Labeling Nonograms

Maarten Löffler Martin Nöllenburg

October 25, 2019

1 Introduction

2 Preliminaries

For the nonogram labeling problem we have the following general input:

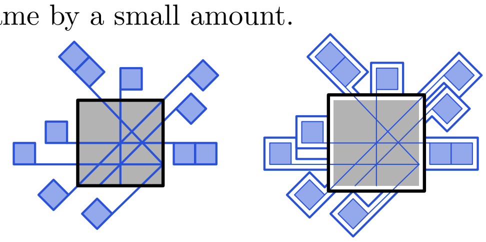
- A *nonogram frame*, which is a simple convex polygon[1] B.
- A set $\mathcal{L}$ of *nonogram lines* passing through B, each $l \in \mathcal{L}$ defining a pair (p_l, q_l) of *ports* in the intersection points of l with B.
- A pair of non-negative integers (a_l, b_l) for each $l \in \mathcal{L}$, where a_l defines the width of the label above l and b_l defines the width of the label below l.

The desired output is a labeling of $\mathcal{L}$ which is defined as follows.

- For each $l \in \mathcal{L}$ we assign each of its two labels to one of the two ports.
- For each label ℓ with a port assignment, we optionally define an extension length d.
- We draw each label ℓ of a nonogram line l as an $a_l \times 1$-rectangle (or $b_l \times 1$-rectangle) aligned with the slope of l and anchored[2] at distance d from its assigned port p_l or q_l.
- The resulting placement of all labels for $\mathcal{L}$ must be free of label overlaps and no two leaders (extensions of a line $l \in \mathcal{L}$ to connect port and label) may intersect outside B.

2.1 Clearance

In fact, 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.

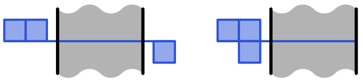


[1] we may also want to allow circles

[2] define precisely which of the four corners is the anchor point

2.2 Sidedness

In principle, there is a choice for each label on which side of the frame it gets placed - both would lead to a valid puzzle.

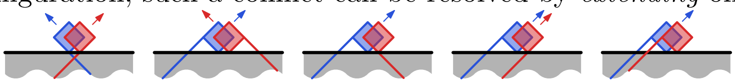


We probably would like the labels to be placed in a consistent way, because it looks nicer. However, putting more restrictions on this might make it impossible to place the labels. Possible features of the label sides we might like are:

- For each port, we only place one label.
- For each port, the label is always on the right when exiting the nonogram frame.

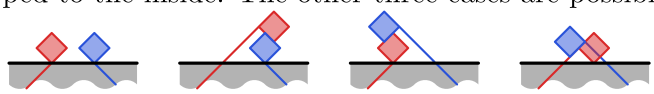
3 Some Interesting Observations

In the case where all ports have 45° angles, there are several ways in which two labels can be intersecting when we place them as close to the border as possible. Depending on the configuration, such a conflict can be resolved by *extending* one or both of the leaders.

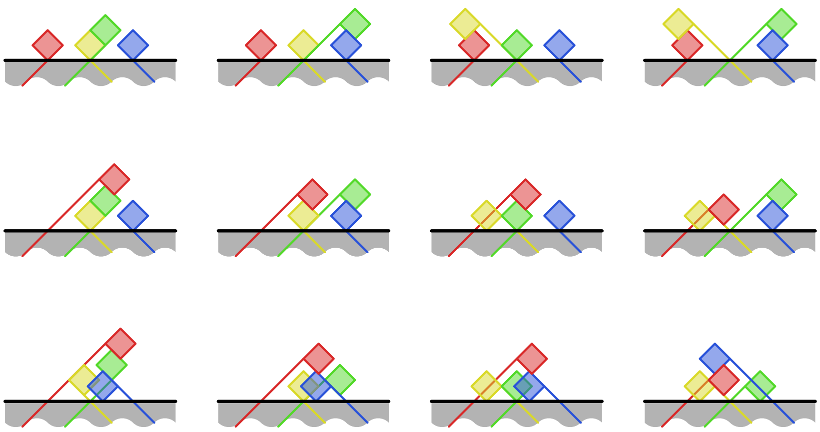


It might also be possible to resolve conflicts by *flipping* one of the labels. Note that this only makes sense if we also flip the label on the opposite side of the line, so this operations creates dependencies between different sides of the puzzle.

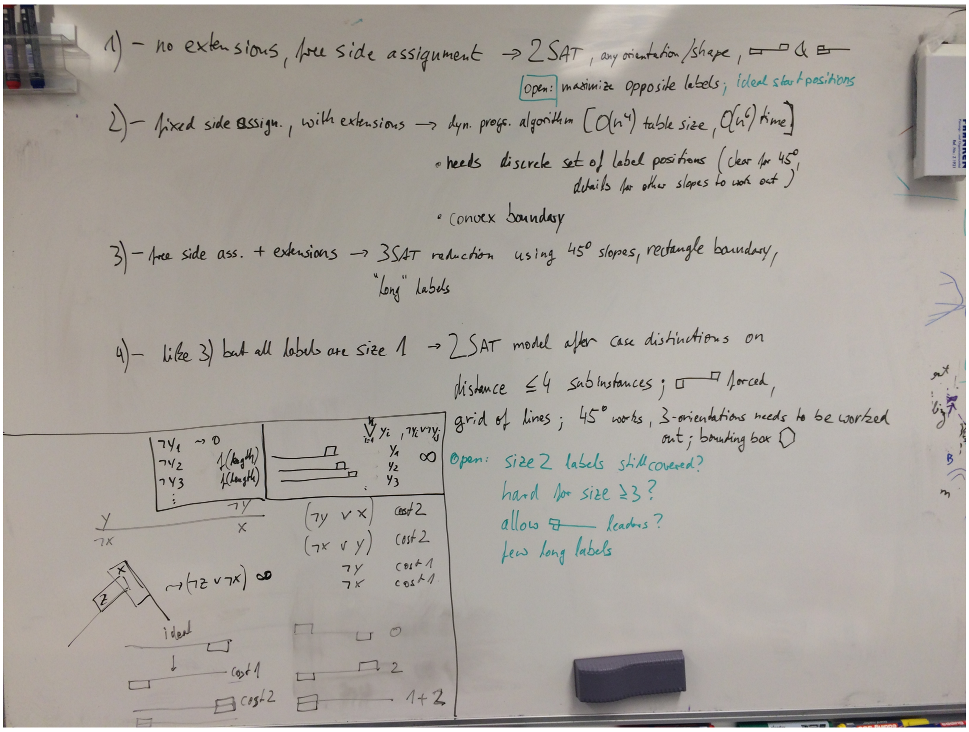
If the two labels come in at distance 1 from each other, labeling is impossible if both are flipped to the inside. The other three cases are possible.



If the two labels come in at distance 2 from each other, whether they can be labeled also depends on other labels between them.



4 Results



4.1 Non-extensible leaders

Theorem 1. *Given a nonogram labeling instance, we can decide in polynomial time whether a labeling exists such that no leader is extended. This is true regardless of whether both labels of a nonogram line can be assigned to the same port or not.*

Proof. Sketch.

The idea is to derive a Boolean 2-Sat formula φ that has a satisfying variable assignment if and only if a valid labeling exists.

For each nonogram line $l \in \mathcal{L}$ we define two variables x_l and y_l, where $x_l = 1$ ($x_l = 0$) indicates that the label above l is assigned to the left (right) port of l. Similarly, $y_l = 1$ ($y_l = 0$) indicates that the label below l is assigned to the right (left) port of l.

For each $l \in \mathcal{L}$ we add the clauses $\neg x_l \vee y_l$ and $x_l \vee \neg y_l$ if we want to exclude two label assignments to the same port of l.

Further, 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 label above l and the label below l' intersect if both assigned to their left ports. Then we add the clause $\neg x_l \vee y_{l'}$.

Now a satisfying assignment for the set of clauses 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.

Solving the 2-Sat instance takes linear time in the size of the formular φ, where the number of clauses is linear in the number of nonogram lines and the number of label conflicts. □

Remarks

- Result holds regardless of the line orientations, the shape of the frame, or even arbitrary curves inside B.
- It is an interesting open question to maximize (efficiently) the number of lines whose labels are placed at opposite ports.

4.2 Fixed side assignment

Theorem 2. *Given a nonogram labeling instance and a fixed side assignment for each label, we can decide in $O(n^6)$ time, whether an assignment of an extension length to each label exists such that the resulting labeling is valid. If this is the case we can find one of minimum total extension length.*

Proof. The idea of the algorithm is to use dynamic programming. Consider one edge e of B and all the ports on e. A first observation is that for any label it is sufficient to consider only a linear number of extension lengths. These positions will be defined by the intersections of each nonogram line with the other nonogram lines (possibly shifted by 1 for the label height).[3] We define a subinstance of the labeling problem for edge e by selecting two boundary lines l_1 and l_2 together with an extension length for each of the two labels. This defines $O(n^4)$ possible subinstances. Any line with a port between those of l_1 and l_2 is restricted to remain between l_1 and l_2, i.e., its extension length must be auch that it does not extend beyond the subinstance's enclosed region. To solve such an instance recursively, we optimize over all labels $\hat{l}$ contained in the instance and all possible and intersection-free extension lengths for that label and recurse into the two subinstances defined by l_1 and $\hat{l}$ as well as $\hat{l}$ and l_2. The optimization step takes $O(n^2)$ time for each subinstance. This yields the overall $O(n^6)$ running time. □

Remarks

- For arbitrary slopes we still need to argue that only a linear number of label positions is relevant.
- Labels of different edges of B may interfere and this still needs attention.

4.3 Hardness

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

Proof. The proof is by reduction from 3-Sat; see Figure 1. Only the clause lines may need extensions. □

4.4 Bounded label size

Theorem 4. *If each label is a 1×1-square, we can decide existence of a valid nonogram labeling in polynomial time. Requirements: $\mathcal{L}$ forms a regular grid with only two slopes intersecting each edge of B; slopes may need to be 45° or behave similarly to that case (hexagonal grid may work). The two labels of each line must be on opposite ports.*

Proof. Again the idea is to use a 2-Sat model. Here the main observation is that some local subinstances with lines up to four grid ports apart have solutions if and only if certain constraints on the side assignment of pairs of the involved labels hold. These can again be expressed by 2-Sat clauses.

A case distinction is necessary, see right side of Figure 1 for some examples. □

Remarks

- Can this be extended to size 1 and 2 labels?
- Can we prove hardness for size 3 labels?
- What happens if we may put two labels to the same port?

[3] We argued about the case of 45° diagonals; for other slopes the contact points of two labels do not immediately produce a discrete set of interesting positions!

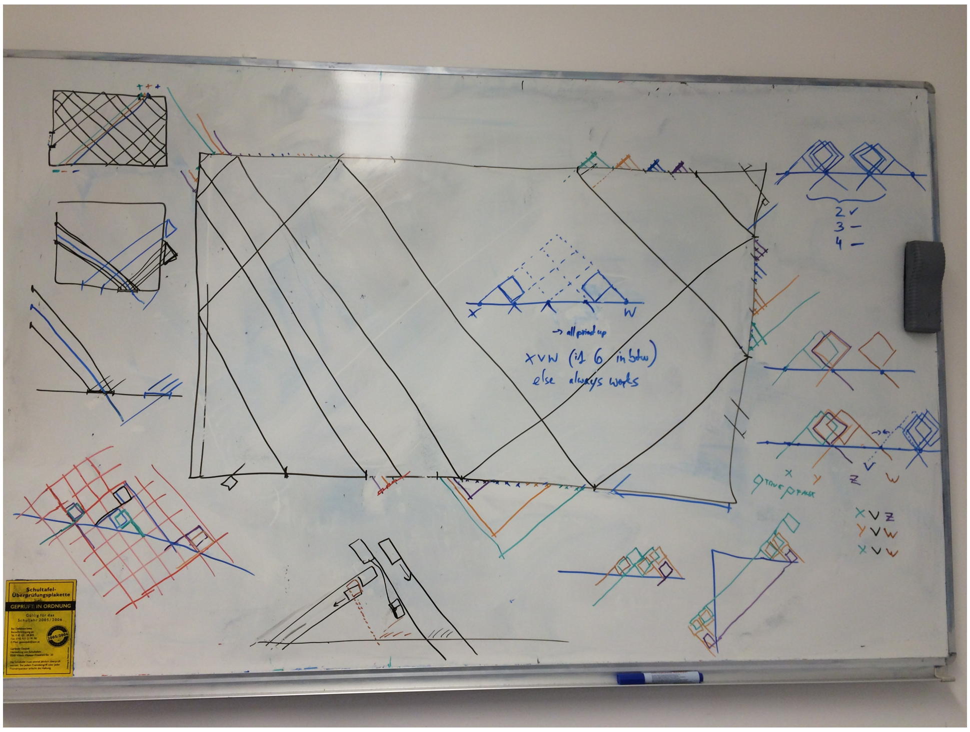


Fig. 1: Hardness proof.