
Report for Programming Assignment 3 (Sudoku)

Matthew Callahan

Suphalerk Lortaraprasert

Abstract

1 Introduction

SuDoKu is a challenge problem, being both a puzzle and a constraint satisfaction problem. In this game, each unit (rows, columns, and boxes) must adhere to the "all-diff" constraint, meaning every digit within a unit must be unique. The 81 squares on the SuDoKu grid are akin to variables, each with a domain from 1 to 9.

2 Environment Descriptions

2.1 Constraint propagation

In the constraint propagation step, the SuDoKu board undergoes a series of rule applications in the following order. This process aims to systematically decrease the domain values associated with each variable:

Rule 1: If a cell has only one value remaining in its domain, assign that value to the cell.

Rule 2: If a value x is not present in the domain of any other cell within the same row, column, or box as a particular cell, assign x to that cell.

Rule 3: Identify k squares within any row, column, or box, where each square shares a domain containing the same k numbers or a subset of those numbers. Then, eliminate those k numbers from the domains of all other cells within that unit. This is referred to as the "naked double" and "naked triple" rule when $k=2$ and $k=3$, respectively.

2.2 Backtracking Search

3 Experimental setup

3.1 Fixed Baseline

3.2 Most Constrained Variable

4 Results

4.1 Backtracking

4.2 Runtime Performance

5 Discussion

puzzle number	Arbitrary Order Backtracks	Most Constrained Variable Backtracks
1	5	0
6	13	0
7	10	0
20	10	0
30	8	0
41	12	0
42	6	0
43	13	0
44	9	0
50	6	0
55	11	0
61	11	0
62	5	0
67	7	0
68	5	0
69	7	0
70	12	1

Table 1: Number of backtracks for various algorithms for the easy puzzles