



## Labeling nonograms: Boundary labeling for curve arrangements

Fabian Klute<sup>a,1</sup>, Maarten Löffler<sup>a</sup>, Martin Nöllenburg<sup>b,\*</sup>

<sup>a</sup> Department of Computing and Information Sciences, Utrecht University, Utrecht, the Netherlands

<sup>b</sup> Algorithms and Complexity Group, TU Wien, Vienna, Austria



### ARTICLE INFO

#### Article history:

Received 27 September 2020

Received in revised form 4 May 2021

Accepted 4 May 2021

Available online 17 May 2021

#### Keywords:

Nonograms

External label placement

Algorithms

Complexity

### ABSTRACT

Slanted and curved nonograms are a new type of picture puzzles introduced by Van de Kerkhof et al. (2019). They consist of an arrangement of lines or curves within a frame  $B$ , where some of the cells need to be colored in order to obtain the solution picture. For solving the puzzle, up to two clues need to be attached as numeric labels to each line on either side of  $B$ . In this paper we study the algorithmic problem of optimizing or deciding the existence of a placement of the given clue labels to such a nonogram. We provide polynomial-time algorithms for restricted cases and prove NP-completeness in general.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

**Nonograms** Nonograms, also known as *Japanese puzzles*, *paint-by-numbers*, or *griddlers*, are a popular puzzle type where one is given an empty grid and a set of clues on which grid cells need to be colored, typically resulting in a picture (see Fig. 1 (left)). There is quite some work in the literature on the difficulty of solving nonograms [1,2], and this remains an active topic of discussion [3].

**Curved nonograms** Van de Kerkhof et al. introduced *curved* nonograms, a variant in which the puzzle is no longer played on a grid but on any arrangement of lines or even curves [4]. Such nonograms have been investigated in several Master theses [5–7]. Of special interest are *slanted* nonograms [6], in which all curves are straight lines, possibly limited to a fixed number  $k$  of orientations (also known as *sloped* nonograms or, in the case of  $k = 4$ , *tangram* nonograms). Fig. 1 illustrates the different variants.

Van de Kerkhof et al. describe heuristics to generate puzzles, but they leave open the question of how to attach labels with clues. It turns out this is actually a non-trivial task, as labels could be placed in several valid locations. Each curve to be labeled enters and leaves the picture *frame* (bold black rectangle in the figures) once, and the information about which incident cells of the arrangement should be filled is summarized in two *clues*, one on each side of the curve (we refer the reader to [4] for a full description of the rules). This gives two logical potential locations for each clue. Furthermore, it may be possible to extend curves outside the frame to make more room for the clues, and not necessarily all clues need to be given for the puzzle to be solvable.

\* Corresponding author.

E-mail addresses: [f.m.klute@uu.nl](mailto:f.m.klute@uu.nl) (F. Klute), [m.loeffler@uu.nl](mailto:m.loeffler@uu.nl) (M. Löffler), [noellenburg@ac.tuwien.ac.at](mailto:noellenburg@ac.tuwien.ac.at) (M. Nöllenburg).

<sup>1</sup> Supported by the Netherlands Organisation for Scientific Research (NWO) under project 612.001.651.



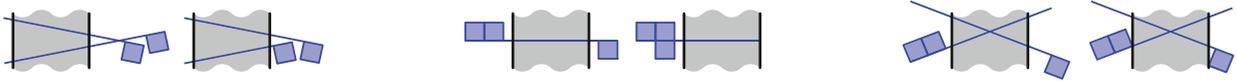


Fig. 2. Left: a crossing and a crossing-free labeling. Center: A balanced and a non-balanced labeling. Right: an extended and a compact labeling.

some considerations on good curve arrangements for nonogram generation [4] do give rise to different patterns on the puzzle frame. Here we briefly review some of them.

Curved nonograms should have curves which are unambiguous to trace, and faces which are sufficiently large and fat to be easily distinguished. This places a natural bound on the density. It is also reasonable to assume that adjacent ports have a certain minimum spacing, or that the angle at which the ports exit the frame have a minimum value.

Van de Kerkhof [7] further observes that nonogram curves are easier to trace when they do not wind too much (for us: the two ports of the same curve are not on the same edge of the frame), and that the rules are easier to understand when no three curves cross in a common point (for us: all ports are at distinct points), or when curves do not have self-intersections (irrelevant to the labeling problem). He notes that it is of interest to be able to generate puzzles with or without these properties, to allow for puzzles of different difficulty.

For slanted nonograms, Parment [6] observes that avoiding points in which multiple lines intersect is difficult. Furthermore, there is some evidence that an arrangement of sufficiently many lines with the property that all faces are sufficiently large must necessarily form (an affine transformation of) some grid-like pattern. In this case, the number of different slopes of ports may be limited, and the spacing between the ports may follow a particular structure: for instance, all distances may be integer, or, as in the example in Fig. 1 (middle), of the form  $a + b\phi$  where  $a$  and  $b$  are integers and  $\phi = \frac{1}{2} + \frac{1}{2}\sqrt{5}$  is the golden ratio.

*Problem statement* In the nonogram labeling problem we are given the following input:

- (i) a nonogram frame, which is a simple convex polygon  $B$ ;
- (ii) a set  $\mathcal{L}$  of  $n$  nonogram lines passing through  $B$ , each  $l \in \mathcal{L}$  defining a pair  $(p_l, q_l)$  of ports at the intersection points of  $l$  with  $B$ ; and
- (iii) a pair of non-negative integers  $(a_l, b_l)$  for each  $l \in \mathcal{L}$ , where  $a_l$  defines the width of the label above  $l$  and  $b_l$  defines the width of the label below  $l$ , and a width of 0 indicates the label is absent.<sup>2</sup>

As output, we ask for a labeling of  $\mathcal{L}$ , such that for each label  $\ell$  of nonogram line  $l \in \mathcal{L}$ :

- (i)  $\ell$  is assigned to one of the two ports  $p_l$  or  $q_l$ ;
- (ii)  $\ell$  is assigned an extension length  $\delta$  (which could be 0);
- (iii)  $\ell$  is assigned a boolean value *right* or *left* indicating whether the label should be placed on the left or right side of the line (when exiting the frame).

Given an output specification, we draw a leader of length  $\delta$  from its assigned port  $p_l$  or  $q_l$  aligned with the slope of  $l$ , and the label itself as an  $a_l \times 1$  (or  $b_l \times 1$ ) rectangle, also aligned with the slope of  $l$ , and anchored at the end of its leader. We say that a label  $\ell$  is *right-flipped* (*left-flipped*) if it is attached to the right (left) side of the leader from its assigned port.

A labeling is *valid* if no two labels overlap each other, no label overlaps a leader of another label, and no label intersects the frame. In addition to being valid, we identify three further desired properties (refer to Fig. 2).

- **Crossing-free.** We disallow intersections between leaders.
- **Balanced.** We require the two labels of a line  $l \in \mathcal{L}$  to be assigned to opposite ports.
- **Compact.** We require all leaders to be of length 0, or the shortest length necessary to avoid an intersection between the label and the frame.

Ideally, we would like to find a crossing-free, compact, and balanced labeling. However, this may not exist (see, e.g., Fig. 1 (middle)). We study the computational problems of testing for the existence of a solution, and minimizing the total leader length among all possible solutions, for several combinations of these properties. Additionally, we investigate whether the maximum length  $c$  of the labels to be placed influences the problem, or, for slanted nonograms, whether the number  $k$  of distinct slopes of lines in  $\mathcal{L}$  does.

*Clearance* In real nonogram puzzles, we do not want to only avoid labels from overlapping, but we would like to have a bit of space between them. We can model this by simply enlarging each label and the nonogram frame by a small amount

<sup>2</sup> Note the difference between a label of width 0, which gives no information, and a label with the string "0", which has width 1 and which gives the information that none of the cells in the sequence should be filled. Both of these occur in actual nonogram puzzles.

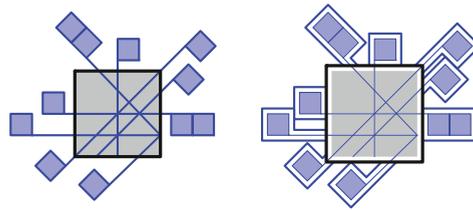


Fig. 3. A labeled instance in which all labels and the nonogram frame observe a minimum amount of clearance  $\delta$  to each other is equivalent to another instance in which the labels and the frame are dilated by an amount of  $\delta/2$ .

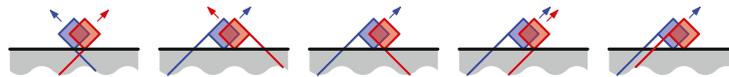


Fig. 4. Depending on the configuration, conflicts can be resolved by extending one or both of the leaders.



Fig. 5. A conflict can be resolved by flipping one or both of the labels.

(refer to Fig. 3). Therefore, we ignore the issue of clearance in the remainder of our discussion. Similarly, we would also like a minimum amount of clearance between leaders. We assume that this follows from the locations of the ports, which are, after all, determined by the interior of the nonogram.

*Initial observations* First, we observe that a balanced solution, which is not crossing-free and not compact always exists: we simply extend the leaders sufficiently far.

**Observation 1.** *Given a nonogram labeling instance, a non-compact potentially crossing labeling always exists and can be computed in linear time. This is true regardless of whether we require it to be balanced.*

Since this does not give a satisfactory result in practice, we focus on more restricted variants in the remainder.

For example, in the special case where the puzzle frame is a rectangle, and all ports have  $45^\circ$  angles with respect to the frame, there are several ways in which two labels can be intersecting when we place them as close to the frame boundary as possible (see Fig. 4). Depending on the configuration, such a conflict can be resolved by *extending* one or both of the leaders.

In the setting where we require a *balanced* solution, it might also be possible to resolve conflicts by *flipping* one of the labels (see Fig. 5). Note that this only makes sense if we also flip the label on the opposite side of the line. Hence, this operations creates dependencies between different sides of the puzzle.

*Results* In Section 3.1, we show that testing whether a compact solution exists for any curved or slanted nonogram is possible in polynomial time. In Section 3.2, we show that for slanted nonograms, a non-crossing balanced solution of minimal total leader length can be computed in polynomial time, if the assignment of labels to ports is given and  $k = 2$  for slope  $\pm 1$ . In Section 4, we show that the problem of testing whether a crossing-free solution exists is NP-complete, even for slanted nonograms when  $k = 2$  for slopes  $\pm 1$ . Interestingly, the hardness proof works even when all labels are *short*, that is, have sizes of at most aspect ratio  $1 : 4$ . This motivates the study of the problem for small labels, and we show in Section 5 that for square labels on a unit grid, the problem can be solved with dynamic programming even without assigning the sides in advance. (See Table 1.)

### 3. Algorithms

We present two polynomial-time algorithms for special cases of nonogram labeling. Section 3.1 considers compact labelings and Section 3.2 assumes that each label is assigned to a fixed port in the input and the extension length of each leader must be computed.

#### 3.1. Compact labeling

In our first result, we assume that each label  $\ell$  must be placed as close to the frame  $B$  as possible, i.e.,  $\ell$  must touch  $B$ . This leaves only one degree of freedom for each label  $\ell$  of a nonogram line  $l$ , namely whether it is placed at port  $p_l$  or  $q_l$ .

**Table 1**

Results for different variants of the nonogram labeling problem, where  $n = |\mathcal{L}|$  is the number of lines. For the regular grid constraint we assume that the ports are spaced at least  $\sqrt{2}$  units apart. Note that the version which requires a solution to be compact, but at the same time allows crossings, is not interesting.

Result	Crossings	Balanced	Compact	Additional constraints	Location
$O(n)$	yes	either way	no		Observation 1
$O(n^2)$	no	either way	yes		Theorem 1
$O(n^2 \log n)$	no	either way	yes	labels can be scaled	Corollary 1
$O(n^{11})$	no	yes	no	fixed side assignment, slopes $\pm 1$ , rectangular frame	Theorem 2
NP-hard	no	no	no		Theorem 3
NP-hard	no	no	no	short labels	Theorem 4
$O(n^3)$	no	yes	no	regular grid, $1 \times 1$ -sized labels, two labels per leader, slopes $\pm 1$ , rectangular frame	Theorem 5

Similarly to more general point labeling problems with two alternative positions per label [17], we define a simple 2-SAT model to determine a feasible nonogram labeling or report that none exists.

**Theorem 1.** *Given a nonogram labeling instance, we can decide in  $O(n^2)$  time whether a compact labeling exists. This is true regardless of whether we require it to be balanced.*

**Proof.** We derive a 2-SAT formula  $\varphi$  that has a satisfying variable assignment if and only if a valid labeling without leader extensions exists. For each nonogram line  $l \in \mathcal{L}$  we define two variables  $x_l^a$  and  $x_l^b$ , where  $x_l^a = 1$  ( $x_l^a = 0$ ) indicates that the label above  $l$  is assigned to the port  $p_l$  ( $q_l$ ) of  $l$ . Similarly,  $x_l^b = 1$  ( $x_l^b = 0$ ) indicates that the label below  $l$  is assigned to  $p_l$  ( $q_l$ ). It is clear that a variable assignment is in bijection to a port assignment of the labels and it remains to add some clauses to  $\varphi$  to model the valid labelings. For each overlap of a label of a line  $l$  with a label of another line  $l'$  (we call that a *conflict*), we add a clause that prevents both labels to be selected simultaneously. As an example consider the case that the label above  $l$  and the label below  $l'$  intersect if both assigned to their ports  $p_l$  and  $p_{l'}$ . Then we add the clause  $\neg x_l^a \vee \neg x_{l'}^b$ . Now a satisfying assignment for  $\varphi$  corresponds to an assignment of each label to a port of its nonogram line such that no two labels intersect each other; otherwise some clause would not be satisfied.

To ensure that the labeling is balanced, we would add the additional clauses  $x_l^a \vee x_l^b$  and  $\neg x_l^a \vee \neg x_l^b$  for each  $l \in \mathcal{L}$ .

Solving the 2-SAT instance takes linear time [18] in the size of  $\varphi$ , where the number of clauses of  $\varphi$  is linear in the number of nonogram lines and the number of label conflicts, which is  $O(n^2)$ .  $\square$

We remark that this 2-SAT model is independent of the type and shape of the curves and the shape of  $B$ . It depends only on the set of conflicting candidate label positions.

A compact labeling does not always exist. If one does insist on a compact and crossing-free labeling, one option is to shrink the label sizes. As an easy corollary of the above result, we show that the largest scale for which a compact labeling exists can also be obtained in polynomial time.

**Corollary 1.** *Given a nonogram labeling instance, we can compute in  $O(n^2 \log n)$  time the largest label size such that a compact labeling exists. This is true regardless of whether we require it to be balanced.*

**Proof.** We create a 2-SAT instance as above. We distinguish two types of clauses: *hard* clauses which must be obeyed regardless of the label size, and *soft* clauses which only need to be obeyed for certain label sizes. The hard clauses include those which describe that each clue is placed exactly once, and, in case of a balanced solution, those that enforce that the labeling is balanced. The soft clauses occur between each pair of label placements which could potentially overlap, if the labels were sufficiently large. The key observation is that, for each such pair, there is a unique scale  $s$  such that for all scales smaller than  $s$  the labels are disjoint, and for all scales larger than  $s$  the labels are intersecting. Hence, we annotate each soft clause with this value  $s$ . The task is now to find the largest value of  $s$  such that all hard clauses and all soft clauses with annotated value smaller than  $s$  together admit a solution. We perform binary search on the value of  $s$  and apply Theorem 1 for each scale; this yields the  $O(n^2 \log n)$  time bound.  $\square$

### 3.2. Fixed side assignment

Our second algorithm allows extensible leaders, but disallows leader intersections and assumes that a balanced assignment of the labels to the ports of each nonogram line is given in the input. We further assume that the nonogram lines are all of slopes  $\pm 1$ .

**Lemma 1.** *For a nonogram labeling instance  $\mathcal{I}$  with  $n$  lines and a balanced fixed side assignment for each label we can discretize the relevant extension lengths of each label to  $O(n^2)$  values such that we can find a labeling of minimum total extension length among*

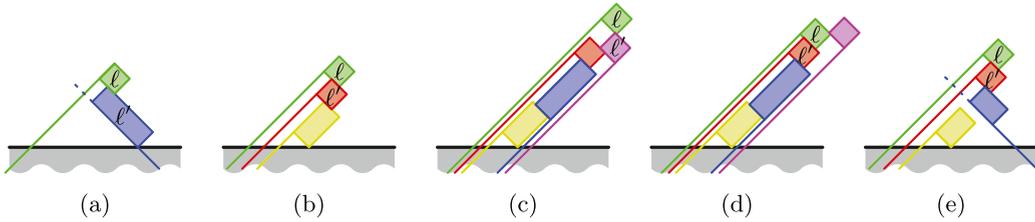


Fig. 6. Possible cases of extension lengths for label  $\ell$  blocked by another label  $\ell'$  in the proof of Lemma 1.

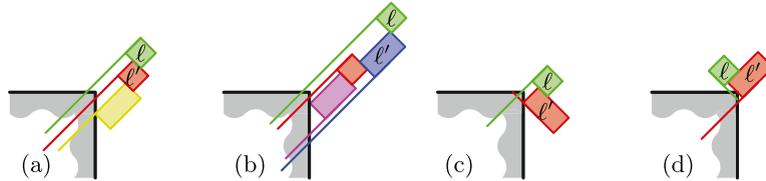


Fig. 7. Possible cases of extension lengths for label  $\ell$  blocked by another label  $\ell'$  across a corner of  $B$  in the proof of Lemma 1.

this set of extension lengths (if  $\mathcal{I}$  has a solution at all). The first or last label on each edge of the frame  $B$  has  $O(n)$  relevant extension lengths.

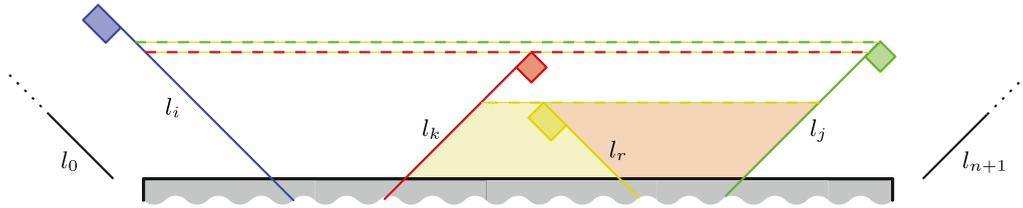
**Proof.** The first observation is that in a minimum-length labeling, every label  $\ell$  of a nonogram line  $l$  is shifted as close to the frame  $B$  as possible without intersecting another label as otherwise a solution with shorter total length would exist. This means that either (i)  $\ell$  touches  $B$  and the extension length is 0 or 1 (depending on which side of  $l$  is labeled and the slope of  $l$ ), or (ii)  $\ell$  touches another label  $\ell'$  blocking  $\ell$  from moving closer to  $B$ . Observe that these are the only two possibilities; in particular, it is not possible that  $\ell$  is blocked by the leader of another label when there are no leader intersections. We show that the number of extension lengths of type (ii) for a single label is bounded by  $O(n^2)$ .

We assume that  $l$  crosses the top edge of the frame  $B$  in a port  $p_l$  to which label  $\ell$  is assigned. The situation for ports on the other edges of  $B$  is symmetric. Let us first consider the case that both  $\ell$  and its blocking label  $\ell'$  are assigned to a port on the top edge of  $B$ . Without loss of generality let the slope of  $l$  be  $+1$  and distinguish two cases for the blocking label  $\ell'$ .

1. In the first case,  $\ell'$  belongs to a line of different slope, meaning that the extension length of  $\ell$  is determined by the intersection point of the two nonogram lines (possibly  $+1$  depending on the side assignment of  $\ell'$ ), see Fig. 6(a). This case can occur only if  $\ell$  is flipped to the right of  $l$ . There are  $O(n)$  such intersection points for  $\ell$ .
2. In the second case, the lines of  $\ell$  and  $\ell'$  are parallel and of distance at most 2 (otherwise  $\ell'$  cannot block  $\ell$ ). Such blocking relations can form chains, where  $\ell'$  itself is blocked again by a third label  $\ell''$  and so on.
  - If the chain of blocking relations comprises only parallel lines and all of them have the labels on the same side (Fig. 6(b)), then we obtain a single extension length for  $\ell$  by summing up the label lengths of all labels in the chain and possibly adding  $+1$  if the labels are flipped to the right of their lines.
  - If the parallel lines come in a group of right-flipped labels followed by a group of left-flipped labels as in Figs. 6(c–d), then the extension length of  $\ell$  is at least the sum of label lengths in its own group (possibly  $+1$ ). Furthermore, any prefix of the sequence of the group of oppositely flipped labels can add to the extension length of  $\ell$ , which again yields  $O(n)$  possible extension lengths for  $\ell$ .
  - The last case combines chains of parallel lines of right-flipped labels with an orthogonal line (Fig. 6(e)). Here  $\ell$  is blocked by a suffix of the chain of right-flipped labels, the last of which is  $\ell'$  and the first of which is blocked by the label of an orthogonal line. Since there are  $O(n)$  possible orthogonal lines and  $O(n)$  possible suffix chains ending in  $\ell$ , this last case can give rise to  $O(n^2)$  different extension lengths.

Finally, we consider the situation of a port  $p_l$  near the top-right corner of the frame  $B$ . For the top-left corner the arguments are symmetric. So let  $\ell$  be a label assigned to a port on the top edge of  $B$  that is blocked by a label  $\ell'$  of the right edge of  $B$  as shown in Fig. 7.

1. If the slope of  $l$  is  $+1$  and  $\ell$  is right-flipped there are three subcases.
  - We can have a chain of right-flipped labels of parallel lines that starts on the top edge and continues on the right edge as shown in Fig. 7(a). As before, there is a single feasible shortest extension length for  $\ell$ . This case can in fact also be triggered from a label on the bottom edge that pushes the first label of the chain on the right edge into its position (combine Fig. 7(a) and Fig. 7(d), rotated clockwise by  $90^\circ$ ).



**Fig. 8.** Illustration of the dynamic programming recursion, where  $l_k$  splits the instance defined by  $l_i$  and  $l_j$  and, subsequently,  $l_r$  splits the instance defined by  $l_k$  and  $l_j$ .

- Alternatively, we can have a group of right-flipped labels followed by a group of left-flipped labels, where the possible extension lengths of  $\ell$  are defined by the sum of lengths of the right-flipped labels plus the sum of label lengths of a prefix of the left-flipped labels. Fig. 7(b) shows an example. Since there are  $O(n)$  prefixes, we obtain  $O(n)$  extension lengths of  $\ell$ .
  - If  $\ell'$  belongs to a line of slope  $-1$  then there is a single possible extension length of  $\ell$  as shown in Fig. 7(c).
2. If the slope of  $l$  is  $-1$  and  $\ell$  is right-flipped, it can be blocked by a left-flipped label  $\ell'$  of a line with slope  $+1$  as shown in Fig. 7(d). This yields a single possible extension length.
  3. Finally, if the slope of  $l$  is  $+1$  or  $-1$  and  $\ell$  is left-flipped, then no label  $\ell'$  assigned to the right edge can block  $\ell$ .

The above arguments imply that the number of relevant positions for a label  $\ell$  to consider is  $O(n^2)$  in the worst case. For a label  $\ell$  interacting with a label  $\ell'$  on an adjacent edge of  $B$ , it is even sufficient to consider  $O(n)$  different extension lengths.  $\square$

Based on the discretization of the relevant extension lengths of Lemma 1 we can now describe a dynamic programming algorithm to solve a nonogram labeling sub-instance defined on one of the four edges of the frame  $B$ . More precisely, we also specify the extension length of the first and last label of the instance so that we can compose compatible sub-solutions for all four edges of  $B$  into a globally optimal solution.

**Lemma 2.** Given a nonogram labeling instance  $\mathcal{I}$  in a rectangular frame  $B$ , a specified edge  $e$  of  $B$  crossed by  $n$  lines, a fixed side assignment for each label, and a fixed extension length for the labels of the two outermost lines crossing  $e$ , we can decide in  $O(n^9)$  time, whether an assignment of an extension length to each label exists such that the resulting labeling is valid and crossing-free. If this is the case we can find one of minimum total extension length.

**Proof.** Assume without loss of generality that  $e$  is the top edge of  $B$  and that the lines are indexed  $l_1, \dots, l_n$  in the left-to-right order of their ports  $p_1, \dots, p_n$  on  $e$ . For the two outermost lines  $l_1$  and  $l_n$  fixed extension lengths  $\delta_1$  and  $\delta_n$  are specified in the input. These two lines play a special role and we will mark any sub-instance invalid, in which they would be assigned any other extension length. This guarantees that we consider only solutions that respect the specified extension lengths of  $l_1$  and  $l_n$ .

To initialize the instance we further introduce two dummy lines  $l_0$  and  $l_{n+1}$ , where  $l_0$  has slope  $-1$  and has its dummy port one unit left of the top-left corner of  $B$  and  $l_{n+1}$  has slope  $+1$  and its port is one unit to the right of the top-right corner of  $B$ . Therefore,  $l_0$  and  $l_{n+1}$  cannot intersect any line or label of  $\mathcal{I}$ . We assign an extension length  $\delta_{\max}$  to  $l_0$  and  $l_{n+1}$  that exceeds the maximum of any of the discretized extension lengths of the lines on  $e$  as defined in Lemma 1. Recall further that by Lemma 1 we know that it is sufficient to consider at most  $O(n^2)$  many extension lengths for each label. For a line  $l_i$  let  $\Delta_i$  be the set of its relevant extension lengths. Note that for the outermost lines  $l_1$  and  $l_n$  we set  $\Delta_1 = \{\delta_1\}$  and  $\Delta_n = \{\delta_n\}$ . We recursively define a sub-instance of  $\mathcal{I}$  by two boundary lines  $l_i$  and  $l_j$  ( $i < j$ ) and a discretized extension length  $a$  and  $b$  for each of them. Since there are  $n$  lines and each of them has at most  $O(n^2)$  many relevant extension lengths, this defines  $O(n^6)$  different sub-instances. In such a sub-instance, any line  $l_k$  with a port  $p_k$  between  $p_i$  and  $p_j$  is restricted to stay within the region bounded by  $l_i$ ,  $l_j$ , and a horizontal line through the topmost point of the shorter of the two lines  $l_i, l_j$ , see Fig. 8. This means that the extension length of  $l_k$  must be such that it does not extend beyond the trapezoidal region defined by the sub-instance. Let  $T[i, j, a, b]$  be the minimum total extension length of the instance defined by  $l_i$  of length  $a$  and  $l_j$  of length  $b$ , if a valid and crossing-free solution exists; otherwise the instance is invalid and we set  $T[i, j, a, b] = \infty$ .

To solve such a sub-instance recursively, we use dynamic programming and optimize over all lines  $l_k$  ( $i < k < j$ ) contained in the sub-instance and all admissible and intersection-free extension lengths  $c$  for  $l_k$  and compose the solution from the stored solutions of the two sub-instances defined by  $l_i$  and  $l_k$  as well as  $l_k$  and  $l_j$ . The recursion stops for  $i = j - 1$ , where the extension length  $T[i, j, a, b] = a + b$ . Accordingly, for  $j - i > 1$ , we can recursively define

$$T[i, j, a, b] = \min\{T[i, k, a, c] + T[k, j, c, b] - c \mid i < k < j, c \in \Delta_k, c \leq \min\{a, b\}, l_k \text{ valid}\}.$$

The optimization of one entry of table  $T$  considers  $O(n^3)$  combinations of a line  $l_k$  and an extension length  $c \in \Delta_k$  and must verify that  $l_k$  with the assigned extension length  $c$  and its label  $\ell_k$  is valid, i.e., it does not intersect neither  $l_i, l_j$  nor their labels. This intersection test takes constant time.

We obtain the minimum-length solution of  $\mathcal{I}$  from cell  $T[0, n + 1, \delta_{\max}, \delta_{\max}]$ , which is computed by the above dynamic programming algorithm in  $O(n^9)$  time. If its value is  $\infty$ , then no valid, crossing-free solution exists.  $\square$

It remains to construct a valid, crossing-free, and minimum-length solution for the entire nonogram by composing it from compatible sub-solutions for the four sides of the frame  $B$ .

**Theorem 2.** *Given a nonogram labeling instance  $\mathcal{I}$  with  $n$  lines and a fixed side assignment for each label, we can decide in  $O(n^{11})$  time, whether an extension length assignment to each label exists such that the resulting labeling is valid and crossing-free. If this is the case we can find one of minimum total extension length in the same time bound.*

**Proof.** In a first step we compute a minimum-length solution for each edge of  $B$  and each combination of extension lengths for its two outermost labels. Due to Lemma 1 each outermost label has  $O(n)$  relevant extension lengths, which implies that we compute optimal solutions for  $O(n^2)$  instances on each edge. Solving one such instance takes  $O(n^9)$  time by Lemma 2 and thus all instances on all four edges require  $O(n^{11})$  time to solve.

In the second step we can now simply consider all combinations of the  $O(n^2)$  sub-solution for each of the four edges of  $B$ , which yields a total of  $O(n^8)$  combinations. For each combination we test whether the outermost labels on adjacent edges are compatible with each other, i.e., whether they yield a valid solution. This requires an intersection test for four pairs of lines with labels, which can be done in constant time. From all valid solutions, we choose the one with minimum total extension length.  $\square$

We remark that in many real-world nonograms the number of relevant extension lengths per line is expected to be much less than  $O(n^2)$ , e.g., since parallel lines will have a certain minimum distance. Moreover, we can have dependencies between different edges of the frame  $B$  only if the ports of their outermost lines are very close to the corners of  $B$ . Otherwise the sub-instance induced by each edge can be solved independently and be combined without further constraints. Combined with the fact that, in practice,  $n$  is not expected to be extremely large, we believe the algorithm may be of practical value despite the high degree of the polynomial.

## 4. Hardness

We will show in Section 4.1 that the problem of testing whether a valid labeling exists, in the setting where we are allowed to choose at which port each label is placed and are allowed to extend the leaders to any desired length, is NP-hard. The proof uses only straight lines of slopes  $+1$  and  $-1$  in the interior of the puzzle, and hence applies already to slanted nonograms with two distinct slopes. It uses lines which have only a single label, but it can easily be extended to the case where we have two labels per line: we simply always place them both at the same port (some distances need to be slightly adjusted). Notice that this does not translate to the balanced setting. The proof also relies on some labels having arbitrary length; we show in Section 4.2 how to adapt the proof so that it uses only labels (and leaders) of constant length.

### 4.1. Unbounded label size

We will construct an instance with a rectangular frame which only has lines of slopes  $+1$  and  $-1$ . We will reduce from 3-SAT: we are given an instance consisting of a set of  $n$  variables  $x_1, \dots, x_n$  and  $m$  clauses  $c_1, \dots, c_m$ , and we will construct a nonogram labeling instance which can be labeled without crossings if and only if the 3-SAT instance is satisfiable.

*Variables* The bulk of the construction consists of *cross gadgets*. A cross gadget consists of two short labels intersecting each other at  $90^\circ$  angles. Fig. 9 shows a single cross gadget.

Note that for a cross gadget it is not relevant on which side of the lines the labels are, and that although labels can be extended, doing so does not change the combinatorial choices of which combinations of sides are possible.

A cross gadget has three possible valid states, and we wish to use them to represent variables, which have two valid states. Furthermore, we would like to enforce multiple cross gadgets to represent the same variable. We can achieve both of these properties by connecting several cross gadgets into a *variable loop*. Fig. 10 illustrates a variable loop.

By connecting the cross gadgets into a loop, we ensure that only two valid solutions remain for each cross gadget. Note that we can increase the number of occurrences of a cross of the same variable by making the frame wider, and we can increase the distance between consecutive crosses by making the frame higher. We can embed multiple independent loops next to each other, as illustrated in Fig. 11.

In the construction, we will also need to place some labels which cannot be removed. We achieve this by creating a special variable loop, for which one of the states is disabled by creating a crossing between a positive and a negative label. We achieve this by making them longer (see Fig. 12). Note that forced (black) labels are only forced to be on a particular side of  $B$ ; they can still be extended, but we will use them in a way where extending black labels is never useful.

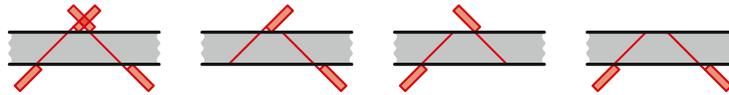


Fig. 9. A variable gadget (cross gadget) and its three possible valid solutions.

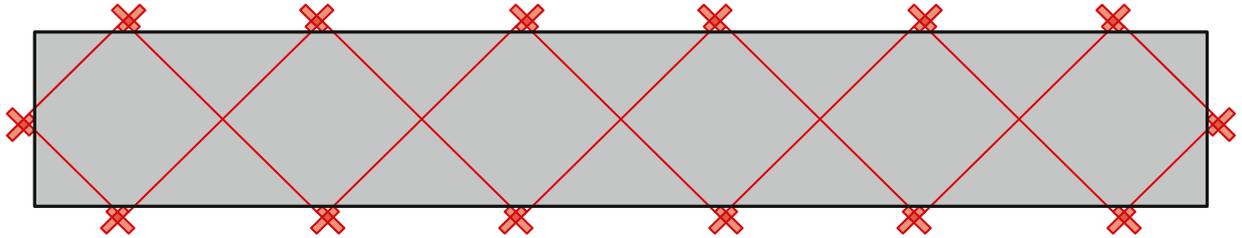


Fig. 10. A single variable loop. There are only two valid crossing-free solutions: placing any label on either side of its line causes a chain reaction of label placements for all remaining lines.

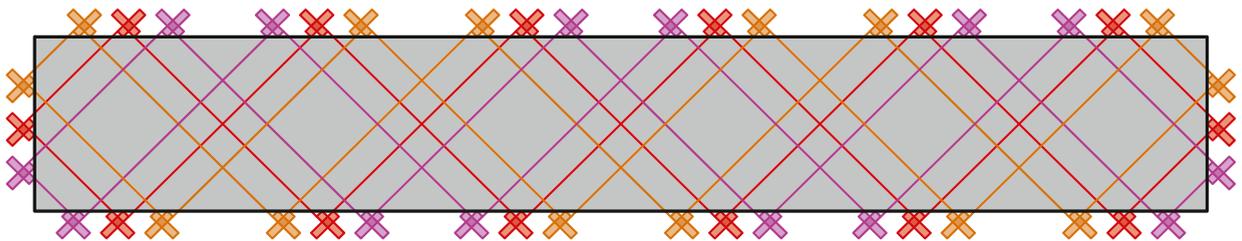


Fig. 11. Multiple independent variable loops.

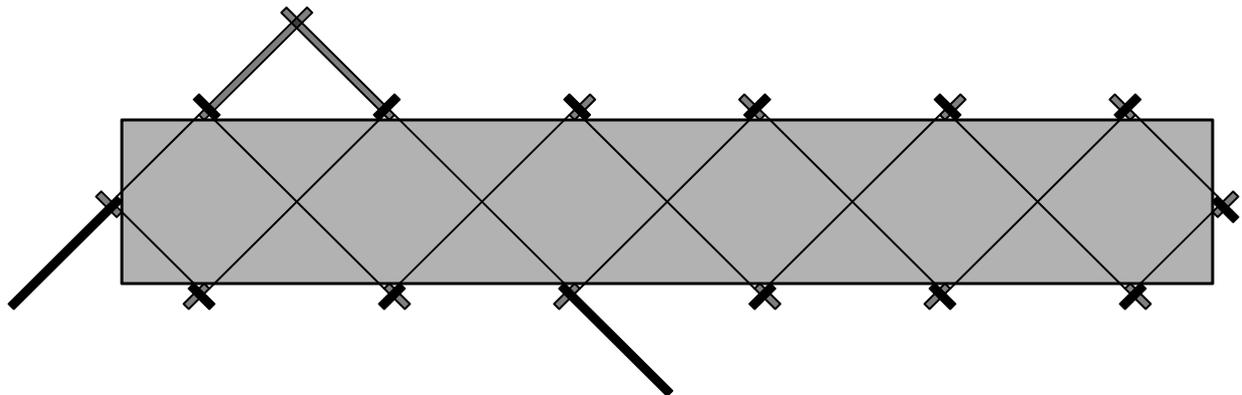
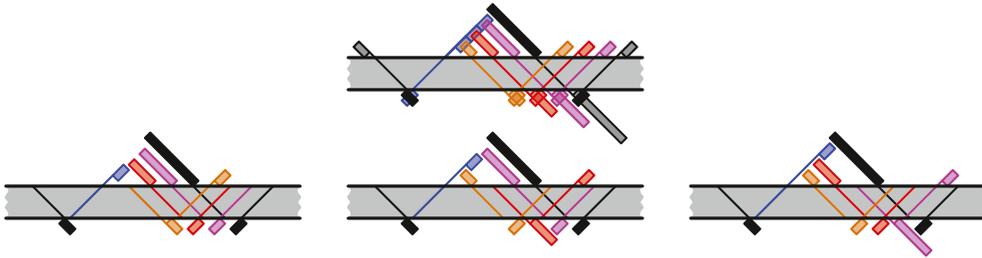


Fig. 12. A variable loop where one state is impossible. Solid black labels are forced; gray labels are alternative states which can never be present in a crossing-free solution.

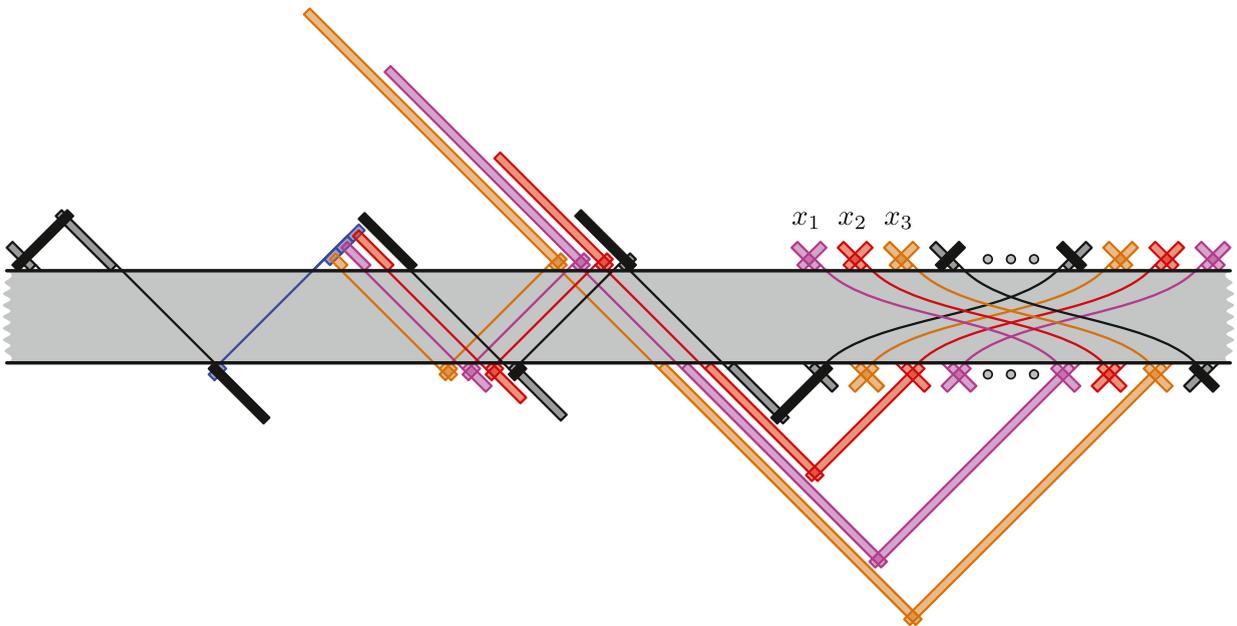
**Clauses** Next, we construct clause gadget. A clause gadget essentially consists of a single *clause* label which is forced to be at a specific port, but we can choose how far we extend it. Depending on how far it is extended, it will intersect different variable labels. The clause label is restricted to only three essentially different positions by two fixed labels. Fig. 13 illustrates a clause gadget and its valid solutions.

Note that for the clause label, it is important on which side of the line the label is placed: it must be faced towards the variable labels. The clause label is also the only label in the construction for which we use the possibility to extend its leader. We need to connect the clause gadget to the correct literals of the three variables involved in the clause, as well as to the fixed loop on two sides. For this, we need to make some very long labels. Fig. 14 illustrates how a clause gadget is connected, showing only the relevant labels. Note that the variables need to be connected to two different groups of cross gadgets in the variable loops, depending on whether or not they occur negated in the clause.

**The global picture** Globally, we embed the different clauses horizontally next to each other. Each clause, including its connections, covers a horizontal distance of a constant number of variable zig-zags. These connections are placed between the



**Fig. 13.** Clause gadget and three possible valid solutions. The black labels are forced. The blue label is the clause label, which must be placed at the top of the construction but can be extended by 0, 1, or 2 units. The orange, red, and purple labels are three different variables which will connect to the variable loops (refer to Fig. 14). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)



**Fig. 14.** Connecting a clause gadget to the correct literals. The picture corresponds to the clause  $\neg x_1 \vee x_2 \vee \neg x_3$ . In particular, note that the groups of variable gadgets on the frame are alternating in which direction corresponds to true and which to false, and note that the red label for  $x_2$  is connected to a different group than the purple and orange labels for  $\neg x_1$  and  $\neg x_3$ .

variable loops, so they do not interfere. Fig. 15 shows how the first clause and the beginning of the second clause could look globally, without hiding any labels.

Now, if the 3-SAT formula is satisfiable, then we put the variables in the corresponding states, and then each clause admits a non-crossing labeling. If the 3-SAT formula is not satisfiable, then no matter how we arrange the variable labels, there must be at least one clause gadget where all three positions for the clause label are blocked; hence, there is no non-crossing labeling.

The construction is polynomial in  $n$  and  $m$ . The total number of nonogram lines we place is  $O(nm)$ . The total horizontal distance covered will be  $O(nm)$ , and the vertical distance is  $O(n)$ , so all locations can be represented with polynomial coordinates.

We conclude:

**Theorem 3.** Given a nonogram instance without side assignment and extensible leaders, it is NP-complete to decide whether a valid labeling exists.

#### 4.2. Bounded label size

The construction above uses mostly only labels of constant size; however, we need a few long labels: they need to be of length linear in the number of variables in order to “jump over” sufficiently many variable gadgets. We next extend the proof to label sizes of 1 by  $c$ ; first we treat  $c$  as an abstract constant and then we argue that the problem remains hard

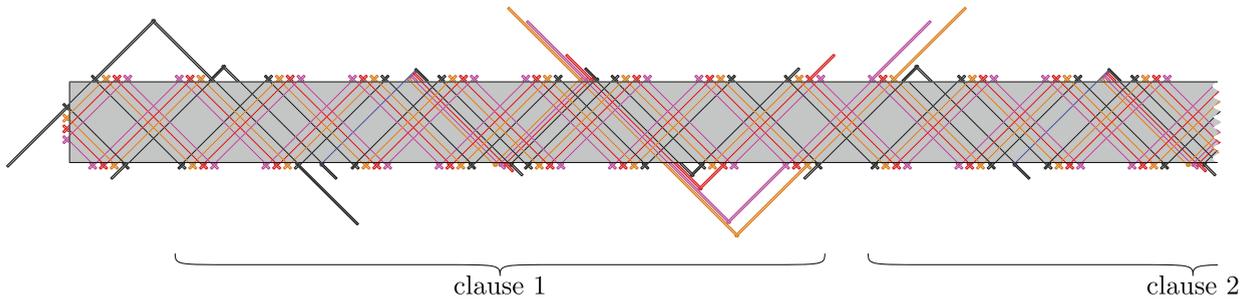


Fig. 15. The global picture.

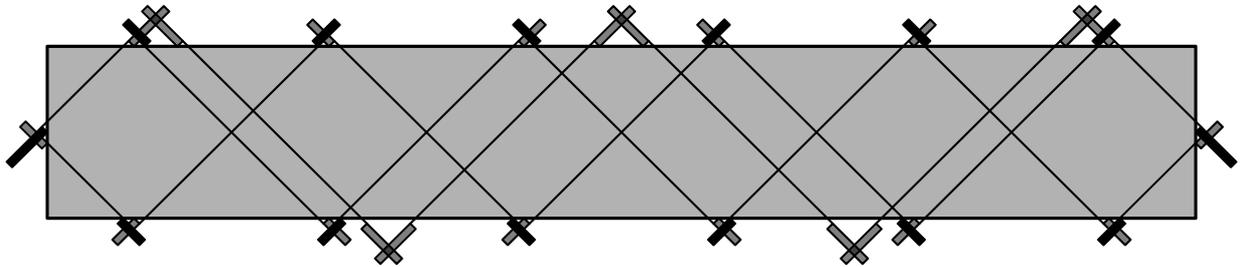


Fig. 16. A variable loop where one state is impossible; solid black labels are *forced*. Note that the additional labels on the path used to make one state infeasible themselves are not forced.

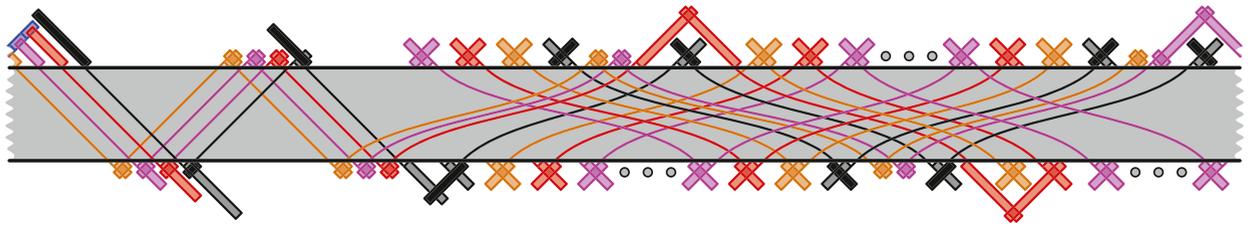


Fig. 17. Connecting clauses to variables using only constant-length labels. Connections are shown schematically.

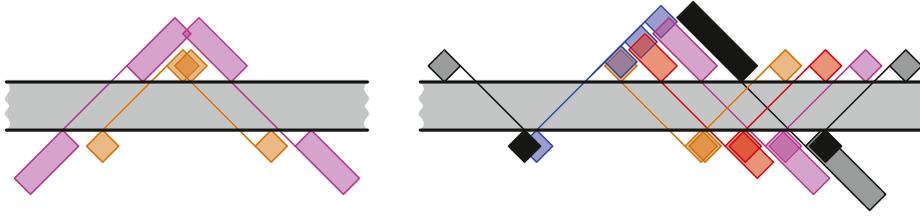
even for  $c = 4$ . The main idea is to replace each pair of long labels by a sufficiently large horizontal section, in which we make several shorter jumps instead. As a result, the total width of the construction will increase by a factor of  $\Theta(n)$ .

*Long forced labels* The first place where we need long labels is in the special variable loop which has a *forced* state. We need two labels of one of the potential states to intersect, in order to make it infeasible. Instead of doing this directly, we instead create a *fixing gadget*: a longer path between variable gadgets, jumping back and forth  $\Theta(n)$  times. Fig. 16 illustrates the idea.

We reserve a section of length  $\Theta(n^2)$  at the left end of the global construction to embed the fixing gadget, without interfering with the rest of the construction. Note that, depending on the exact length of the labels, we can freely choose any (constant) horizontal displacement, and it will be possible to “interweave” the fixing gadget with the other variable loops.

*Long labels to connect to clauses* The second place where we need long labels is in the connection of the clause gadget to the correct variable loops. Again, we replace each such connection by a longer *connection gadget*, which has, for each literal in the clause, a path which jumps between the top and bottom as often as necessary ( $\Theta(i)$  times for literal  $x_i$ , and  $\Theta(n + i)$  times for literal  $\neg x_i$ ), and we reserve a sufficiently long horizontal distance between consecutive clause gadgets to make this possible. Since each clause consists of three literals, at the start of each connection gadget three of these paths need to be embedded simultaneously, therefore requiring slightly wider gaps between consecutive variable loops. Fig. 17 illustrates the idea.

*Required label sizes* In order to make the adaptations above, we require labels to be “nested” by one level: a sequence of labels  $abcd$  such that  $a$  crosses  $d$  and  $b$  crosses  $c$ , but there are no crossings between  $a$  or  $d$  and  $b$  or  $c$ . This can be created using  $1 \times 1$  labels for  $b$  and  $c$ , and  $1 \times 3$  labels for  $a$  and  $d$ ; refer to Fig. 18 (left).



**Fig. 18.** (left) Two disjoint nested crossing pairs of labels require labels of size  $1 \times 3$ , and this is sufficient. Note that it does not matter whether the leaders are attached to the left or right size of the labels. (right) The clause gadget using only labels of size  $1 \times 1$ ,  $1 \times 2$ ,  $1 \times 3$ , and  $1 \times 4$ .

However, we need larger labels than this for the clause gadget. The clause gadget requires labels of three different sizes for the literals in the clause: we may use  $1 \times 1$ ,  $1 \times 2$  and  $1 \times 3$ . Then, the intersecting clause label with variable leader length can also be a  $1 \times 1$  label. Finally, we need a forced label that blocks the variable leader label, which must be disjoint from the three literal labels but does not need to extend beyond them; therefore, this is possible with a  $1 \times 4$  label. Fig. 18 (right) shows a precise construction.<sup>3</sup>

**Theorem 4.** *Given a nonogram instance without side assignment and extensible leaders, it is NP-complete to decide whether a valid labeling using labels of sizes up to  $1 \times 4$  exists.*

### 5. Unit-sized labels

In this section we consider labels of bounded size. By Theorem 4 we know that for labels of size at most  $1 \times 4$  the problem is NP-complete. In the following we prove that a special case for labels of size  $1 \times 1$  can be resolved in time  $O(n^3)$ .

For this special case we assume that the set of lines  $\mathcal{L}$  forms a regular grid, i.e., ports can only appear at equally spaced intervals. Note that this does not mean that the coordinates of the ports are rational, as the unit of the grid could be irrational, e.g., square-root two. As discussed in Section 2 this is not as big of a restriction as it might seem at first glance. To then make sure that the labels do not intersect on their boundaries we shrink them by an epsilon-amount compared to the unit of the grid. Moreover, we assume that all lines in  $\mathcal{L}$  have slope  $\pm 1$ , that the frame  $B$  is a rectangle, that the labeling we search is balanced, and every line  $l \in \mathcal{L}$  has two labels attached to it, i.e., in a valid nonogram labeling every port has a label placed at it. Now, the only remaining degree of freedom in this setting is to choose for each label on which side of the frame it is placed. As the labels are all  $1 \times 1$  boxes though and since every line has two labels and we require the labeling to be balanced, this can be thought of as choosing if at a port the label is right- or left-flipped. Note that, if labels of size two or three were allowed, this would not be the case and we would need to consider which of the two labels of a line is placed on which port of their line.

In the following we will call a port  $p$  full if there exist two lines  $l, m \in \mathcal{L}$  with slopes  $+1$  and  $-1$ , respectively, that intersect in  $p$ . Let  $p_l$  and  $p_m$  be ports on edge  $e$  of  $B$  at distance  $d \in \mathbb{N}$  (this is well defined as we assume the underlying grid to be regular), then there exist at most  $d - 1$  other ports  $p_{l+1} \dots p_{m-1} = p_{m+d-1}$  on  $e$  that lie in between  $p_l$  and  $p_m$  if  $d \geq 2$  and no other port if  $d = 1$ . To ease notation we sometimes also speak of the lines  $l$  and  $m$  being at distance  $d$  if the considered edge is clear from the context. For an edge  $e$  of  $B$  we call a mapping  $\sigma$  from the labels on ports along  $e$  to values in  $\{\text{left-flipped}, \text{right-flipped}\}$  a flip-assignment. Note that this is well defined since we only deal with unit-size labels and every line has exactly two labels.

The following definitions are described only for the top-edge of  $B$ ; for the other edges they can be defined analogously. Two lines  $l_1, l_2 \in \mathcal{L}$  with corresponding ports  $p_{l_1}$  and  $p_{l_2}$  are called opposing if  $p_{l_1}$ , and hence  $l_1$ , is left of  $p_{l_2}$  and  $l_1$  has slope  $+1$  and  $l_2$  has slope  $-1$ .

**Definition 1.** A flip-assignment  $\sigma$  is called extendable, if we find that for every two labels  $\ell_1, \ell_2$  on opposing lines  $l_1, l_2 \in \mathcal{L}$  at distance  $d \in \mathbb{N}$  with  $\ell_1$  right-flipped and  $\ell_2$  left-flipped it holds that  $d \neq 1$  and

$$\begin{cases} \text{if } d = 2 \Rightarrow \text{no third label is assigned to a port between } p_{l_1} \text{ and } p_{l_2}, \\ \text{if } d = 3 \Rightarrow \text{there are at most two labels at the ports between } p_{l_1} \text{ and } p_{l_2}, \\ \text{if } d \geq 4 \Rightarrow \text{not all ports between } p_{l_1} \text{ and } p_{l_2} \text{ are full.} \end{cases}$$

Definition 1 provides a set of conditions that must hold for a flip-assignment to be extendable without providing a way to compute the extension lengths. The definition of extendable does this by listing all forbidden patterns for two labels at

<sup>3</sup> Note that, if we require the labels to be aligned to an integer grid, we cannot realize the construction with these sizes anymore, and we will need larger constant-size labels.

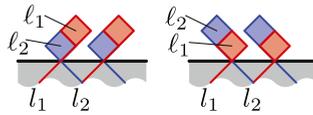


Fig. 19. Two two-blocks. The left figure shows a right-flipped two-block, the right figure a left-flipped one.

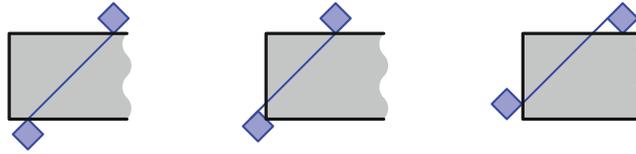


Fig. 20. For a straight line of slope  $\pm 1$  that crosses two opposing edges of  $B$ , it is possible for both labels to be flipped away from the frame, but for a line that crosses two adjacent edges, one of the labels must be flipped towards the frame.

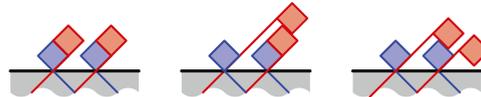


Fig. 21. All configurations that resolve two ports if there is no line at slope  $-1$  with its label left-flipped.

opposing lines where the one on the line with slope  $+1$  is right-flipped and the one on the line with slope  $-1$  is left-flipped. As we will see in the two subsequent lemmas (Lemma 3 and 4) this is sufficient, since only in the presence of one of these forbidden configurations the labeling might not be possible.

Finally, let  $p$  be a full port and  $l_1$  and  $l_2$  labels on lines  $l_1$  and  $l_2$  intersecting in  $p$ , such that  $l_1$  has slope  $+1$  and  $l_2$  slope  $-1$ , then we say  $l_1$  and  $l_2$  form a *two-block* if  $l_1$  is left-flipped and  $l_2$  is right-flipped. Two such labels can then be drawn in two ways such that the sum of the lengths of the extensions of  $l_1$  and  $l_2$  is one; see Fig. 19. If  $l_1$  is extended we call the two-block *right-flipped* and if this is true for  $l_2$  we denote it as *left-flipped*.

**Remark 1.** Intuitively, placing labels that are flipped to point away from a frame  $B$  are easier to place. Especially, if all labels could be flipped away from  $B$  we only need series of two-blocks to find a valid labeling. If  $B$  consisted only of two parallel lines this would be possible as all labels could be flipped such that they point away from the frame (see Fig. 20 on the left). In case  $B$  consists not just of two parallel edges, e.g., if  $B$  has edges orthogonal to each other, finding such a trivial labeling is not possible as for one line crossing two orthogonal edges and its two labels one label has to be flipped towards the frame (see Fig. 20 on the right). This makes the problem non-trivial.

We begin by considering nonogram instances with given flip-assignment per edge. For simplicity, our descriptions are written for the top-edge of  $B$ , but the statements hold analogously for every edge.

**Lemma 3.** Given a flip-assignment  $\sigma$  such that for no two labels  $l_1$  and  $l_2$  at opposing lines  $l_1$  and  $l_2$  we find that  $l_1$  is right-flipped and  $l_2$  is left-flipped, then there exist extension lengths such that the result is a valid nonogram labeling. Furthermore, such extension lengths can be found in linear time in the number of lines.

**Proof.** Straight-forwardly, if for no line  $l \in \mathcal{L}$  with slope  $+1$  its label is right-flipped and for no line  $l' \in \mathcal{L}$  with slope  $-1$  its label is left-flipped, the labels can be placed as a series of non-intersecting two-blocks.

Now, let the labels  $l_1, \dots, l_k$  at lines  $l_1, \dots, l_k$  with slope  $+1$  be right-flipped and assume that  $l_1, \dots, l_k$  are all such labels. Observe that, regardless of how the corresponding lines are extended, no two distinct labels  $l_i, l_j$  intersect. This means we only need to guarantee that none of the labels  $l \notin \{l_1, \dots, l_k\}$  and their lines intersect any label. This can be done by simply placing all such labels either with no extension, or, if the corresponding block is full, as a right-flipped two-block. Fig. 21 illustrates this. The labels  $l_1, \dots, l_k$  can then always be resolved by extending their lines  $l_1, \dots, l_k$  by one if there is no port one unit to the right, by two if there is one label at the port one unit to the right, and by three if that port is full. To compute such a labeling we can simply traverse the ports from right to left in  $O(n)$  time. We can argue symmetrically for left-flipped labels at lines with slope  $-1$  and no right-flipped labels at lines with slope  $+1$ , by exchanging right- and left-flipped as well as letting the algorithm go from left to right.

Finally, if there are labels  $l_1$  and  $l_2$  at lines  $l_1, l_2 \in \mathcal{L}$  with slopes  $+1$  and  $-1$  respectively, such that  $l_1$  is right-flipped and  $l_2$  is left flipped, then we know that  $l_1$  and  $l_2$  are not opposing by assumption. Hence, the port corresponding to  $l_2$  is never right of  $l_1$ 's port. Consequently, we can split the instance at the rightmost such label  $l_1$  and line  $l_1$  when traversing the ports along the boundary from left to right.  $\square$

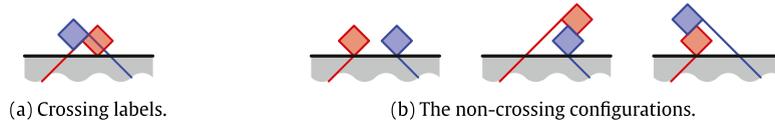


Fig. 22. The four possibilities how two lines at distance one can interact.

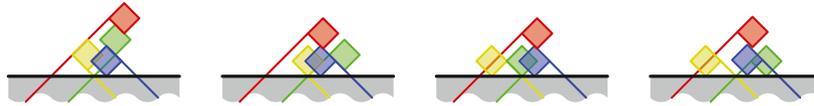


Fig. 23. The cases of two opposing lines at distance two and how they interact with a full port in-between.

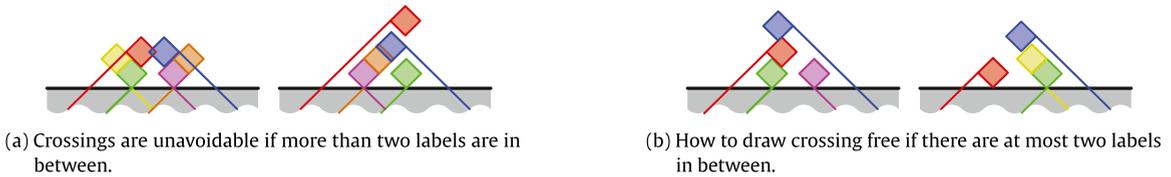


Fig. 24. The cases of two opposing lines at distance three and how they interact with four, three, and two labels in between.

Next we show that for one edge and a given flip-assignment a valid nonogram labeling with the desired constraints exists if and only if the flip-assignment is extendable.

**Lemma 4.** *Given a nonogram labeling instance with fixed side assignment  $\sigma$ , extensible leaders, and all labels of size  $1 \times 1$ , then there exists a valid nonogram labeling of the instance respecting  $\sigma$  if and only if  $\sigma$  is extendable.*

**Proof.** Let  $\mathcal{L}$  denote the set of lines given with the nonogram labeling instance. We begin by assuming that there exists a valid nonogram labeling in which the labels are flipped according to  $\sigma$ , yet  $\sigma$  is not extendable. By Definition 1 not being extendable requires that there are at least two labels, say  $\ell_1$  and  $\ell_2$ , at opposing lines  $l_1, l_2 \in \mathcal{L}$  such that the corresponding labels  $\ell_1$  and  $\ell_2$  are right-flipped and left-flipped respectively. Note, that  $l_1$  and  $l_2$  are extended by at least one unit in any valid labeling of this instance. Let  $d \in \mathbb{N}$  be the distance between  $l_1$  and  $l_2$  and  $p_1$  and  $p_2$  the two ports of  $l_1$  and  $l_2$ . Recall that then there are at most  $d - 1$  other ports between  $p_1$  and  $p_2$ . We proceed via case analysis over  $d$ .

- $d = 1$  As noted,  $l_1$  and  $l_2$  are extended by at least one unit, yet there is only one unit of space between the corresponding ports and  $\ell_1$  and  $\ell_2$  are of size  $1 \times 1$ . Hence, they have to intersect, see also Fig. 22a for an illustration.
- $d = 2$  Since  $\sigma$  is not extendable we find that there is a non-empty port between  $p_1$  and  $p_2$ , say  $p'$ . Let  $l' \in \mathcal{L}$  be the line corresponding to  $p'$  and  $\ell'$  its label at  $p'$ . Without loss of generality assume  $l'$  has slope  $+1$ . By the previous case we know that  $\ell'$  must be left-flipped. For  $\ell'$  and  $\ell_1$  to not intersect we extend  $l_1$  by another unit to two units. Note that now  $l_2$  cannot be extended more than one unit without  $\ell_1$  and  $\ell_2$  intersecting. Though, if  $l_2$  is extended only one unit we get that  $\ell_2$  is intersecting  $\ell'$ . Since we must extend  $l_2$  this is inevitable. See Fig. 23 for an illustration.
- $d = 3$  Let  $p'_1$  and  $p'_2$  be the two ports in between  $p_1$  and  $p_2$ , which exist by assumption. Moreover, again since  $\sigma$  is not extendable we find that there are more than two labels at the ports  $p'_1$  and  $p'_2$ , consequently, at least one of them is full. Without loss of generality let  $p'_1$  be the full port. Clearly, in any valid labeling we have to place its two labels in a right-flipped two-block, see also Fig. 24a. Now we can argue as in the case of  $d = 2$  that, w.l.o.g.  $l_1$  has to be extended three units. As a consequence  $\ell_2$  intersects either the label at  $p'_2$ , a label in the two-block at  $p'_1$ , or  $\ell_2$ .
- $d \geq 4$  By Definition 1 we get that all ports between  $p_1$  and  $p_2$  are full. Without loss of generality we can assume that all the labels at these ports are arranged as two-blocks and that they all are right-flipped. Clearly, we can extend  $l_1$  by three units such that  $\ell_1$  is above the first two-block. Though for  $l_2$  this is not possible as the two-block directly left of it would need to be left-flipped. But this is impossible as the left-flipped two-block will intersect the previous right-flipped two-block. Fig. 25a illustrates this case.

The above case distinction leads to a contradiction. Consequently,  $\sigma$  has to be extendable.

For the other direction of the proof, assume that  $\sigma$  is extendable. We claim that then there always exists a valid nonogram labeling and that the labels are flipped as prescribed by  $\sigma$ . By Lemma 3 we know that the case of no two opposing lines having their labels right- and left-flipped can be dealt with.

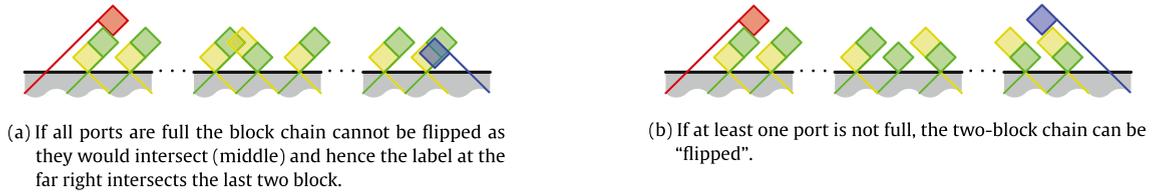


Fig. 25. The case of two opposing lines at distance four and how they interact with a chain of two-blocks.

Now, consider two opposing lines  $l_1, l_2 \in \mathcal{L}$ , with labels  $\ell_1$  and  $\ell_2$  at ports  $p_1$  and  $p_2$ . Moreover, assume that  $\ell_1$  is right-flipped and  $\ell_2$  left-flipped. Since  $\sigma$  is extendable we know that  $l_1$  and  $l_2$  have distance  $d > 1$ . Denote with  $q_1, \dots, q_{d-1}$  the at most  $d - 1$  ports between  $p_1$  and  $p_2$ . If  $d = 2$  or  $d = 3$  we can easily find extension lengths for the leaders and additionally no label at a port left of  $p_1$  or right of  $p_2$  can intersect a label at a port  $q_i, i \in \{1, 2\}$ . See Fig. 24b for an illustration of case  $d = 3$ .

Let us assume  $d \geq 4$  and that for no  $q_i$  we find a line intersecting the edge of  $B$  at  $q_i$  such that it is opposing to  $l_1$  or  $l_2$  and its label is left-flipped or right-flipped, respectively. Let  $q_i, i \in \{1, \dots, d - 1\}$ , be a non-full port. Using Lemma 3 we know that all the labels at  $q_1, \dots, q_{i-1}$  can be drawn as right-flipped two-blocks and all labels at ports  $q_{i+1}, \dots, q_{d-1}$  as left-flipped two-blocks. Extending  $l_1$  and  $l_2$  the necessary two or three units yields the labeling. Fig. 25b illustrates this.

Next, assume that there is at least one port  $q_i$  with a line  $m_i$  intersecting the edge in  $q_i$  and such that  $m_i$  has slope  $+1$  and its label is right-flipped or it has slope  $-1$  and its label is left-flipped. Let  $q_i$  be such a port with line  $m_i$  at slope  $+1$  and its label right-flipped or  $-1$  and its label left-flipped. Without loss of generality assume that  $m_i$  has slope  $+1$ . We assume that between  $p_1$  and  $q_i$  there are no more such lines. Using the above discussion as an induction basis, we can assume by induction that all labels attached to lines intersecting ports between  $q_i$  and  $p_2$  are already drawn. Then, by Lemma 3 we can draw all the labels attached to lines intersecting the edge in ports  $q_1, \dots, q_i$ . Moreover, none of these labels or their leaders can intersect a label at a line intersecting the edge in port  $q_h$  that lies between  $q_i$  and  $p_2$ , as we never extend a leader more than three units. Hence, we can also determine the extensions of the lines  $l_1$  and  $l_2$  to be one, two, or three units. □

It remains to prove our main theorem. On a high level we are going to construct a 2-SAT formula with one variable  $x_l$  per line  $l \in \mathcal{L}$ . This formula models the conditions of a flip-assignment being extendable. From a satisfying truth-assignment we can then read off how to flip the labels. Using Lemma 4 we then know a valid nonogram labeling exists and we compute it recursively by considering sections of labels between opposing lines where the label at the left line is right-flipped and the label on the right line is left-flipped.

**Theorem 5.** Given a nonogram labeling instance without side assignment, extensible leaders with slopes  $\pm 1$ , ports at distance at least  $\sqrt{2}$ , and all labels of size  $1 \times 1$ , we can decide in time  $O(n^3)$  if a balanced labeling exists.

**Proof.** We construct a 2-SAT formula as follows. For every line  $l \in \mathcal{L}$  we define one variable  $x_l$ . To treat the other edges of  $B$  correctly we consider them all rotated such that they are parallel to the top edge of  $B$  and such that if we traverse the ports from left to right the nonogram interior would be on the right-hand side. Consequently, the slope of lines relative to the left and right edge flip their sign, i.e., a line with slope  $+1$  has slope  $-1$  when we consider the left or right edge. We describe the creation of the formula  $\varphi$  for one edge  $e$  of  $B$ . Note that in the end we take the conjunction of the formulas created for each edge and in this way guarantee that the obtained labeling is consistent around the whole frame. For every pair of opposing lines  $l, m \in \mathcal{L}$  with  $l \neq m$  and both intersecting  $e$ , let  $d$  be the distance between the ports  $p_l$  and  $p_m$  of  $l$  and  $m$  along  $e$ . We introduce a clause  $x_l \vee x_m$  to the 2-SAT formula whenever one of the following holds: (i)  $d = 1$ , (ii)  $d = 2$  and there is a third port between  $p_l$  and  $p_m$ , (iii)  $d = 3$  and there are two ports between  $p_l$  and  $p_m$  such that more than two lines pass through these, or (iv)  $d \geq 4$  and all ports between  $p_l$  and  $p_m$  are full.

We interpret the truth-values of an assignment of  $\varphi$  as follows. If for a label  $\ell$  at a line  $l \in \mathcal{L}$  with slope  $+1$  we find  $x_l$  set to *true* in the assignment we flip  $\ell$  to the left and to the right otherwise. If  $l$  was at slope  $-1$  and  $x_l$  set to *true* we would flip  $\ell$  to the right and to the left otherwise. In this manner, we can immediately conclude that whenever a clause evaluates to *false* in an assignment of the formula  $\varphi$  we violate Definition 1 as we then constructed two opposing lines with labels not fulfilling one of its four conditions. Now, if the assignment is satisfying no clause evaluates to *false* and for every clause evaluating to *true* we find that the conditions of Definition 1 are in fact met, i.e., there are no two opposing lines with labels fulfilling any of its four conditions. Consequently, we obtain the following. Let  $\varphi$  be the formula constructed in this manner from a nonogram labeling instance with the above conditions. If  $\varphi$  is satisfiable, then the induced flip-assignment is extendable for each edge and hence with Lemma 4 there exists a valid nonogram labeling with the desired conditions.

Next we argue that if there exists an extendable flip-assignment  $\sigma$  for each edge of  $B$ , then  $\varphi$  is satisfiable. We begin with the top edge of  $B$ . Let  $\ell$  be a label on a line  $l \in \mathcal{L}$  that intersects the top edge of  $B$ . We set the truth value of the corresponding variable  $x_l$  to *false* if  $\ell$  has slope  $+1$  and its label is right-flipped or it has slope  $-1$  and its label is left-flipped and to *true* otherwise. We do the same for all other edges. Note that this can never lead to a contradictory

truth-assignment as the labeling has to be balanced. Now assume that this derived assignment is not satisfying  $\varphi$ . This means there exists a clause which evaluates to false. Let  $\ell_1$  and  $\ell_2$  be two labels of lines  $l_1$  and  $l_2$  in  $\mathcal{L}$  such that their corresponding variables form this clause  $x_{l_1} \vee x_{l_2}$ . By construction, this clause was only introduced if one of the conditions in Definition 1 was fulfilled. Both literals being *false* though implies that the labels are such that one of these conditions is violated, a contradiction with Lemma 4 as the resulting flip-assignments are not extendable. Hence, the assignment is satisfying  $\varphi$ .

It remains to argue the running time. Computing the side assignment  $\sigma$  is done by solving the 2-SAT formula which is clearly in size  $O(n^2)$ . Hence solving it takes also time  $O(n^2)$  [18]. The extension lengths can then be computed in linear time if the conditions of Lemma 3 are also fulfilled or by using one of the constant number of possibilities for ports at distance two or three. In every other case we recursively compute the solution of the labels in between pairs of opposing lines with their labels being right-flipped and left-flipped. Recall that since we do not need any extension lengths of more than three units when applying Lemma 3, the extension lengths for the two surrounding opposing lines can be easily computed and are independent from the remaining lines and labels. Doing so in the fashion of Lemma 4 yields a running time of  $O(n^3)$  as there are at most  $O(n^2)$  such pairs. Hence the whole algorithm runs in time  $O(n^3)$ .  $\square$

**Remark 2.** As in Section 3.2 the labels could intersect around corners. For the here considered  $1 \times 1$  unit size this is not a problem as the spacing of the labels is of the same size. For longer label lengths a similar argumentation as in Section 3.2 would be required.

## 6. Conclusions

In this paper we initiated the study of labeling nonograms. While the arguably most natural version of the problem proved to be NP-complete already for constant sized labels, we identified several restricted cases which can be solved in polynomial time.

Even though the versions we solve apply only in restricted settings and the polynomials are of rather high degree, we do believe our algorithms may be of practical value. The reason for this is three-fold:

1. in nonogram puzzles  $n$  is typically not very large, since otherwise it would not be fun as a puzzle anyway;
2. the high dependence on  $n$  is to a large degree based on worst-case configurations which arguably are not likely to occur;
3. some of our restrictions, such as having slopes of  $45^\circ$  and uniform spacing, may in fact be met by real puzzles because of aesthetical considerations.

We remark that the labeling problem is independent of the interior of a curved nonogram, and depends only on the set of ports. Hence, our results apply in principle to both curved and slanted nonograms. We do note that some restrictions such as slopes  $\pm 1$  or equally spaced ports are more likely to arise for slanted nonograms.

*Future work* Several interesting questions in nonogram labeling remain open. When aiming for a compact labeling it is an interesting open question if the number of lines whose labels are placed at opposite ports can be maximized in an efficient manner. Further, we considered here the objective of minimizing the total leader length. For practical puzzle design, however, it may also be interesting to minimize the space on a piece of paper occupied by the labels outside the actual puzzle. This translates into minimizing the area of the bounding box, which is determined by the longest leaders on each edge of the frame. In regard to the NP-completeness result, recall that we could show in Theorem 4 that the problem remains NP-complete even if we restrict all labels to be of size  $1 \times 4$ . Naturally, one can ask if the length 4 is required or if the problem remains NP-complete for even smaller labels. Moreover, since the result in Theorem 5 requires that the labeling is balanced and each line has to have two labels attached to it, it is an interesting open question if the problem actually remains NP-complete for constant-size labels under the same restrictions.

Another important direction for future research is to investigate to what extent the polynomial-time solutions extend to nonograms with more, or different, slopes, or even to curved nonograms where each port may have a distinct slope. For curved nonograms, it would also be interesting to investigate a model where the extensions outside the frame may also be curved, giving more options to avoid crossings. Similarly, it would be interesting to see to what extent other boundary shapes can be supported.

Finally, it would be of interest to test our algorithms on real puzzles, to investigate whether the running times are acceptable and whether the resulting layouts are usable in practice.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful suggestions and comments. The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

## References

- [1] K.J. Batenburg, W.A. Kosters, On the difficulty of nonograms, *ICGA J.* 35 (4) (2012) 195–205, <https://doi.org/10.3233/ICG-2012-35402>.
- [2] D. Berend, D. Pomeranz, R. Rabani, B. Raziel, Nonograms: combinatorial questions and algorithms, *Discrete Appl. Math.* 169 (2014) 30–42, <https://doi.org/10.1016/j.dam.2014.01.004>.
- [3] Y. Chen, S. Lin, A fast nonogram solver that won the TAAI 2017 and ICGA 2018 tournaments, *ICGA J.* 41 (1) (2019) 2–14, <https://doi.org/10.3233/ICG-190097>.
- [4] M. van de Kerkhof, T. de Jong, R. Parment, M. Löffler, A. Vaxman, M.J. van Kreveld, Design and automated generation of Japanese picture puzzles, *Comput. Graph. Forum* 38 (2) (2019) 343–353, <https://doi.org/10.1111/cgf.13642>.
- [5] T.K. de Jong, The concept and automatic generation of the curved nonogram puzzle, Master's thesis, Utrecht University, 2016, <https://dspace.library.uu.nl/handle/1874/337632>.
- [6] R.J. Parment, Generation of sloped nonograms, Master's thesis, Utrecht University, 2015, <https://dspace.library.uu.nl/handle/1874/323196>.
- [7] M.A. van de Kerkhof, Improved automatic generation of curved nonograms, Master's thesis, Utrecht University, 2017, <https://dspace.library.uu.nl/handle/1874/357864>.
- [8] M.A. Bekos, B. Niedermann, M. Nöllenburg, External labeling techniques: a taxonomy and survey, *Comput. Graph. Forum* 38 (3) (2019) 833–860, <https://doi.org/10.1111/cgf.13729>.
- [9] M.A. Bekos, M. Kaufmann, A. Symvonis, A. Wolff, Boundary labeling: models and efficient algorithms for rectangular maps, *Comput. Geom. Theory Appl.* 36 (3) (2007) 215–236, <https://doi.org/10.1016/j.comgeo.2006.05.003>.
- [10] M. Benkert, H. Haverkort, M. Kroll, M. Nöllenburg, Algorithms for multi-criteria boundary labeling, *J. Graph Algorithms Appl.* 13 (3) (2009) 289–317, <https://doi.org/10.7155/jgaa.00189>.
- [11] M. Bekos, M. Kaufmann, M. Nöllenburg, A. Symvonis, Boundary labeling with octilinear leaders, *Algorithmica* 57 (2010) 436–461, <https://doi.org/10.1007/s00453-009-9283-6>.
- [12] P. Kindermann, B. Niedermann, I. Rutter, M. Schaefer, A. Schulz, A. Wolff, Multi-sided boundary labeling, *Algorithmica* 76 (1) (2016) 225–258, <https://doi.org/10.1007/s00453-015-0028-4>.
- [13] B. Niedermann, M. Nöllenburg, I. Rutter, Radial contour labeling with straight leaders, in: *IEEE Pacific Visualization Symposium, PacificVis'17, 2017*, pp. 295–304.
- [14] M.A. Bekos, M. Kaufmann, A. Symvonis, Efficient labeling of collinear sites, *J. Graph Algorithms Appl.* 12 (3) (2008) 357–380, <https://doi.org/10.7155/jgaa.00170>.
- [15] M.A. Bekos, M. Kaufmann, K. Potika, A. Symvonis, Multi-stack boundary labeling problems, in: S. Arun-Kumar, N. Garg (Eds.), *Foundations of Software Technology and Theoretical Computer Science, FSTTCS'06*, in: LNCS, vol. 4337, Springer, 2006, pp. 81–92.
- [16] A. Gemsa, J.-H. Haunert, M. Nöllenburg, Multi-row boundary-labeling algorithms for panorama images, *ACM Trans. Spat. Algorithms Syst.* 1 (1) (2015) 1, <https://doi.org/10.1145/2794299>.
- [17] M. Formann, F. Wagner, A packing problem with applications to lettering of maps, in: *Computational Geometry, SoCG'91, ACM, 1991*, pp. 281–288.
- [18] B. Aspvall, M.F. Plass, R.E. Tarjan, A linear-time algorithm for testing the truth of certain quantified Boolean formulas, *Inf. Process. Lett.* 8 (3) (1979) 121–123, [https://doi.org/10.1016/0020-0190\(79\)90002-4](https://doi.org/10.1016/0020-0190(79)90002-4).