# Comparison of Stochastic and Deterministic SAT Solvers in the Domain of Regular vs. Irregular Sudoku Puzzles

Ece Kamer Takmaz

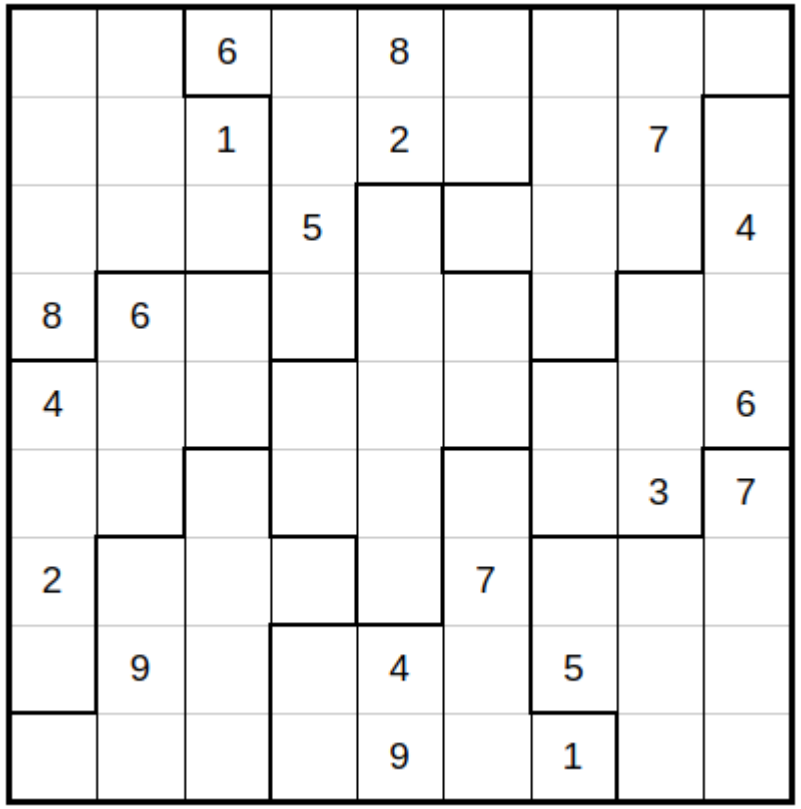
ecekt028@gmail.com

11823453

# Links

- https://github.com/ecekt/KRsudoku
- https://www.youtube.com/channel/UCgwBmGRaHx9sV6Vw7oe68cg
  - https://youtu.be/-K-avcHs4VY

# Introduction




Showing puzzle number: 11696

# Research questions

- Types of sudoku puzzles – We expect that the irregular sudoku puzzles would require more levels, decisions or steps in order for a solver to be able to find a satisfying assignment.
- Types of solvers – We would like to see the difference between solvers in finding a solution

# Methodology – Data set

- http://www.menneske.no/sudoku/eng/
- 18345 irregular puzzles
- 10000 regular puzzles of 5 different difficulty levels
- Region encoding from implicit user interface descriptions of boundaries
- Propagating information via search through neighbouring cells

# Methodology – Data set

- Conversion to CNF and DIMACS
- Constraints on:
  - Cells
  - Numbers
  - Rows
  - Columns
  - Blocks
- Dictionary for converting 111 -> 1

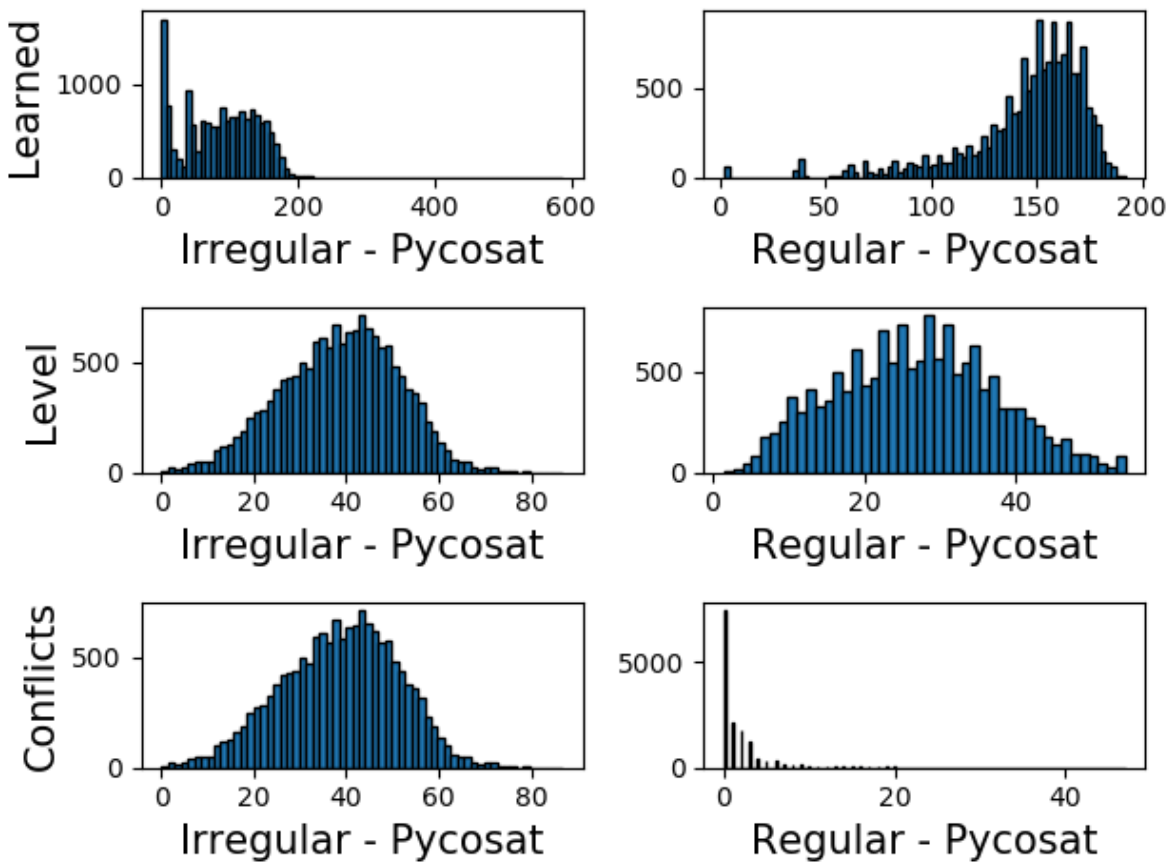
# Methodology - Solvers

- Deterministic – conflict driven clause learning
  - pycosat: 0.6.2 https://pypi.python.org/pypi/pycosat [10]
  - zChaff: 2007.3.12 https://www.princeton.edu/~chaff/zchaff.html [7]
- Stochastic – random moves and random variable assignments
  - UBCSAT: version 1.1.0 (Sea to Sky Release) http://ubcsat.dtompkins.com/ [6]
  - Walksat: v51 https://www.cs.rochester.edu/u/kautz/walksat/ [5]

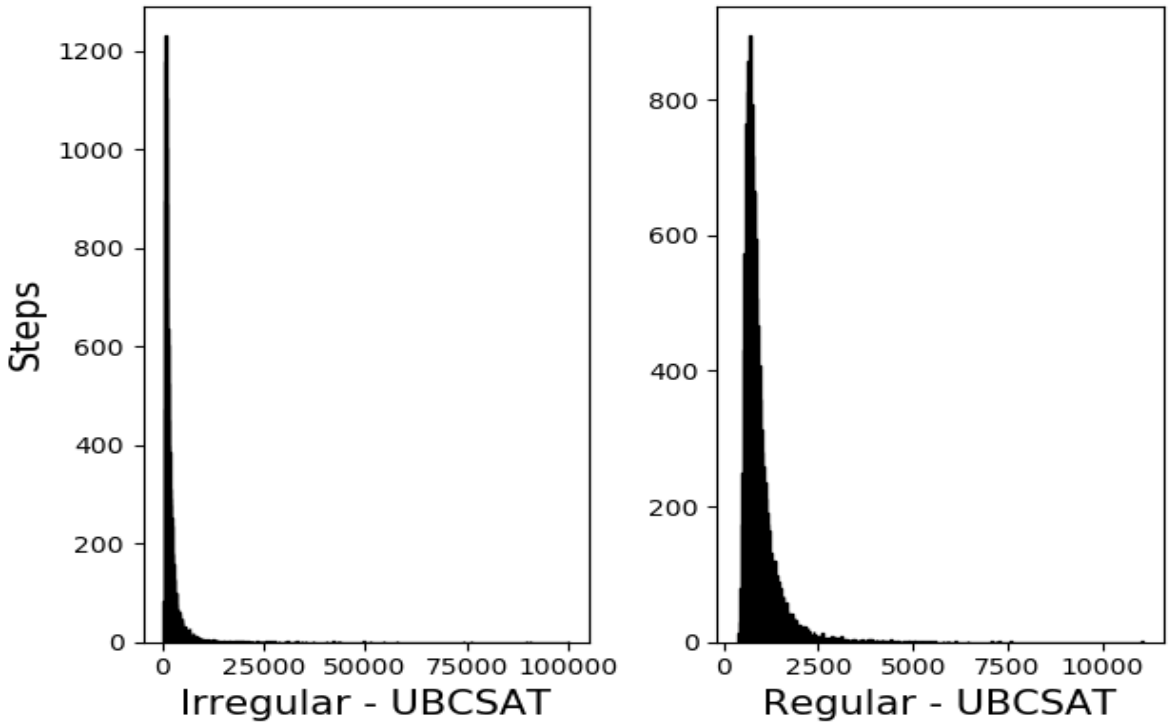
# Methodology – Solver metrics

- For pycosat:
  - Learned: Number of learned clauses when the solver arrives at a solution
  - Level: Number of levels the algorithm goes through to arrive at a solution
  - Conflicts: Number of conflicts that occurred during the solving of the problem at hand
- For UBCSAT:
  - Steps: Number of steps the algorithm goes through to arrive at a solution
- For zChaff:
  - Level: Maximum decision levels the algorithm goes through to arrive at a solution
  - Number of decisions: Number of decisions the algorithm makes to arrive at a solution
- For Walksat:
  - Avg Length of Trials: Average of the length of 10 trials performed by the stochastic algorithm

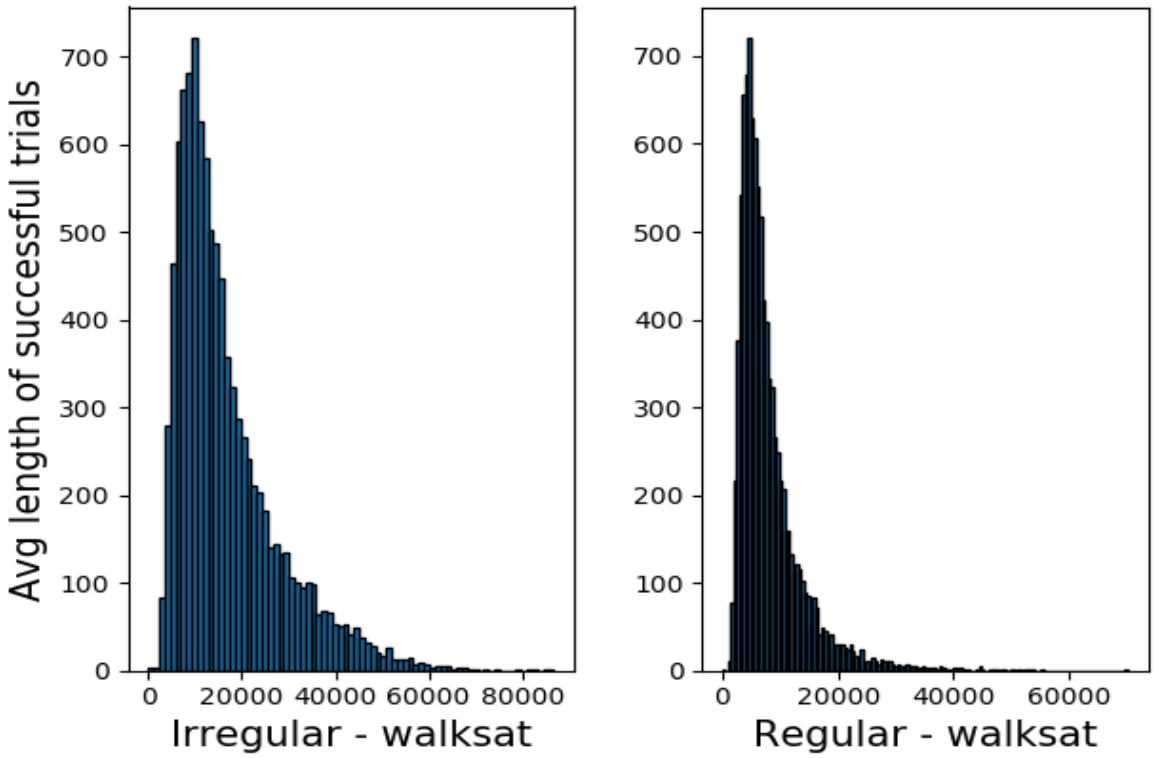
# Results


Learned
1000
0
0
200
400
600
Irregular - Pycosat
500
0
0
50
100
150
200
Regular - Pycosat
Level
500
0
0
20
40
60
80
Irregular - Pycosat
500
0
0
20
40
Regular - Pycosat
Conflicts
500
0
0
20
40
60
80
Irregular - Pycosat
5000
0
0
20
40
Regular - Pycosat

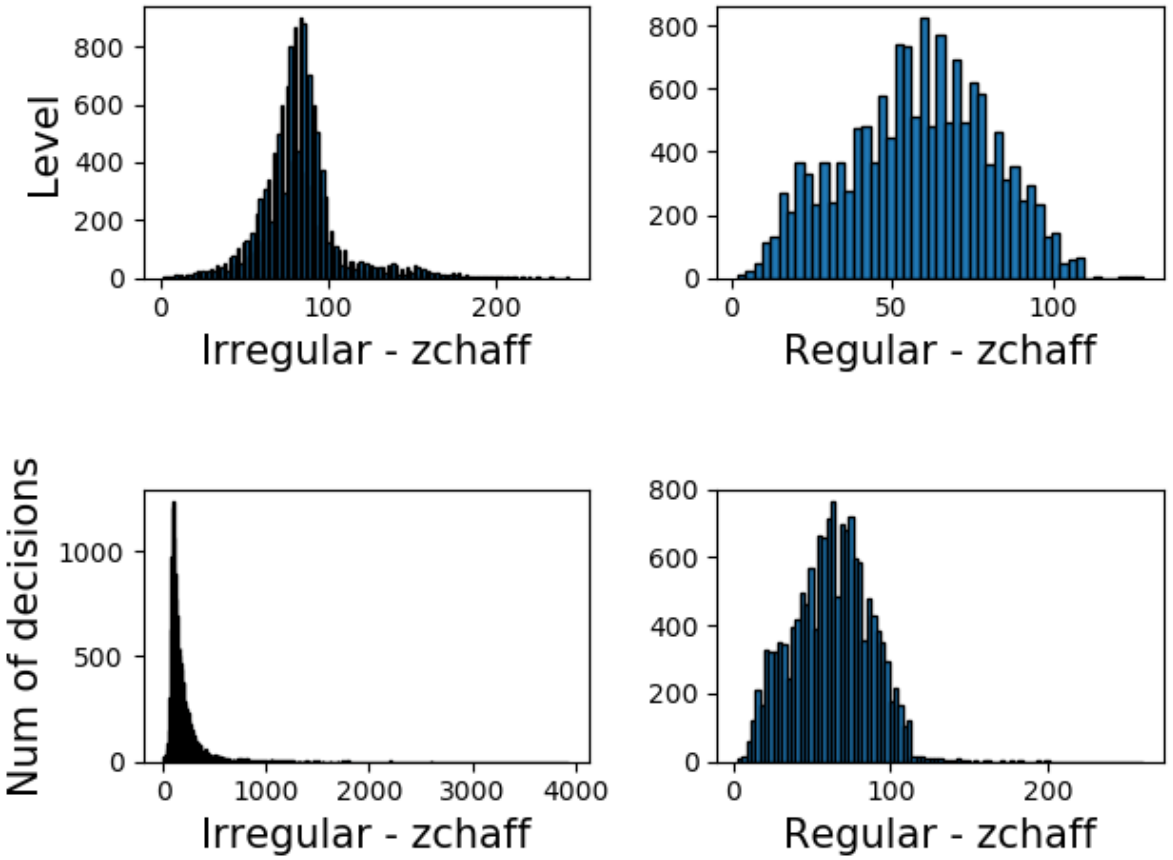
# Results


Steps
1200
1000
800
600
400
200
0
0
25000
50000
75000
100000
Irregular - UBCSAT
800
600
400
200
0
0
2500
5000
7500
10000
Regular - UBCSAT

# Results


Avg length of successful trials
0
100
200
300
400
500
600
700
0
20000
40000
60000
80000
Irregular - walksat
0
100
200
300
400
500
600
700
0
20000
40000
60000
Regular - walksat

# Results


Level
800
600
400
200
0
0
100
200
Irregular - zchaff
800
600
400
200
0
0
50
100
Regular - zchaff
Num of decisions
1000
500
0
0
1000
2000
3000
4000
Irregular - zchaff
800
600
400
200
0
0
100
200
Regular - zchaff

# Results

OUTCOMES FOR 15000 RUNS FOR EACH TYPE OF PUZZLE

| Solver | Sudoku Type | Parameter | Mean | Std |
|---|---|---|---|---|
| Pycosat | Irregular | Learned | 85.708 | 53.373 |
| Pycosat | Regular | Learned | 145.649 | 28.725 |
| Pycosat | Irregular | Level | 38.621 | 12.563 |
| Pycosat | Regular | Level | 25.807 | 10.511 |
| Pycosat | Irregular | Conflicts | 11.602 | 14.394 |
| Pycosat | Regular | Conflicts | 2.008 | 3.704 |
| UBCSAT | Irregular | Steps | 2063.506 | 3206.003 |
| UBCSAT | Regular | Steps | 930.679 | 510.775 |
| zChaff | Irregular | Level | 82.661 | 23.813 |
| zChaff | Regular | Level | 58.038 | 22.371 |
| zChaff | Irregular | Number of Decisions | 210.122 | 257.058 |
| zChaff | Regular | Number of Decisions | 62.667 | 24.900 |
| Walksat | Irregular | Avg Length of Trials | 17297.035 | 11595.851 |
| Walksat | Regular | Avg Length of Trials | 7263.001 | 5452.748 |

# Discussion

- Type of the puzzle can have an effect on the values of the solver metrics, different distributions
- Due to the differences in the calculation of parameters, it is not quite possible to indicate the difference between the solvers; yet, it is possible to compare each solver's performance given a sudoku puzzle type.
- For all the solvers, irregular puzzles require a greater amount of decisions, levels and steps
- Stochastic solvers require a greater amount of decision steps
- The average number of conflicts encountered is also greater for the irregular puzzles as compared to classic sudoku puzzles.
- The number of learned clauses, however, is fewer than that of the regular puzzles.
- These results seem to support the hypothesis of irregular puzzles requiring a higher number of levels
    - Number of literals is 729, clauses is around 23660

# Conclusions

- Results indicated that the irregular sudoku puzzles may require more steps or levels of decision-making in order for solvers to make satisfactory assignments.
- Reliable statistical test results
- Using universal metrics for solvers
- Different difficulty levels
- Distance parameter to indicate the scale of irregularity

# References


[1] T. Weber, A SAT-based sudoku solver. In 12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2005, pages 1115, 2005.

[2] I. Lynce and J. Ouaknine, Sudoku as a SAT problem. In Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics, AIMATH 2006, Fort Lauderdale. Springer, 2006.

[3] T. Davis, The Mathematics of Sudoku, Unpublished, http://www.geometer.org/mathcircles/sudoku.pdf

[4] S.B. Varun, Sudoku Squares as Experimental Designs, International Journal of Engineering Trends and Technology, 28(5), 229-235, 2015.

[5] B.Selman, H.A. Kautz, and B. Cohen, Noise strategies for improving local search, 337343. MIT press, 1994.

[6] D.A.D. Tompkins and H.H. Hoos, UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT, Revised Selected Papers from the Seventh International Conference on Theory and Applications of Satisfiability Testing, 306- 320, Springer Berlin / Heidelberg, 2005.

[7] M.W. Moskewicz , C.F. Madigan , Y. Zhao , L. Zhang , S. Malik, Chaff: engineering an efficient SAT solver, Proceedings of the 38th annual Design Automation Conference, p.530-535, Las Vegas, Nevada, USA, 2001.

[8] C.P. Gomes, H.A. Kautz, A. Sabharwal, B. Selman, Satisfiability Solvers. Handbook of Knowledge Representation, 89-134, 2008.

[9] M. Perez, and T. Marwala, Stochastic optimization approaches for solving Sudoku, Proc. IEEE Congress on Evolutionary Computation, pp 256279, Vancouver: Piscataway, 2011.

[10] A. Biere, PicoSAT Essentials. Journal on Satisfiability, Boolean Modeling and Computation (JSAT), vol. 4, pages 75-97, Delft University, 2008.

[11] U. Pfeiffer, T. Karnagel, G. Scheffler, A Sudoku-Solver for Large Puzzles using SAT. LPAR short papers(Yogyakarta), 52-57, 2010.

[12] M.L. Littman, S.M. Majercik, T. Pitassi, Stochastic Boolean Satisfiability, Journal of Automated Reasoning, v.27 n.3, p.251-296, 2001.