Labeling Sloped Nonograms

Eva Timmer

January 28, 2021

1 Introduction

2 Related Work

2.1 Nonograms

There have been several studies on generating and solving nonograms.

Simple nonograms are nonograms that have an unique solution and can be solved by looking at only one row or column at a time.

There is a classification of difficulty of simple nonograms [1].

Different methods have been proposed for solving nonograms using heuristics [5], depth-first search [9] and combining relaxations [2].

Solving nonograms has been proven to be NP-hard [6].

There have been studies on the generation of classic nonograms as well as on newer variations of nonograms, like colored nonograms, curved nonograms [7] and sloped nonograms. With the latter two types, the problem of labeling becomes considerably more complicated.

2.2 Labeling

The problem of labeling nonograms is related to the problem of boundary labeling.

Algorithms have been proposed to solve the problem of boundary labeling. There are algorithms that optimize different aspects, like minimizing bends or minimizing length [4].

There have been defined several leader types, like straight (s), orthogonal (o), perpendicular (p), diagonal (d) and variations thereof (op, opo) [3].

Nonogram labeling is like boundary labeling with straight leaders that are restricted in their direction, but can have variable length and are restricted to two possible ports on either side of the line. With nonogram labeling whatever is going on inside the boundary cannot be changed to suit the labeling.

2.3 Puzzle/label aesthetics

Vollick et al. [8] have defined several properties of good label layouts and state that preventing overlap between labels is absolutely crucial for a legible layout. Another property that could be applied to nonogram labeling are that they should be as close to the figure as possible up to a minimum buffer distance.

rephrase (basically copied)

They also provide an enegry function that can be used to measure the quality of a label configuration.

3 Definitions

3.1 Nonograms

Classic nonograms are grid puzzles with a uniform grid. Each row and column has an associated clue, which is a sequence of numbers. Each number represents an uninterrupted sequence of cells in that row or column that should be colored in. Together these clues describe how the whole grid is to be colored in. A simple nonogram has exactly one valid unique solution.

The clues for a classic nonogram are generally placed above or to the left of the corresponding column or row.

Sloped nonograms do not not adhere to a uniform grid, but includes lines with a range of orientations. In this case each line has two descriptions, one for the adjecent cells above the line, and one for the adjecent cells below the line.

For sloped nonograms, the clues are placed at the start or end of each line. The line can be extended past the boundary, to avoid overlapping clues.

Curved nonograms are quite similar to sloped nonograms, but have curved instead of straight lines. The clues are placed the same way.

3.2 Nonogram labeling

A set of clues for one line/row/column we call a *label*. Each label can be placed on the boundary in one of two ways. For a classic nonogram, the label can be placed on either side of each row or column. For sloped and curved nonograms, each label can be placed on either intersection of the line with the boundary. These points on the boundary are called *ports*. In the case of classic nonograms, the labels can be placed in any way and will never overlap, because of the uniform grid. However, for the other two types, this is not usually the case.

Unless the line is perpendicular to the boundary, it needs to be extended in order to avoid overlapping the label with the boundary. This part of the line that is extended past the boundary we call the *leader*. The leader can be further extended to avoid overlapping between labels. A choice of labels and leabel leaders is called a *labeling*. There may not always be a labeling that has no overlapping labels.

4 Nonogram labeling quality

What about using leaders that don't follow the direction of the line but are perpendicular to the boundary? Is this less readable? Does this make the problem easier?

What about leaving out labels to avoid overlap, how does this effect the quality of the puzzle?

What about allowing a small amount of overlap, is there something to say about the negative effect on aesthetics vs the positive effect of labeling efficiency?

5 Algorithms

5.1 Overview

5.2 Input

B: the boundary this can be a rectangle or a different shape depending on the type of nonogram and the limits of the algorithm

Set of unplaced labels For each side of a line there are two possible ports, and the clues for that side of the line. Each port has a position and direction.

5.3 Non-extensible leaders

It uses SAT2

There is a variable for every possible placement of a label.

There is a clause that makes sure exactly one port is chosen for every line-side.

There is a clause for every pair of overlapping labels, that makes sure at most one of them is chosen.

5.3.1 SAT and discrete amount of leader lengths

Is it possible to use a SAT2 impentation with extensible leaders? With the possible lengths being 1..max

So the clauses would be:

Variable for every possible placement of a label.

Clause that makes sure exactly one port-leader length combo is chosen.

Clause for every pair of overlapping labels, that makes sure at most one of them is chosen.

5.4 Fixed side assignment

It uses dynamic programming

5.5 Relaxed direction

Is it possible to create an algorithm for sloped nonograms that has a range of allowed directions instead of one?

6 Experiments

6.1 Puzzles

Weinig tot veel lijnen.

Weinig tot veel orientaties. (Curved = veel)

6.2 Complexity

How well does algorithm x work with 10,20...n labels?

How much easier does the problem become with more allowed directions?

6.3 Simplifying the problem: allow overlap

How much easier does the problem become when you allow 1,2...n cases of overlap.

6.4 simplifying the problem: leave out labels

How much easier does the problem become when you leave out 1,2..n labels?

References

[1] K Joost Batenburg, Sjoerd Henstra, Walter A Kosters, and Willem Jan Palenstijn. Constructing simple nonograms of varying difficulty. *Pure Mathematics and Applications (Pu. MA)*, 20:1–15, 2009.

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

[3] Michael A Bekos, Michael Kaufmann, Martin Nöllenburg, and Antonios Symvonis. Boundary labeling with octilinear leaders. *Algorithmica*, 57(3):436–461, 2010.

[4] Michael A Bekos, Michael Kaufmann, Antonios Symvonis, and Alexander Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry*, 36(3):215–236, 2007.

[5] Sancho Salcedo-Sanz, Emilio G Ortiz-Garca, Angel M Pérez-Bellido, Antonio Portilla-Figueras, and Xin Yao. Solving japanese puzzles with heuristics. In *2007 IEEE Symposium on Computational Intelligence and Games*, pages 224–231. IEEE, 2007.

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

[7] MA van de Kerkhof. Improved automatic generation of curved nonograms. Master's thesis, 2017.

[8] Ian Vollick, Daniel Vogel, Maneesh Agrawala, and Aaron Hertzmann. Specifying label layout style by example. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 221–230. ACM, 2007.

[9] Chiung-Hsueh Yu, Hui-Lung Lee, and Ling-Hwei Chen. An efficient algorithm for solving nonograms. *Applied Intelligence*, 35(1):18–31, 2011.