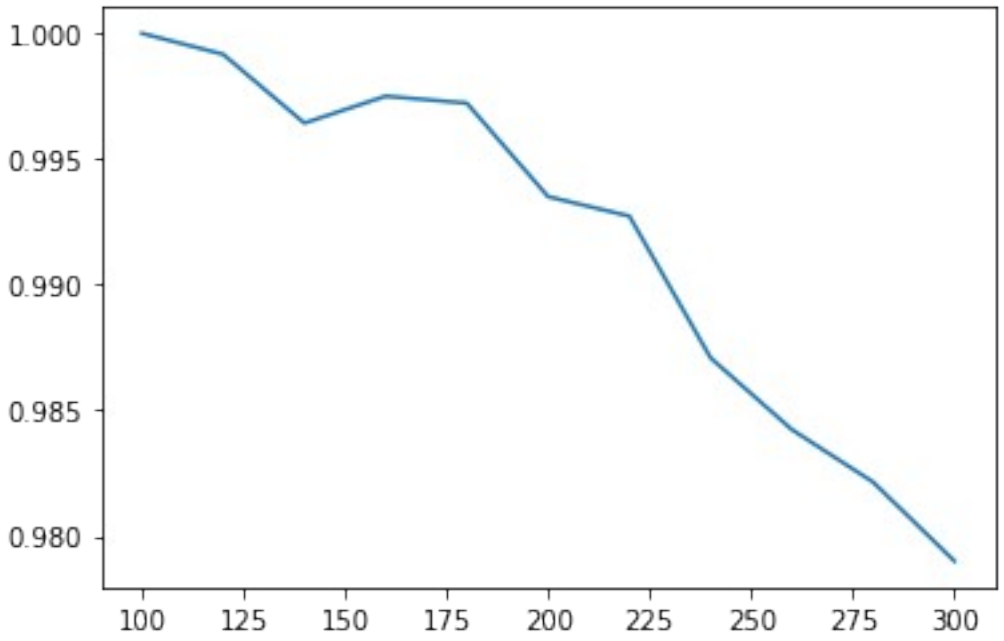
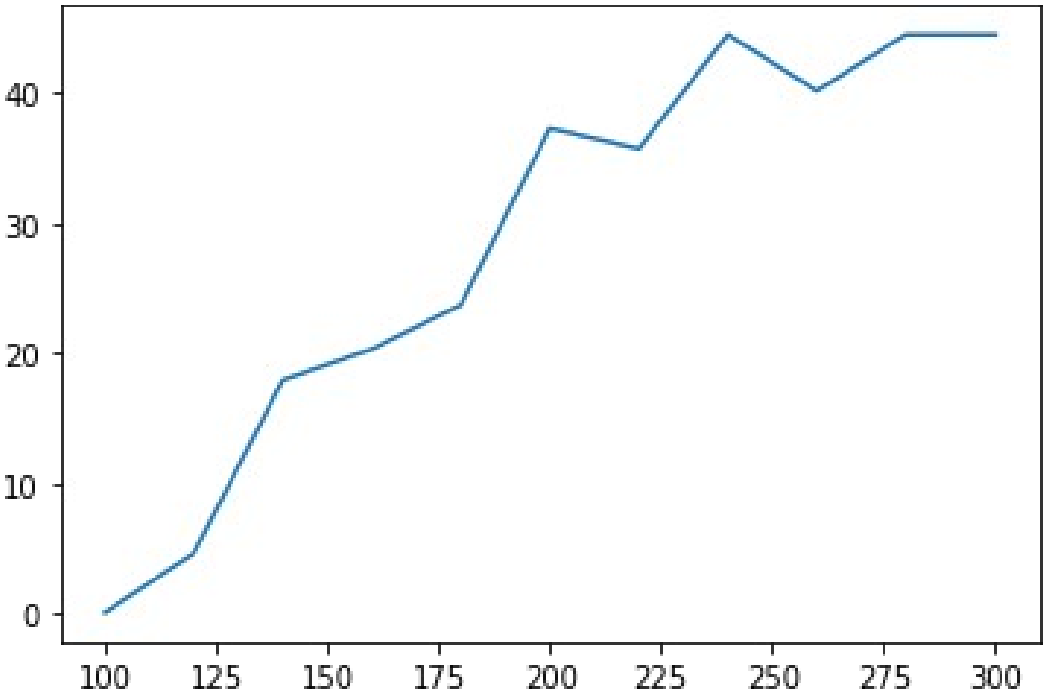
1) Average fitness function for best model found for different values of number of clauses m



2) Average running time for best model found for different values of number of clauses m



3) The genetic algorithm was improved by first increasing the number of parents to 3 rather than 2. This showed substantial improvement over the original algorithm. Then, I had tried to increase the mutation rate, which gave positive results. However, I then tried to set the mutation rate as decreasing with the time elapsed, which did not give good results and in fact hindered the algorithm in some cases. After this, I tried improving the algorithm by increasing the number of parents to 4, which gave a slight improvement over using 3 parents. I also implemented elitism, by having the most fit 20% of the population continue to the next generation without generating children. This greatly improved the speed of my code.

4) From the above graphs, it would seem that the genetic algorithm finds it difficult to obtain a good solution when the solution space is small, or when there are only a few possible states which satisfy the requirements. It appears to become stuck at local maximums and is unable to improve its fitness value after several generations for cases with a higher value of m.

5) A 3 CNF sentence becomes difficult to satisfy when there are clauses that conflict with each other. If a clause is only true when a certain other clause is false, it is impossible to satisfy the sentence. The likelihood of this happening increases when the number of clauses are increased.