Can’t Get No Satisfaction

Lorenzo Cappetti, Fabio Sucameli

03/06/2025

UNIVERSITÀ
DEGLI STUDI
FIRENZE

P and NP Problems

- A problem is said to belong to the class **P** if there exists a **polynomial-time** algorithm to solve it.
- **NP** problems (nondeterministic polynomial time) are decision problems for which no polynomial-time algorithm is known, but there is also no proof that such an algorithm cannot exist. However, a proposed solution can be verified in polynomial time.
- **NP-complete** problems are a special class of NP problems: if a polynomial-time algorithm could be found for even just one of them, that algorithm could be adapted to solve all problems in NP.

SAT and 3-SAT

- The satisfiability problem, or SAT, consists in determining whether there exists an assignment of values to variables that makes a given Boolean formula true.
- Boolean expressions are usually written in a format called Conjunctive Normal Form (CNF).
- A specific case of the SAT problem where each clause in the Boolean formula contains exactly three literals.
- Example:

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_4 \vee x_5) \wedge (\neg x_2 \vee \neg x_3 \vee x_6)$$

- 3-SAT was the first problem proven to be NP-complete.

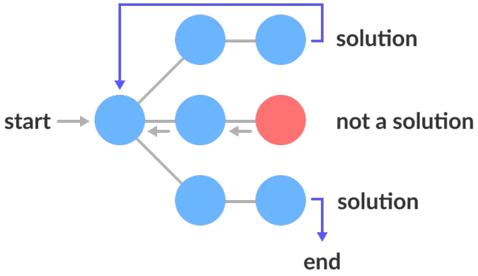
UNIVERSITÀ DEGLI STUDI FIRENZE

Clause-to-Variable Ratio

- Complexity analysis is based on worst-case scenarios; for this reason, examining the distribution of easy and hard cases across the problem space is of great interest.
- The clause-to-variable ratio (M/N) has proven to be a crucial parameter in determining the distribution of easy and hard instances of the 3-SAT problem.
  - Low $\frac{M}{N} \rightarrow$ high probability that the formula is satisfiable
  - High $\frac{M}{N} \rightarrow$ high probability that the formula is unsatisfiable

Backtracking

- The basic strategy involves exploring a branch of the solution tree until a dead end is reached. At that point, the algorithm backtracks to a previous decision point to try a different branch. If that path also fails, it backtracks further to an earlier choice. This process continues until a solution is found or all possible branches have been exhausted.



Instance Generation

- To generate a clause in a random 3-SAT instance, three distinct variables are selected from the total set of N variables. Each selected variable is then either negated or left positive with a probability of $\frac{1}{2}$. To construct a 3-SAT formula with M clauses, this process is repeated M times.

Heuristics

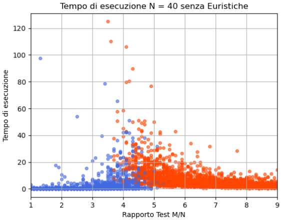
- A **heuristic** is a practical approach to problem solving that uses methods based on experience or intuition to find solutions more quickly, even though it does not guarantee optimality.
- Unit Propagation Heuristic:
  - If a clause has **only one unassigned literal**, then that literal **must be true** to satisfy the clause.
  - Remove the clause in which the literal appears.
  - Remove the negation of the literal from all other clauses.
- This technique can trigger **deterministic cascades of simplifications**, drastically reducing the search space and computational time.

Experimental Setup

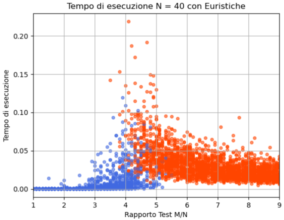
- We generated random 3-SAT instances with $N = 10, 20, 30, 40$ variables.
- For each value of M/N from 1 to 9, we generated k random CNF formulas.
- Three solving methods were compared:
  - **Backtracking without heuristics**
  - **Backtracking with heuristic unit propagation**
  - **MiniSat** (external industrial solver)
- Goals: compare execution time, percentage of satisfiable formulas, and average computational cost.

Average Execution Time

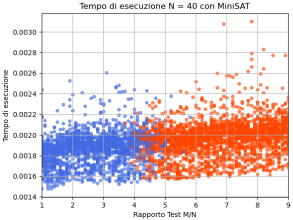
- Execution time increases with M/N, reaching a peak near $M/N \approx 4.2$.
- MiniSat is significantly faster than the other two methods.
- The heuristic reduces the execution time, especially in the critical region.
- Satisfiable — Unsatisfiable



(a) Without heuristic



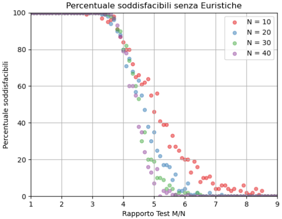
(b) With heuristic (Unit Propagation)



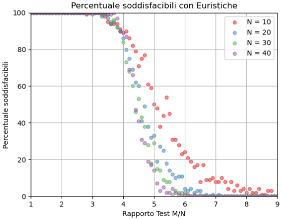
(c) MiniSat

Percentage of Satisfiable Formulas

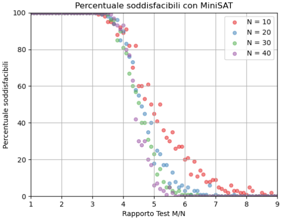
- The transition curve from “almost always satisfiable” to “almost never” is clearly visible.
- The results are consistent across all algorithms: the transition is observed in every case.
- As N increases, the transition becomes sharper.



(d) Without heuristic



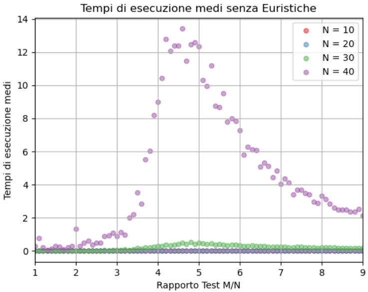
(e) With heuristic (Unit Propagation)



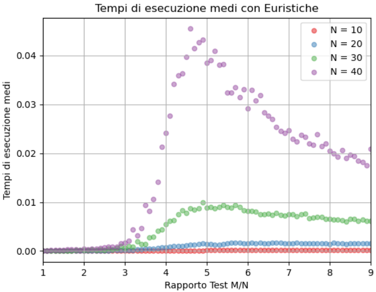
(f) MiniSat

Average Computational Cost

- The cost is minimal in both underconstrained and overconstrained regions.
- A peak appears near the critical region ($M/N \approx 4.2$).
- The use of heuristics significantly reduces the number of computational steps.



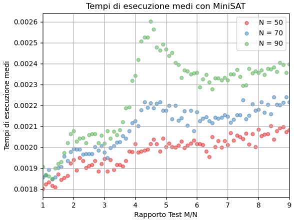
(g) Without heuristic

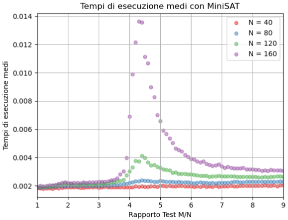


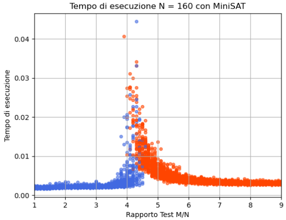
(h) With heuristic (Unit Propagation)

MiniSAT

- MiniSAT turned out to be the best-performing method.
- We compared its performance by increasing both the number of variables.
- For MiniSAT the results of the paper come out for a very large number of variables

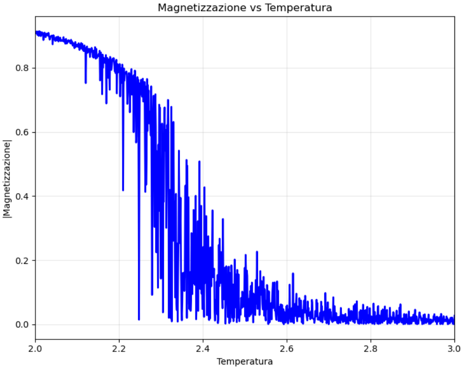


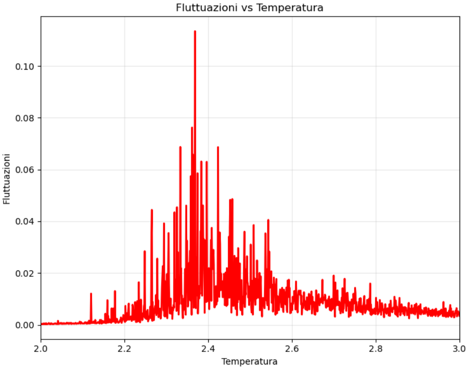




Phase Transition

- In 3-SAT, as the clause-to-variable ratio increases, instances exhibit a sharp transition from satisfiable to unsatisfiable: a phenomenon strikingly similar to the phase transition in physical systems like the Ising model, where magnetization abruptly vanishes beyond a critical temperature.





Experimental Conclusions

- All the methods analyzed confirm the presence of a **phase transition** in the 3-SAT problem: as the M/N ratio increases, instances abruptly shift from satisfiable to unsatisfiable.
- The **unit propagation heuristic** significantly reduces both the **execution time** and the **computational cost**, making the search more manageable in the critical region.
- **MiniSat** proved to be the **most efficient** method tested, thanks to advanced optimizations and internal pruning strategies.
- The numerical results reflect well-known phenomena from **statistical physics**, such as phase transitions, highlighting the interdisciplinarity between theoretical computer science and physical models.

References

Niklas Eén and Niklas Sörensson.
Minisat: A sat solver with conflict-clause minimization.
`http://minisat.se`, 2003.

Brian Hayes.
Can't get no satisfaction.
American Scientist, 85(2):108–112, 1997.

Wikipedia contributors.
Boolean satisfiability problem — wikipedia, the free encyclopedia, 2024.
[Online; accessed May 31, 2025].

Can’t Get No Satisfaction

Thank you for listening!