Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Min-Conflicts Against Single-Worker CP-SAT on Dense Blocked n-Queens Instances: A Small Replication Note

Dikshant Bikram Thapa

dikshantthapa3@gmail.com

Kathmandu University https://orcid.org/0009-0002-5551-7296

Research Article

Keywords: n-Queens, Blocked n-Queens, local search, min-conflicts, CP-SAT, SAT, phase transition, solver comparison

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11855600/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Min-Conflicts Against Single-Worker CP-SAT on Dense Blocked n-Queens Instances: A Small Replication Note

Dikshant Bikram Thapa
Kathmandu University

Abstract

Blocked n-Queens asks for n non-attacking queens on an n x n board where each cell is forbidden at random with probability p. Gent, Jefferson and Nightingale (2017) showed that hardness peaks at a solvability transition. We ask a practical question on the satisfiable side of that transition: how do an exact CP-SAT model (AllDifferent, one worker), min-conflicts local search, restart backtracking and a hybrid compare as n grows? We use 1,440 instances with n from 30 to 70 (4 s limit, 3 s for n = 40) and 360 instances with n from 100 to 400 (10 s). For n up to 50 CP-SAT is best or within two instances of the best. For n >= 100 and hole density up to 0.88 to 0.94, min-conflicts finds solutions that one-worker CP-SAT does not find in 10 s: at n = 200 and p = 0.85 to 0.90 it solves 20 of 20 instances, all within 0.4 s, against at most 1 of 20 for CP-SAT. The advantage is confined to the satisfiable side of the transition: at n = 100 and p = 0.90 to 0.92 almost no solver finds solutions (min-conflicts 1 of 20 at p = 0.90 and 0 of 20 at p = 0.92), and CP-SAT proves 441 of 1,440 small instances unsatisfiable, mostly at p >= 0.84, which local search cannot do. A phase map of 240 further instances with a SAT-solver baseline shows that the advantage of local search ends at the satisfiability transition, where no solver we tried decides within 5 s, and that the complete solvers (CP-SAT, glucose4 and the backtracking-based hybrid) are the ones that prove unsatisfiability, CP-SAT the most within the limit. A learned per-instance solver selector gives no clear gain. The instances are random, one CP-SAT model and limit were used, and the effect of min-conflicts far from the threshold is well known; this note quantifies it on a documented grid and is not a new result on the threshold.

1 Introduction

The n-Queens completion problem and the variants Blocked n-Queens and Excluded Diagonals were studied by Gent et al. (2017), who proved NP-completeness and found phase transitions with hard instances near them. Local search by conflict minimisation is known to solve very large n-Queens instances quickly (Minton et al. 1992; Sosic and Gu 1994), restarts help backtracking on heavy-tailed instances (Gomes et al. 2000), and random constraint problems are easy far from the transition (Kirkpatrick and Selman 1994). We measure how these facts play out for Blocked n-Queens against a general-purpose solver. Our contribution is a small, documented empirical comparison with a clean rerun of the large-n part.

2 Method

Instances. Each cell is blocked independently with probability p; seeds are fixed. Small set: n in {30, 40, 50, 70}, 1,440 instances in total (n = 40: p in {0.6, 0.7, 0.75, 0.8, 0.85, 0.9}, 40 per cell, limit 3 s for the local solvers; other n: p in {0.70, 0.74, 0.78, 0.80, 0.82, 0.84, 0.86, 0.90}, 50 per cell, 4 s). Large set: n in {100, 200, 400}, p in {0.80, 0.85, 0.88, 0.90, 0.92, 0.94}, 20 per cell (360), 10 s wall-clock, run sequentially in one process on an Apple laptop (8 cores) with OR-Tools CP-SAT.

Solvers. CP-SAT: one integer variable per row over its free columns, AllDifferent on q, q+r and q-r, one worker; wall time includes model construction. A Boolean AtMostOne encoding was weaker on the small set (0.39 of instances solved against 0.58 for AllDifferent). Min-conflicts: greedy start, repair moves with a penalty for blocked cells, random restarts. Restart backtracking: forward checking, minimum remaining values, random value order, Luby restarts. Hybrid: backtracking to 80% depth, then min-conflicts repair, alternated with restarts. A timeout is not a claim of unsatisfiability. Solutions are not independently validated in the pipeline; solver code checks constraints by construction.

3 Results

Table 1 gives the fraction of instances solved (a solution found) and the median time of solved instances; CP-SAT proved unsatisfiability on 441 of the 1,440 small instances (the hybrid and restart backtracking also report proofs, 292 and 350 times, always agreeing with CP-SAT), and 19 of the 360 large ones (95% of the n = 100, p = 0.94 instances), which Table 1 and Figure 1 do not count as solved. For n <= 50 CP-SAT is best or within two instances of the best (n = 50, p = 0.82: CP-SAT 0.84, hybrid 0.40, min-conflicts 0.14). At n = 70 the three are close at p = 0.82 and 0.84 (hybrid 0.98 and 0.68), but at p = 0.86 CP-SAT is ahead. For n >= 100 the order reverses on the satisfiable side: at n = 100 and p = 0.88, min-conflicts solves 0.90 and the hybrid 0.95 against 0.40; at n = 200 and p = 0.85 to 0.90 both local methods solve 20 of 20 (all within 0.4 s) against 1 of 20 and 0 of 20; at n = 400 and p up to 0.94 min-conflicts solves 20 of 20 in every cell (hybrid 19 of 20 at p = 0.94) and CP-SAT none. The advantage ends where instances stop being solvable: at n = 100 and p = 0.90 and 0.92 min-conflicts solves 0.05 and 0.00 and almost no solver finds solutions. A four-worker CP-SAT solved 5 of 5 instances at n = 100, p = 0.88 (one worker: 1 of 5) but 0 of 5 at n = 200, where min-conflicts solved 5 of 5. We checked whether the n = 200 failure is a model-size or setting effect on 10 instances at p = 0.88 with a 20 s limit: the default one-worker model, the same model without presolve, one worker with randomised search, and four workers each solved 0 of 10, and building the model took 0.01 to 0.02 s, so the time goes into search and not into construction. These checks cover one model and one cell of the grid.

Solver selection. A random forest over 16 features of the blocked set, trained on the small set (five solvers including the Boolean CP-SAT, timeouts counted as 8 s for every n, feature time included), costs 1.49 (five-fold) and 1.37 (leave-one-n-out) against 1.43 for always running the best single solver and 0.95 for the oracle. The paired bootstrap interval of the mean cost difference to always running CP-SAT is [-0.02, +0.16] for the five-fold selector and [-0.12, +0.01] for leave-one-n-out, so neither differs clearly from zero, while the oracle is 0.48 lower [0.42, 0.55]. This is not a clear gain, and the comparison is dominated by the instances that CP-SAT proves unsatisfiable, where local search always times out, so it says little about choosing among solvers on satisfiable instances.

Phase map. To see where each kind of solver works, we ran a further set of 240 instances (n = 50, 100 and 200; hole density p from 0.80 to 0.96; 10 per cell; 5 s per solver) with min-conflicts, the hybrid, one-worker CP-SAT and the glucose4 SAT solver on a pairwise CNF encoding (Table 2). The cells where no solver finds a solution and none proves unsatisfiability are consistent with the transition region: n = 50 between p = 0.84 and 0.88, n = 100 at p = 0.90 and 0.92 (all 20 instances undecided within 5 s), and n = 200 at p = 0.94 (all 10 undecided). On the satisfiable side, min-conflicts finds a solution in every instance of n = 200 up to p = 0.92 and of n = 100 up to p = 0.86, where CP-SAT solves 6 of 10 at n = 100 and none at n = 200. On the unsatisfiable side only the complete solvers answer: CP-SAT proves 9 of 10 instances at n = 100, p = 0.94 within 5 s, and the SAT solver is weaker than or equal to CP-SAT except for one n = 200, p = 0.80 instance (at n = 50, p = 0.88 it proves 9 of 10 against CP-SAT's 10 of 10, and it solves no n = 200 instance at p >= 0.84); its 5 s timer starts after the clauses are built, which favours it, and the pairwise encoding is large at n = 200, so this comparison is specific to our encoding. The advantage of local search is therefore confined to the satisfiable side, and the difficult region is the narrow band at the transition, where every solver we tried fails within 5 s.

4 Discussion

When n is large and the board is dense but still satisfiable, a few lines of min-conflicts finish in milliseconds where a one-worker AllDifferent model does not finish in ten seconds. This is the usual easy-region behaviour of local search. Our data say nothing about instances at the threshold and beyond, where local search cannot answer and CP-SAT can, and the crossover moves with the CP-SAT model, workers and limit.

Limitations

Random instances from one generator; satisfiability unknown for timeouts; one CP-SAT model, one worker for the main runs, and multi-worker and setting checks on one grid cell; hybrid parameters not tuned; short limits; 20 instances per large cell; different p grids for different n; selector evaluated on n <= 70 with five-fold splits; n = 40 CP-SAT

n	p	instances	min-conflicts	hybrid	CP-SAT (1 worker)
50	0.78	50	0.54 (1.76 s)	0.88 (0.72 s)	0.90 (0.33 s)
50	0.82	50	0.14 (1.61 s)	0.40 (1.70 s)	0.84 (0.72 s)
70	0.82	50	0.94 (0.69 s)	0.98 (0.50 s)	0.96 (0.93 s)
70	0.84	50	0.54 (1.36 s)	0.68 (1.07 s)	0.58 (1.13 s)
100	0.85	20	1.00 (0.22 s)	1.00 (0.06 s)	0.95 (0.95 s)
100	0.88	20	0.90 (0.96 s)	0.95 (1.77 s)	0.40 (5.40 s)
200	0.85	20	1.00 (0.02 s)	1.00 (0.03 s)	0.05 (9.27 s)
200	0.90	20	1.00 (0.05 s)	1.00 (0.06 s)	0.00
400	0.88	20	1.00 (0.04 s)	1.00 (0.08 s)	0.00
400	0.92	20	1.00 (0.11 s)	1.00 (0.15 s)	0.00

Table 1: Fraction of instances solved (median time of solved instances in parentheses). Limits: 4 s (n <= 70, 3 s for the local solvers at n = 40), 10 s (n >= 100). CP-SAT proofs of unsatisfiability are not counted as solved.

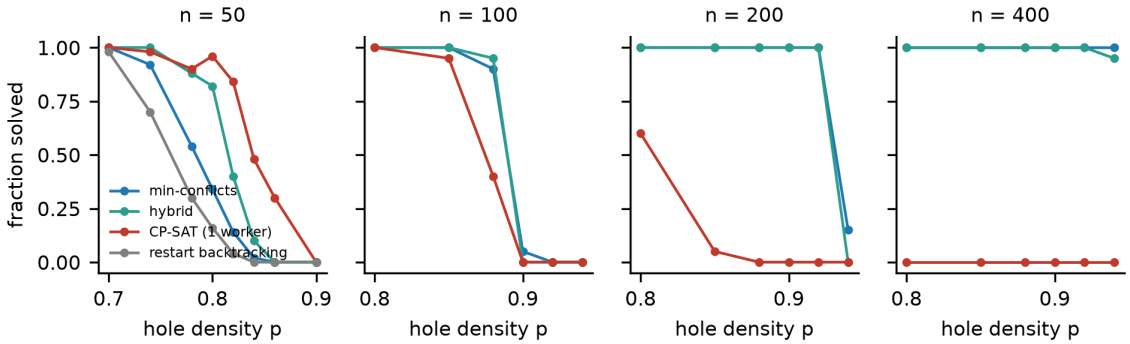

Figure 1: Fraction of instances solved against hole density for n = 50, 100, 200 and 400 (n >= 100: clean sequential rerun, 20 instances per point; n = 50: 50 per point). Proofs of unsatisfiability by CP-SAT are counted as not solved.

n	p	SAT found by any solver	proved UNSAT	undecided	min-conflicts	CP-SAT (SAT / UNSAT)	glucose4 (SAT / UNSAT)
50	0.80	10	0	0	3	10 / 0	4 / 0
50	0.84	7	0	3	0	7 / 0	0 / 0
50	0.86	6	4	0	0	6 / 4	0 / 0
50	0.88	0	10	0	0	0 / 10	0 / 9
50	0.90	0	10	0	0	0 / 10	0 / 10
50	0.92	0	10	0	0	0 / 10	0 / 10
50	0.94	0	10	0	0	0 / 10	0 / 10
50	0.96	0	10	0	0	0 / 10	0 / 10
100	0.80	10	0	0	10	10 / 0	6 / 0
100	0.84	10	0	0	10	10 / 0	0 / 0
100	0.86	10	0	0	10	6 / 0	0 / 0
100	0.88	7	0	3	7	1 / 0	0 / 0
100	0.90	0	0	10	0	0 / 0	0 / 0
100	0.92	0	0	10	0	0 / 0	0 / 0
100	0.94	0	9	1	0	0 / 9	0 / 1
100	0.96	0	10	0	0	0 / 10	0 / 10
200	0.80	10	0	0	10	0 / 0	1 / 0
200	0.84	10	0	0	10	0 / 0	0 / 0
200	0.86	10	0	0	10	0 / 0	0 / 0
200	0.88	10	0	0	10	0 / 0	0 / 0
200	0.90	10	0	0	10	0 / 0	0 / 0
200	0.92	10	0	0	10	0 / 0	0 / 0
200	0.94	0	0	10	0	0 / 0	0 / 0
200	0.96	0	1	9	0	0 / 1	0 / 1

Table 2: Phase map (the hybrid was also run and counts towards the first column, but is not shown): for each cell of 10 instances, the number where any solver found a solution, where a complete solver proved unsatisfiability, and where nothing was decided in 5 s, then the counts per solver.

was run with a 4 s limit while the local solvers had 3 s, which favours CP-SAT; a few CP-SAT runs finished up to 0.6 s after the limit because of model-building time. Earlier large-n runs overlapped on one machine and some wall times were inflated by the laptop sleeping; we reran them

cleanly. The location of the transition is not our contribution (Gent et al. 2017). Code and instance seeds are available from the author on request.

Ethical considerations

No human data. Generative AI use (disclosure): Claude (Anthropic) was used for code, analysis and drafting; we reviewed the content and take responsibility.

References

Carla P. Gomes, Bart Selman, Nuno Crato, and Henry Kautz. 2000. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. Journal of Automated Reasoning 24:67-100. doi:10.1023/A:1006314320276.

Ian P. Gent, Christopher Jefferson, and Peter Nightingale. 2017. Complexity of n-Queens completion. Journal of Artificial Intelligence Research 59:815-848. doi:10.1613/jair.5512.

Scott Kirkpatrick and Bart Selman. 1994. Critical behavior in the satisfiability of random Boolean expressions. Science 264:1297-1301. doi:10.1126/science.264.5163.1297.

Steven Minton, Mark D. Johnston, Andrew B. Philips, and Philip Laird. 1992. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. Artificial Intelligence 58:161-205. doi:10.1016/0004-3702(92)90007-K.

Google. OR-Tools CP-SAT solver. https://developers.google.com/optimization (accessed November 2026).

Rok Sosic and Jun Gu. 1994. Efficient local search with conflict minimization: A case study of the n-queens problem. IEEE Transactions on Knowledge and Data Engineering 6(5):661-668. doi:10.1109/69.317698.