Optimized CNF Encoding for Sudoku

Ashwot Acharya , Lakki Thapa , Bishesh Bohora , Supreme Chaudhary

Your University

February 19, 2026

Motivation

Sudoku can be encoded as a SAT problem.

- Each cell (r, c) contains a value $v \in \{1, \ldots, n\}$
- Represent using Boolean variables:

$$x_{r,c,v}$$

- True iff cell (r, c) has value v

Goal:

- Convert Sudoku constraints into CNF
- Solve using a SAT solver

Variable Encoding

For an $n \times n$ Sudoku:

$$V = \{(r, c, v) \mid 1 \leq r, c, v \leq n\}$$

Total variables:

$$n^3$$

Example (9×9):

$$9^3 = 729 \text{ variables}$$

Core Constraints

Cell constraints

- Each cell has at least one value
- Each cell has at most one value

Row constraints

- Each value appears once per row

Column constraints

- Each value appears once per column

Block constraints

- Each value appears once per block

Previous Encodings

From: *Optimized CNF Encoding for Sudoku Puzzles*
Kwon & Jain :contentReference[oaicite:1]index=1
Three encodings:

- Minimal
- Efficient
- Extended

All produce:

$$O(n^4) \text{ clauses}$$

Problem:

- Very large CNF for large grids (e.g., 81×81)

Extended Encoding

Extended encoding includes:

- Cell definedness
- Cell uniqueness
- Row definedness
- Row uniqueness
- Column definedness
- Column uniqueness
- Block definedness
- Block uniqueness

Produces many redundant clauses.

Optimized Encoding Idea

Key observation:
Fixed cells imply:

- Some variables are already TRUE
- Many others are forced FALSE

Therefore:

- Remove redundant literals
- Remove satisfied clauses

This reduces:

- Number of variables
- Number of clauses

Variable Partition

Variables are partitioned:

$$V = V^+ \cup V^- \cup V'$$

- V^+ : true variables (fixed cells)
- V^- : forced false variables
- V' : remaining unknown variables

Reduction operators eliminate:

- Clauses already satisfied
- Literals known to be false

Effect of Optimization

From experimental results :contentReference[oaicite:2]index=2:

- Variables reduced up to 12×
- Clauses reduced up to 79×
- Significant SAT solving speedup

Example (81×81):

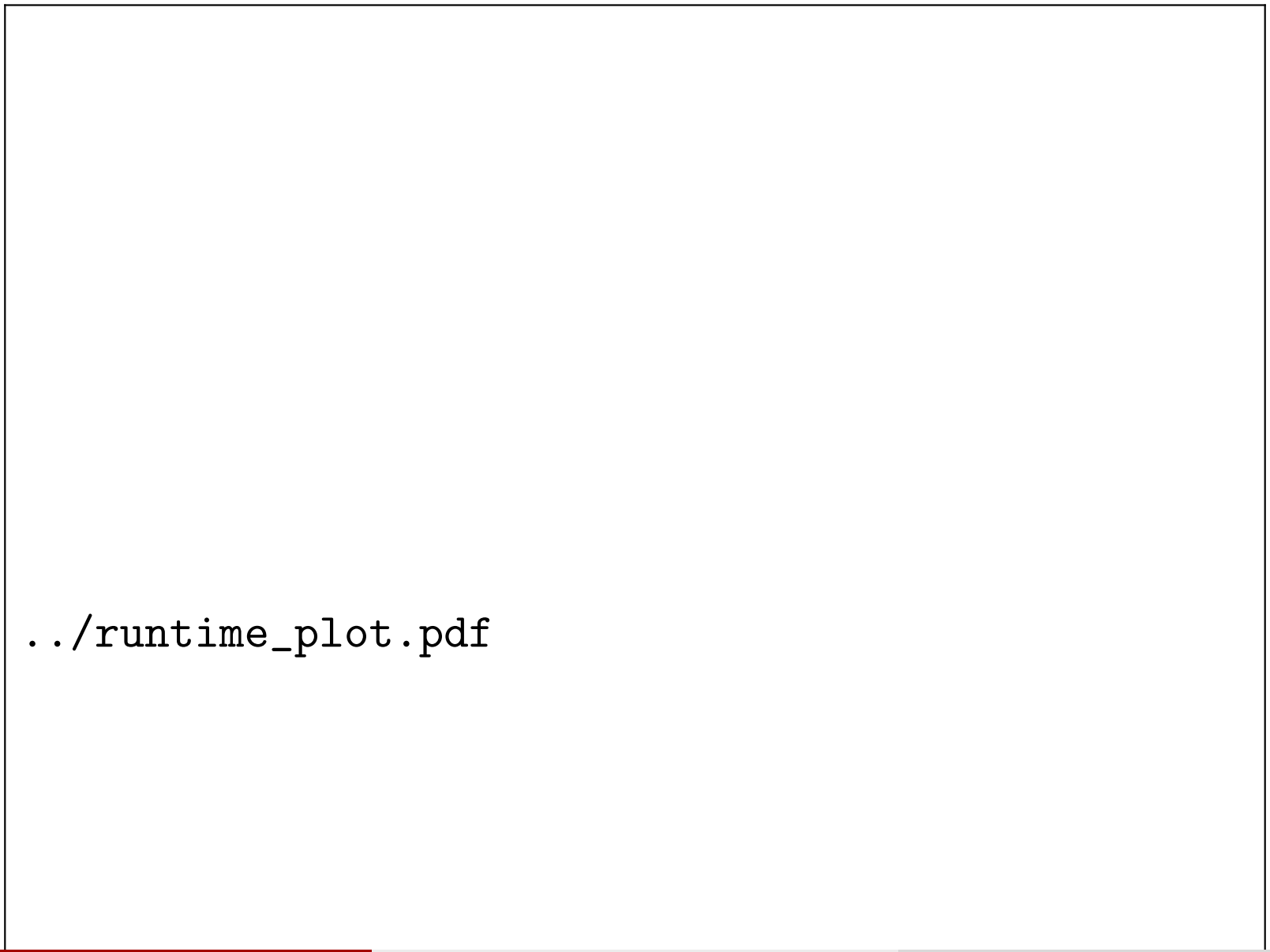
- Extended: 85M clauses
- Optimized: 266K clauses

Our Implementation

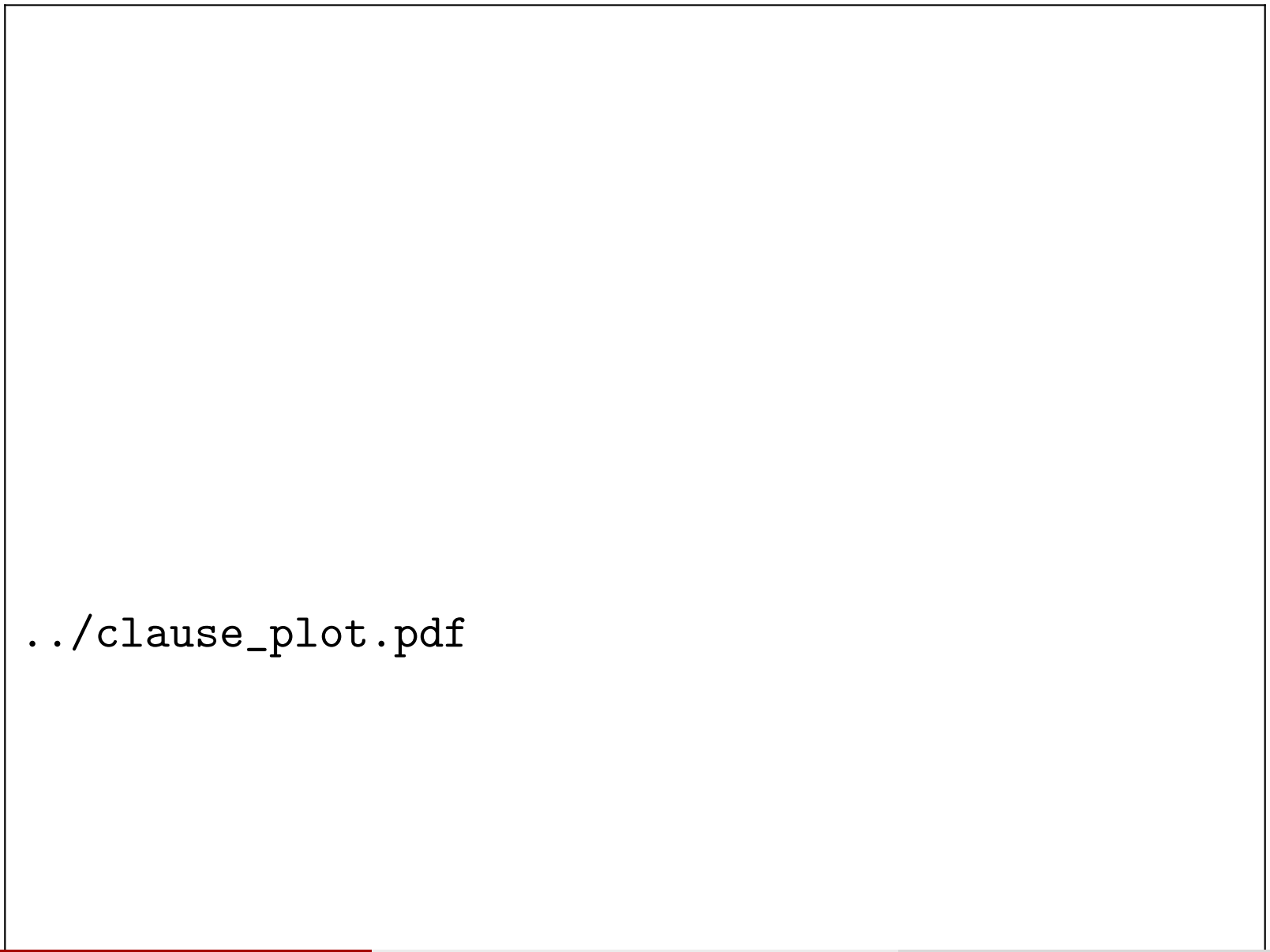
We implemented:

- CNF encoding in DIMACS format
- Optimized preprocessing for fixed cells
- Benchmarks on:
  - 9×9 (17 clues)
  - 9×9 (20+ clues)
  - Larger grids

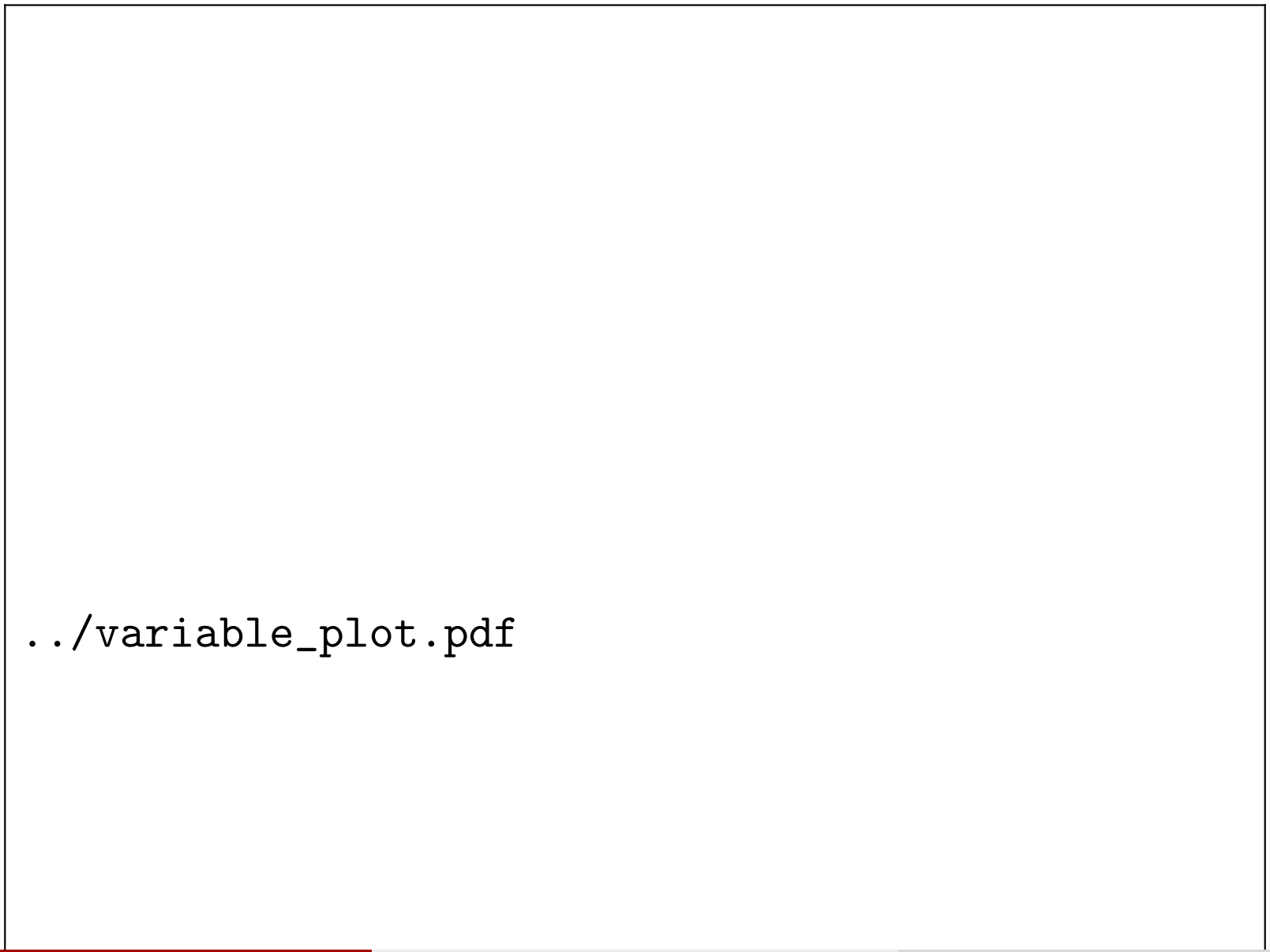
Benchmark Results – Runtime



Benchmark Results – Clause Count



Benchmark Results – Variable Count



Observations

- More clues $\Rightarrow$ fewer variables
- More clues $\Rightarrow$ fewer clauses
- Optimized encoding significantly reduces CNF size
- SAT solver runtime strongly correlates with clause count

Conclusion

- Sudoku naturally maps to SAT
- Naive encodings produce large CNF formulas
- Using fixed-cell reduction:
  - Shrinks formula
  - Improves solver performance
- Effective for larger Sudoku sizes

References

Kwon, G., Jain, H.
Optimized CNF Encoding for Sudoku Puzzles :contentReference[oaicite:3]index=3
Lynce, I., Ouaknine, J.
Sudoku as a SAT Problem
Moskewicz et al.
Chaff: Engineering an Efficient SAT Solver