Proceedings of the Ninth International
Symposium on Combinatorial Search (SoCS 2016)

Stochastic Local Search over Minterms on Structured SAT Instances

Wenxiang Chen and **Darrell Whitley** and **Adele Howe** and **Brian Goldman**
Department of Computer Science
Colorado State University
Fort Collins, CO 80523
{chenwx, whitley, howe}@cs.colostate.edu, brianwgoldman@acm.org

Abstract

We observed that Conjunctive Normal Form (CNF) encodings of structured SAT instances often have a set of consecutive clauses defined over a small number of Boolean variables. To exploit the pattern, we propose a transformation of CNF to an alternative representation, *Conjunctive Minterm Canonical Form (CMCF)*. The transformation is a two-step process: CNF clauses are first partitioned into disjoint subsets such that each subset contains CNF clauses with shared Boolean variables. CNF clauses in each subset are then replaced by *Minterm Canonical Form* (i.e., partial solutions), which is found by enumeration. We show empirically that a simple Stochastic Local Search (SLS) solver based on CMCF can consistently achieve a higher success rate using fewer evaluations than the SLS solver WalkSAT on two representative classes of structured SAT problems.

Introduction

Satisfiability (SAT) is an NP-complete problem of determining whether there exists a variable assignment such that a propositional logic formula $\mathcal{F}$ is true. SAT problem instances are usually defined in Conjunctive Normal Form (CNF): a conjunction of clauses $\mathcal{F} = \bigwedge_{c_i \in \mathbb{C}} c_i$, where each clause c_i is a disjunction of literals $c_i = \bigvee_{l_j \in \mathbb{L}_i} l_j$ and each literal is either a Boolean variable b or its negation $\bar{b}$.

Encoding structured problems into SAT problems often introduces *dependent variables* (Kautz, McAllester, and Selman 1997), whose values are defined by a Boolean function of other variables. These dependent variables are usually required by the well-known Tseitin encoding to achieve linear size conversion of propositional logic formulas to CNFs. Developing Stochastic Local Search (SLS) techniques that can effectively handle variable dependencies has been considered to be a fundamental challenge in propositional reasoning and search (Kautz and Selman 2007).

Exploiting Modularity: From CNF to CMCF

Our approach to handling variable dependencies is motivated by the fact that structured SAT instances often contain subsets of clauses consisting of a small set of variables with different combinations of polarities. To illustrate the modularity, we show a snippet (indices represent Boolean variables) of the problem of factoring semiprime 1003 in Equation 1 of Example 1. The four CNF clauses are taken from the problem in their *original order*; they only touch three Boolean variables: 24, 33 and 124. These CNF clauses represent a Boolean function $\mathcal{F}_i$, which we call a "module". *Minterms* is a conjunction (AND) of literals, e.g., $-24 \wedge -33 \wedge -124$. $\mathcal{F}_i$ can also be represented by a disjunction (OR) of the 4 minterms in Equation 2, which is called *Minterm Canonical Form (MCF)*. In MCF, the minterms are simply solutions to $\mathcal{F}_i$. Replacing each subset of CNF clauses with MCF, we obtain an alternative representation of $\mathcal{F}$, *Conjunctive Minterm Canonical Form (CMCF)*, which is a conjunction of MCFs.

$$\begin{array}{l} \qquad\qquad \ldots \qquad \wedge \\ (-24 \vee -33 \vee 124) \wedge \\ (\ 24 \vee 33 \vee -124) \wedge \\ (\ 24 \vee -124) \wedge \\ (\ 33 \vee -124) \wedge \\ \qquad\qquad \ldots \end{array} \tag{1}$$

$$\begin{array}{l} \qquad\qquad \ldots \qquad \wedge \\ (-24 \wedge -33 \wedge -124) \vee \\ (\ 24 \wedge -33 \wedge -124) \vee \\ (-24 \wedge 33 \wedge -124) \vee \\ (\ 24 \wedge 33 \wedge 124) \wedge \\ \qquad\qquad \ldots \end{array} \tag{2}$$

Example 1: Replace a subset of CNF clauses with MCF.

Roussel (Roussel 2004) proposed a transformation that uses a local model enumeration similar to ours. Roussel's encodes sets of clauses using prime implicants, while ours encodes them as minterms. However, Roussel did not show the utility of the new representation in search.

SLS Over CMCF

CMCF-Based Local Search (CMCF-LS) is a CMCF based SLS analogous to the well-known WalkSAT (Selman, Kautz, and Cohen 1994). WalkSAT has a straightforward algorithm, is very effective at solving uniform random SAT instances, and is a good basis for designing a SLS on CMCF that exploits problem structure. CMCF-LS first converts the CNF into a compact CMCF and then runs SLS.

Conversion The original CNF is translated to CMCF using a linear-time algorithm that exploits the natural ordering of CNF clauses. A module is limited to contain at most 6 variables. Auxiliary variables are introduced to break the clauses until they are within the limit. Iterating over clauses, if M_i can take the clause c_j without breaking the limit, c_j is added

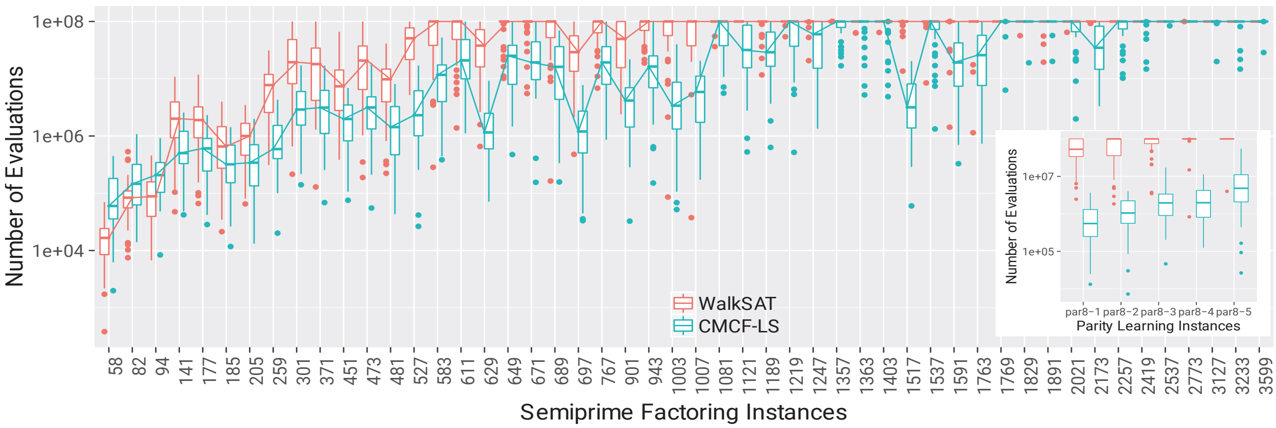

Figure 2: Number of Evaluations spent by WalkSAT (left red boxes) and CMCF-LS (right blue boxes) on Semiprime Instances and Parity Problem (the lower the better). Y-axis is a log10 scale. Boxplot shows the distribution over 50 runs.

to M_i. Otherwise M_{i+1} is created to take c_j. Then, two types of efficient constraint processing are performed.

1) *Value constraint propagation.* A Boolean variable b_i that has a consistent truth value in a MCF M_j indicates that it is the only way b_i can be set to satisfy M_j. We then propagate the assignment to other MCFs.

2) *Equivalency constraint propagation.* When two variables b_i and b_j satisfy the equivalency or negative equivalency constraint $b_i = b_j$ or $b_i = \bar{b}_j$ in M_k, b_j can be eliminated by replacing all occurrences of b_j with b_i or $\bar{b}_i$.

Stochastic Local Search With the CMCF representation, every MCF M_i contributes exactly one minterm $s_{i,j}$, where $s_{i,j} \in M_i$. A candidate solution in the search space of CMCF-LS is a vector $\mathbf{V}$ of minterms $s_{i,j}$.

Instead of flipping one Boolean variable to satisfy one CNF clause, we *change multiple variables* at a time to guarantee *all of the clauses* in a module are satisfied. Even though we are not looking for any specific variable dependencies, dependencies are automatically respected.

To select which module (set of variables) to change, we define an evaluation function that counts the support for each value (positive and negative) for each variable across the MCFs. We use the smaller of the two counts to represent the degree of disagreement on each Boolean variable.

A minterm $s_{i,j}$ from M_i effectively *votes* for the truth assignment for all variables in M_i. Each variable b_k imposes a *constraint* on a candidate solution such that b_k should only receive a single type of vote (true or false). A candidate solution that does not violate any constraint is a valid solution.

Similar to WalkSAT, CMCF-LS first selects a Boolean variable b_i that has disagreement. With probability $p = 0.1$, CMCF-LS makes a random move that forces modules containing b_i to reach a consensus state on a random truth value; with probability $(1 - p)$, CMCF-LS makes a greedy move that selects a minterm from modules containing b_i such that it yields the highest improvement in the evaluation function.

Results and Conclusions

Two classes of structured problems are used in our evaluation: Semiprime Factoring (https://toughsat.appspot.com/) and Parity Learning (Crawford, Kearns, and Shapire 1994). We ran CMCF-LS and WalkSAT each for up to 100 million evaluations of candidate solutions, or when a satisfying assignment is found. Figure 2 presents the number of evaluations required by each of the solvers on the 50 semiprime and the 5 parity instances. CMCF-LS does equal to or better than WalkSAT on all instances larger than 100. The shape of the two curves after 100 is also similar, which makes sense since the algorithmic framework of random and greedy moves is common to both.

For the parity learning problems, there is only one instance (par8-1) for which WalkSAT has a median lower than the 100 million cutoff. On par8-1, the median number of evaluations spent by WalkSAT is more than 95 times that of CMCF-LS. On par8-5, the median CMCF-LS run required 26,521 evaluations, which is over $150\times$ fewer than WalkSAT's best (and only) successful run.

We have shown that a SLS with the CMCF representation can solve two classes of structured problems more effectively than the CNF-based WalkSAT. CMCF-LS uses fewer evaluations and is more likely to yield a satisfying solution. The results presented can also be improved. We used a brute force form of greedy "best improving" SLS. Methods to track the greedy pick far more efficiently, additional constraint pre-processing and a better heuristic would likely improve the performance. But that also means that the results reported here leave considerable room for optimizations.

References

Crawford, J. M.; Kearns, M. J.; and Shapire, R. 1994. The minimal disagreement parity problem as a hard satisfiability problem. *AT&T Bell Labs Technical Report*.

Kautz, H., and Selman, B. 2007. The state of SAT. *Discrete Applied Mathematics* 155(12):1514 – 1524.

Kautz, H.; McAllester, D.; and Selman, B. 1997. Exploiting variable dependency in local search. In *Poster Sessions of IJCAI*.

Roussel, O. 2004. Another SAT to CSP conversion. In *Proc. of ICTAI*, 558–565. IEEE.

Selman, B.; Kautz, H. A.; and Cohen, B. 1994. Noise strategies for improving local search. In *Proc. of AAAI*, 337–343. AAAI.