

John Ventura

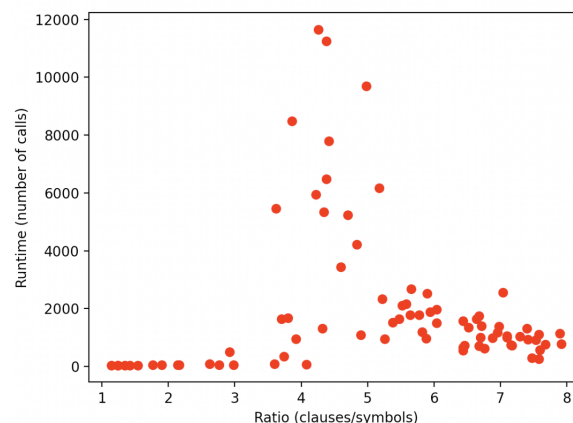
ECE 4524

May 1st, 2022

Project 5 Report

For this final project, I believe that has satisfied the specifications. We were tasked with creating an SAT solver and test it against test cases provided to us. My program was able to test these five test cases and run them much faster than five minutes each. The test cases in which my program takes much longer to run on are the dubois*.txt test cases, so I only test my program on the five that were specified to us. My program is also modular and satisfies that criteria. There are no parts of the project that I did not implement.

In regards to what was difficult, I would say that the most difficult part of the project was debugging the code when a test case gave the wrong result. This led to me having to go through all of my methods and checking to see if there were any errors I could spot. The main issue I was having was how I was writing my main DPLL function which made all the recursive calls. Debugging was a tedious process which took a while for me to find errors. I would say the part I found surprisingly easy was making each helper method which handled pure symbols and unit clauses. I thought these parts would be the most time consuming but they were fairly straightforward and easy to understand and implement. I did not test on additional test cases and the only other test cases my code ran on were the randomly generated test cases. My program tends to run much slower with more than 60 variables, however it can handle upwards of 1000 clauses



The graph above shows the runtimes of 3-CNF sentences vs. the ratio of their number of clauses and number of symbols. This is very similar to the graph shown in the assignment specifications as it shows that the hardest problems have a ratio of around 4.3. It was very interesting to see how we can graph the performance of the algorithm and make it visually understandable.