Preliminary Results on Exploration-Driven Satisfiability Solving

Md Solimul Chowdhury and **Martin Müller** and **Jia-Huai You**
Department of Computing Science, University of Alberta
Edmonton, Alberta, Canada.
{mdsolimu, mmueller, jyou}@ualberta.ca

Abstract

In this abstract, we present our study of exploring the SAT search space via random-sampling, with the goal of improving Conflict Directed Clause Learning (CDCL) SAT solvers. Our proposed CDCL SAT solving algorithm *expSAT* uses a novel branching heuristic `expVSIDS`. It combines the standard VSIDS scores with heuristic scores derived from exploration. Experiments with application benchmarks from recent SAT competitions demonstrate the potential of the *expSAT* approach for improving CDCL SAT solvers.

Introduction

An inherent issue of any heuristic guided search method is the inaccuracy of heuristic estimation, which may result in poor search guidance. Exploration can potentially make a search more robust by mitigating "early mistakes" caused by inaccurate heuristics (Xie et al. 2014). Examples of exploration methods are Monte Carlo Tree Search (MCTS) and random walk techniques, which have been successfully applied to deterministic planning (Nakhost and Müller 2009). Perhaps the best known example, which combines many of the recent advances in both MCTS and machine-learned heuristics, is the super-human strength Go-playing program AlphaGo (Silver et al. 2016).

Complete SAT solvers based on the DPLL framework employ heuristics-guided state space search. Conflict Driven Clause Learning (CDCL) SAT solvers, such as GRASP and Chaff substantially altered the DPLL framework by adding conflict analysis and clause learning. The key decision-making step in a CDCL SAT solver is using a *branching heuristic* to select a variable from the current set of unassigned variables, and making a boolean assignment to that variable. Variable selection has a dramatic effect on search efficiency. The conflict-driven *Variable State Independent Decaying Sum (VSIDS)* heuristic and its variants have been the leading branching heuristics for over 15 years. Their dominance has recently been challenged by heuristics which model variable selection as an online Multi-Armed Bandit problem (Liang et al. 2016), where the propagated assignments are viewed as deterministic exploration in the MAB framework.

In this abstract, we present *expSAT*, a novel exploration-driven extension to CDCL SAT solvers. In *expSAT*, exploration via random walks can potentially uncover valuable information about variables based on sampling *future* search states, as opposed to VSIDS, which utilizes conflict information from *past* search states. We report the empirical evaluation of the *expSAT* approach on the application instances from recent SAT competitions.

Exploration in CDCL SAT

The state of the art CDCL SAT solver *Glucose* uses an aggressive clause reduction strategy, which is primarily based on the LBD scores of the learned clauses. Clauses with LBD score of 2 or less are never deleted in Glucose. The lower the LBD score of a learned clause, the better is its quality.

Similar to VSIDS, the goal of exploration in *expSAT* is to identify branching variables, which quickly lead to conflicts. Unlike VSIDS, exploration scores are derived from sampling future search states, not from the tree search so far. We reward variables based on the quality of conflicts they generate during sampling, favoring variables that generate conflicts from which high-quality clauses with low LBD are derived during conflicts in sampling.

The *expSAT* Solver

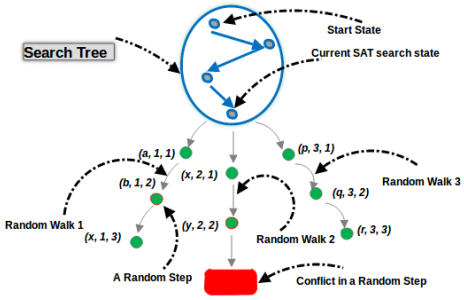
Given a CNF SAT formula $\mathcal{F}$, let $vars(\mathcal{F})$, $uVars(\mathcal{F})$ and $assign(\mathcal{F})$ denote the set of variables in $\mathcal{F}$, the set of currently unassigned variables in $\mathcal{F}$ and the current partial assignment, respectively. In addition to $\mathcal{F}$, *expSAT* also accepts five *exploration parameters* $nW, lW, \theta_{stop}, p_{exp}$ and ω, where $1 \leq nW, lW \leq uVars(\mathcal{F})$, $0 < \theta_{stop}, p_{exp}, \omega \leq 1$. These parameters control the exploration aspects of *expSAT*.

Given a CDCL SAT solver, *expSAT* modifies it as follows: (I) Before each branching decision, if the search-height[1] $\leq \theta_{stop}$, with probability p_{exp}, *expSAT* performs an *exploration episode*, consisting of a fixed number *nW* of random walks. Each walk consists of a limited number of *random steps*. Each such step consists of (a) the uniform random selection of a currently unassigned *step variable* and assigning a boolean value to it using a standard CDCL *polarity* heuristic, and (b) a followed by Unit Propagation (UP). A walk

[1] We define search-height as follows: $\frac{|assign(\mathcal{F})|}{|vars(\mathcal{F})|}$.

terminates either when a conflict occurs during UP, or after a fixed number lW of random steps have been taken. After each walk, the search state is restored and the next walk begins. Figure 1 illustrates an exploration episode. (II) In an exploration episode of nW walks of maximum length lW, the *exploration score expScore* of a decision variable v is the average of the *walk scores* $ws(v)$ of all those random walks within the same episode in which v was one of the randomly chosen decision variables. $ws(v)$ is computed as follows: (a) $ws(v) = 0$ if the walk ended without a conflict. (b) Otherwise, $ws(v) = \frac{\omega^d}{lbd(c)}$, with decay factor $0 < \omega \leq 1$, $lbd(c)$ the LBD score of the clause c learned for the current conflict, and $d \geq 0$ the *decision distance* between variable v and the conflict which ended the current walk: If v was assigned at some step j during the current walk, and the conflict occurred after step $j' \geq j$, then $d = j' - j$. Note that the values of *ws* and *expScore* for a variable are in the interval $(0, 1)$. (III) The novel branching heuristic `expVSIDS` additively combines VSIDS score and *expScore* of the unassigned variables. At the current state of the search, the variable bumping factor of VSIDS is g^z, where $g > 1$ and $z \geq 1$ is the count of conflicts in the search so far. To achieve a comparable scale for $expScore$ and VSIDS score, we scale up the *expScore* by g^z before adding these scores. Finally, a variable v^* with maximum combined score is selected for branching. (IV) All other elements, such as unit propagation, conflict analysis, restarts, and backjumping, remain the same as in the underlying CDCL SAT solver.

Figure 1: An exploration episode with $nW = 3$ walks and a maximum of $lW = 3$ random steps per walk. (v, i, j) represents that the variable v is randomly decided at the j^{th} step of i^{th} walk.



Evaluation and Future Works

Experiments and Analysis: Out first empirical evaluation of the potential of the *expSAT* approach compares two prototype systems, *expMiniSAT* and *expGlucose*, against their baselines, the well-known CDCL SAT solvers *MiniSAT* and *Glucose*. We run our experiments on 755 application instances from SAT competition 2014, SATRACE-2015 and SAT competition 2016, on a machine with 96GB RAM, 16 cores of 2.3 GHz clock speed. We used the standard 5000 second time limit. For exploration parameters, we have selected 20 representative parameter settings by performing small scale experiments on the instances from SATRACE-2015. For instances from each test set and parameter settings, we have performed experiments with *expMiniSAT* and *expGlucose*. We summarize our experimental results as follows: (I) *expMiniSAT* solves 21 more instances than *MiniSAT*, a 5.13% improvement. *expGlucose* performs par with *Glucose*. Overall solves 1 less instance, a 0.17% decline. (II) Both *expMiniSAT* and *expGlucose* solve harder instances faster than their baseline solvers.

Analysis of our experimental results reveals the some interesting insights: (I) For all three sets of application instances, we observe a small change in the performance of *expMiniSAT* and *expGlucose* as a function of the exploration parameters. (II) Compared to the other test sets, the performance of both *expGlucose* and *expMiniSAT* is better for SATRACE-2015. This result is not surprising, as the exploration parameters were tuned based on experiments on a small subset of SATRACE-2015 instances. We expect that *expSAT* can similarly be tuned for the characteristics of other test sets. Automated tuning of parameters remains a topic for future study. (III) We attribute part of the good performance of *expSAT* to its ability to select variables which can generate conflicts at a faster rate. The quality of the learned clauses, which is taken into account in the walk score computation of *expSAT*, may also contribute to its efficiency, but this requires further study.

Future Works: Currently, exploration aspects of *expSAT* is governed by parameters of fixed values. How to make those parameters adaptive during the search is an interesting question. In *expSAT*, the selection of step variables follows the uniform random distribution. A potential research direction is to use other methods, such ϵ-greedy method for step variable selection.

References

Liang, J. H.; Ganesh, V.; Poupart, P.; and Czarnecki, K. 2016. Learning rate based branching heuristic for SAT solvers. In *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings*, 123–140.

Nakhost, H., and Müller, M. 2009. Monte-Carlo exploration for deterministic planning. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, 1766–1771.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T. P.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; and Hassabis, D. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Xie, F.; Müller, M.; Holte, R.; and Imai, T. 2014. Type-based exploration with multiple search queues for satisficing planning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, 2395–2402.