COS 516: Class Project Outline
Testing SAT Heuristics

Natalia Perina, Pranjit Kalita

November 24, 2016

1 Introduction

Describe in a few sentences WHAT problem you will be addressing. Feel free to use the text in the list of projects as a starting point, but make it specific to the scope of your project.

In our project, we will be studying the effects of changing decision heuristics and formula processing techniques on three different classic NP-complete problems. Specifically, we will study the effects of random decision making, VSIDS decision making (default heuristic) and DLIS decision making, and formula preprocessing in the miniSAT solver individually for the three aforementioned classic problems. We intend to use the finding of our experiments to draw parallels between termination and solvability in terms of efficient memory usage (if applicable) and time taken to execute.

2 Proposed Work

Describe briefly HOW you will be solving the problem. Provide as many relevant details as you can about the following items (not all of these might apply to your project).

- Tools: MiniSAT solver, Trace Generator and Trace Analyzer developed by Princeton Senior Thesis
- Examine the heuristics on 3 NP-Complete problems: traveling salesman, K-coloring, and longest common subset problem.
- Modules added would be the implemented for each heuristic that we test along with miniSAT implementation for the 3 problems. Implement modules/code for random decision, DLIS decision, VSIDS decision (default in miniSAT), and pre-processing on/of.
- Experiments that we conduct will be running each of the problems with the varying heuristics for the SAT solver and comparing across the boards. Experiments include the random decisions, VSIDS, DLIS decision heuristics, and pre-processing turned on and off. We can test varying outputs for these experiments which would include running time, termination of program, variable assignments, the useful work performed as produced by the trace analyzer. We can test these on increasing size of the problems, and if the size becomes to large for the memory of the trace analyzer then we can simply record performance.
- Deliverables will be the code for each module/heuristics, so implementations for the random, VSIDS, and DLIS decisions heuristics, and pre-processing on/off in the SAT solver. Scripts for running the tests on each of the heuristics and for testing large values will also be added to the deliverables. Code for traveling salesman and for largest common subset implementation of miniSAT would also be a deliverable. For example, this could include working examples, working code, scripts for running tests/experiments, result output from tools, etc. Result output will be the output of the tests, and we could potentially visualize some of the outputs with graphs.

3 Proposed Schedule

- Nov. 7th: Research heuristics/different options, get a basic idea of DLIS and preprocessing on/off and how they're done
- Nov. 14th: miniSAT implementations of 3 test problems, finish literature review for DLIS decision heuristic
- Dec 12th: Have implemented code for different heuristics/options in miniSAT and produced early results
- Jan 5th: created and run final testing scripts and results for analysis
- Jan 17th: Final Project Report

4 Progress Report

- Things we have completed: We already have the MiniSat implementation for the k-coloring program from the homework. We have also created MiniSat implementations for the longest common sub sequence problem. There are some bugs to work out for the longest common sub sequence problem over the Thanksgiving holiday.

  The traveling salesman problem, being a minimization problem and not just a decision problem which the miniSat solver is designed to model, would require a lot of peripheral code to represent the greedy algorithm that determines the next node traversed at each step. Due to these constructive heuristics, upon consulting with Professor Gupta, we decided to not waste too much of our time on it. Instead, we can just alter what we already had finished up for setting up clauses of the traveling salesman problem to implement a similar decision problem - the Hamiltonian Cycles problem.

- Expected to complete by December 12th: We wrote our original report thinking that all had to be finished by December 12th, but now aim to have the code implemented for the decision heuristics and options in the miniSAT and have produced some results but not all by the 12th.

  We are committed to replacing the Traveling Salesman problem with the Hamiltonian Cycles problem, which we will work on over the Thanksgiving holiday.

  Lastly, both of us have committed ourselves to reading up on Victor Ying's B.S.E thesis, which we believe will help us be more efficient with the project in the final weeks ahead. Accordingly, we will take the Thanksgiving break to go over his thesis carefully.

- Plan alterations: As noted above, we now are aiming to have the code written for the different decision heuristics by the 12th but finish the testing into Dec/ early January to have it wrapped up to write the final report by January 17th. We also decided to remove the Traveling Salesman problem from being one of the problems we would test our heuristics on, and replace it with the Hamiltonian Cycles problem, time willing.