Matthew Harrison
18/10/20
COMP 4418

Finding the “Hardness-Peak” of the 3-SAT Problem Empirically

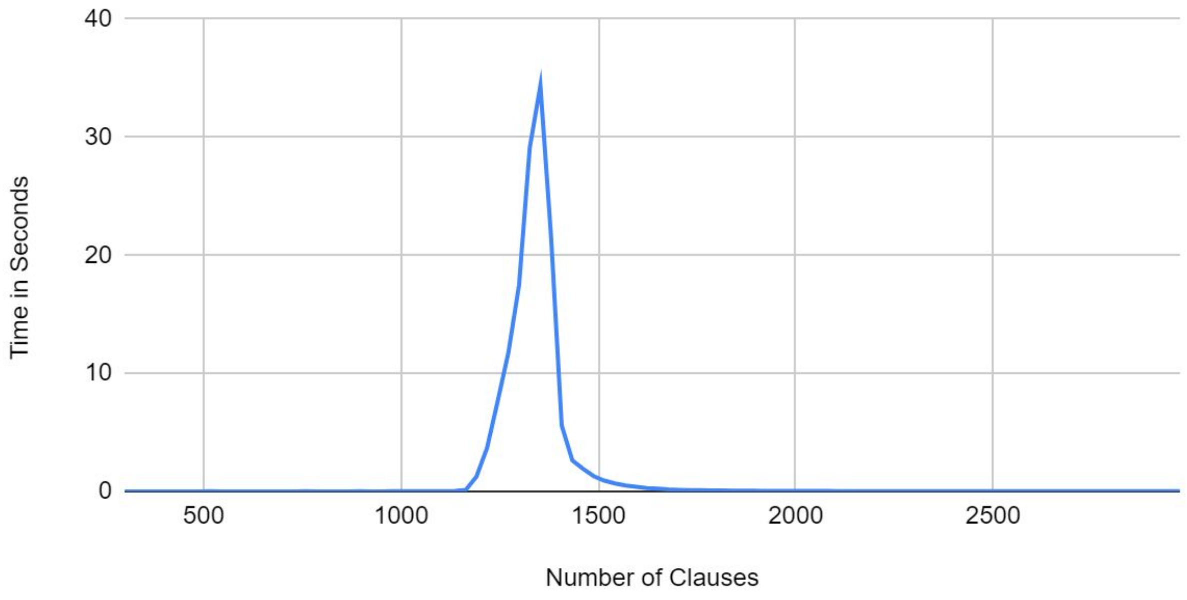
The 3-Satisfiability problem follows an easy-hard-easy curve with respect to the number of clauses, as described in the project specification. To find the peak of this curve, which represents the longest time taken to solve a 3-SAT problem, I used Python to generate numerous random 3-SAT problems. Then, I used shell scripts to run the **Minisat** satisfiability problem solver on these randomly generated files and average the results.

I ran three tests, each of which took between four and eight hours in total. These tests had different values of *p*, the number of propositional variables, and different ranges and step sizes for *c*, the number of clauses.

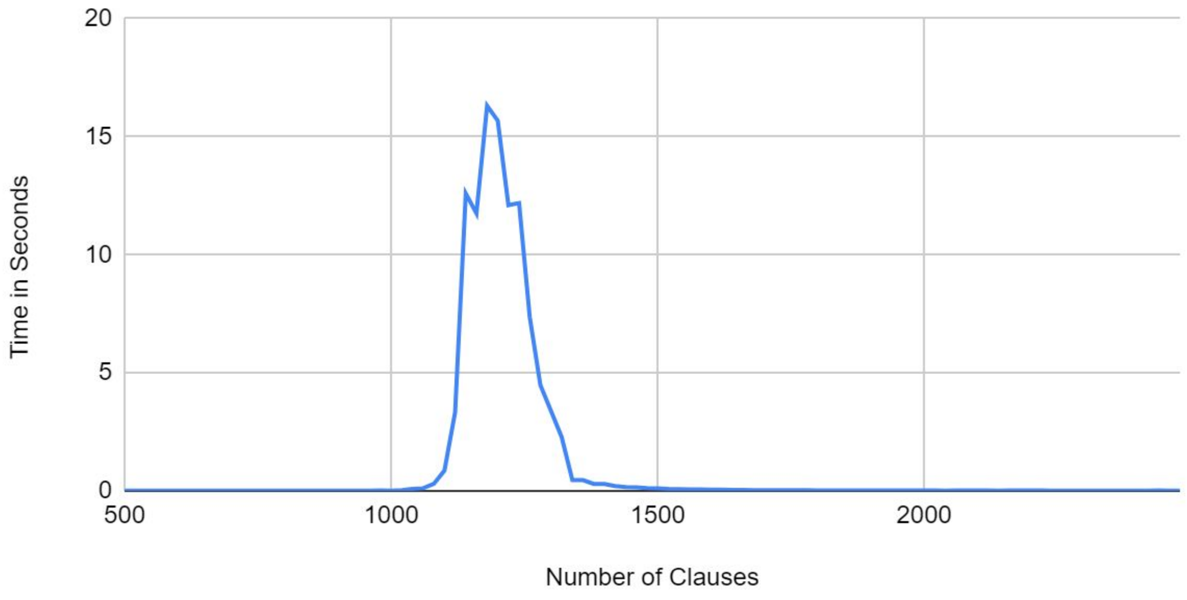
For the first test, **runTests.sh** called **tsg.py** to generate 100 **.cnf** files (named **threeSAT*x*.cnf**, where x was some value in the range *[0, 99]*) with *p = 300* and *c = 300*. Then, **Minisat** was called on each **.cnf** file, and the output was piped into **timeData.txt.** Next, **avgData.py** was called to open **timeData.txt**, read the times of each test into an array, average the 100 times, and write the result to **testTimes.txt**. This process was repeated for a further 99 values of *c*, until *c = 3000*. Then, with **runTests.sh** concluded, I took the values from **testTimes.txt** and used a spreadsheet to plot the average time to solve a problem against the number of clauses in the problem. These plots are shown below.

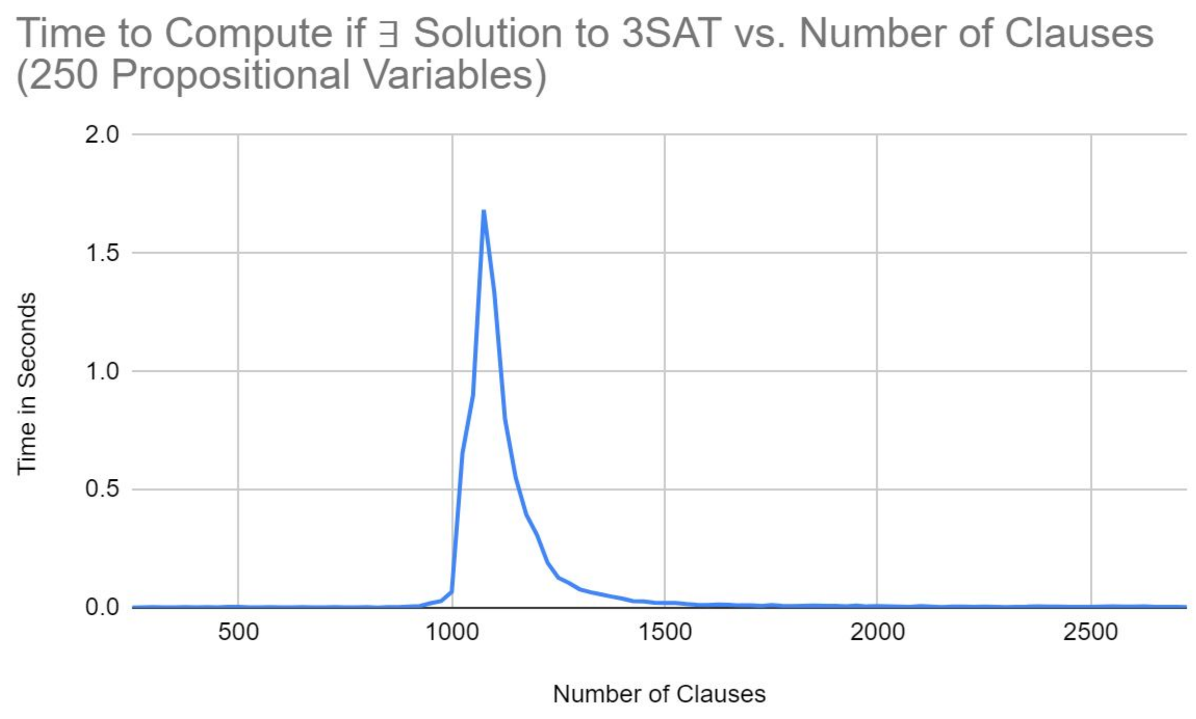
This process was repeated for tests with *p = 275* and *p = 250*, though to save time, the number of **.cnf** files produced by **tsg.py** was reduced from 100 to 50.

Time to Compute if ∃ Solution to 3SAT vs. Number of Clauses (300 Propositional Variables)



Time to Compute if ∃ Solution to 3SAT vs. Number of Clauses (275 Propositional Variables)





After the data was plotted, the ratio of the number of clauses to the number of propositional variables was calculated for each test, averaged, and found to be equal to *4.37*.

Number of Propositional Variables	**Ratio of (Number of Clauses / Number of Propositional Variables) for Peak difficulty**
250	*4.30*
275	*4.29*
300	*4.51*
Average	*4.37*

While conducting my research, I had numerous problems with the data collection. One notable issue was a bug in my shell script, where I had neglected to remove the previous **testTimes.txt** file before averaging the results of the next test. This led to a running average of all the times leading up to a given number of clauses, giving my plots a bizarre shape. This was fixed by including the **clean.sh** file to clean the directory after each iteration of testing.

Another issue I encountered, and perhaps the most significant one, was finding useful values for *p* before I even started testing. To find the values of *p* that I ended up using, I ran a much smaller version of the tests I used to collect my data. Instead of collecting data at 100 intervals, I collected data at 5 intervals - *1 * c, 2 * c, 3 * c, 4 * c*, and *5 * c*. I started with *p = 1000,* and then decreased it until some of my tests ran quickly and others took longer. I repeated the tests a few times at each stage to try to minimize the effect of randomness, and then I ended up honing in on *p = 300* as my upper bound.