SAT Results

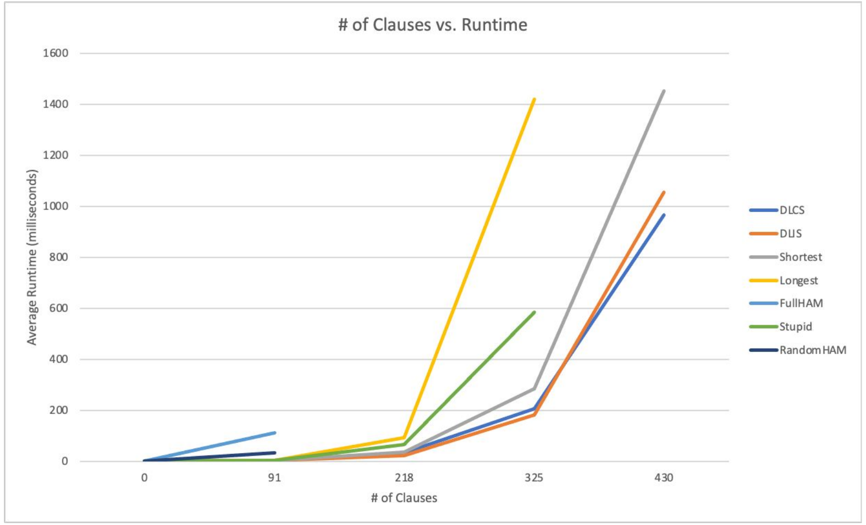
Let’s talk data

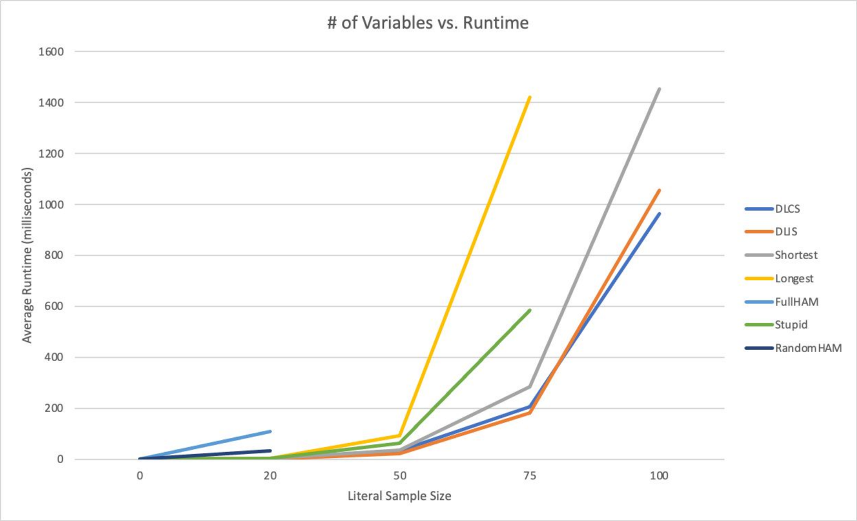
- I updated the algorithms to accept files (formulas) from the SATLIB database because they had great data
- I learned an important lesson about infinite sample spaces and SATs
    - What constitutes a representative sample is not so straightforward
    - My original testing was not robust enough. The algorithms failed on SATLIB sets. I was able to debug everything but rec7Alg.
    - Phase transition regions are a big deal in SAT solver testing and the SATLIB data is sampled from that region.

DPPL Variations

- DLCS: Choosing the literal with the most instances of it or its negation
  - Intuitive choice to reduce the formula as much as possible
- DLIS: Choosing the pos literal with the most instances
  - Maybe better because you get rid of the most clauses potentially
- Shortest: Choose a literal from one of the smallest length clauses
  - Allows reduction to lean on STAND unit clause case
- Longest: Choose a literal from one of the longest length clauses
  - Allows reduction to 2SAT as fast as possible
- Stupid (random): Chooses the first literal that appears in the the clause

First Experiment: Pulse Check



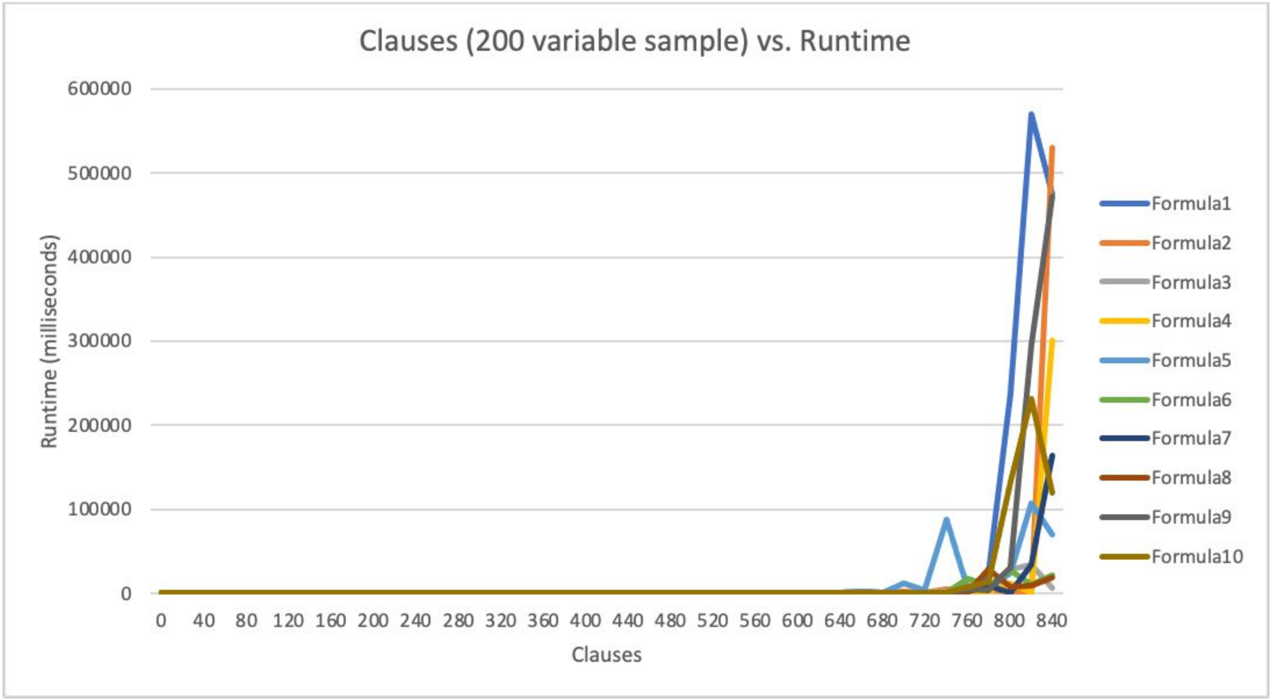


** These are two projections of a 3D graph. The runtimes where collected on couples (literal sample size, # of clauses) to be in phase transition region

Observations

- Sharp exponential behavior (as expected)
- DPPL is much more efficient in practice than HAM algorithms
- Random HAM is more efficient that full HAM
- Over time DLCS becomes the most efficient heuristic
- Picking literals from the longest clauses is worse than ‘random’ selection!

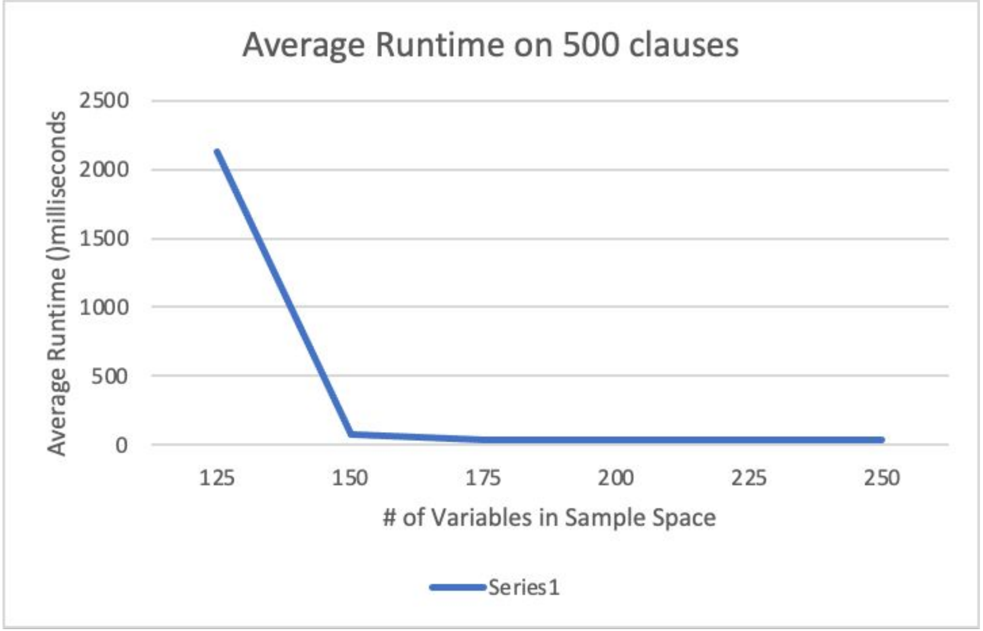
Second Experiment: Effect of Clauses



Observations

- For a given sample space of variables the effectiveness of DPPL has a critical threshold, after which only a few clauses can make a drastic difference.
- This can be understood as the point at which literals become very constrained
- Eventually the problems would become over constrained and the number of satisfiable formulas would tend to 0.

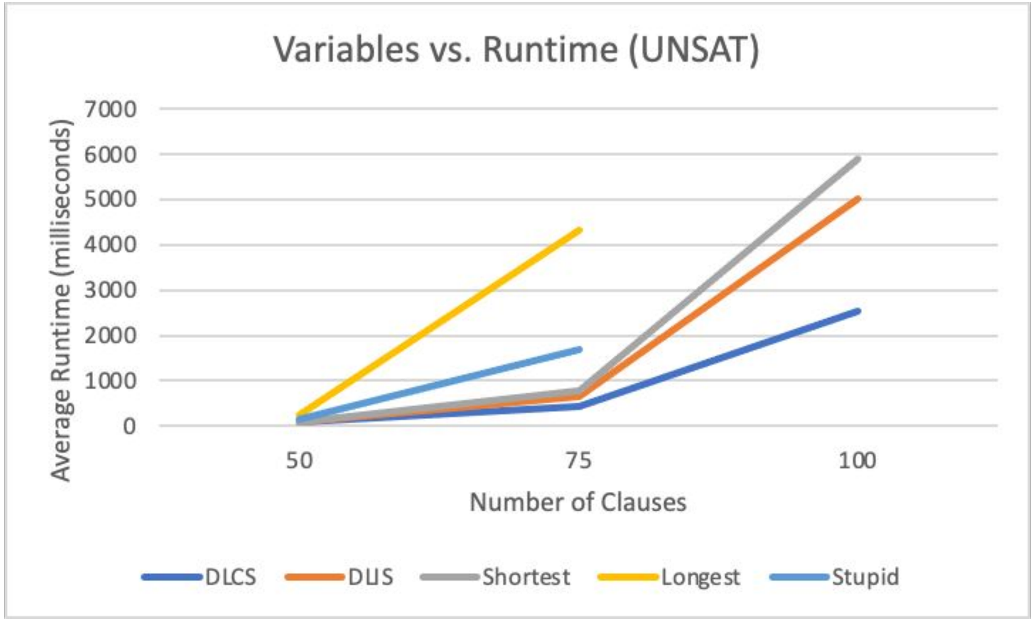
Third Experiment: Effect of Variables



Observations

- Unlike clauses, more variables does not mean harder! WHY?
- After a threshold, more variables just means STAND can efficiently take care of more pure literals. With less variables, there is more constraints.
- The graph is incomplete because of a lack of data. If we could continue the graph to the left, I would suspect that the graph would dip again sharply (when there are too few variables any selection simplifies the formula a lot).

Fourth Experiment: Unsatisfiability



Observations

- Unsatisfiability is HARD!!! with these algorithms
    - Exponential behavior but hard to tell because the next iteration of data couldn’t finish in a reasonable time.
- I tried random HAM and it couldn’t even do the ‘small’ clause efficiently in practice.
- Interestingly, DLCS is significantly the **best** heuristic for solving unsatisfiable formulas.
- Once again we see that choosing literals from the longest clause is actually worse than random and that picking from the shortest clause is not bad.