Ranking Clues for SuDoKu SAT Solvers

Gulfaraz Rahman, University of Amsterdam
Tharangni H Sivaji, University of Amsterdam

The aim of this paper is to analyze the variations of clues in terms of orientation, existence, arrangement and position. It discusses about the parameters of the initial sudoku state which is achieved by running it through PycoSAT in non-deterministic mode. The metrics generated by PycoSAT are used to rank each variation of the given sudoku to compare which variation (spatial or non-spatial) is favoured by the SAT solver.

Categories and Subject Descriptors: SAT Solver [**Decision Logic**] Weighted Ranking

Additional Key Words and Phrases: Ranking clues, Sudoku SAT Solvers, PycoSAT

1. EXPERIMENTAL SETUP

Assigning the ranks begins with getting an input sudoku as a sequence of 81 digits with 0 denoting an empty cell. For ex: 007000000 500000006000920300098000005700200900000800470009305007640000000100006030 is an input sudoku. The input sudoku is passed to an intermediary input stage where variations of the original are created by: a) rolling (position) and b) single clue absent variations (clue values). These intermediate sudokus are paased through the SAT Solver (PycoSAT) which generates metrics for each variation. These metrics serve as the inputs for ranking the different variations of the input sudoku. We finally visualize the comparisons as a scatter plot from which inferences can be drawn as to which variation of the sudoku is favoured by the SAT Solver.

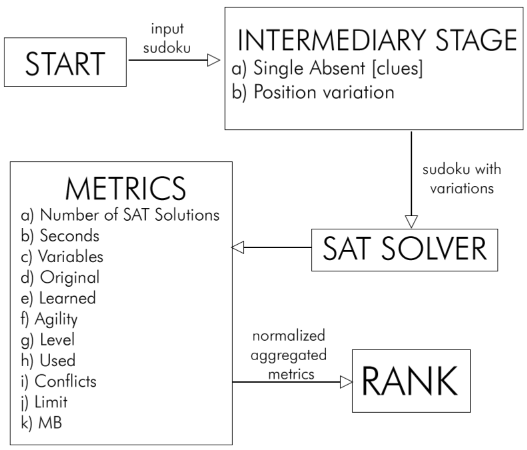

Fig. 1: Ranking Workflow

2. OBSERVATION

Results are formulated after the sudokus with variation are passed to the SAT Solver which generates a *consolidated log* file of the metrics stated in Fig 1. Consolidated metrics are computed from the raw output generated by the SAT Solver (Table I). These metrics are then normalized to assess the ranks of the sudoku clues (Table II).

2.1 Metric Table

The above table displays some of the consolidated metrics for two clues referenced as **S2** and **C1**. **S2** refers to the sudoku variation that has been rolled twice. **C1** refers to the sudoku variation with one absent method. * refers to the original input sudoku.

ID	# of SAT Solutions	Learned Clauses			Conflict			Conflict per Decision Ratio		
		AVG	MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX
*	8	16	12	20	136	132	140	11.56	2	18
S2	0	1	1	1	324	324	324	0	0	0
C1	57	61.62	28	96	220.5	192	249	193.45	5	335

Table I. : Consolidated Metrics

2.2 Rank Table

Clue Category	Clue Type	Clue	Minimum	Dense
0	2	S2	10	4
1	82	C1	90	24

Table II. : Ranks

Ranks are formulated by normalizing the consolidated metrics while having * as the median.

2.3 Plots

When the ranks are plotted, we see that the 'rolled' sudoku variations (in gray) have a higher rank than the 'single absent' variations (in red). This behaviour is consistent for increasing number of input sudokus (8, 499, 999). The ranks are calculated from the metrics output by the SAT solver. The rankings are shown in density and minimum ranking strategies to highlight the consistency in representations of the variations. The two ranking methods are plotted against the sudoku variations and shown in Figures 2 - 7.

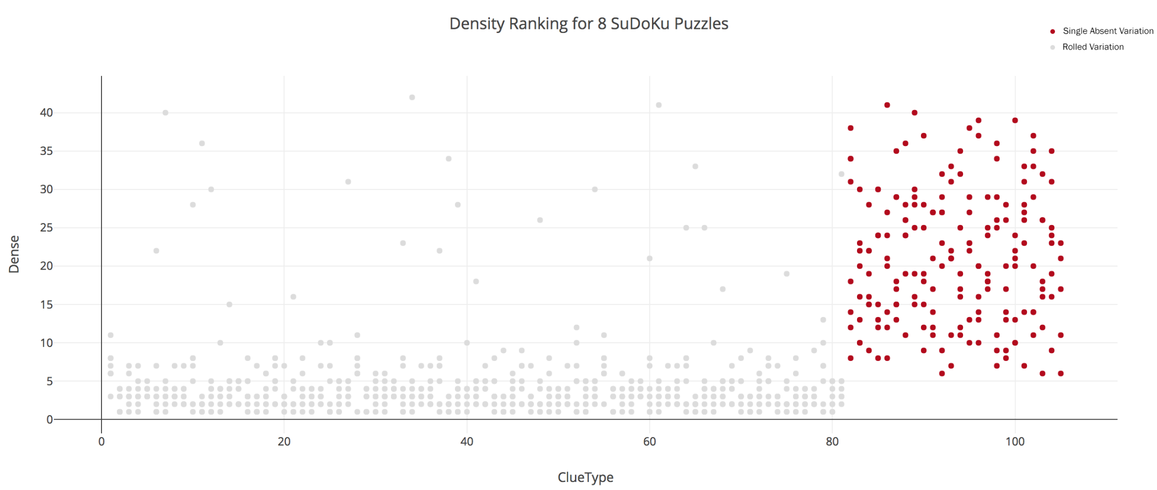

Fig. 2: 8 Sudoku Density

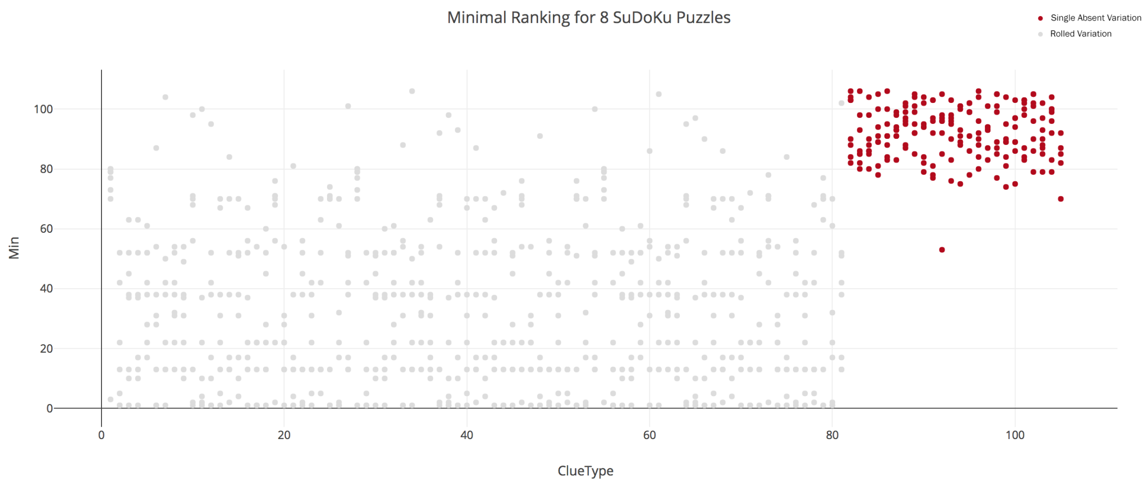

Fig. 3: 8 Sudoku Minimum

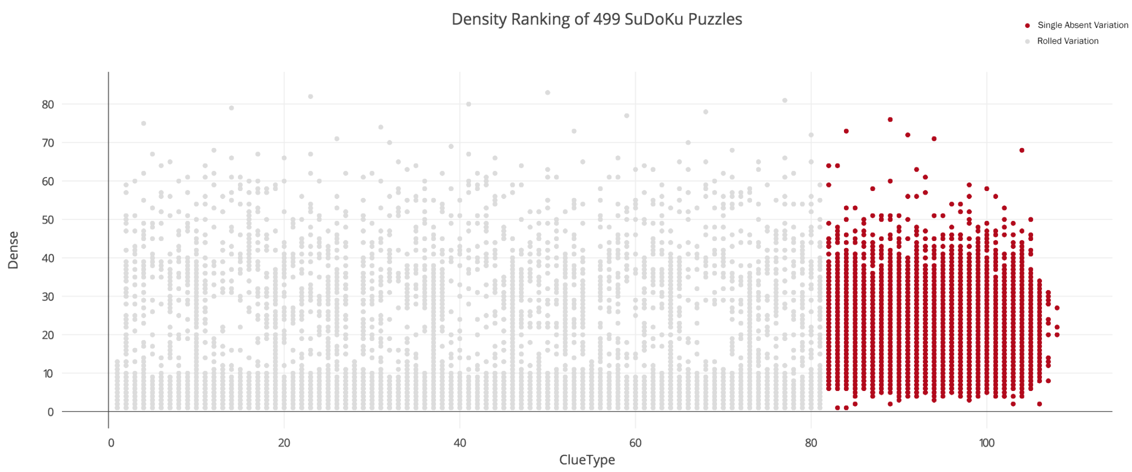

Fig. 4: 499 Sudoku Density

3. SOURCE CODE

(1) The altered version of an existing pycosat implementation [1] of a sudoku solver is available at: https://github.com/gulfaraz/sudoku-sat-solver

(2) A modified version of pycosat is required to run the experiment for improved readability and some additional metrics. This is available at: https://github.com/gulfaraz/pycosat/tree/alter-logging

(3) A walkthrough of the experiment for one sudoku is explained in a presentation: https://github.com/gulfaraz/sudoku-sat-solver/blob/master/docs/walkthrough.pptx

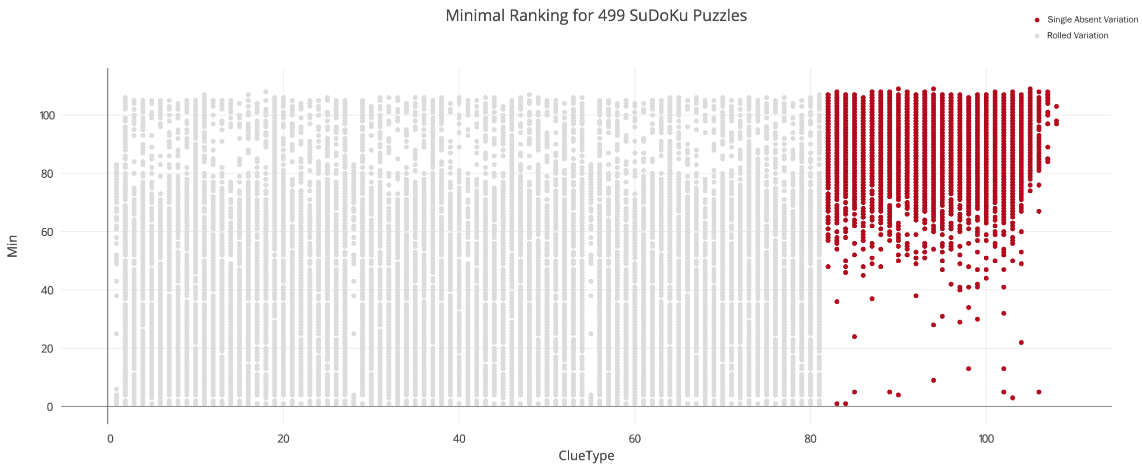

Fig. 5: 499 Sudoku Minimum

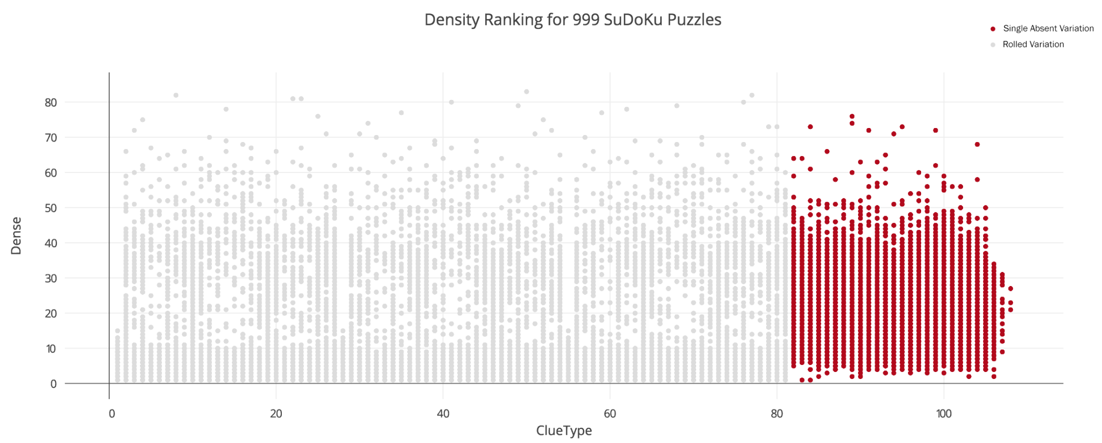

Fig. 6: 999 Sudoku Density

4. CONCLUSION

The results show that more variations of 'rolled' sudokus have a higher ranking than the variations of 'single absent' sudokus. Apart from exceptions in specific sudoku arrangements (outliers), the results indicate that the positional arrangement of the clues have a profound impact on the performance of the SAT solver when compared to the non-spatial variations of the input sudoku.

5. LIMITATIONS

The input sudoku used are minimal in order to reduce complexity in the variations at the intermediate level. This can be extended to any type of sudoku at a run-time cost.

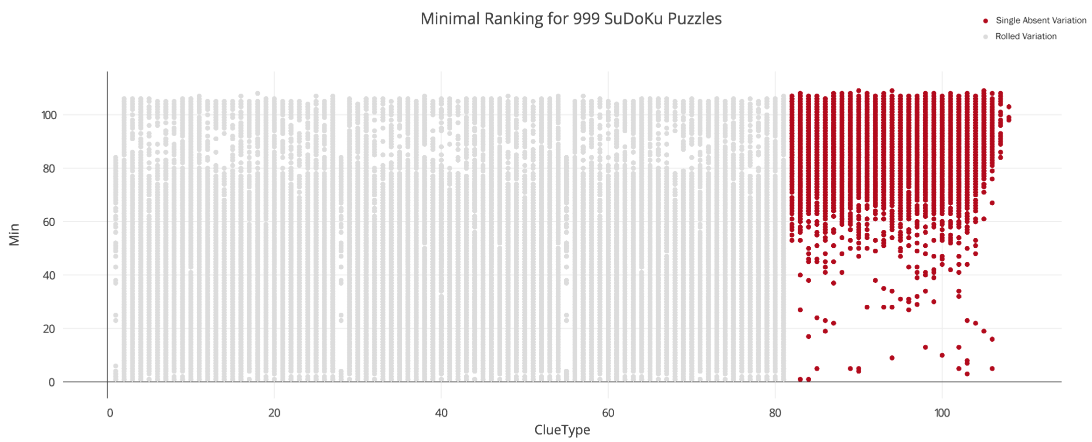

Fig. 7: 999 Sudoku Minimum

The ranking system is trivial, our implementation gives equal weights to all the metrics which is output by the SAT solver. If there are multiple solutions for a given variation then the metrics are aggregated to create a AVG/MIN/MAX version of those metrics. These two rankings assumptions heavily bias the comparison of ranks and can be further improved. A major reason to avoid fine tuning the ranking system is to prevent forced biases which could favour the sudokus that have unique solution. Such an approach may not be the ideal ranking preference in many applications.

6. APPLICATIONS

Any system that involves decision making will benefit from this weighted feature comparison approach of ranking. The most significant impact would be in assignments of ambulances for medical emergencies. Other potential applications are listed as follows:

(1) Courier delivery networks
(2) Classroom assignment for universities
(3) Strategy games

7. FUTURE RESEARCH

There is much to improve in the experiment. Every component can be improved drastically, the sat solver metrics can be fine tuned and weighted more proportionally. The number of variations are limited to one type for spatial (rolled) and valued (single absent) sudokus. There are a reasonable number of additional variations which can lead to more insightful results. For instance, we can generate more variations by rotating the sudoku puzzle. In the ranking system, we can account for the symmetries of sudoku. For example we can assign a huge favour to the variation that leads to a symmetrically equivalent solution to the solution of the original sudoku. The whole experiment can be repeated on other kinds of SAT solvers and different sudoku encodings.

REFERENCES

Nicholas Pilkington, 2015, https://gist.github.com/nickponline/9c91fe65fef5b58ae1b0
Peter Norvig, 2010, http://norvig.com/sudoku.html