Difficulty in solving a M x N Bridge Sudoku

Akash Raj Komarlu Narendra Gupta
Jeroen Baars

University of Amsterdam

Abstract

In this report, we investigate the difficulty in solving a bridge sudoku with an arbitrary overlap. We encode bridge sudoku based on the minimal encoding scheme. The statistics given by the minisat solver - number of conflicts, conflict literals, decisions and unit propagations are used to comment on the difficulty of solving the sudoku. We found that considering the above statistics, it is easier to solve a bridge sudoku against solving the corresponding standard sudokus. The addition of more constraints reduces the size of the solution space. On exploring the relationship between the bridge-givens and the difficulty, we could not find a correlation and requires additional analysis.

1. The Bridge Sudoku Problem

Bridge sudokus are defined as two sudokus having a common overlapping (bridge) region of size M x N. M and N denote the row and column overlap respectively. For a valid solution, the bridge follows the constraints of both the sudokus.

$$0 \leq M, N \leq \ \ 9$$
$$0 \leq M\text{x}N \leq 81$$

Our goal is to investigate the difficulty of solving a M x N bridge sudoku as compared to solving a normal sudoku puzzle. Due to the additional constraints in the bridge sudoku, we can expect that it is easier to solve a normal sudoku.

We intuitively claim that a bridge sudoku with no overlap is only as difficult as solving two normal sudokus. A bridge sudoku with a total overlap is also as 'difficult' as solving a normal sudoku. Therefore, as the overlap area increases, we can expect the difficulty initially increases upto a maximum and then decreases.

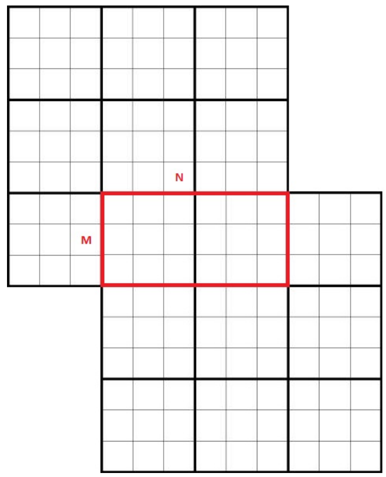

Figure 1: Figure shows a bridge Sudoku with M x N overlap area

Our claim -

1. It is more difficult to solve a bridge sudoku compared to a standard sudoku with the same givens
2. The plot for difficulty against overlap area follows an inverted parabolic structure

2. SAT Encoding for the Bridge Sudoku Problem

Our encoding is based on the minimal encoding strategy introduced by Lynce & Ouaknine [1]. This encoding technique suffices to characterize the bridge sudoku and compare its difficulty against solving a normal sudoku. Encoding a bridge sudoku puzzle requires $9 * 9 * 9 * 2 = 1458$ variables. We denote the variable s_{xyzp} as True if and only if the number z is assigned to the row x and the column y of the sudoku p, $p \in \{1, 2\}$.

$$\wedge_{x=1}^{9} \wedge_{y=1}^{9}, \wedge_{z=1}^{9} s_{xyzp} \qquad \text{(one number per entry)}$$
$$\wedge_{x=1}^{9} \wedge_{z=1}^{9}, \wedge_{y=1}^{8} \wedge_{i=y+1}^{9} (\neg s_{xyzp} \vee \neg s_{xizp}) \qquad \text{(row constraint)}$$
$$\wedge_{y=1}^{9} \wedge_{z=1}^{9}, \wedge_{x=1}^{8} \wedge_{i=x+1}^{9} (\neg s_{xyzp} \vee \neg s_{iyzp}) \qquad \text{(column constraint)}$$

The block constraint is given by

$$\wedge_{z=1}^{9} \wedge_{i=0}^{2} \wedge_{j=0}^{2} \wedge_{x=1}^{3} \wedge_{y=1}^{3} \wedge_{k=y+1}^{3} \left(\neg s_{(3i+x)(3j+y)zp} \vee s_{(3i+x)(3j+k)zp}\right)$$
$$\wedge_{z=1}^{9} \wedge_{i=0}^{2} \wedge_{j=0}^{2} \wedge_{x=1}^{3} \wedge_{y=1}^{3} \wedge_{k=x+1}^{3} \wedge_{l=1}^{3} \left(\neg s_{(3i+x)(3j+y)zp} \vee s_{(3i+k)(3j+l)zp}\right)$$

In addition to the constraints required for solving a normal sudoku, we have another constraint for the elements in the bridge area. The variables in the bridge area of the first sudoku must be equal to the variables in bridge area of the second sudoku.

$$\left(\wedge_{x=9-m+1}^{9} \wedge_{y=9-n+1}^{9} \wedge_{z=1}^{9} s_{xyz}\right)_1 = \left(\wedge_{x=1}^{m} \wedge_{y=1}^{n} \wedge_{z=1}^{9} s_{xyz}\right)_2$$

Using the above minimal encoding, the CNF formula will have a total of $(8829 * 2 + 9 * m * n)$ clauses. Of these, 162 are nine-ary clauses, 17496 are binary clauses and $(9*m*n)$ are four-ary clauses. The four-ary clauses result from the bridge constraint.

3. Dataset and SAT solver

The dataset used for testing contains about 80000 bridge sudokus with different overlap areas. On an average there are 52 total givens. For generating the various statistics we have used the minisat solver, which is based on the two literal watch-scheme [2].

Minisat gives the following statistics - number of conflicts, restarts, decisions, propagations, conflict literals, cpu time and memory used. We do not use the cpu time as a statistic to explain difficulty since it is not sandboxed. On performing a Bayesian Correlation test, we also found that cpu time is not correlated to the overlap area and the difficulty in solving a bridge sudoku (See appendix for this result).

In order to compare the bridge sudoku statistics with standard sudokus', we used the same dataset without the bridge constraint. This is equivalent to solving the two sudokus separately.

4. Results

The main goal of this section is to compare the difficulty of solving a standard sudoku with one that has more constraints. All the results, code and datasets used can be found in our Github repository [3]. We observed that for the difficulty comparison problem, similar results were obtained when using minimal encoding and extended encoding. All the results reported below are for minimal encoding. The extended encoding results can be found in the above mentioned link.

We performed a Bayesian Correlation Test using JASP [4] on the various statistics (decisions, restarts, conflicts, propagations and conflict literals) with the overlap area. For all these statistics, we found a high negative correlation with the overlap area.

	r	BF
area - restarts	-0.600	15.203
area - conflicts	-0.791	35.818
area - decisions	-0.810	39.067
area - propagations	-0.799	37.192
area - conflict literals	-0.729	27.274

Table 1: Table shows the Pearson correlation coefficient and the log of Bayes factor in favor of the alternative hypothesis for the various statistics considered with the overlap area

For the correlation test, we assume the pairs of variables follow a bivariate normal distribution. We define the null hypothesis that the population Pearson product-moment correlation between the pairs of variables equals 0. The Bayes factor values in Table 1 show a strong evidence for the alternate hypothesis which represents a high correlation. An interesting result that we observed was the high correlation between the number of conflicts and propagations (Pearson's coefficient, $r = 0.993$)

We generated the above mentioned statistics for the standard sudokus using the same datasets (without the bridge constraint). On average, the total number of givens were about 21% of the total area. For each overlap area value, we generated the statistics for about 1000 different sudokus and took the average for each of the mentioned statistics. We find, consistently, that it is easier to solve bridge sudoku compared to a standard sudoku.

5. Conclusion and Future Work

In this report we investigated the difficulty of solving a bridge sudoku compared to a standard sudoku. The experimental results falsify our original hypothesis. We believe that is due to the extra constraints imposed on the

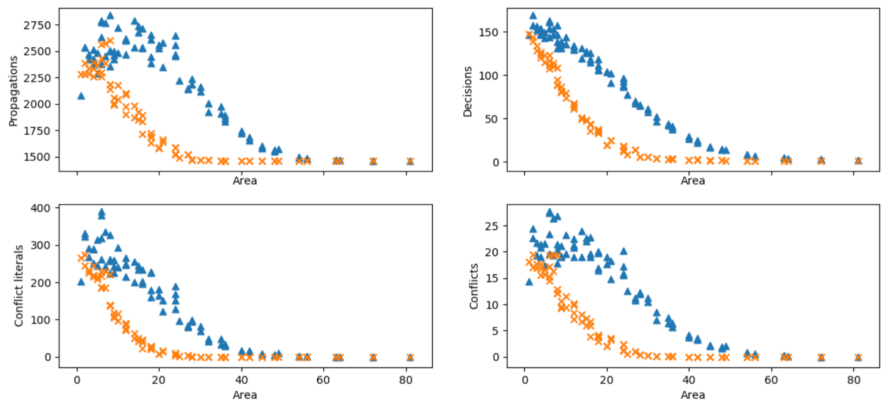

Figure 2: Number of conflicts, propagations, conflict literals and decisions plotted against overlap area. Orange points belong to the set of bridge sudokus and the blue points belong to the corresponding standard sudokus. Note that there are lesser conflicts while solving the bridge sudokus.

bridge sudokus. Assignment of a value to one of the variables in the overlap area in effect reduces more number of clauses in the case of bridge sudokus. In addition, the solution space is more constrained.

Our github repository contains all the analyses, results and the code (https://github.com/akashrajkn/bridge-sudoku-solver). Presentation video (https://www.youtube.com/watch?v=KkghV1t61uI&feature=youtu.be)

For future work, we would like to carry out analysis to explore the impact of the bridge givens on the difficulty. An extension to this point is to test the heuristic of selecting variables in the bridge area with a higher preference.

References

[1] I. Lynce, J. Ouaknine, Sudoku as a SAT Problem, ISAIM (2006).

[2] N. Sorensson, N. Een, Minisat v1.13 - a SAT solver with Conflict-Clause Minimization (2005).

[3] A. Raj, J. Baars, Solver for bridge sudoku, 2017.

[4] E.-J. Wagenmakers, et al., JASP (Version 0.8.2) [computer software], 2017.

Appendix A. Bayesian Correlation Test results for choice of difficulty measure

To conduct the Bayesian correlation pairs test, we made the assumption that the pairs of variables follow a normal distribution in the population. Null hypothesis states that the Pearson correlation between overlap area and the test statistic is 0. (Two-sided alternative hypothesis that the correlation is not 0 is considered).

One apparent issue with using the cpu time as a difficulty measure is that the simulations were not sandboxed and the cpu time reported could be wrong. Also, on conducting a Bayesian Correlation Test, we get a very small correlation. On further conducting a Bayes factor robustness check, we found that for different values of the prior width, the Bayes factor is generally in favor of the null hypothesis.

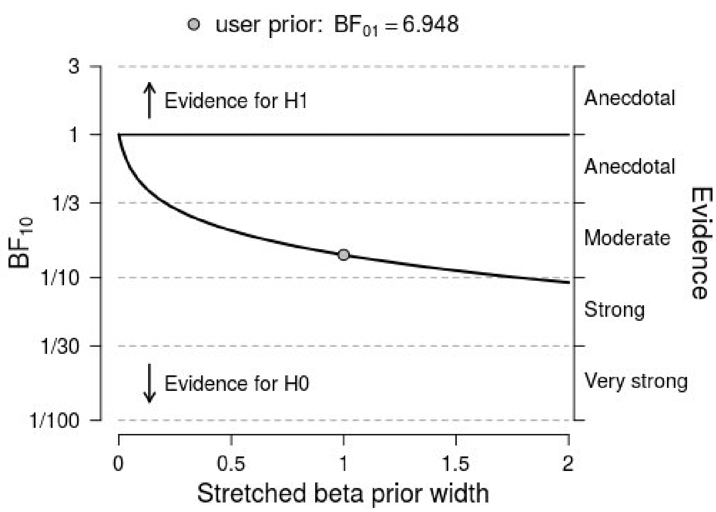

Figure A.3: Bayes factor robustness check for area-cpu time correlation test