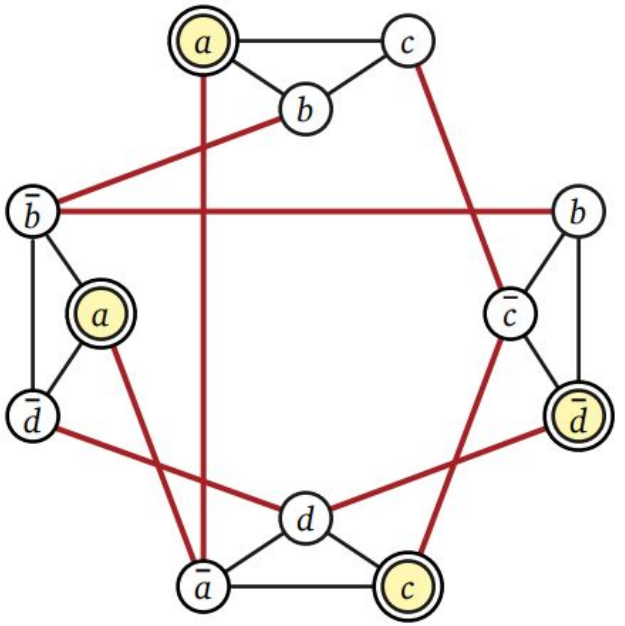
Reduced Solution

Max 3-SAT to Max Independent Set (MIS)

Max 3-SAT Reduction Algorithm

1. Initialize Graph
2. Map Literals
3. Identify Clause Conflicts (△)
4. Identify Logical Conflicts
5. Build Edge List

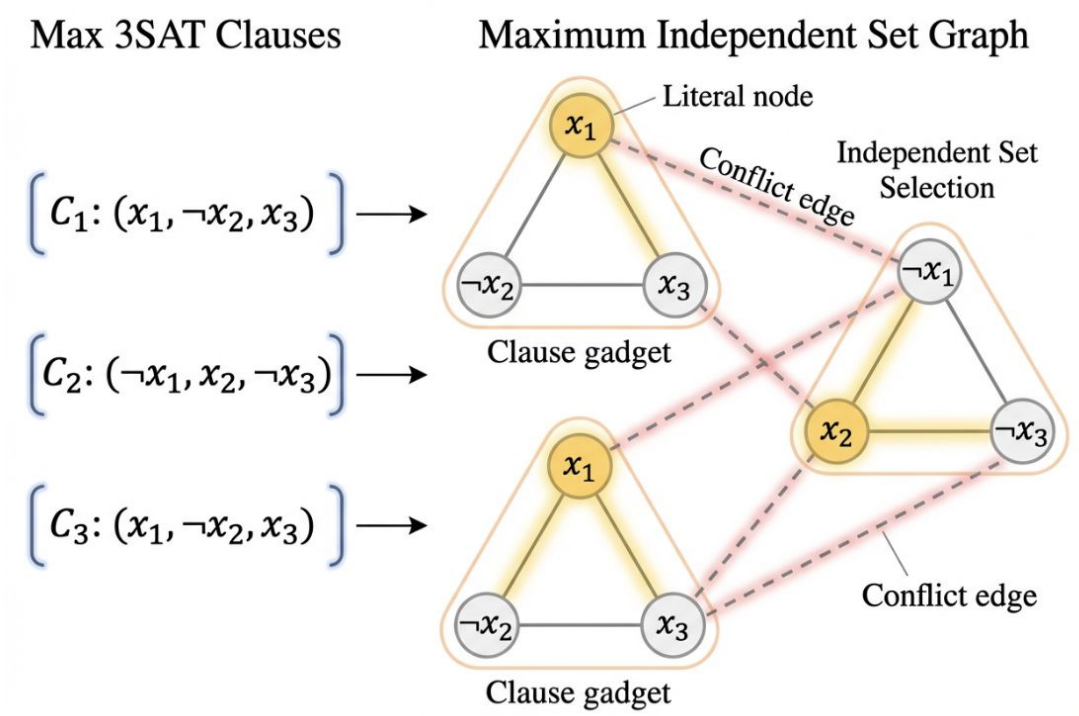


$$(a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})$$

The runtime complexity of this reduction is O(c^2), where c is the number of clauses, due to the nested loops for edge construction.

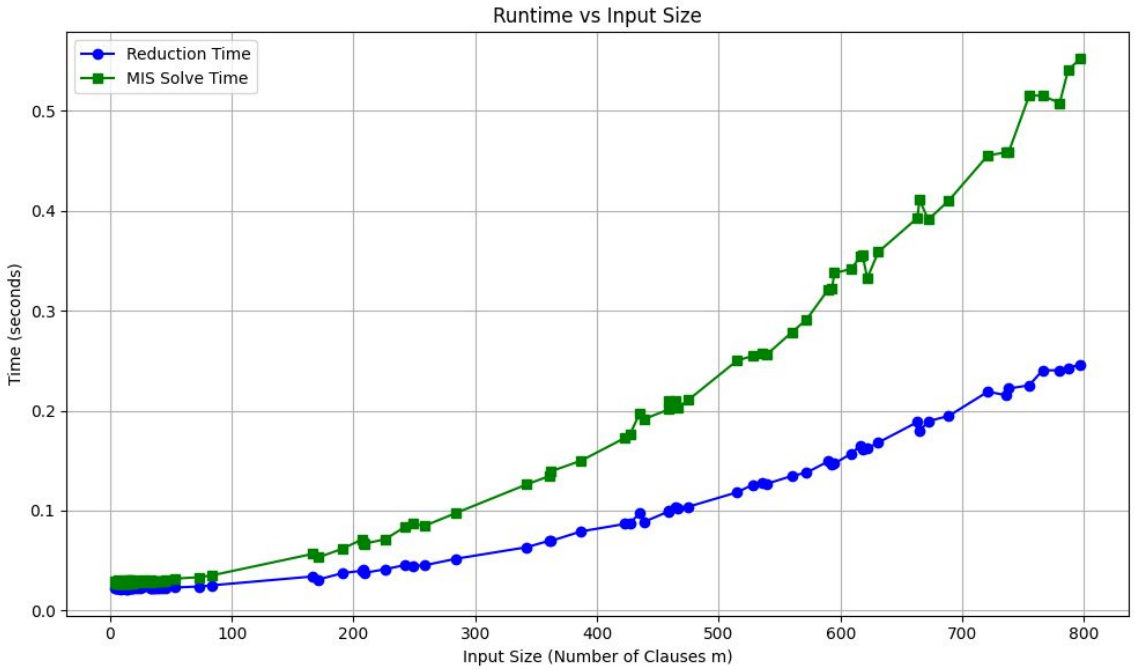
Reduction Explanation

- Clause Gadget (Triangles): Each 3-literal clause forms a triangle. This complete subgraph ensures only one vertex (literal) can be selected per clause.
- Conflict Edges (Dashed): Edges connect every literal x to its negation ¬x. This prevents contradictory literals from existing in the same independent set.
- Core Intuition: Selecting one non-conflicting literal per clause creates an independent set. The size of this set equals the number of satisfiable clauses.



Runtime For Reduction vs Total After MIS

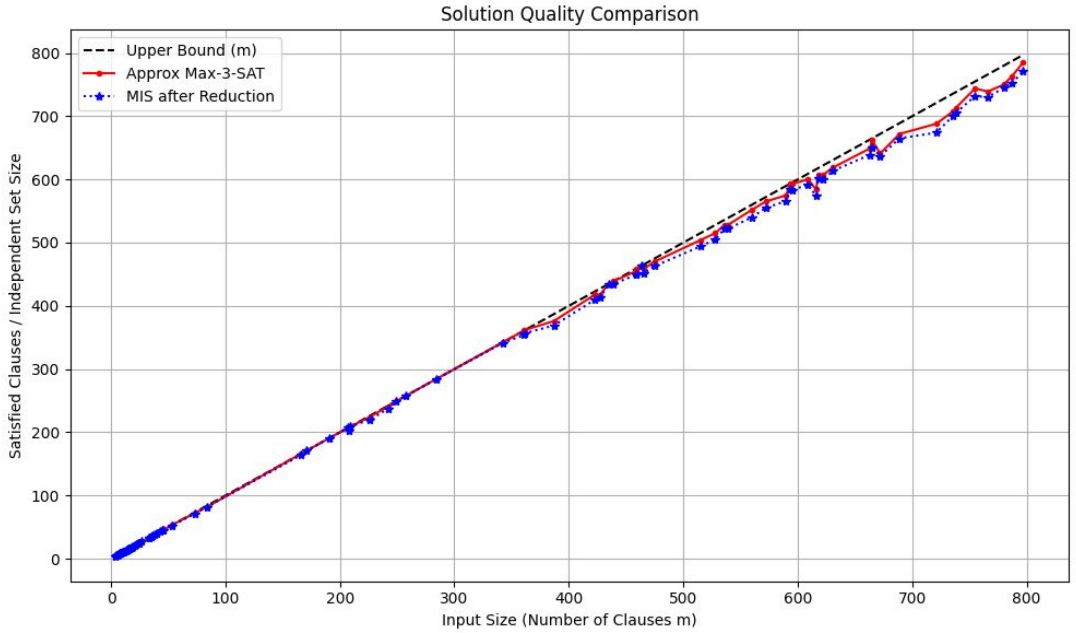
- The runtime complexity of this reduction is O(c^2), where c is the number of clauses, due to the nested loops for edge construction. The outer loop is clause construction and inner is checking for logical conflicts.
- Solving the Maximum Independent Set problem is NP-hard. The best algorithms are still exponential in the worst case.
- If it were to be an exact solver it would be exponential time, but the solution I was given was an approximate.



Runtime For Reduction vs MIS

Max 3-SAT Upper Bound

- Due to Max 3-SAT being an optimization/maximization problem we are looking to find an upper bound.
- I landed on a trivial yet effective solution for my upper bound, return the total number of clauses.
- Assuming each clause has a satisfiable literal, the maximum clauses that can be satisfied is C (all of them).
- Disadvantage: Can become looser with less ideal clauses or large quantities.



More Results