FourierSAT: A Fourier Expansion-Based Algebraic Framework for Solving Hybrid Boolean Constraints (Student Abstract Version)*

Anastasios Kyrillidis[1], Anshumali Shrivastava[1], Moshe Vardi[1], Zhiwei Zhang[2]
[2]Rice University, Houston, TX, 77005, USA, 832-366-1300
https://www.cs.rice.edu/~zz59/
zhiwei@rice.edu
[1]Rice University, Houston, TX, 77005, USA
{anastasios, anshumali, vardi}@rice.edu

Abstract

We design `FourierSAT`, an incomplete SAT solver based on Fourier analysis of Boolean functions, a technique to represent Boolean functions by multilinear polynomials. By a reduction from SAT to continuous optimization, we propose an algebraic framework for solving systems consisting of different types of constraints. The idea is to leverage gradient information to guide the search process in the direction of local improvements. Empirical results demonstrate that `FourierSAT` is more robust than other solvers on certain classes of benchmarks.

1 Introduction

Despite of the fact that CNF is most commonly used in SAT solving, non-`CNF` constraints are playing important roles in theoretical computer science and other engineering areas, *e.g.*, `XOR` constraints in cryptography (Bogdanov, Khovratovich, and Rechberger 2011) and cardinality constraints (`CARD`) in discrete optimization (Costa et al. 2009). The combination of different types of constraints enhances the expressive power of Boolean formulas. Nevertheless, compared to that of `CNF`-SAT solving, efficient SAT solvers that can handle non-`CNF` constraints are less well studied.

One way to deal with non-`CNF` constraints is to encode them in `CNF`. However, different encodings differ from the size, the ability to detect inconsistencies by unit propagation and solution density (Prestwich 2009). It is generally observed that the running time of SAT solvers relies heavily on the detail of encodings. (Quimper and Walsh 2007)

Another way is to extend the existing SAT solvers to adapt to non-`CNF` clauses (Soos, Nohl, and Castelluccia 2009) (Elffers and Nordstrm 2018). Such specialized extensions, however, often require different techniques for different types of constraints. Meanwhile, general ideas for solving hybrid constraints uniformly are still lacking.

The primary contribution of this work is the design of a novel algebraic framework as well as a versatile, robust incomplete SAT solver—`FourierSAT`—for solving hybrid Boolean constraints. The main technique we used is the Fourier transform on Boolean functions (O'Donnell 2014). By transforming Boolean functions into "nice" polynomials, numerous properties can be analyzed mathematically. One of the attractive properties of our method is, different types of constraints are handled uniformly.

2 Problem Definition

Let $x = (x_1, ..., x_n)$ be a sequence of n Boolean variables. A Boolean function $f(x)$ is a mapping from $\{-1, 1\}^n$ to $\{-1, 1\}$, where -1 stands for `True` and 1 for `False`. A vector $a \in \{-1, 1\}^n$ is called an assignment. A formula $f = c_1 \wedge c_2 \wedge \cdots \wedge c_m$ is the conjunction of m Boolean functions, where each c_i is called a clause and belongs to a type from {`CNF` (`or`-ing), `XOR`, `CARD` (cardinality constraints), `NAE` (Not-all-equal)}.

Let the set of clauses of f be $C(f)$ with $m = |C(f)|$ and n be the number of variables of f. A model of f is an assignment that evaluates all the clauses in $C(f)$ to -1. We aim to design a framework to find a model of f.

3 A Reduction from SAT to Continuous Optimization

We use Fourier transform, a technique for transforming a Boolean function into a multilinear polynomial, to construct our objective function.

Instead of computing Fourier coefficients of a monolithic logic formula (#P-hard), we take advantage of factoring, constructing a polynomial for a formula by the Fourier expansions of its clauses. Excitingly, Fourier expansions of four types of clauses mentioned in Section 2 all have closed form representations. For a formula f, we define the *objective function* associated with f, denoted by F_f, by the sum of Fourier expansions of f's clauses, i.e.,

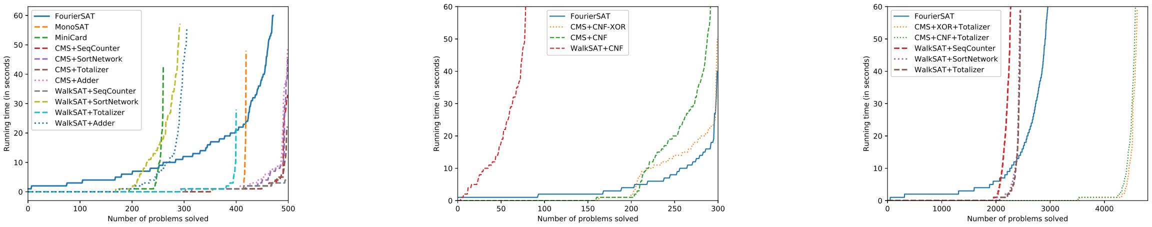
$$F_f = \sum_{c \in C(f)} \mathtt{FE}_c$$

where $\mathtt{FE}_c$ denotes the Fourier expansion of clause c.

Example 1. *Suppose* $f = (x_1 \vee x_2) \wedge (x_2 \vee \neg x_3)$*. Then,* $C(f) = \{x_1 \vee x_2,\ x_2 \vee \neg x_3\}$ *and*

$$F_f = -1 + \tfrac{1}{2}x_1 + x_2 - \tfrac{1}{2}x_3 + \tfrac{1}{2}x_1x_2 - \tfrac{1}{2}x_2x_3.$$

[*] The author list has been sorted alphabetically by last name; this should not be used to determine the extent of authors contributions.



(a) Results on vertex covering

(b) Results on parity learning with error

(c) Results on random `CNF-XOR-CARD`

Figure 1: Experimental Results (best viewed online)

Then we relax the domain from discrete to continuous. Theorem 1 reduces SAT to a multivariate minimization problem over $[-1,1]^n$. All the proofs and algorithms can be found in the supplemental material.

Theorem 1. *(Reduction) f is satisfiable if and only if*

$$\min_{x\in[-1,1]^n} F_f(x) = -m.$$

4 Theoretical and Experimental Results

We first claim that when we minimize our objective function, we are in fact maximizing the expectation of the number of satisfied clauses, after a randomized rounding.

Theorem 2. *For $a \in [-1,1]^n$, let R be a randomized rounding function defined by $\mathbb{P}[R(a)_i = -1] = \frac{1-a_i}{2}$ and $\mathbb{P}[R(a)_i = 1] = \frac{1+a_i}{2}$. Let $m_{SAT}(R(a))$ be the number of satisfied clauses by $R(a)$, then*

$$\mathbb{E}[m_{SAT}(R(a))] = \frac{m-F_f(a)}{2}$$

Compared to local search SAT solvers which only make progress when the number of satisfied clauses increases, our framework makes progress as long as the value of the polynomial decreases, even by a small amount.

Finding the global minimum of a multilinear polynomial is NP-hard in general. Theorem 3 indicates that local minima are also meaningful. It also shows how we relate a continuous assignment to a discrete one.

Theorem 3. *Suppose $a \in [-1,1]^n$ is a local minimum of F_f on $[-1,1]^n$. We construct $b \in \{-1,1\}^n$ as follows: let $b_i = a_i$ where $a_i \in \{-1,1\}$ and assign -1 or 1 arbitrarily to other coordinates of b, then*

$$m_{SAT}(b) = \frac{m - F_f(a)}{2}$$

Since the objective function is constrained, continuous and differentiable, projected gradient descent (PGD) (Nesterov 2014) is a candidate for solving our optimization problem. Theorem 4 shows that PGD converges to an approximate critical point in polynomial time.

Theorem 4. *(Convergence speed) With the step size $\eta = \frac{1}{nm}$, PGD converges to a ϵ-projected-critical point (where $||G(x)|| < \epsilon$, $G(x)$ is the projected gradient) of F_f in $O(\frac{nm^2}{\epsilon^2})$ iterators.*

A critical point can be either a local minimum or a saddle point. In the supplemental material we design algorithms to escape saddle points and identify local minima.

We implemented our method as `FourierSAT` and compared it with several SAT solvers and `CARD` encodings on three classes of benchmarks. Results are shown in Figure 1. Due to the maturity of modern SAT solvers such as CryptominiSAT (CMS), it is not surprising that `FourierSAT` is not the best solver on some benchmarks, especially for long `XOR` clauses. Nevertheless, as a versatile solver, `FourierSAT` is comparable with many well-developed, specialized solvers. We also notice that `FourierSAT` seems to suffer less from scaling exponentially.

5 Conclusion

We propose a novel algebraic framework for solving Boolean formulas consisting of hybrid constraints. Our study on multilinear polynomial leads to the discovery of attractive features of this method. The experimental results indicate that `FourierSAT` is comparable with state-of-the-art solvers on certain hybrid Boolean benchmarks. We believe this work provides a new case and perspective for bridging continuous and discrete optimization.

References

Bogdanov, A.; Khovratovich, D.; and Rechberger, C. 2011. Biclique Cryptanalysis of the Full AES. In *ASIACRYPT 2011*, 344–371.

Costa, M.-C.; de Werra, D.; Picouleau, C.; and Ries, B. 2009. Graph Coloring with Cardinality Constraints on the Neighborhoods. *Discrete Optimization* 6(4):362 – 369.

Elffers, J., and Nordstrm, J. 2018. Divide and Conquer: Towards Faster Pseudo-Boolean Solving. In *IJCAI*.

Nesterov, Y. 2014. *Introductory Lectures on Convex Optimization: A Basic Course*. 1 edition.

O'Donnell, R. 2014. *Analysis of Boolean Functions*. New York, NY, USA: Cambridge University Press.

Prestwich, S. 2009. *CNF Encodings, Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications*.

Quimper, C.-G., and Walsh, T. 2007. Decomposing global grammar constraints. In *CP 2007*, 590–604.

Soos, M.; Nohl, K.; and Castelluccia, C. 2009. Extending SAT Solvers to Cryptographic Problems. In *SAT*, 244–257.