Coding Progress

Aaryan Gupta

July 2022

1 27 June - 1 July

This week I mainly focused on these three tasks:

1.1 RoughMC

This program returns the minimum number of XORs hashes m_{rough} that make our CNF formula UNSAT from SAT. Keep in mind that this is a randomised algorithm so it might return a different answer every time. I have uploaded an implementation with a few bugs in the roughmc branch of the approxmc_private_improvements fork. I will try and debug it and run tests by next week.

1.2 Reduced binary search values

Given the value of m_{rough} from the previous part (hardcoded in the configuration file for now as the RoughMC part is under construction), we output m_1 and m_2 such that the value of m found earlier by logSATsearch, m_{true}, is such that $m_{rough} - m_1 \leq m_{true} \leq m_{rough} + m_2$. The values of m_1 and m_2 were found theoretically in an earlier report, depending on only δ. I have changed the binary search for m in logSATsearch from $0 \implies n$ to $m_{rough} - m_1 \implies m_{rough} + m_2$. This code (again with a few minor bugs due to the dependencies) has been implemented in the main branch of the approxmc_private_improvements fork.

1.3 Hashing Probability values higher than 3600

The approach 2 I worked on last week led to probability values slightly higher than the ones that are currently implemented (probably due to an optimisation that I am missing out on). However, the code runs much faster than the old opt.py code. So we can calculate probability values for rows higher than 3600. I suggest using the old values for rows 0-3600 and using the approach 2 values for rows 3600-6000(roughly) and approach 1 values after that as it is extremely fast. I have implemented the code which converts the probability values into the table format in constants.cpp in ApproxMC. I am running the code for 24 hours and will create a push request to the main public ApproxMC code.