable, an example of which can be seen in Figure 7. In the configuration on the right, the starting position causes the puzzle to contain no valid push moves, therefore the target can never be reached by an ice block.

Because the starting position matters, it is important to clarify how we position the ice blocks in our experiments. Ice blocks will always start in the top left corner of the grid, placed from left to right along the upper edge of the ice floor. An example of this is illustrated in Figure 7 on the left. There are cases where this is not possible, for example a 3×3 board with 4 blocks. We have decided to omit these cases from the results, since they provide no meaningful insights. This means that in general we will not consider cases where the number of ice blocks is greater than the width of the board.

The choice in starting position does affect the acquired results. Naturally, some tiles will require more moves to reach when starting from a different position. When the ice blocks start in the top left corner, it takes more moves to reach the tiles on the right. This causes our tables to be a bit skewed to the right when looking at the hardest tile to reach. We considered starting the ice blocks in the corners of the board, however, that would cause ambiguity when there are more than four ice blocks. Therefore we decided the top row, left to right approach is the best.

Keeping these considerations about the initial ice block positions in mind, this section contains specific ice block movement patterns and specific puzzle configurations we investigated, like trains, square boards, boards of height two and height three, and attempts at creating images using ice blocks. From here on, a move will mean an ice block being pushed.

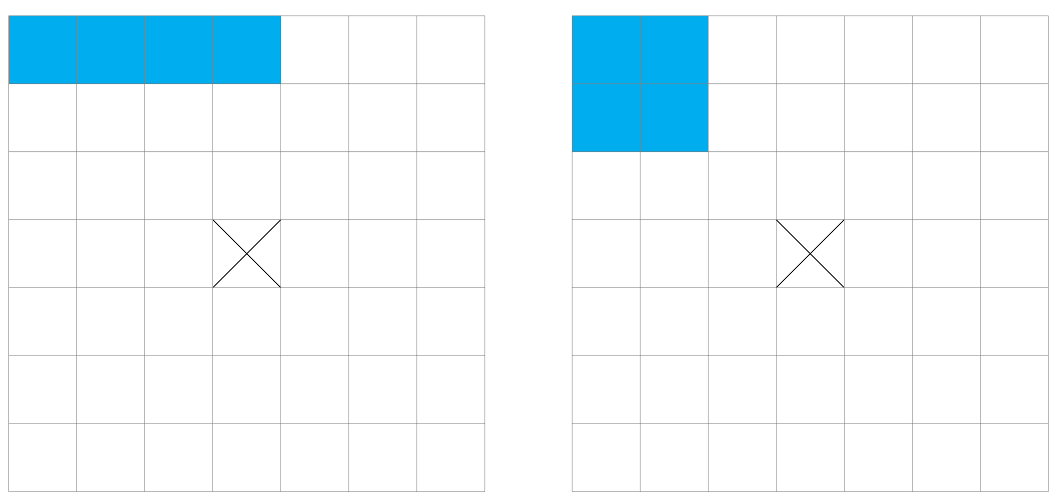

Figure 7: The configuration on the left is solvable, but the configuration on the right is unsolvable.

4.1 Trains

Kluiver discusses a certain type of puzzle always being solvable [Klu20]. These types of puzzles contain three ice blocks, have a rectangular board, and one target in any position. An example of such a puzzle can be seen in Figure 8. The way to solve this puzzle is by using what Kluiver calls "trains". We will also refer to this movement as trains from now on. The train uses four moves to move one tile forwards or backwards. In this case, to move to the right, the leftmost block is moved as can be seen in Figure 8 on the left. This looping motion to move the blocks, or the row of blocks themselves, is called a "train".

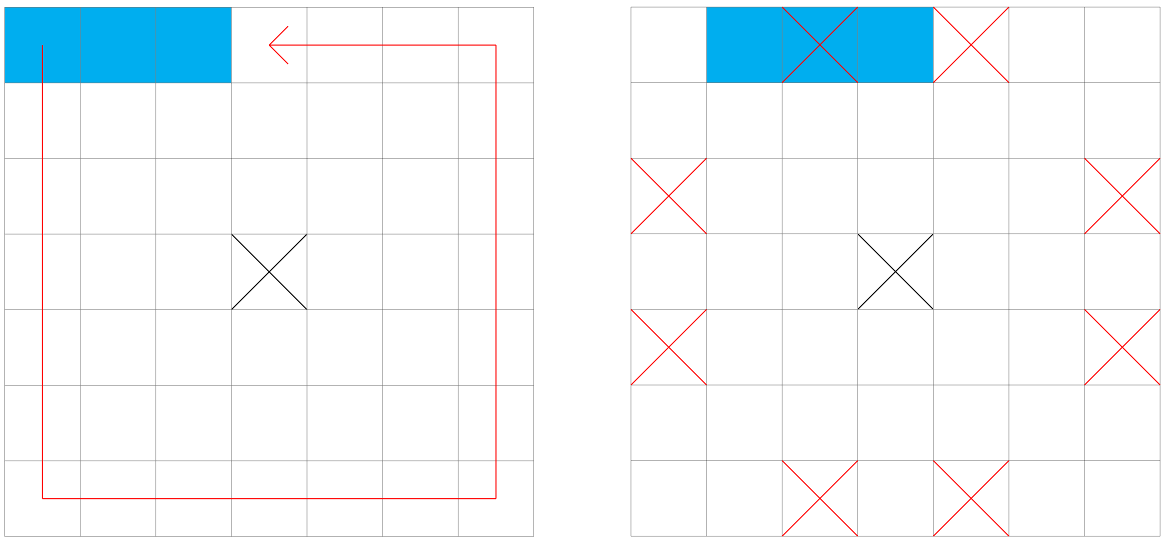

Figure 8: Train start.

In order for the target to be reachable, a block needs to be placed one row or column next to it, stuck to the outer edge. In Figure 8 these tiles are highlighted with a red cross. Using the train, we can move to one of these tiles, and then start setting up the second loop. In this case we already have a block on one of the necessary tiles. The block on the red cross will be used to allow a train running through the target.

The next step is to run the train through the target. The start of this second loop can be seen in Figure 9 on the left. Using the block marked with a red cross as an anchor, we can start looping the other two blocks downwards, to eventually have an ice block standing still on the target in Figure 10. The setup for this train is completed in Figure 9 on the right.
The last step to solving this puzzle using trains is to keep looping the bottom block around as can be seen in Figure 10. Every cycle of four moves, the train moves one tile. This means only two cycles are needed to solve the puzzle, reaching the final solution in Figure 10.

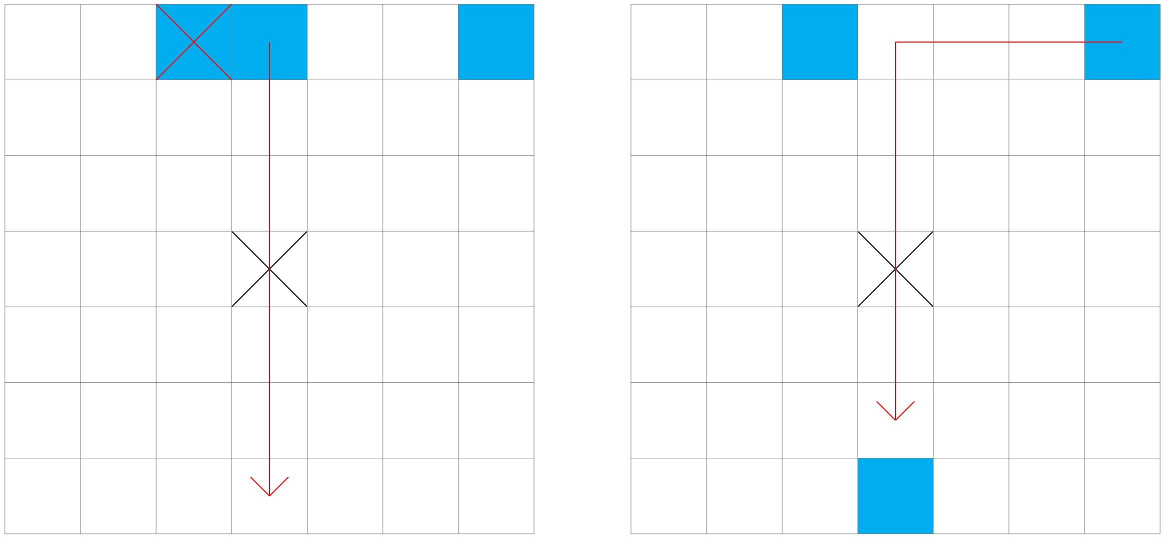

Figure 9: Train step two.

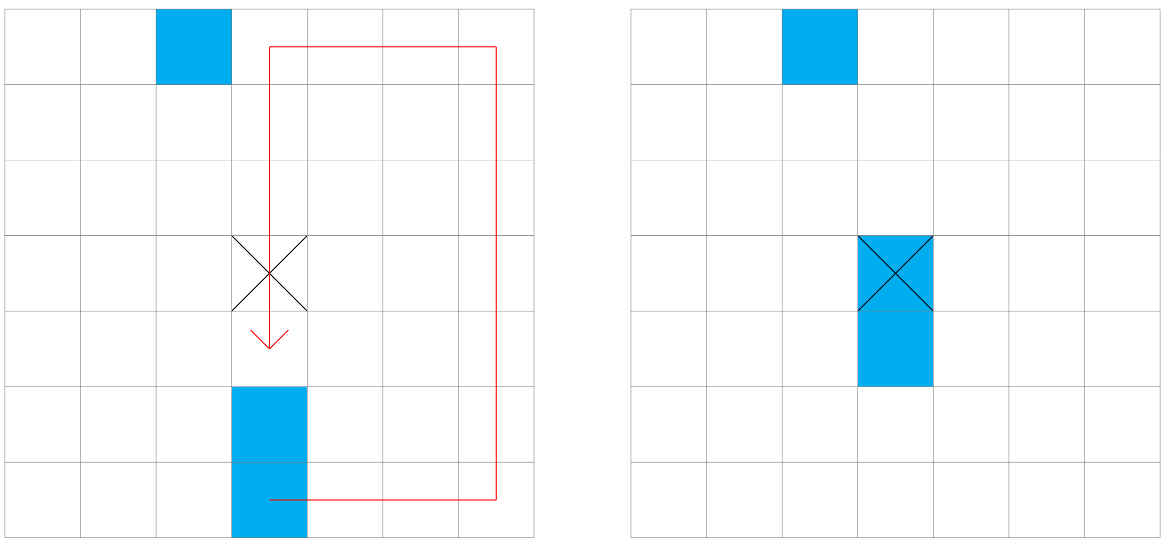

Figure 10: Train solution.

4.2 Square boards

In the previous section an algorithm is described that can be used to solve any $n \times n$ puzzle that contains one target and three ice blocks. While the trains do work to solve the puzzle, we wanted to test if trains are also the fastest way to reach a solution. A bruteforce algorithm is used to determine the minimal number of moves to reach a tile. This is done for every tile on the board. The hardest tiles are then the tiles with the largest minimal number of moves. Tables 1 through 8 show the minimal number of moves needed to reach any tile on the board. The hardest tiles are coloured green. Tiles that become harder to reach with more ice blocks are coloured red.

0	0	0
3	3	3
1	1	1

Table 1: Boards of size 3×3 with 3 blocks. The highest numbers are highlighted green, any tiles that are harder to reach with more blocks are highlighted red.

0	0	0	2
3	3	5	4
3	3	5	4
1	1	1	2

0	0	0	0
3	3	3	3
3	3	3	3
1	1	1	1

Table 2: Boards of size 4×4 with 3 and 4 blocks.

0	0	0	3	2
3	3	5	6	4
5	7	9	10	6
3	3	5	6	4
1	1	1	4	2

0	0	0	0	2
3	3	3	5	4
5	7	7	8	6
3	3	3	5	4
1	1	1	1	2

0	0	0	0	0
3	3	4	3	3
5	6	6	6	5
3	3	4	3	3
1	1	1	1	1

Table 3: Boards of size 5×5 with 3, 4 and 5 blocks.

0	0	0	4	3	2
3	3	5	8	6	4
6	8	9	13	10	7
5	7	10	12	10	6
3	3	5	8	6	4
1	1	1	4	4	2

0	0	0	0	4	2
3	3	3	5	7	4
6	7	7	9	9	7
5	7	7	9	9	6
3	3	3	5	7	4
1	1	1	1	4	2

0	0	0	0	0	2
3	3	4	3	5	4
6	7	7	7	8	7
5	6	6	6	8	6
3	3	4	3	5	4
1	1	1	1	1	2

Table 4: Boards of size 6×6 with 3, 4 and 5 blocks.

0	0	0	4	5	3	2
3	3	5	8	9	6	4
6	8	9	13	14	10	7
8	11	13	16	17	13	9
5	7	10	12	13	10	6
3	3	5	8	9	6	4
1	1	1	4	5	4	2

0	0	0	0	4	4	2
3	3	3	5	8	7	4
6	7	7	9	12	9	7
7	11	11	13	15	12	8
5	7	7	9	11	9	6
3	3	3	5	8	7	4
1	1	1	1	4	4	2

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

Table 5: Boards of size 7×7 with 3, 4 and 5 blocks.

0	0	0	4	7	5	3	2
3	3	5	8	11	9	6	4
6	8	9	13	15	14	10	7
9	12	13	17	19	18	14	10
8	11	14	16	20	17	13	9
5	7	10	12	16	13	10	6
3	3	5	8	11	9	6	4
1	1	1	4	7	5	4	2

0	0	0	0	4	6	4	2
3	3	3	5	8	9	7	4
6	7	7	9	13	13	9	7
9	11	11	13	17	16	13	10
7	11	11	13	15	15	12	8
5	7	7	9	11	12	9	6
3	3	3	5	8	9	7	4
1	1	1	1	4	6	4	2

Table 6: Boards of size 8×8 with 3 and 4 blocks.

0	0	0	4	7	8	5	3	2
3	3	5	8	11	12	9	6	4
6	8	9	13	15	16	14	10	7
9	12	13	17	19	20	18	14	10
12	15	17	20	23	24	20	16	12
8	11	14	16	20	21	17	13	9
5	7	10	12	16	17	13	10	6
3	3	5	8	11	12	9	6	4
1	1	1	4	7	8	5	4	2

0	0	0	0	4	7	6	4	2
3	3	3	5	8	11	9	7	4
6	7	7	9	13	15	13	9	7
9	11	11	13	17	19	16	13	10
10	14	15	17	19	22	18	15	11
7	11	11	13	15	19	15	12	8
5	7	7	9	11	15	12	9	6
3	3	3	5	8	11	9	7	4
1	1	1	1	4	7	6	4	2

Table 7: Boards of size 9×9 with 3 and 4 blocks.

0	0	0	4	7	10	8	5	3	2
3	3	5	8	11	14	12	9	6	4
6	8	9	13	15	19	16	14	10	7
9	12	13	17	19	23	20	18	14	10
13	16	17	21	23	27	24	21	17	13
12	15	18	20	24	26	24	20	16	12
8	11	14	16	20	22	21	17	13	9
5	7	10	12	16	18	17	13	10	6
3	3	5	8	11	14	12	9	6	4
1	1	1	4	7	10	8	5	4	2

0	0	0	0	4	7	7	6	4	2
3	3	3	5	8	11	11	9	7	4
6	7	7	9	13	15	15	13	9	7
9	11	11	13	17	19	19	16	13	10
12	15	15	17	20	23	23	20	17	13
10	14	15	17	19	23	22	18	15	11
7	11	11	13	15	19	19	15	12	8
5	7	7	9	11	15	15	12	9	6
3	3	3	5	8	11	11	9	7	4
1	1	1	1	4	7	7	6	4	2

Table 8: Boards of size 10×10 with 3 and 4 blocks.

	3 blocks	4 blocks	5 blocks
3×3	No		
4×4	No	No	
5×5	No	Yes	No
6×6	Yes	Yes	No
7×7	Yes	No	No
8×8	Yes	No	No
9×9	Yes	No	-
10×10	Yes	No	-

Table 9: Are trains part of the quickest solution?

The bruteforce algorithm mentioned previously remembers the fastest solutions, which were then used to determine whether or not a train was used. A cell in Table 9 reads "Yes", if a train was used in a fastest solution for the hardest to reach tiles. If not, the cell reads "No". Some results are left out because of a board size that is too small to fit the ice blocks in their starting positions, and some contain just a '-', because the results took too long to compute. Using the results in Tables 1 through 8 we can calculate the results shown in Tables 10 and 11. Table 10 shows the number of moves needed to reach the hardest tile. Table 11 shows the average number of moves necessary to reach a tile.

	3 blocks	4 blocks	5 blocks
3×3	3		
4×4	5	3	
5×5	10	8	6
6×6	13	9	8
7×7	17	15	12
8×8	20	17	-
9×9	24	22	-
10×10	27	23	-

Table 10: Largest number of moves to reach a tile per board.

	3 blocks	4 blocks	5 blocks
3×3	1.3		
4×4	2.3	1.8	
5×5	3.7	3.1	2.6
6×6	5.1	4.3	3.7
7×7	6.6	5.7	5.1
8×8	8.2	7.0	-
9×9	9.9	8.6	-
10×10	11.5	10.0	-

Table 11: Average moves per board.

An interesting observation is that in general, an additional block speeds up the solution. This can be seen by comparing the hardest tiles in the same row in Table 10, or by looking at the average number of moves necessary to reach a tile, as seen in Table 11. When comparing the number of moves necessary to reach a tile with one more block, some tiles become harder to reach. These tiles that become harder to reach with more blocks are coloured red. This observation can be explained by the blocking behaviour these ice blocks can have on each other. A simple example of this can be seen in Figure 11. Adding an extra block in this situation means one of the two blocks needs to be moved out of the way in order to reach the target, which results in an extra move being used.

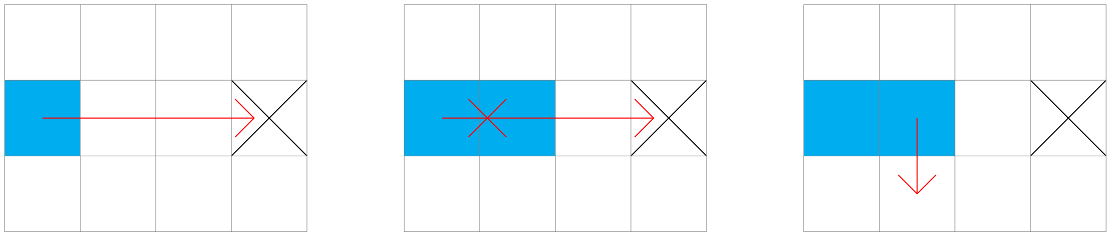

Figure 11: Introducing more blocks can increase the length of the solution.

When researching these different board sizes and their quickest solutions, we made some interesting observations. Firstly, instead of, or in addition to, using the trains, a new method of further moving the group of blocks was seen. We will call this movement pattern "zigzag". Figure 12 shows the setup, Figure 13 shows what one cycle of the zigzag looks like, and in Figure 14 the result and a further cycle is illustrated. As can be seen, the zigzag only requires three moves, rather than the train's four, to move the whole formation one tile to the right. The zigzag is therefore very useful to bypass the first step of the train. The zigzag does have a weakness however, it requires constant bouncing between two edges. Tiles that are further from the edges, in other words, tiles that have a distance from the edges of at least one, are impossible to reach with the zigzag. Therefore the zigzag can not be used to reach the middle of the board. In solutions where the zigzag is used, the target is often still reached using the second step of a train. Secondly, in some cases with three blocks, the train is not used, but when a fourth block is added, we do see train usage.

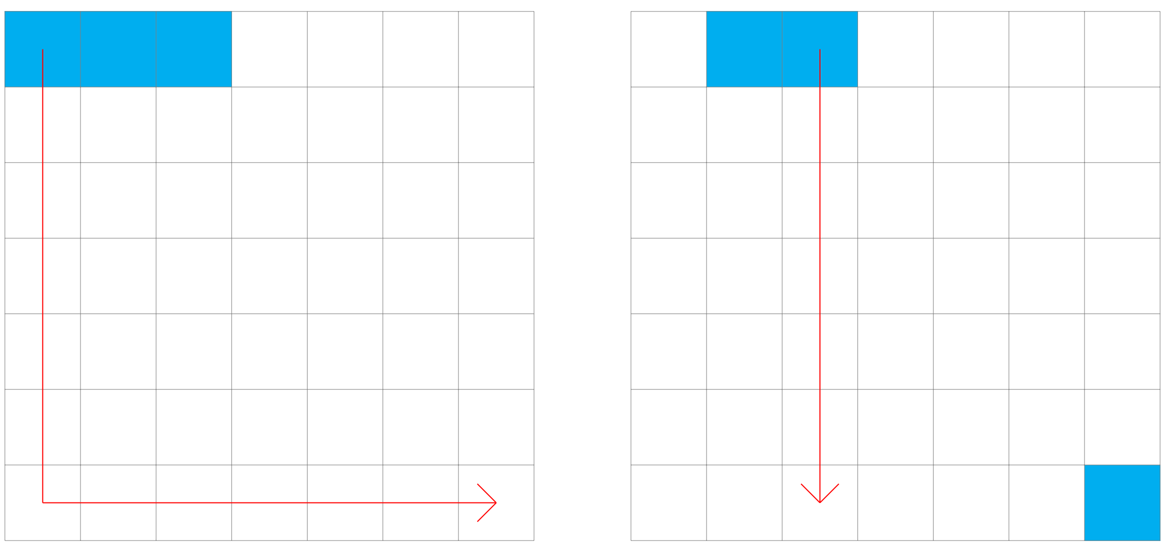

Figure 12: Zigzag setup, requiring 3 moves.

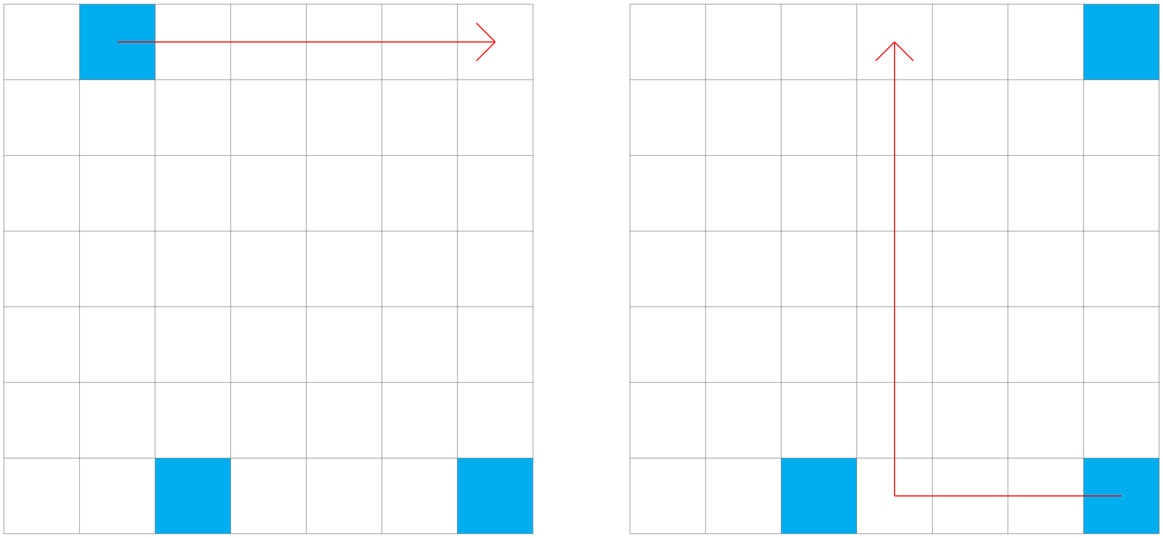

Figure 13: Zigzag cycle, with 3 moves.

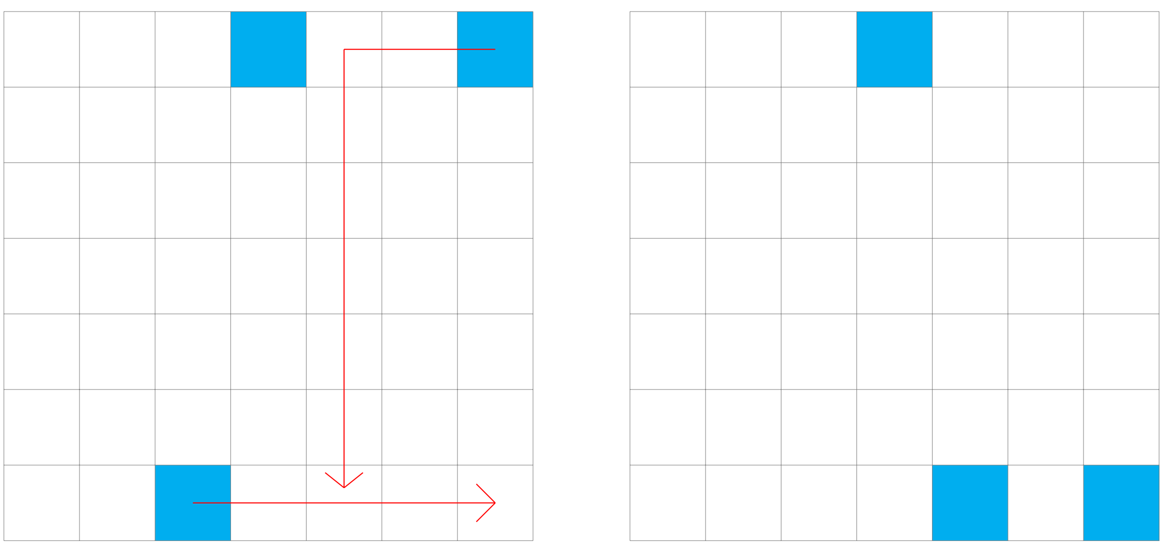

Figure 14: Zigzag result after one cycle (left), and one cycle further (right).

4.2.1 Four block solutions

When investigating the 7×7 board with four blocks, an interesting new fastest solution was found. This solution did not make use of trains, or zigzags. Rather, the solution made use of conveniently placed blocks to efficiently reach the target, and can be used for other targets as well. This solution can be seen in the images in Figure 15. This specific solution may no longer be possible on bigger boards, where the train or zigzag will have to be used again to get to the row or column necessary to reach the target.

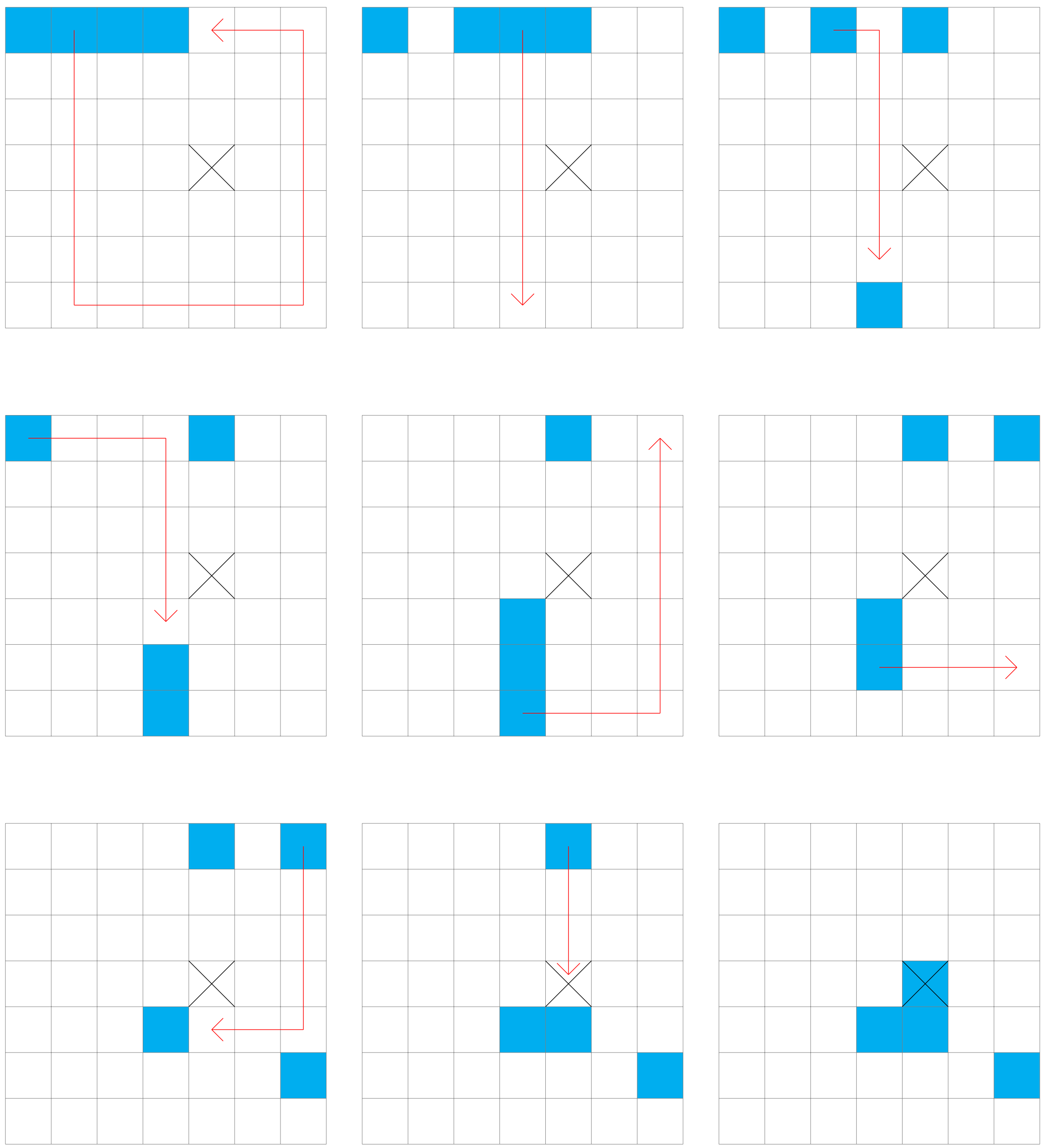

Figure 15: Interesting four block solution, with 15 moves.

4.3 Boards with height 2 and 3

In general, boards of height two can be represented as a $2 \times n$ board. We experimented with boards containing three ice blocks, and varying values for n. The number of moves the hardest tile took can be seen in Table 12. To reach all fastest solutions, the zigzag movement is used. Starting from a width of six, a pattern starts to emerge, that continues even after the values shown in Table 12. The number of moves compared to the previous value of n goes up by a set amount. When n is even, the number of moves increases by two, and when n is odd, the number of moves increases by one. This means the number of moves $moves(n)$ for the hardest tile on a $2 \times n$ board using three ice blocks can be calculated by:

$$moves(n) = n + \lfloor n/2 \rfloor - 5, \; n \geq 6$$

To determine whether adding ice blocks has a significant effect on the number of moves required to reach the hardest tile, we used a 2×15 board and varied the number of ice blocks. We chose an arbitrary value for n that would be at least twice the number of blocks. The results are shown for $n = 15$ in Table 13. We found that the decrease in moves depends on whether n is even or odd, but the maximum decrease is always three. This decrease can be explained by the fact that one more ice block means starting one tile further right. This tile means one fewer cycle of the zigzag is needed, which corresponds to three moves less. Overall this means no significant increase in speed when adding extra blocks is observed, other than the obvious.

n	height 2	height 3
3	1	4
4	2	5
5	4	6
6	4	8
7	5	9
8	7	11
9	8	12
10	10	14

Table 12: Hardest tiles with 3 ice blocks on a $2 \times n$ and $3 \times n$ board.

ice blocks	height 2	height 3
3	17	21
4 ice blocks	16	20
5 ice blocks	13	17
6 ice blocks	10	-

Table 13: Moves for different numbers of ice blocks on a 2×15 board.

We found that $3 \times n$ boards are very similar to $2 \times n$ boards. Tables 12 and 13 show the results for $3 \times n$ boards as well. The fastest solutions found use the zigzag to reach every tile, and increasing the number of ice blocks has no significant effect on the number of moves needed to find solutions. A 3×15 board with 6 ice blocks proved too computationally expensive. The number of moves required to reach the hardest tiles show a pattern, just like with boards of height two, but this pattern is found earlier, at $n > 3$. The following formula can be used to calculate the number of moves $moves(n)$ needed to reach the hardest tile on a board with height three and width n when using three ice blocks:

$$moves(n) = n + \lfloor n/2 \rfloor - 1, \; n \geq 4$$

5 Image creation using ice blocks

On a square board, we know that any tile on the board is reachable when given at least three ice blocks. Since it is possible to place blocks in specific positions on the board, this can be used to create images. An example of such an image can be seen in Figure 16. Attempts at creating images in games have been researched before, like in Tetris for example [HK05]. For now we will limit the images to consist of two colours, since a tile in the image can either contain an ice block or just be ice floor. This could be generalised by having ice blocks in different colours. In this section we discuss the possibilities and limitations of ice block puzzle images.

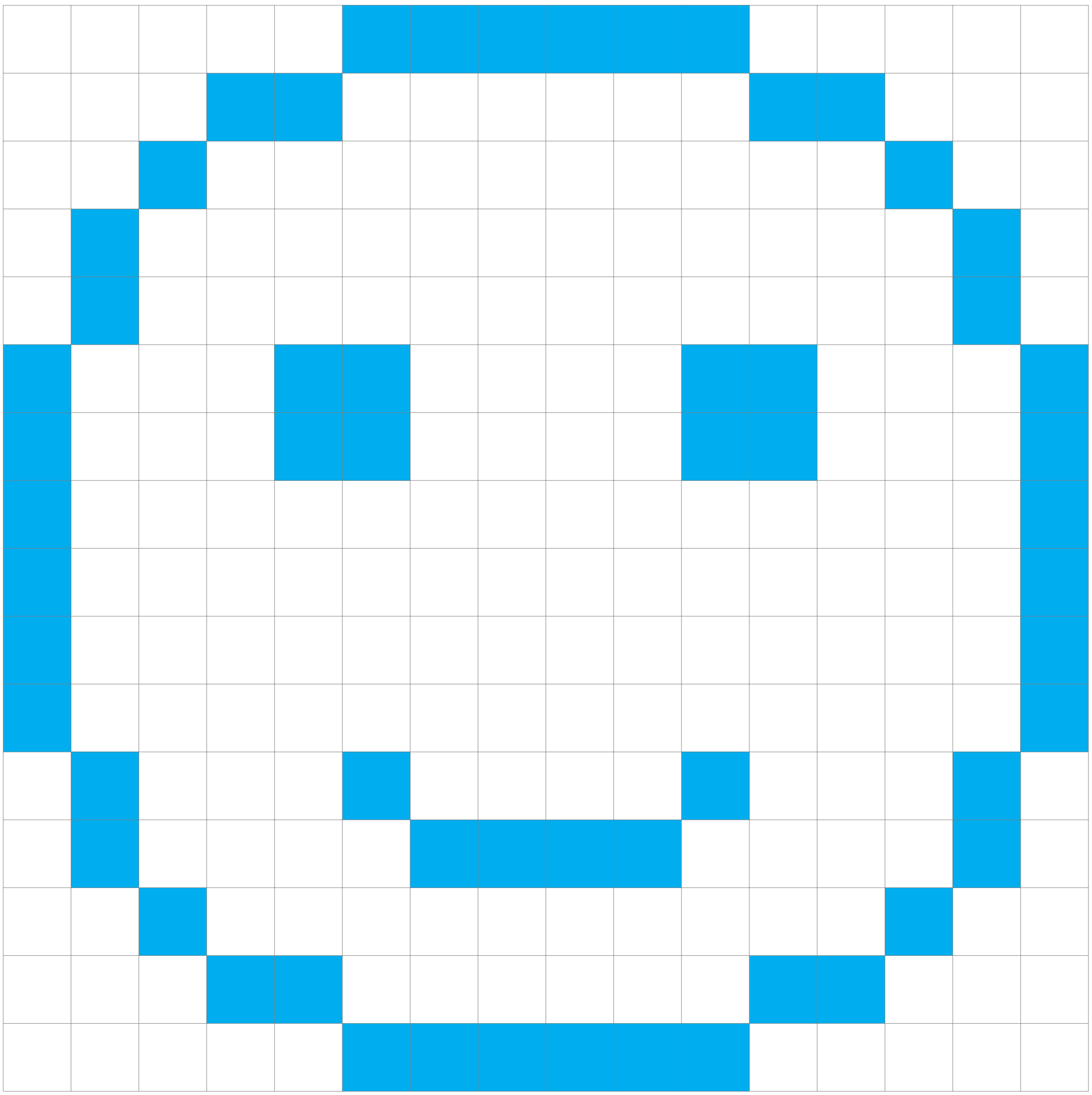

Figure 16: Image with 58 blue pixels created using ice blocks.

5.1 Long train

The first option of creating images using ice block puzzles we considered, was to have a long train loop through the entire board, and leave ice blocks behind on every target required to create the image. For an image with n targets, $n+2$ ice blocks are needed, because the train requires two extra blocks. To create Figure 16, a total of 60 ice blocks would be needed. The image itself has size 16×16, but in order for the train to fit on the board, a board of at least 60 tiles wide would be needed. Therefore, although having a long train loop through the board and leaving one of its "wagons" on each of the targets seems like a good way to generate images, it is not very practical, and most images are even impossible to create this way, for reasons we will explain in Section 5.2. This option has therefore been discarded as a viable option to create images.

5.2 Short train

The second option uses a train consisting of three ice blocks on the board. This train places an ice block on a target, after which a new ice block is provided, which then forms a new train with the remaining two ice blocks. This way, the problem of creating an image can be solved using the following algorithm:

1. Convert all pixel locations into a list of targets.
2. Take one target from the list.
3. Place an ice block on that target.
4. Consider that ice block to be a wall from now on.
5. Receive a new ice block.
6. Repeat from step 2 until the list of targets is empty.

Kluiver has shown that any tile is reachable on a square board using three ice blocks [Klu20]. However, this method only works on otherwise empty boards. This means the middle of the board contains no walls or normal ground. In our image creation process, we add walls to the middle of the board. This opens up options for placing ice blocks but also provides some limitations when trying to reach certain targets. Because of this, the order in which we place the targets in our list matters. A simple example of this can be seen in Figure 17. The left image is the image to be created, and the image on the right shows that trying to place an ice block on the middle tile as the last target is impossible. But, if the middle tile is drawn first, the image can be created.

Placing blocks on targets can now be done in two ways. The first is to run a train through the target, just like described in Section 4.1. This method allows us to place a block in the middle of the board, with the following limitations: Firstly, there needs to be a line directly to the target, from one of the sides of the board, unobstructed by walls. Secondly, in order for the train movement to be possible, the currently moving block must be retrievable throughout the entire train loop. A block is retrievable when the block is able to reach one of the sides of the board, without moving the other blocks, which allows the train loop to continue.

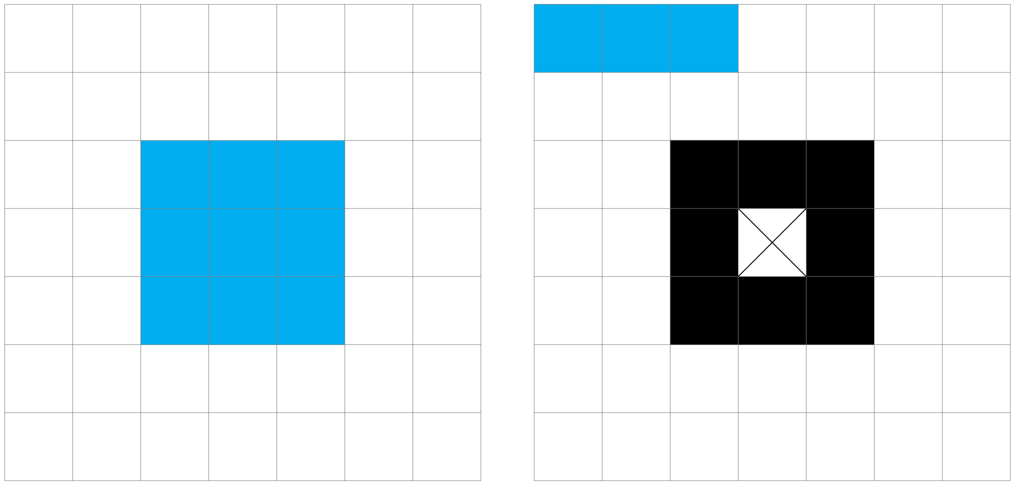

Figure 17: Left image is the goal, right image shows a target that is not reachable.

Figure 18 shows examples in which the back block, depicted in green for clarity, is not retrievable. In the image on the left, the green block is not movable at all without moving the block below it, and in the second image moving the block would result in the block getting stuck permanently.

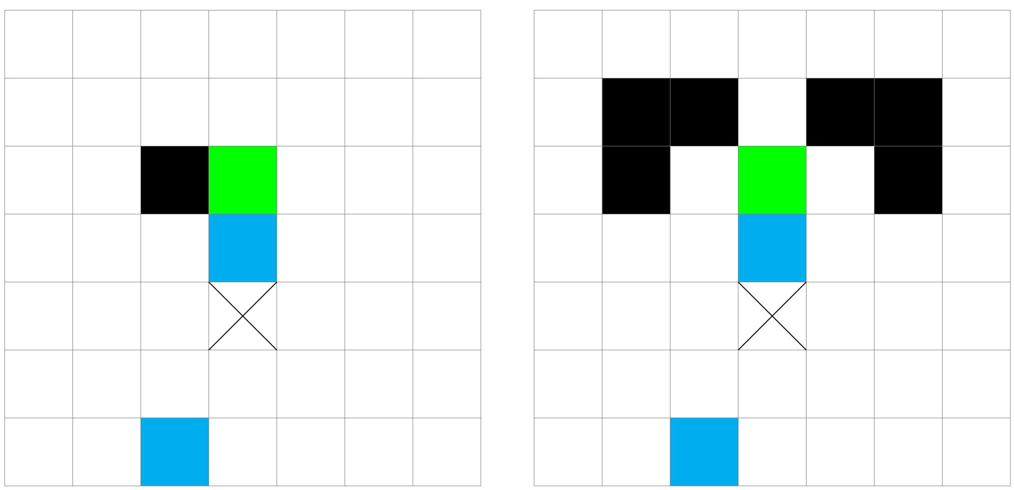

Figure 18: Irretrievable green blocks.

The second way of placing blocks is by stacking them onto walls already present in the middle of the board. This method could also be described as using only the setup of a train, so it can be seen as a part of the first method, the train. This method only requires two of the three ice blocks, and can be used to "shoot" ice blocks from the walls, to the middle of the board. The obvious limitation this method has is that it requires a wall, which is a block placed in previous iterations of our image creation algorithm, to be present already, so that the new block can be shot against it.

To facilitate the creation of images, remove certain edge cases, and allow new ice blocks to be provided, we use a slightly modified board. In order to create an image of size $m \times n$, we use a board consisting of a rectangle of size $(m + 4) \times (n + 4)$, with a "Block feeding tube" leading to a storage area at the top. This storage area can be resized depending on the number of ice blocks needed for the image to be created. An example for $m = n = 8$, with 14 ice blocks is shown in Figure 19. The drawing area in this image consists of the red square. When a block is placed, it is considered a wall and not moved anymore. The first short train is already present, and after every block has been placed, a new block can be pushed down from the storage area, through the gate.

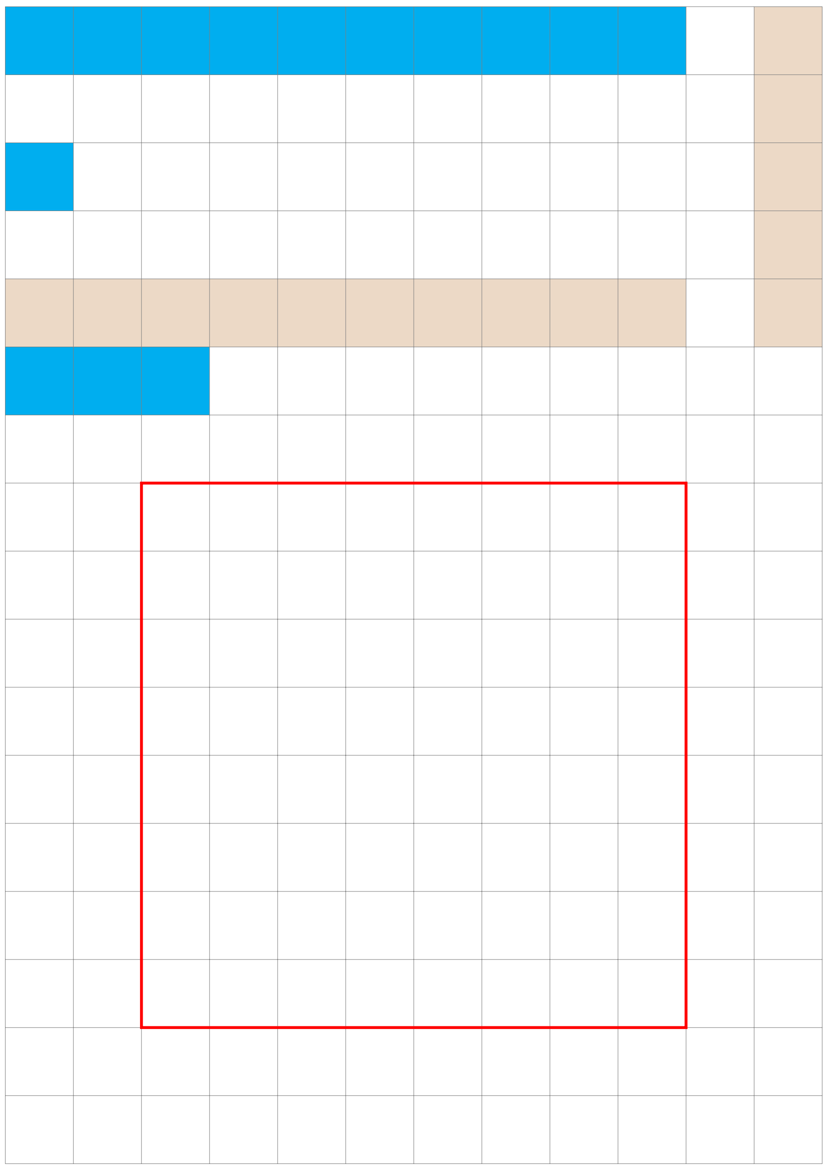

Figure 19: Starting position of an image creation board for an 8×8 board with 14 ice blocks. The short train is already present, and the upper left part shows the storage from which new ice blocks can be pushed down.

Drawing an image in an ice block puzzle is not a trivial problem. It cannot be solved by simply filling in every target from top left to bottom right. The order matters, and the direction from which a target is approached matters as well. Direction approached here means the direction where the anchor block for the train is, not the direction the train is moving in. Some images are impossible to create due to the limitations provided by ice block movement. Figure 20 shows one of the possible solutions to draw the smiley shown in Figure 16, with the order and direction for each block placed, to prove it is indeed possible to draw this image in an ice block puzzle using the short train option.

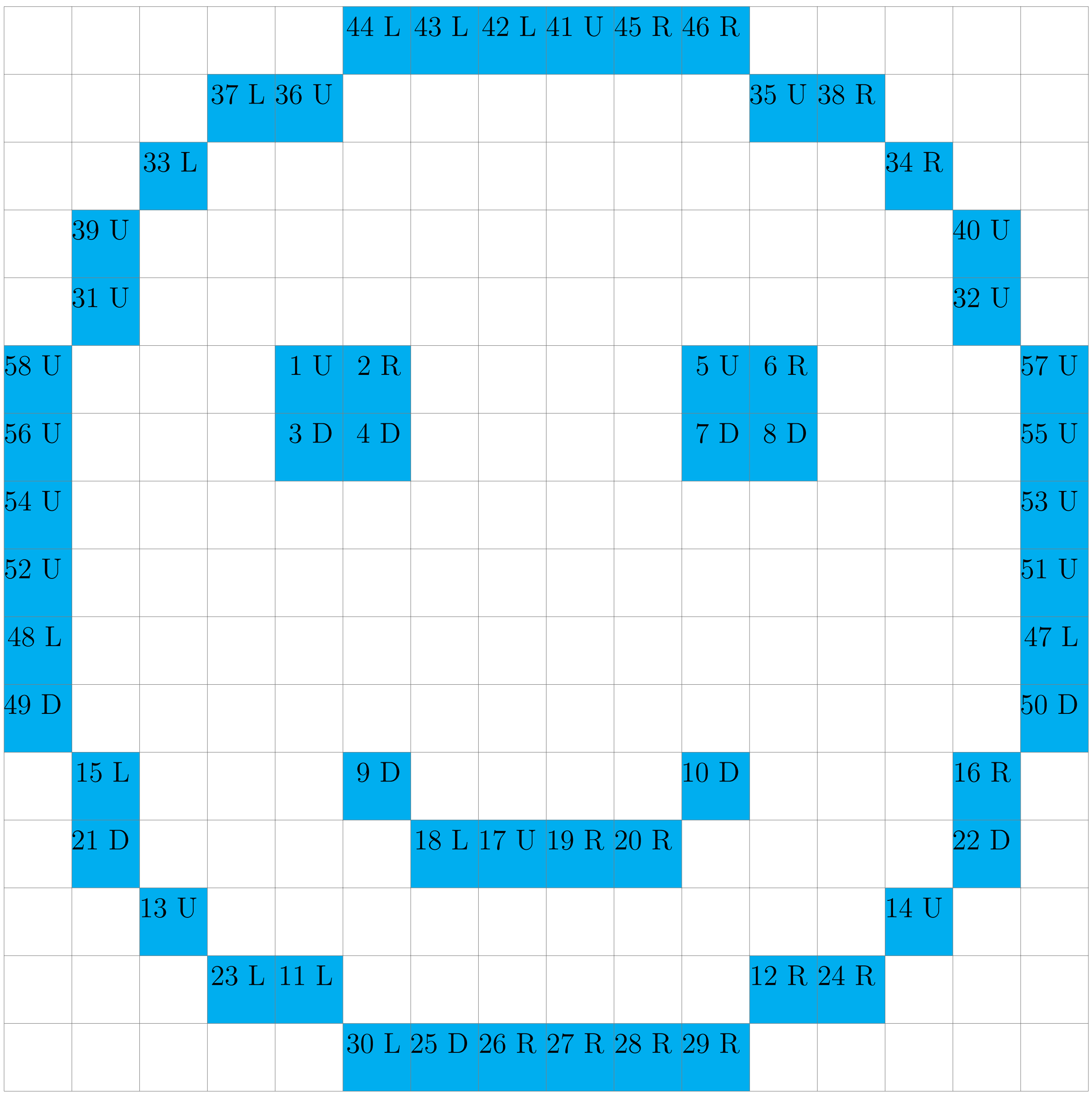

Figure 20: Smiley image solution. The numbers indicate the order in which the blocks get placed, and the letters indicate the direction from which the blocks are pushed into the image as follows. U = Up, R = Right, D = Down, L = Left.

The given solution begins with the eyes and upper corners of the mouth. Then we position a few blocks that are part of the lower half of the face. This is done now because placing blocks that only have diagonal connections to other blocks requires the train method. The train method requires more room to be empty, which is why these blocks need to be placed first. After this we place one block for the mouth, and stack the rest against it using the openings left in the outer ring. After the mouth and eyes have been placed, the rest of the ring can be filled in using trains to place the first block, then stacking blocks against it where possible. This solution was found by hand, using trial and error to determine the order in which the blocks need to be placed.

5.3 Possible configurations

It is possible to create any 1×1 image, and any 2×2 black and white image using our algorithm as described in section 5.2. However, due to the limits of our two methods of positioning blocks, two configurations that are impossible to create with our algorithm can be found when looking at 3×3 images. These configurations can be seen in Figure 21. It is important to notice that the empty spaces are part of the configurations, placing a block on any of these tiles would result in a configuration that is possible to create.

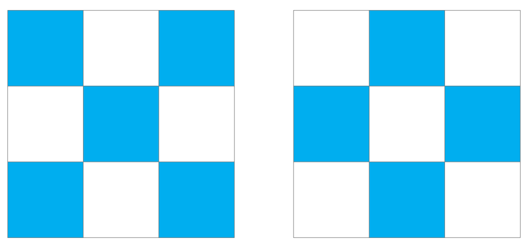

Figure 21: Impossible configurations on a 3×3 board.

Due to the nature of our algorithm, the following statement is always true: An image is creatable if and only if there exists an image that consists of one ice block less, and there is a way to place the last block in that smaller image on the target tile. Using this property of our algorithm, it becomes easy to prove that the configurations in Figure 21 are impossible to create. We can remove one of the blocks in the configuration, and show that it is impossible to get the last block on the target. Due to the symmetric nature of these configurations, we only have three possibilities to check for these images, that can be seen in Figure 22.

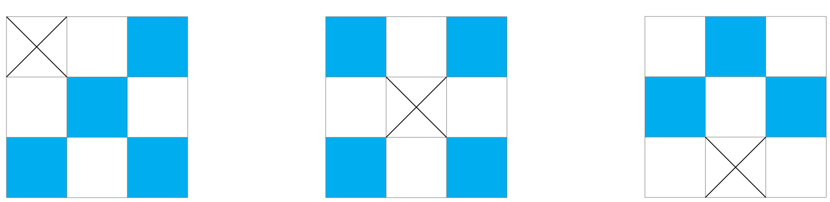

Figure 22: Smaller configurations, with a target where a block should be placed.

We have two ways of placing an ice block, but we can rule out the stacking method, since in all three cases, the target has no adjacent ice blocks. The only possibility left would be to use a train to reach the tiles. Figure 23 shows the positions (in red) where an irretrievable ice block would be placed as a result of trying the train method. One of these blocks will inevitably be placed down and become irretrievable, depending on what direction the train method is started from. These positions are shown for all possible directions a train could approach the target from. When there is an ice block in the way, like in the leftmost configuration moving upwards for example, this train will not be considered, and is therefore not shown in the image.

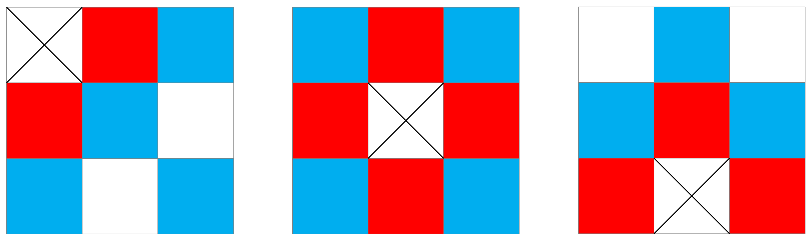

Figure 23: Irretrievable block positions, shown in red.

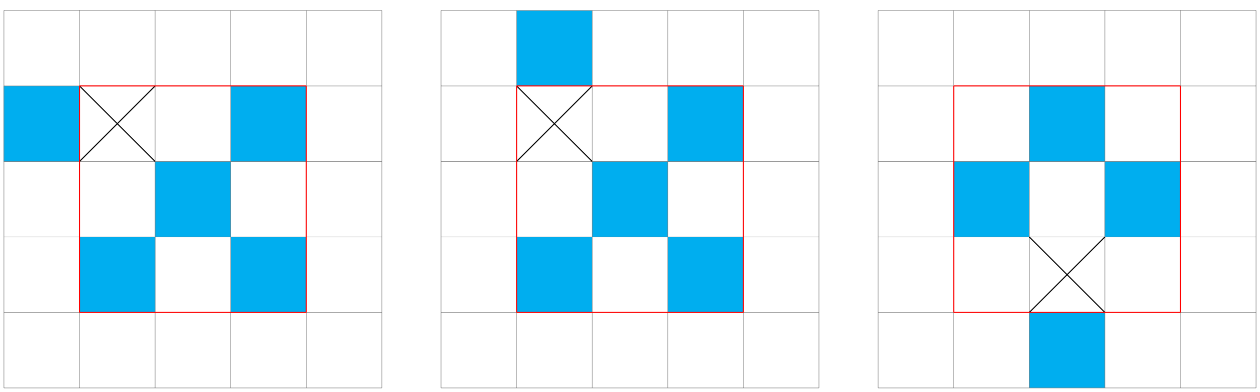

Figure 24: Bigger images containing impossible configurations.

We have shown that the configurations in Figure 21 are impossible to create in an ice block puzzle, because there is no method that can be used to place the last ice block. However, we have only proven this is the case for 3×3 images. In general, any image bigger than 3×3 that contains one of the configurations in Figure 21, is impossible to create in an ice block puzzle. We have already shown that any attempt to reach the necessary target tiles using the train method is impossible without leaving irretrievable blocks in Figure 23. The only other way to reach the targets would be through the stacking method, which requires a directly adjacent block. This converts the images in Figure 22 to the corresponding 5×5 possibilities with a block directly adjacent to the target, as seen in Figure 24. In this figure, the original configuration is enclosed in a red border. In all three images it is impossible to stack a block against the ice block that is positioned outside the red border, because the path there is already blocked by an ice block that is part of the configuration. The middle image seen in Figure 22 is left out, since any block placed adjacent to the target would mean a change to the configuration, which means it is no longer impossible to create.

Since there are no other methods of placing an ice block, we have proven that any image bigger than 3×3, containing one of the configurations in Figure 21, is impossible to create in an ice block puzzle. However, we have not been able to show that these are the only subconfigurations that cause an image to be noncreatable.

6 Conclusion and further research

In this thesis we discussed ice block puzzles with a specific ruleset. We looked at different movement patterns used to solve these puzzles, such as the “trains”, and “zigzag”.

We calculated the number of moves needed to reach any tile on a number of different sizes of square boards, and found that using an increasing number of ice blocks led to some interesting new methods of reaching a fastest solution.

We researched boards of height two, and height three, and we came up with formulas to calculate the number of moves needed to reach the hardest tile, using the width as a parameter.

Lastly we discussed using ice block puzzles to create black and white pixel images. We found that not every image is creatable, and proved that any image containing specific configurations is impossible to create in an ice block puzzle.

Future work

As we have seen, increasing the number of ice blocks speeds the fastest solutions up in most cases. It also introduces new interesting movement patterns in that fastest solution, that were not possible or at least not as fast with fewer ice blocks. Perhaps increasing the number of ice blocks even further can provide even more different patterns.

As of now, the algorithm to create an image in an ice block puzzle consists largely of trial and error. There may be a way to determine whether or not an image is creatable just by looking at the image, or by use of a recursive program. When we know an image is creatable, there may be a systematic algorithm that always works, like starting in the middle and working our way outwards, or perhaps working from left to right, top to bottom.

We have proven that any image containing specific configurations is not creatable in an ice block puzzle. Perhaps it is also true that any image that is not creatable must contain one of these two configurations, which we have not proven. If this were the case, determining whether or not an image is creatable would become a matter of checking the image for these configurations, which would greatly simplify the problem.

In this thesis, we have considered ice block puzzles with a specific set of rules. There are, however, many more possibilities, like allowing diagonal movement of blocks for example, or having the board be three-dimensional. One could even consider puzzles that include new tile types, like a portal tile that transports an ice block to a different tile. It would be interesting to see what effect the ruleset chosen has on many properties, like the solvability of ice block puzzles.

References

[Cul97] Joseph Culberson. Sokoban is PSPACE-complete. Technical Report TR 97-02, Department of Computer Science, University of Alberta, 1997.

[DDF+18] Paul Dorbec, Eric Duchêne, André Fabbri, Julien Moncel, Aline Parreau, and Eric Sopena. Ice Sliding Games. *International Journal of Game Theory*, 47 (2):487–508, 2018.

[DDO00] Erik D. Demaine, Martin L. Demaine, and Joseph O'Rourke. PushPush and Push-1 are NP-hard in 2D. *Proceedings of the 12th Annual Canadian Conference on Computational Geometry*, pages 211–219, 2000.

[Edi16] Luc Edixhoven. Attacking the n-Puzzle Using SAT Solvers. Bachelor Thesis LIACS, Leiden University, 2016.

[HK05] Hendrik Jan Hoogeboom and Walter A. Kosters. The Theory of Tetris. *Nieuwsbrief van de Nederlandse Vereniging voor Theoretische Informatica (Newsletter of the Dutch Organization for Theoretical Computer Science)*, 9:14–21, 2005.

[Klu20] Sem Kluiver. Solving and Generating Ice Block Puzzles. Bachelor Thesis LIACS, Leiden University, 2020.