DEPARTMENT OF COMPUTER SCIENCE, ROCHESTER INSTITUTE OF TECHNOLOGY

Performance Evaluation of $k-SAT$ Solvers
Applied to Graph Arrowing

Christopher Wood
Advisor: Professor Stanisław Radziszowski

March 26, 2013

1 Introduction

In his 1972 seminal paper entitled, "Reducibility Among Combinatorial Problems," Karp introduced a list of 21 NP-complete problems, including Boolean satisfiability, the maximum cut of a graph, and 0-1 integer programming [1]. The complexity of these problems was proven by deriving a polynomial-time reduction from CIRCUIT-SAT = $\{\langle C\rangle : C$ is a satisfiable Boolean combinational circuit $\}$, the first problem shown to be NP-complete by Cook in 1971 [2], starting the rush of complexity theory research.

The problem $3-SAT$, or more formally, $3-CNFSAT$, is a special case of satisfiability. It is a decision problem in which takes as input a 3-CNF Boolean formula and returns YES if the formula is satisfiable, and NO otherwise [3]. A 3-CNF formula, more formally known as a Boolean formula in 3-conjunctive normal form, is expressed as the Boolean AND of arbitrarily many clauses, where each clause is the Boolean OR of three literals, which is a Boolean variable or its negation. Such a Boolean formula is said to be satisfiable if and only if there exists an assignment of truth values to the variables such that substituting them into the literals of the formula will cause it to evaluate to true (or 1). Expressed as a formal language, we have that $3-SAT = \{\langle\phi\rangle : \phi$ is satisfiable $\}$.

In 2002 Hans van Maaren of and John Franco initiated the public SAT competition in search of optimal performing SAT solvers judged by a variety of criteria and specializations, including their ability to demonstrate satisfiability and exhaustively prove unsatisfiability. In addition, since SAT is a problem that often arises in academia and the industry, each of the candidate solvers are rigorously tested with massive application-specific, crafted, and random Boolean formulas as input. With three different solver specializations tested against three different types of inputs, and a first, second, and third place awarded to the candidates, a total of 27 possible trophies are awarded each year. In the most recent competition held in 2011, solvers were tested using CPU time and world-clock time as a basis for their results, thus expanding the trophy space to 54 slots. The next competition is slated to take place in 2013.

The SAT problem is particularly interesting when applied to graph arrowing. It can be shown how to reduce the question $G\overbrace{\rightarrow}^{?}\rightarrow(3,3)^e$ to an equivalent $3-CNF$ formula ϕ_G such that $G\not\rightarrow(3,3)^e \Leftrightarrow \phi_G$. Intuitively, this is a very promising technique for determining if $G\rightarrow(3,3)^e$ for K_4-free graphs G.

The immediate application of this technique is to attack the upper bound of the Folkman number $F_e(3,3;4)$. In particular, to lower this bound, we will need to find a graph G on n vertices where $G\rightarrow(3,3)^e$. It has been conjectured that $G_{127}=G(127,3)=(\mathbb{Z}_{127}, E=\{(x,y)|x-y=\alpha^3 \mod 127\})$ is a prime candidate for witnessing an upper bound of 127 because of its denseness and large number of triangles. To determine whether G_{127} is indeed a witness we will decompose the problem of arrowing into problems on subgraphs H that witness $H\not\rightarrow(3,3)^e$, and then carefully extend H to encompass all of G. For each subgraph H will generate a corresponding 3-CNF formula ϕ_G by mapping the edges in $E(H)$ to variables in $\phi_H\in$ 3-SAT, and for edge adding the following clauses to ϕ_H:

$$(x+y+z)\wedge(\bar{x}+\bar{y}+\bar{z})$$

If H can be extended to encompass all of G and ϕ_G is shown to be unsatisfiable, then $G\rightarrow(3,3)^e$, and so $F_e(3,3;4)=127$.

Unfortunately, while this approach seems simple upfront, the complexity of the formulas ϕ_H with $n\approx 85$ has proven to be very difficult for modern SAT solvers to handle. In this study, we will attempt to determine the structure of these formulas that makes them so difficult to solve. We will also present a comprehensive performance comparison for the popular SAT solvers entered into the SAT competition, including Minisat, zChaff, glucose, ppfolio //, ppfolio seq, contrasat hack, and 3S.

2 $k-SAT$ Algorithms

TODO: most solvers use a Minisat variation - articulate that here for completeness

3 Selected $k-SAT$ Solvers

TODO: discuss each

4 Performance

TODO: determine test cases first

References

[1] Richard Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations, (RE Miller and JM Thatcher, eds.)* (1972), 85âĂŞ103.

[2] Stephen A. Cook. The Complexity of Theorem-Proving Procedures. *In Proceedings of the third annual ACM symposium on Theory of computing (STOC '71)*. ACM, New York, NY, USA (1971), 151-158. DOI=10.1145/800157.805047. `http://doi.acm.org/10.1145/800157.805047`

[3] Gormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms. MIT Press 44 (1990), 97-138.

[4] Niklas Sörensson and Niklas Eèn. Minisat v1.13 - A SAT Solver with Conflict-Clause Minimization. *SAT 2005* (2005) 53.

[5] Yogesh Mahajan, Zhaohui Fu, and Sharad Malik. Zchaff2004: An Efficient SAT Solver. *Theory and Applications of Satisfiability Testing.* Springer Berlin/Heidelberg (2005).