

Project Milestone 1

Edward He
<https://ehe9991.github.io/>
Marijin Heule, CS Department

February 3, 2022

1 Progress Report

So far, I have been looking to familiarize myself with SAT solving. This involves reading papers about SAT solvers in general, and using some for small test cases. Currently, I am studying the original ProbSAT paper, and reading more on how I can modify the source code of YalSAT accordingly. I have experimented with running the binary, but have not started modifying the source code as of yet.

In the ProbSAT paper, we see that there is the potential to improve performance dramatically as compared to existing methods of SAT solving [1]. This is done by randomly selecting based on a heuristic, instead of deterministically selecting an unsatisfied literal to flip. The paper points out that the constant (c_b) involved in the heuristic varies based on each problem, and that this value can have dramatic effects on performance.

There is also another constant that appears in some heuristics, called (c_m), but researchers found that simply adjusting c_b suffices. Adjusting the heuristic to a polynomial instead of exponential heuristic, we see that for low values of k , it performs worse, but for higher numbers of literals per clause, we see that it actually performs better at times.

Another paper relevant to my project that I read was Biere and Armin's paper on YalSAT [2]. Although not particularly detailed, it provides context of the solver in a competition pitting it up against deterministic algorithms. They found that in general, YalSAT performed well compared to alternative methods. This lends credibility to the importance of probabilistic local search (stochastic local search).

Finally, another influential paper I looked over was Lorenz and Nickerl's paper on the possibility of restarts in ProbSAT [3]. They performed statistical analysis on how ProbSAT could be sped up by restarting the process. Intuitively, this can improve the performance because ProbSAT can get stuck in some nearly optimal states/local optimums. Their potential calculated speedup was 39%, a significant improvement. However, they state the difficulty of actually finding this optimal restart time. As a result, this warrants future research, and experimentation.

In the future, I intend to modify this c_b value to see the effects for myself. As of now, I have experimented with various deterministic SAT solvers just to simplify the thought process moving forwards. Then, I will have the prerequisite knowledge to modify the stochastic SAT solvers (such as YalSAT).

Although I have not met my goal, I do believe that I am making good progress. I believe I have solidified my knowledge on many SAT algorithms, and can compare and contrast their effectiveness. For example, I have read up on DPLL and its differences from CDCL SAT solvers, and how these differ from the local search that I am performing in this project. This allows me to weigh the benefits of SLS.

2 Reflection on Initial Plan

Major Changes:

So far, no major changes have been made to my research plan. I intend to still study ProbSAT, and to modify YalSAT in order to improve performance of the stochastic local search solver.

Meeting your Milestone:

So far, I have almost reached my milestone, but not quite. Although I am still looking through the source code, I have not started experimenting with modifying it. The source code was a little bit hard for me to understand, so it takes me more time to look through. Furthermore, finals and projects have been mounting up, so I have been dedicating more time to other classes. To remedy this, I will start modifying the source code and just experimenting soon. Hopefully, I can begin running some benchmarks on my own laptop.

Surprises:

One surprise, as mentioned previously, was that the source code was a little bit hard to understand at time. Although modestly commented, a lot of it is in C, which I am rusty in. I have been touching up on my C knowledge, and will practice writing some low level programs to improve my knowledge.

Revisions to your 07-400 milestones:

I have not made many revisions to my milestones. One thing I could consider is pushing back some of the deadlines due to me taking some more time to look at the source code, but I will discuss this with my mentor further.

Resources needed:

I have obtained the source code for my required resources, and will be looking to modify this. On top of this, I may need more powerful computing hardware, but this will be more important later. For now, I can just run the binaries on my local machine or on the Andrew clusters.

References

- [1] Adrian Balint and Uwe Schöning. Choosing probability distributions for stochastic local search and the role of make versus break. In *SAT*, 2012.
- [2] Armin Biere. CaDiCaL, Lingeling, Plingeling, Treengeling, YalSAT Entering the SAT Competition 2017. In Tomáš Balyo, Marijn Heule, and Matti Järvisalo, editors, *Proc. of SAT Competition 2017 – Solver and Benchmark Descriptions*, volume B-2017-1 of *Department of Computer Science Series of Publications B*, pages 14–15. University of Helsinki, 2017.

- [3] Jan-Hendrik Lorenz and Julian Nickerl. The potential of restarts for probsat. *Lecture Notes in Computer Science*, page 352–360, 2020.