A handle is enough for a hard game of Pull

Oscar Temprano
oscartemp@hotmail.es

Abstract

We are going to show that a variant of a puzzle called pull were the boxes have handles (i.e. we can only pull the boxes in certain directions) is np-hard

1 Introduction

In [1], Marcus ritt proved that two variants of pushpush [2], called pull and pullpull, were np-hard. He asked about the complexity of a variant of pull in which the boxes have either horizontal (left and right) or vertical (up and down) handles. In sections 2 and 3 of this document we are going to prove that this variant is np hard and we will extend this result to the storage version of pull. Also, we are going to prove that the path version of pull in which the boxes have either down or left handles is np-hard as well. We will do this by reducing satisfiability to these problems. In section 4, we will prove that the path version of pull where all boxes have handles in opposite directions (i.e. left and right handles) is np hard. In section 5, we will prove that the path version of pull in which all boxes can only be pulled in a single direction is np-hard.

2 Description of gadgets

2.1 No return gadget or Limited no return gadget

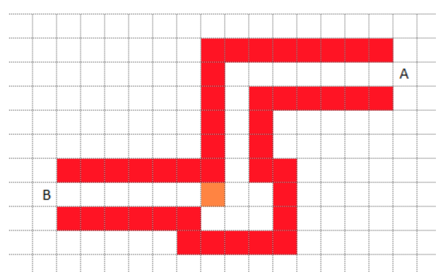

Figure 1: Limited no return gadget.

Depending on the number of handles that the boxes can have, this is a gadget that will behave in different ways:

If we allow left and right handles (and/or top and down handles), this is a gadget that allows us to go to from A to B and then we cannot go from B to A , but we can go from A to B all the times that we want. For these variants we will need another aditional gadget, the no return gadget. Because this gadget can be placed in its original position after traversal, we will use this gadget in the storage versions of sokoban.

If we restrict the handles, so that there is only a handle in one direction(i.e. a down handle and a left handle, but not an up handle or a right handle), then this gadget behaves like a no return gadget, allowing passage from A to B a single time and once we traverse the gadget we cannot return from B to A.

2.2 No return gadget

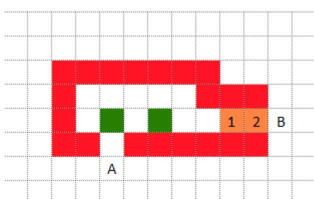

Figure 2: Limited no return gadget (the green squares are places in which the player has to put the boxes in the storage version of pull).

This is a gadget that we will only use for those variants in which the boxes have left and right handles (or/and up and down handles).

This gadget allows passage from A to B, forbids passage from B to A, and can only be traversed once

Lemma 1. *The no return gadget can be traversed from A to B , but after it is traversed, te player cannot go back from B to A.*

Proof. To go to B, the player has to move the boxes 1 and 2 out of the way. In figure 2, the only way to do it is by pulling the blocks to the left (it can be pulled to any other direction if the gadget is rotated). To be able to pull box 2 out of the narrow corridor, the player has to pull the block 1 to the position marked by the leftmost green square, blocking the way to A, the player can then pull the other block to the other green square and exit through B ☐

2.3 Two way gadget

This is an optional gadget that we can use in the path versions of pull to make sure there is no leakage between literals

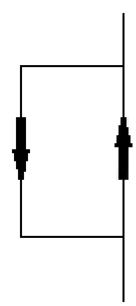

Figure 3: Two way gadget. The arrows represent no return gadgets.

The gadget is formed by two wires, each wire has a no return gadget that has to be traversed in opposite direction to the other. This gadget can only be used two times. The first time to go from a position A to a position B, and the second time to return from B to A. Once the player returns to A, he cannot go to B anymore.

This gagdet can be placed at the entrances and exits of a cossover gadget (to ensure that there is no leakage between wires) and before the player enters a clause from a literal wire.

2.4 Crossover (for path version)

This is a crossover for the path version of pull. If the player enters the gadget A1, he can only exit through A2. And if the player comes from B1 he can only exit through B2.

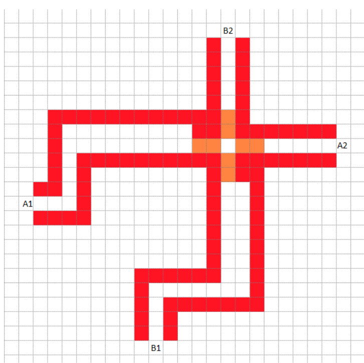

Figure 4: The crossover in its original state

If the player enters the gadget from A1, he can reach A2 by pulling all boxes that are blocking the way to the left.

If the player enters the gadget from B1, he can reach B2 by pulling downwards all boxes that are blocking the way.

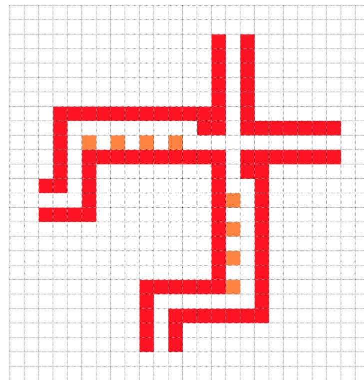

Figure 5: A crossover that has been crossed from both directions

Lemma 2. *In its original state, the crossover gadget can be traversed from A1 to A2 or it can be traversed from B1 to B2. Those are the only possible traversals of the gadget*

Proof. Let´s suppose that the player enters the gadget from A1, the other direction is simmetrical. When the player enters, the only thing he can do is pull the two boxes that he finds at the entrance. From there, he cannot go to B2 or B1 because the player has not enough space to pull down the boxes that are blocking the path to B2, and he has not enough space to pull up the boxes that are blocking the way to B1 □

2.5 Crossover (for storage version)

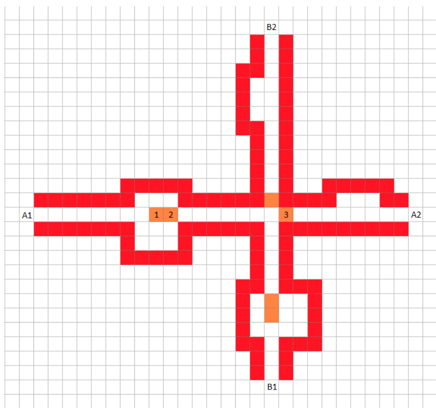

Figure 6: The crossover in its original state

This is a crossover for the storage version of pull. If the player enters the gadget from A1, he can only exit through A2. And if the player comes from B1 he can only exit through B2.

We are now going to describe how to cross the gadget. We will suppose that the player enters the gadget from A1, the other case is symmetrical.

When the player enters, he pulls down the boxes 1 and 2

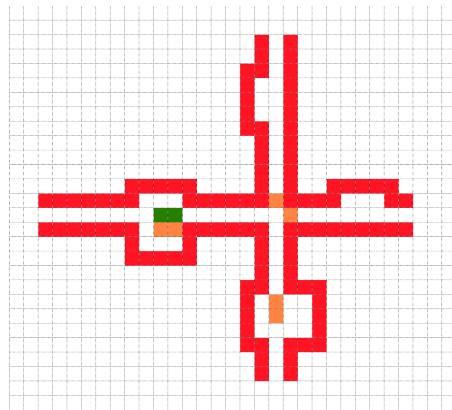

Figure 7: The player has pulled down boxes 1 and 2

After doing that, the player has opened a gap in which he can place box 3

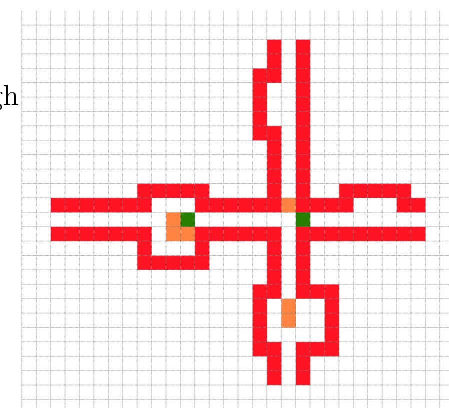

Figure 8: The player has moved box 3 to the left gap

Now, the player can exit through A2. Once he returns, he has to place all boxes in their original positions (marked by green squares in the previous figures).

To do that, the first thing he does

is pull block 3 to the gap at the right:

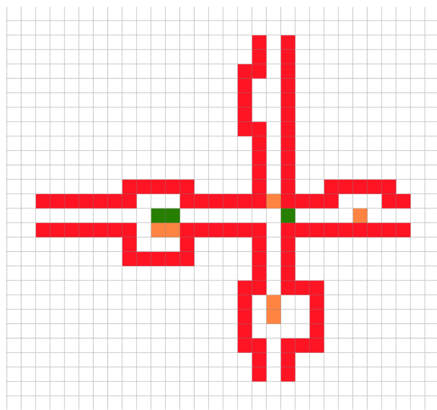

Figure 9: The player has moved box 3 to the right gap

Then he can pull to the left this block until he reaches the righmost green square, its original position.

Then he pulls upwards blocks 2 and 1 to its original positions, after doing this he can return through A1

Lemma 3. *In its original state, the crossover gadget can be traversed from A1 to A2 or it can be traversed from B1 to B2. Those are the only possible traversals of the gadget*

Proof. Let´s suppose that the player enters the gadget from A1, the other direction is simmetrical. When the player enters, the only thing he can do is pull down the two boxes that he finds at the entrance. From there, he cannot pull down the box blocking the way to B2 because he will get trapped between that box and the two boxes below him. He can´t exit through B1 because he can only pull upwards one of the two boxes that are blocking the way to B1 □

2.6 Clause gadget (for path version)

This is a clause gadget for all those path versions of pull where the player can come and return through the same literal wire he came to the gadget.

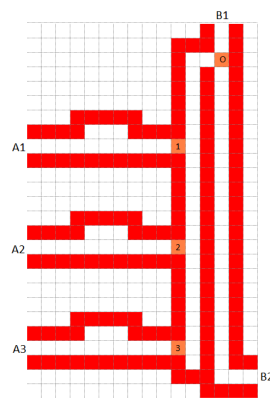

Figure 10: A picture of a clause gadget with three literals, the entrances A1, A2 and A3 correspond to entrances that are connected to a literal wire

The gadget has $n + 1$ entrances where n is the number of literals in the clause.

The gadget has a no return gadget before B1. This is to force the player to exit from B2 if he enters the gadget from B1

Lemma 4. *If the player wants to go from B1 to B2 in figure 9, the player had to access the gadget previously from* *an entrance that belongs to a literal wire*

Proof. When the player enters the gadget, there is a box (with the label O in figure 9) that it´s blocking the way to B2. In order to get this box out of the way, the player has to access the gadget from one of the other entrances (A1, A2 or A3) and pull the box one square, so that the box doesn´t block the coridor anymore. □

Lemma 5. *When the player enters the gadget from an entrance that belongs to a literal wire he can´t exit through a literal wire that hasn´t been visited before*

Proof. When the player enters the gadget, he has to pull a block to enter (These blocks are labeled with numbers in Figure 9). When he enters the gagdget, he cannot exit through a literal wire that hasn´t been visited before because there is a box blocking the way to that wire. The player cannot pull this obstacle out because there isn´t enough space to pull the block. □

2.7 Clause gadget (for storage version)

This is a clause gadget for the storage version of pull

The gadget has n entrances where n is the number of literals in the clause.

To win the game, the player has to place the O box in Figure 10 to the green square.

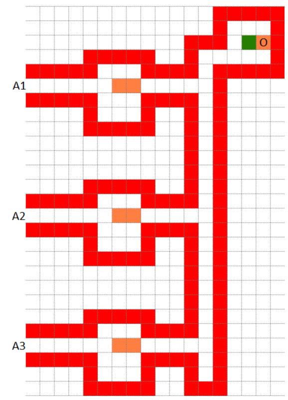

Figure 11: A picture of a clause gadget with three literals, the entrances A1, A2 and A3 correspond to entrances that are connected to a literal wire

Lemma 6. *When the player enters the gadget from an entrance that belongs to a literal wire he can´t exit through a literal wire that hasn´t been visited before*

Proof. When the player enters the gadget, he has to pull down two blocks to enter (when he returns he has to pull them up again). When he enters the gagdget, he cannot exit through a literal wire that hasn´t been visited before because there are two boxes blocking the way to that wire. The player can only pull one of the two obstacles out. □

2.8 Variable gadget

This gadget represents a variable that belongs to the SAT instance.

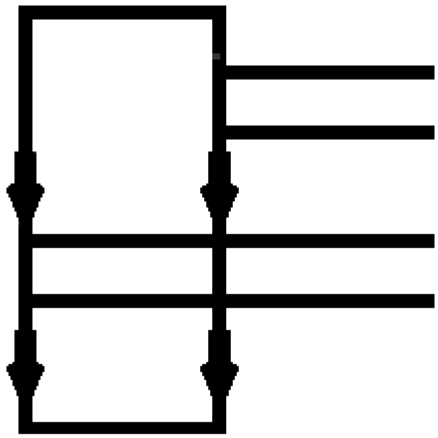

Figure 12: A picture of a variable gadget. The arrows represent no return gadgets in the path versions of pull and limited no return gadgets for the storage versions of pull

When the player enters the gadget, there is a bifurcation that divides the path in two paths: a left path and a right one.

If the player takes the right path, that means the player has chosen to assign the variable a negative value. If he chosses the left path, that means the player has chosen to assign the variable a possitive value.

When the player enters the left path or the right path he crosses a no return gadget (section 2.2) in the case of the path versions of pull and a limited no return gadget (section 2.1) in the case of the storage versions of pull.

After that he can traverse the literal wires that correspond to the literal, there will be one for each ocurrence of the literal in the clauses. After that, he will traverse another no return gadget in the case of the path versions of pull and another limited no return gadget in the case of the storage versions of pull. This is to ensure that the player cannot access the literal wires of the other path.

Finally, the player traverses a no return gadget at the end of the variable gadget.

3 General view of the board

We have seen the gadgets that we are going to use. Now we are going to shown how to arrange them to create a board.

In the board, the variable gadgets are at the left of the board and the clause gadgets are at the right of the board. Each variable gadget is put below the next one, the same for the clause gadgets. For the first variable in the SAT formula, we put a variable gadget in the left and we put one clause gadget for every clause in which the variable appears. First we put a clause gadget for every clause in which there is a negated literal of that variable and then we put the clauses in which there is a positive literal of that variable. Then we connect the corresponding literals to the clauses

For the second and remaining variables, we follow this procedure:

We define two kind of clauses, "old clauses" and "new clauses". Old clauses are those that have a clause gadget already in the board and new clauses are those that dont have a clause gadget in the board yet. That is, old clauses are those that belong to a variable gadget that has been put in the board and new clauses are those in which the current variable appears but not the previous variables.

For the negated literals of the variable, we connect the wires in the variable gadget to the old clauses if there is any old clause that contains the negated literal. Then we put the variable gadgets corresponding to the new clauses (if there is any) and we connect the literals to them.

We follow the same procedure for the positive literals of the variable.

We decided to set the board this way to force the player to cross through horizontal crossovers to go to new clauses and to force the player to cross vertical crossovers to go to old gadgets. In this way, if the player traverses a crossover and then he traverses the crossover again from another literal wire, he will only be able to visit the variable gadget corresponding to that literal wire or another variable gadget that he has already visited.

For the path versions of pull,note that we can place two way gadgets at the beginning and at the end of the crossovers. This prevents leakage between wires

Theorem 1. *The path and storage versions of pull in which all boxes can only be pulled either left and down or left to right is NP-hard*

Theorem 2. *The path version of pull in which all boxes can only be pulled either down or left is NP-hard*

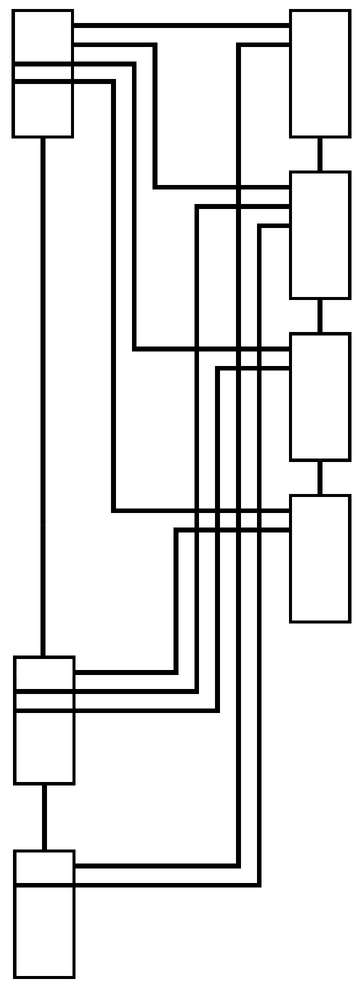

Figure 13: An example board. The variables at the left are x_1 x_2 and x_3. The clauses at the right are $(\neg x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)$

4 Extending the result

In previous sections, we showed that all versions pull where handles are in different directions (e.g. left and down, right and up etc.) are NP-hard, now we are going to prove that the path variants of pull in which the player can only pull boxes in the same direction (e.g. all boxes have only left and right handles) are NP-hard.

To do that, we will describe now the changes that we must do to the gadets we presented in section 2 so that they work correctly in this variant.

4.1 Variable gadget

For this variant of the game, we replace every literal wire with two parallel wires. One is used to go to the clause and the other is used to return from the clause. We have to ensure that the player can only use these wires if he comes from the right place.

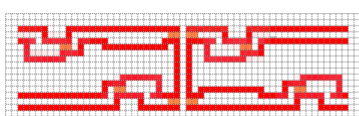

Figure 14: This picture represents the intersection point between an horizontal and a vertical wire inside a variable gadget. The two horizontal wires are a closeup of the two literal wires that the player has to use to go to a clause (up wire) and return from that clause (down wire)

If the player comes from the path of a positive literal, we don´t have to do any change, since the one way gadget that is just after the literal wires of the negative literal prevent him from accessing those literal wires.

Now if the player comes from the negative literal path, we have to ensure that the player cannot access any of the literal wires of the other path.

We cannot use the crossovers of the previous section, so we introduce the "half crossover", a crossover that fobids passage from a vertical wire to an horizontal one or viceversa (see figure 14).

4.2 Clause gadget

This is clause gadget similar to the one in section 2, but now we have to account for the fact that for each literal there are two literal wires connected to the gadget.

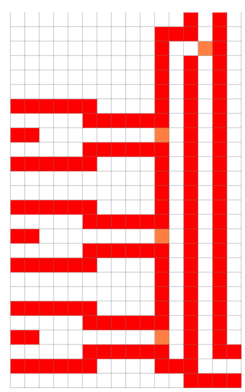

Figure 15: Clause gadget

4.3 Crossover gadget

This crossover is different from the other crossovers. It can be traversed from down to up and then up to down, without allowing the player to access the horizontal wires. If the player traverses it from the horizontal direction, he can traverse the gadget left to right and right to left. The player can also travel left to up or left to right and right to up but if the player does that, he will get trapped.

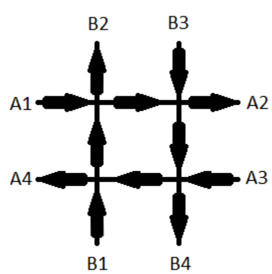

Figure 16: Schematic drawing of the crossover

Lemma 7. *If the player enters the crossover gadget from B1 he must return through B3 and he cannot enter to the horizontal wires*

Proof. When the player enters the gadget, he has to pull to the left a block that is inside a corridor that connects the two vertical wires. He must pull this block to be able to return from B4. If he doesn´t pull this block, he will be trapped when he returns from B3 to B4. Notice that this block makes sure that the player can´t use this corridor to travel from one vertical wire to the other.

After doing this, the player traverses the wire forward. The one way gadgets that the player traverses make sure that the player cannot return to a previously visited position and the horizontal crossovers (if the player has not traversed horizontally) or the no return gadgets (if he has traversed the gadget previously) make sure that the player can´t access the horizontal wires. This forces the player to return through B3. From B3, the player is forced by the no return gadgets to keep moving downwards until he reaches B4 □

Lemma 8. *If the player tries to access a vertical wire from an horizontal one, he will get trapped*

Proof. If the player traverses the crossover from the horizontal direction, he has to enter through A1. After traversiing the first no return gadget and opening the first horizontal crossover. He will be able to go through B2. Because of the no return gadget, he will be forced to keep moving upwards. As shown, in lemma seven, whenever he enters a crossover from B1 he will be forced to go to B2 and he wont be able to enter the horizontal wires of that crossover. He will be forced to keep going until he reaches the clause gadget of that literal wire. Because, of the properties of the clause gadget, he will be forced to return from the other literal wire. After traversing all crossovers in the opposite direction, the player will reach the crossover he visited at first. He will enter it from B3, and the horizontal crossovers and the no return gadgets will force the player to keep moving downwards. Until he traverses the last no return gadget in his way to B4, there the player will get trapped between this no return gadget and a block that is below it (see figure 17). This block can only be pulled out if the player entered into the gadget from B1 and therefore the player will get trapped.

If the player decides to keep moving forwards, he can open the second horizontal crossover and go downwards to B4, but this again will left the player trapped between a no return gadget and a block.

When the player goes from A3 to A4, the player will get trapped if he tries to exit from B4 or B2 by the same argument than before and he cannot access B1 or B3 because of the no return gadgets. So he must exit the gadget from A4

□

With these components, we can prove the hardness of pull of those versions where the player can only pull the handles from opposite directions

Theorem 3. *The path version of pull in which all boxes can only be pulled from opposite directions is NP-hard*

5 Pull with one handle is NP-hard

In this section, we are going to prove the main claim of our paper. That is we are going to prove that the version of pull where all boxes can only be pulled from one direction (in our examples, the left direction) are NP-hard.

As we did in the previous section, we are going to describe the changes that must be done to the gadgets so that they work in this section

5.1 Variable gadget

For this variant of the game, we replace every literal wire with two parallel wires. One is used to go to the clause and the other is used to return from the clause. We have to ensure that the player can only use these wires if he comes from the right place.

If the player comes from the path of a positive literal, we don´t have to do any change, since the one way gadget that is just after the literal wires of the negative literal prevent him from accessing those literal wires.

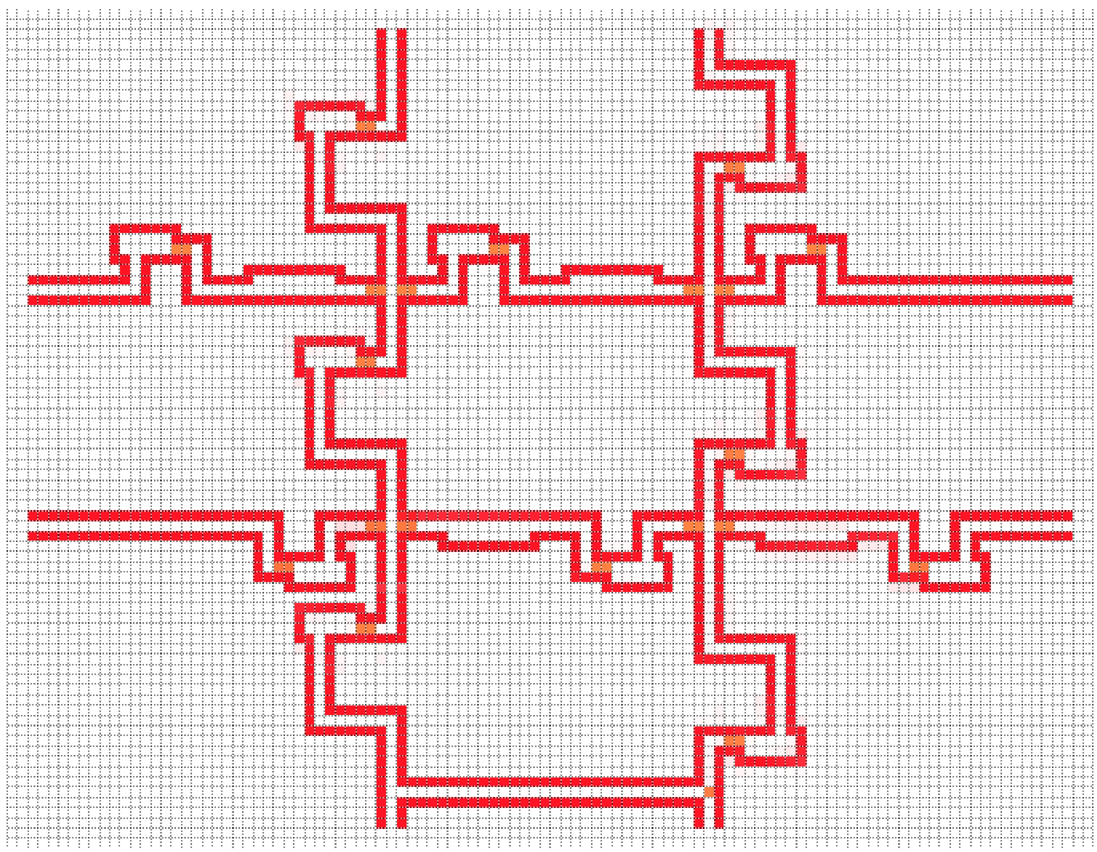

Figure 17: Complete drawing of the crossover. In this crossover, the player can only pull the blocks left and right

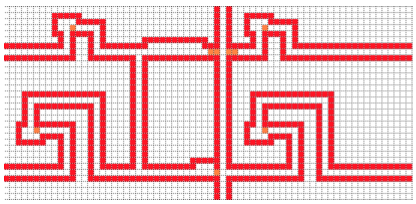

Figure 18: This picture represents the intersection point between an horizontal and a vertical wire inside a variable gadget. The two horizontal wires are a closeup of the two literal wires that the player has to use to go to a clause (up wire) and return from that clause (down wire)

Now if the player comes from the negative literal path, we have to ensure that the player cannot access any of the literal wires of the other path.

In the example shown in figure 18, this is done by placing an horizontal crossover in the wire that goes to the clause.

For the return wire, a single block prevents the player from accesing the wire. This block can be unlocked from a corridor that comunicates the two wires. This corridor can only be traversed from the interior of the other wire (see figure 18).

The clause gadget is the same that the one we shown in section 4.

5.2 Crossover gadget

This crossover is similar to the one from the previous section. It can be traversed from down to up and then up to down, without allowing the player to access the horizontal wires. If the player traverses it from the horizontal direction, he can traverse the gadget left to right and right to left. The player can also travel left to up or left to right and right to up but if the player does that, he will get trapped.

Lemma 9. *If the player enters the crossover gadget from B1 he must return through B3 and he cannot enter to the horizontal wires*

Proof. When the player enters the gadget, he has to pull to the left a block that is inside a corridor that connects the two vertical wires. He must pull this block to be able to return from B4. If he doesn´t pull this block, he will be trapped when he returns from B3 to B4. Notice that this block makes sure that the player can´t use this corridor to travel from one vertical wire to the other.

After doing this, the player traverses the wire upwards. After traversing the first no return gadget in his way upwards, notice that the player can traverse a no return gadget that is at the left (if the player has not traversed the gadget horizontally before), if he decides to traverse this gadget, he finds a corridor where a box is blocking his way and he is trapped between this block and the no return gadget he has just traversed. This is because this block can only be moved out of the way only if the player enters the gadget from A1. If the player decides to keep moving upwards, he traverser a second no return gadget from there, he has access to a second corridor, in it there is another box that he must pull in order to be able to return from B3. After doing this, he can keep moving upwardds. The no return gadgets that the player traverses make sure that the player cannot return to a previously visited position and the horizontal crossovers (if the player has not traversed the gadget horizontally previously) or the no return gadgets (if he has traversed the gadget previously) make sure that the player can´t access the horizontal wires. This forces the player to return through B3. From B3, the player is forced by the the horizontal crossovers (if the player has not traversed the gadget horizontally previously) or the no return gadgets (if he has traversed the gadget previously) to keep moving downwards until he reaches B4 □

Lemma 10. *If the player tries to access a vertical wire from an horizontal one, he will get trapped*

Proof. If the player traverses the crossover from the horizontal direction, he has to enter through A1. After traversiing the first no return gadget he can acces a corridor that is below him, in this corridor he can pull a box to the left, he needs to do this in order to be able to return from A3. After that, he can only keep moving to the right where he traverses a no return gadget and opens the first horizontal crossover. From there, he will be able to go through B2. Because of the no return gadget, he will be forced to keep moving upwards. As shown, in lemma twelve, whenever he enters a crossover from B1 he will be forced to go to B2 and he wont be able to enter the horizontal wires of that crossover. He will be forced to keep going until he reaches the clause gadget of that literal wire. Because, of the properties of the clause gadget, he will be forced to return from the other literal wire. After traversing all crossovers in the opposite direction, the player will reach the crossover he visited at first. He will enter it from B3, and the horizontal crossovers and the no return gadgets will force the player to keep moving downwards. When he traverses the second no return gadget in his way to B4, he will get trapped between this no return gadget and a block that is below it (see figure 17). This block can only be pulled out if the player entered into the gadget from B1 and therefore the player will get trapped.

If the player decides to keep moving forwards, he can open the second horizontal crossover and go downwards to B4, but this again will left the player trapped between a no return gadget and a block.

When the player goes from A3 to A4, the player will get trapped if he tries to exit from B4 or B2 by the same argument than before and he cannot access B1 or B3 because of the no return gadgets. So he must exit the gadget from A4 □

Theorem 4. *The path version of pull in which all boxes can only be pulled from a single direction is NP-hard*

6 Open questions and future directions

Can the results shown in sections 4 and 5 be extended to the storage version of pull?. That is, can we show that the storage versions of pull where the player can only pull blocks in opposite directions or the one where all boxes may be pulled in the same direction are NP-hard?

We think that the version where the player can only pull boxes in opposite directions is NP-hard as well. If there exists a variant of planar SAT where the negative clauses are outside the cycle of variables and the positive ones are inside the cycle and this variant of SAT is NP-hard, then we can reduce that variant of SAT to the version of storage pull where the player can only pull boxes in opposite directions.

We think that the storage variant of pull where all boxes can only be pulled in a single direction is tractable.

Extend the results of this paper to the game of Push.

Is the variant of pull where the player cannot revisit a square (called pull-X), still hard if the player can only pull boxes in certain directions.If the problem is tractable, this will allow us to solve the open question of [3] about finding an interesting but tractable block-moving puzzle.

References

[1] Marcus Ritt *"Motion planning with pull moves"*. http://arxiv.org/abs/1008.2952

[2] Demaine, E. D., Demaine, M. L., and O'Rourke, J. *"PushPush and Push-1 are NP-Hard in 2D"*. Proc. 12th Canad. Conf. Comput. Geom. (2000), 211-219.

[3] Erik D. Demaine, Martin L. Demaine, Michael Hoffman and Joseph O'Rourke *"Pushing blocks is hard"*. Computational Geometry, 26:21-36, 2003

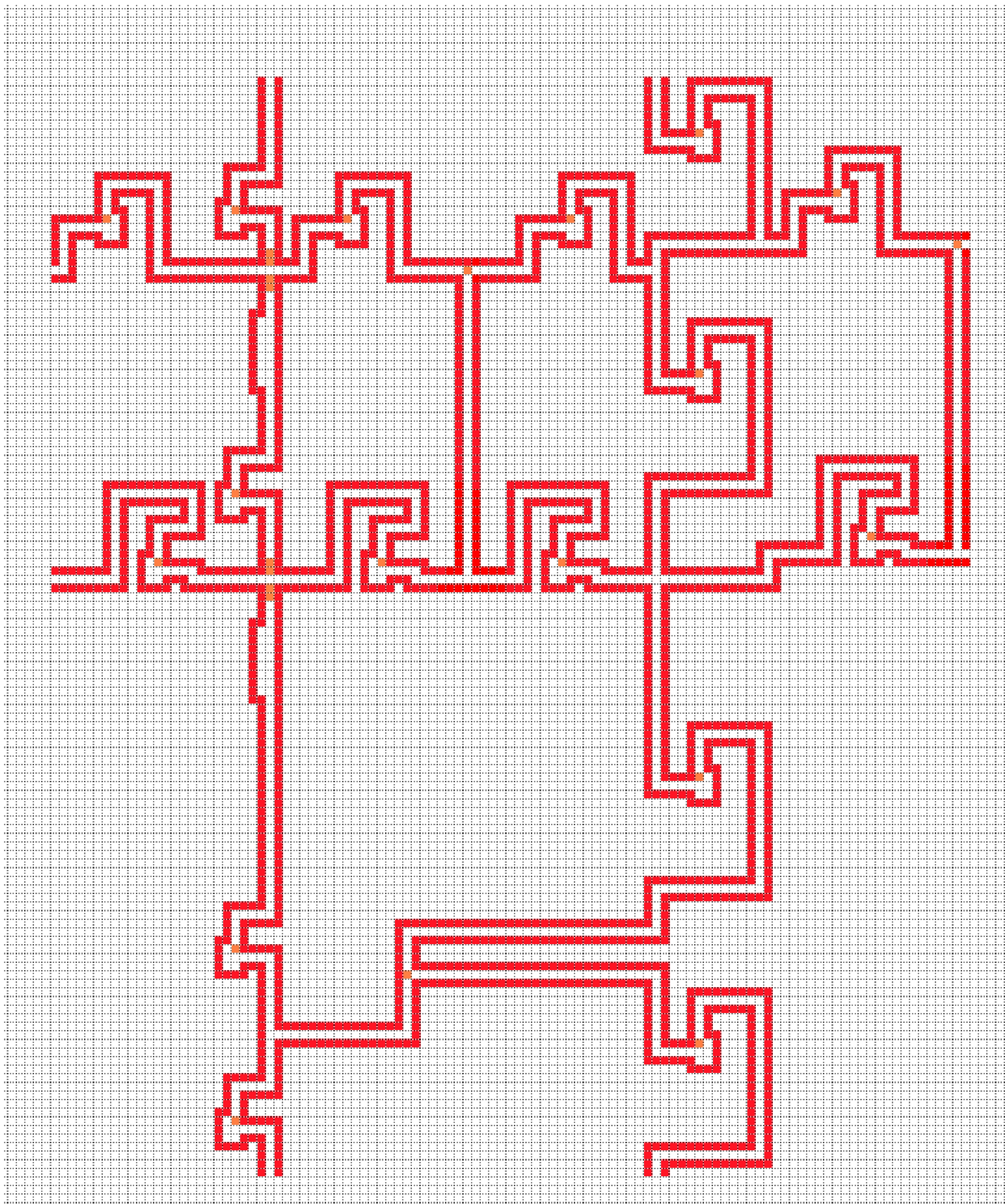

Figure 19: Complete drawing of the crossover. In this crossover,the player can only pull the blocks leftwards (the image is rotated 90 degrees, A1 is at the bottom left)