

Human and Computational Measures of Nonogram Puzzle Difficulty

Changdao He, Yibing Ju, Juliana Zhang | Supervisors: Jonathan Calver, Alice Gao

Introduction

- A Nonogram is a grid-based logic puzzle where row and column clues indicate lengths of filled blocks, separated by at least one empty cell. This structure makes it a constraint satisfaction problem.
- Nonograms can be formulated as **Boolean satisfiability (SAT)** problems to analyze difficulty using automated solvers. However, solver-based difficulty may not reflect human experience.

How do SAT solver statistics correlate with human evaluations of puzzle difficulty?

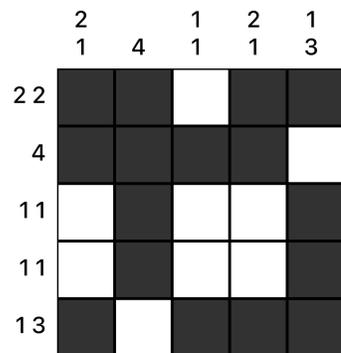


Figure 1. Example 5 x 5 Nonogram

SAT Problem Formulation

High-Level Idea

Each row and column clue restricts where filled blocks can appear. We encode these restrictions as **Boolean variables and constraints**, allowing a SAT solver to search for valid assignments. The encoding enforces block placement, order, spacing, and consistency with filled cells.

Figure 2. A row with clues 1 4

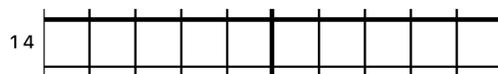


Figure 3. Earliest feasible placement



Figure 4. Latest feasible placement



Example Encoding

Consider a row of length 10 with clue [1,4]. The two blocks must appear in order and be separated by at least one empty cell. By packing the blocks as far left as possible, we get the **earliest feasible placement**; by shifting the pattern as far right as possible, we get the **latest feasible placement**. These two extreme placements determine the valid start range for each block.

In this example, the 4-block cannot start before cell 3, and the 1-block cannot start after cell 5. We use these bounds to create Boolean variables only for feasible block starts, which keeps the SAT encoding compact.

We combine these block-start variables with cell variables that indicate whether each grid cell is filled. The SAT constraints then enforce valid block placement, correct ordering, required separation, and consistency between row and column clues.

References & More Information

- Gilles Audemard, Laurent Simon. On the Glucose SAT solver. *International Journal on Artificial Intelligence Tools (IJAIT)*, 2018, 27 (1), pp.1-25. (hal-03299473)
- Sörensson, N. MINISAT 2.2 and MINISAT++ 1.1; 2010.



SCAN ME

Access the participant information collection form here.

Solver-Derived Difficulty Measures

Solvers Used

- Primary:** MiniSat2.2 (Sörensson, 2010)
Secondary: Glucose4.2 (Audemard & Simon, 2018)

Collected Metrics

Decisions	solver chooses a variable value based on a heuristic
Propagations	solver deduces a variable assignment that must be true
Conflicts	a clause becomes false under the current partial assignment

We generated 1000 random, unique solution 50% density puzzles, and selected the puzzles used in the experiment based on the conflicts metric.

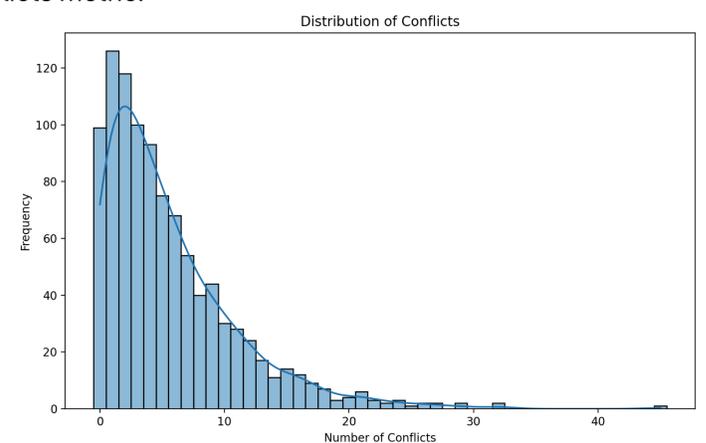


Figure 5. Distribution of the conflicts solver metric over 1000 puzzles

Human-Subject Study

Experimental Design

- Aim to recruit 80 participants
- Study completed individually using a web-based platform
- Each session includes a tutorial, a warm-up puzzle, a pre-survey, three 50% density puzzles, and a post-survey
- Different sequences of puzzles are presented to reduce learning and fatigue effects

Measured Variables

Behavioural	solving time, actions, hints, incorrect submissions
Subjective	difficulty rating, guessing frequency
Background	puzzle experience, skill level

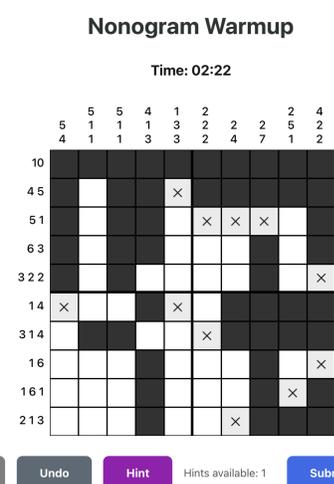


Figure 6. Nonogram puzzle UI, implemented by Python and React