Relative Changes in removal of constraints in 3D Sudokus

Samarth Bhargav, Alexander Geenen

October 3, 2017

University of Amsterdam

Video: https://youtu.be/dTnoKvseOFI

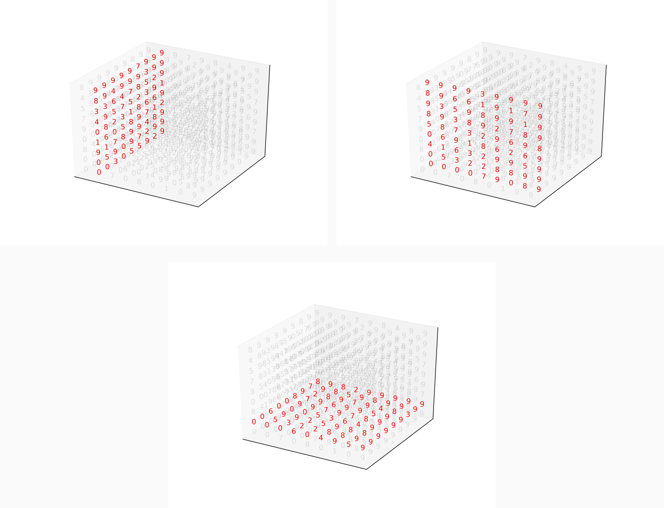
Github Link:

https://github.com/samarthbhargav/constraint-removal-3d-sudoku

Table of Contents

Table of Contents

Example



The 3 views of a 3D Sudoku

Constraints

- The `digit` constraint: Only one number can be placed in a position

Constraints

- The `digit` constraint: Only one number can be placed in a position
- The `x` direction constraint: In the x direction, all n numbers should be present, and should appear exactly once

Constraints

- The `digit` constraint: Only one number can be placed in a position
- The `x` direction constraint: In the x direction, all n numbers should be present, and should appear exactly once
- The `y` direction constraint: In the y direction, all n numbers should be present, and should appear exactly once

Constraints

- The `digit` constraint: Only one number can be placed in a position
- The `x` direction constraint: In the x direction, all n numbers should be present, and should appear exactly once
- The `y` direction constraint: In the y direction, all n numbers should be present, and should appear exactly once
- The `z` direction constraint: In the z direction, all n numbers should be present, and should appear exactly once

Constraints

- The `digit` constraint: Only one number can be placed in a position
- The `x` direction constraint: In the x direction, all n numbers should be present, and should appear exactly once
- The `y` direction constraint: In the y direction, all n numbers should be present, and should appear exactly once
- The `z` direction constraint: In the z direction, all n numbers should be present, and should appear exactly once
- The `box` constraint: In the x-y plane, each cell of size $\sqrt{n} \times \sqrt{n}$ should contain all n numbers exactly once. These cells are arranged just like the box constraint of a 2D Sudoku: n boxes tiled in a $\sqrt{n} \times \sqrt{n}$ grid

Table of Contents

Hypothesis

The *relative change* in **complexity** between subsequent degrees of constraints will be the same.

Dataset

- Sourced from `http://www.menneske.no/sudoku3d/eng/`
- 5354 3D Sudokus

Dataset

- Sourced from `http://www.menneske.no/sudoku3d/eng/`
- 5354 3D Sudokus
- Block constraint on the x-y plane

PICOSAT

- Easy to use - it has `python` bindings!

PICOSAT

- Easy to use - it has `python` bindings!
- Fast

SAT Solver

PICOSAT

- Easy to use - it has `python` bindings!
- Fast
- Deterministic

Metric

The `level` metric is defined as the total number of levels divided by the number of decisions. The lower the level, the more complex the Sudoku.

Table of Contents

Encoding

At Most One: $\wedge_{i=1}^{n-1} \wedge_{j=1}^{n} (\neg x_i \vee \neg x_j)$

Encoding

At Most One: $\wedge_{i=1}^{n-1} \wedge_{j=1}^{n} (\neg x_i \vee \neg x_j)$

At Least One: $\vee_{i=1}^{n} x_i$

Encoding

At Most One: $\wedge_{i=1}^{n-1} \wedge_{j=1}^{n} (\neg x_i \vee \neg x_j)$

At Least One: $\vee_{i=1}^{n} x_i$

Exactly One: $\left(\wedge_{i=1}^{n-1} \wedge_{j=1}^{n} (\neg x_i \vee \neg x_j)\right) \wedge (\vee_{i=1}^{n} x_i)$

Table of Contents

Setup - 1

- `all`: All 4 constraints were enforced
- `3-constraint`: Only 3 constraints out of 4 were enforced
- `2-constraint`: Only 2 constraints out of 4 were enforced
- `1-constraint`: Only 1 constraint out of 4 was enforced

Setup - 2

- Ran all combinations of the constraints
- 15 combinations $\times$ 5354 $=$ 80310

Setup - 2

- Ran all combinations of the constraints
- 15 combinations $\times$ 5354 $=$ 80310
- Some puzzles time out!

Setup - 2

- Ran all combinations of the constraints
- 15 combinations $\times$ 5354 $=$ 80310
- Some puzzles time out!
- Capped at 30 seconds of computation time

Setup - 3

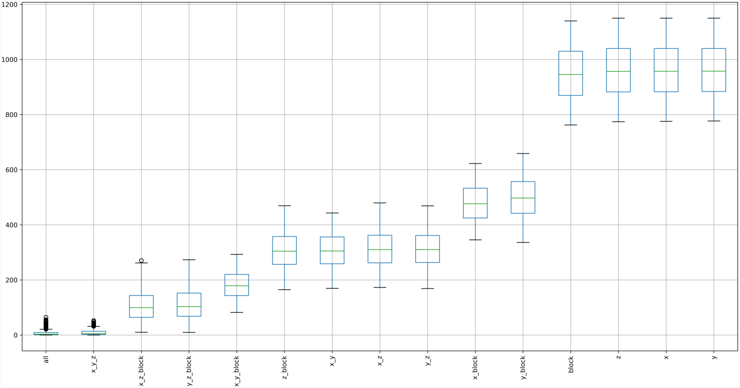
`level` metrics were then averaged per constraint combination

Table of Contents

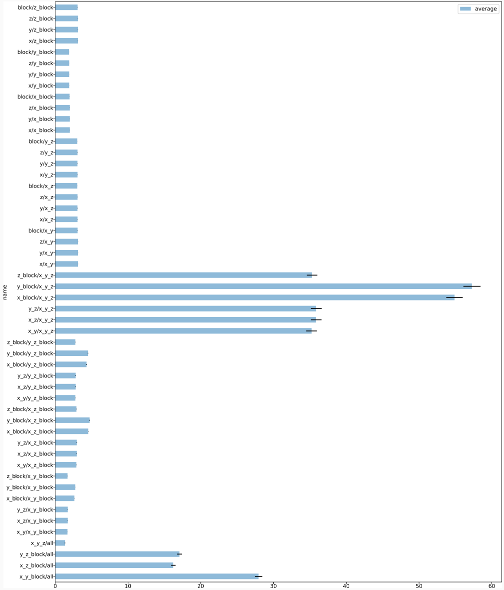
Results

Constraint	Timeouts	Average `level`
`all`	367	6.507894
`x-y-block`	0	181.843220
`x-z-block`	0	105.605603
`y-z-block`	0	111.127232
`x-y-z`	2980	8.734983
`x-y`	0	307.721815
`x-z`	0	313.153474
`y-z`	0	313.309301
`x-block`	0	479.311524
`y-block`	0	500.234516
`z-block`	0	308.272768
`x`	0	960.703399
`y`	0	960.182013
`z`	0	960.736272
`block`	0	948.700504

Box plot for `level`



The average `level` for all constraints



Relative change in the average `level`

Table of Contents

Conclusions - 1

Since there is a large variation in these levels, we can conclude that our original hypothesis is false

Conclusions - 2

Considering the average of the `level` metrics for the 2 have overlapping uncertainty bounds, these can be considered close

Conclusions - 2

Considering the average of the `level` metrics for the 2 have overlapping uncertainty bounds, these can be considered close

Considering the timeouts, the results indicate that the removal of the `block` constraint appears to make the problem **more** difficult for the picosat solver

Conclusions - 3

$\implies$ The search space is far more nuanced than we assumed
The most difficult constraint combination was not the combination of all of the constraints, but was found to be the combination of all 3 row constraints

Future Work - 1

- Optimized encoding schemes

Future Work - 1

- Optimized encoding schemes
- Larger dataset

Future Work - 1

- Optimized encoding schemes
- Larger dataset
- Remove or increase runtime cap

Future Work - 2

- Higher-dimensional Sudokus

Future Work - 2

- Higher-dimensional Sudokus
- Larger dataset

Future Work - 2

- Higher-dimensional Sudokus
- Larger dataset
- $>$ order

Future Work - 2

- Higher-dimensional Sudokus
- Larger dataset
- $>$ order
- $<$ order

Thank you!