Reactive Dynamic Local Search algorithms for the Satisfiability Problem

CPSC 532D Course Project Proposal

Frank Hutter
mail@fhutter.de

Dave Tompkins
davet@ece.ubc.ca

Department of Computer Science
University of British Columbia
Vancouver

February 4, 2002

1 Motivation & Background

Stochastic Local Search (SLS) algorithms are of interest to a large number of research areas as well as business applications because they constitute some of the current best approaches to efficiently solve hard (NP-complete) combinatorial problems approximatively [1].

Satisfiability (SAT) problems, forming one of the most prominent type of NP-complete problems, are especially suitable for the study of SLS algorithms because they are easy to understand, and can represent a wide variety of problems. Numerous algorithms are available to solve SAT problems, and they often have varied results, depending on the type, characteristics and the source of the problem [2].

One promising family of SLS algorithms is known as Dynamic Local Search (DLS) [2]. While all SLS approaches have an evaluation function to guide the local search, DLS approaches are generally characterized as having dynamic evaluation functions that can change over time. When a DLS algorithm is trapped in a local optima (and possibly in the process of getting stuck), it can escape by changing the evaluation function such that improvement is possible. For SAT solving DLS algorithms, the evaluation function typically uses weights or penalties assigned to the clauses.

Some recent DLS algorithms have been very successful, and are considered amongst the current state of the art, such as the Exponentiated Subgradient Algorithm (ESG) [3]. It would appear that there are opportunities to explore and analyze existing DLS algorithms, and to research the effects of combining some successful DLS techniques with traditionally non-dynamic approaches. Furthermore, many of the DLS techniques are very sensitive with respect to their tunable parameters. There is potential for a reactive DLS approach that could eliminate this dependency on tuned parameters.

2 Project Description

The main objective of our project is to investigate the effect of DLS techniques on current non-dynamic SAT algorithms such as GSAT, WALKSAT, TABU, and their variants. For this purpose, we will examine existing dynamic weighting schemes and survey their results. We will then extract promising approaches and incorporate them into an encompassing framework. One important point is to make the choice of evaluation function and dynamic weighting scheme explicit, so that the performance of different approaches can easily be compared.

In addition to our main objective, we will also look at memory schemes and approximation techniques. Finally, we plan on examining the sensitivity of the various algorithms to their parameters, and employing Machine Learning techniques for reactive parameter optimization.

Our intended work can be divided into several stages that might not necessarily be worked through consecutively.

2.1 Examine Existing Weighting Systems

We plan on examining existing DLS weighting systems to determine when they are successful and (often more importantly) unsuccessful. Some work has already been undertaken to understand the aforementioned ESG work from Schuurmans et. al. so at least one scheme is available. However, it would be beneficial to seek out and obtain some additional algorithms, if for no other reason than to have results to compare and contrast against.

Depending on how generally we define DLS algorithms, we can also include algorithms that may not traditionally be considered DLS. For example, TABU-based algorithms could be classified as weighted DLS algorithms, where the literal weights are binary (forbidden / allowed).

We see a potential benefit in having weights applied to individual literals in a TABU-like manner, even though traditional DLS based SAT algorithms look at clause weights exclusively [7, 4, 5, 6]. There could be some potential for these two approaches to be combined; however, we may find that this approach has already been attempted, or that it will be largely unsuccessful.

2.2 DLS in Traditionally Non-weighted Algorithms

The main focus of our project will be to study the effect of DLS techniques when incorporated in already successful algorithms. The primary objective will be to reduce the number of search steps, and measure the sensitivity of the algorithm to the search parameters. As with many complicated hybrid approaches, the interaction of the different features will be difficult to understand. We plan on designing an encompassing framework to facilitate interchangeability between the various approaches, and to experiment with different evaluation functions and dynamic weighting schemes.

2.3 Memory Operations

In some DLS weighted schemes, there are memory mechanisms in place so that the algorithms can 'forget' weights. This can be seen in the smoothing mechanism in the ESG algorithm, similar to the decay found in Ant Colony Optimization (ACO). We plan on conducting some work to see if a successful memory scheme found in one algorithm will have similar effects in other algorithms.

2.4 Fast Approximations

One large drawback of DLS approaches is the added complexity required to calculate the dynamic weights. We will look at the time performance of different systems, and explore various approximation methods and bookkeeping techniques to reduce their execution times.

2.5 Machine Learning

As with most SLS techniques, the success of an algorithm will depend on tuned parameters. Initially, we will collect statistical data from numerous algorithm executions over various instances, correlating parameter settings, runtime behaviour, and problem instance characteristics.

Once the data is collected, we will look into Machine Learning techniques to reactively change the parameters of an algorithm during execution. These techniques will take into account the features of the current problem instance, and the dynamic behaviour of the algorithm.

3 Timeline

Week of	Completed task
February 11th	Literature review
February 18th	Snowboarding week :-)
February 25th	Frameworks [code] *
March 4th	Statistics collection & memory experiments
March 11th	ML framework & speed optimization
March 18th	Final optimization & empirical evaluation
March 25th	Final report

*progress report

References

[1] H.H. Hoos, *Stochastic Local Search - Methods, Models, Applications*, PhD thesis, Department of Computer Science, Darmstadt University of Technology, 1998.

[2] H.H. Hoos and T. Stützle, *Stochastic Local Search - Foundations and Applications*, Morgan Kaufmann Publishers, to appear in 2003.

[3] D. Schuurmans, F. Southey, and R.C. Holte. The exponentiated subgradient algorithm for heuristic boolean programming. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 334-341. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2001.

[4] J. Frank. Learning short-term clause weights for GSAT. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pages 384-389. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1997.

[5] P. Morris. The breakout method for escaping from local minima. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 40-45. AAAI Press / The MIT Press, Menlo Park, CA, USA, 1993.

[6] B. Selman and H. Kautz. Domain-independent extensions to GSAT: Solving large structured satisfiability problems. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 290-295. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1993.

[7] B. Cha and K. Iwama. *Adding new clauses for faster local search*, In Proceedings of the 15th National Conference on Artificial Intelligence, pages 332-337. AAAI Press / The MIT Press, Menlo Park, CA, USA, 1996.