MaxSAT Resolution with the Dual Rail Encoding *

Maria Luisa Bonet[1], Sam Buss[2], Alexey Ignatiev[3,4], Joao Marques-Silva[3], Antonio Morgado[3]
[1] Computer Science Department, Universidad Politécnica de Cataluña, Barcelona, Spain
[2] Department of Mathematics, University of California, San Diego, USA
[3] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
[4] ISDCT SB RAS, Irkutsk, Russia

Abstract

Conflict-driven clause learning (CDCL) is at the core of the success of modern SAT solvers. In terms of propositional proof complexity, CDCL has been shown as strong as general resolution. Improvements to SAT solvers can be realized either by improving existing algorithms, or by exploiting proof systems stronger than CDCL. Recent work proposed an approach for solving SAT by reduction to Horn MaxSAT. The proposed reduction coupled with MaxSAT resolution represents a new proof system, DRMaxSAT, which was shown to enable polynomial time refutations of pigeonhole formulas, in contrast with either CDCL or general resolution. This paper investigates the DRMaxSAT proof system, and shows that DRMaxSAT p-simulates general resolution, that AC^0-Frege+PHP p-simulates DRMaxSAT, and that DRMaxSAT can not p-simulate AC^0-Frege+PHP or the cutting planes proof system.

Introduction

The practical success of Conflict-Driven Clause Learning (CDCL) SAT solvers hinges on what can be construed as a relatively weak proof system, at least when compared with several others (Beame and Pitassi 2001; Buss 2012; Nordström 2015). This proof system (CDCL) is as powerful as general resolution (RES). One approach to improving SAT solvers is to exploit proof systems stronger than CDCL/RES; however, it is open whether a proof system stronger than CDCL/RES can yield more efficient SAT solvers, as attempts at exploiting extended resolution (Huang 2010; Audemard, Katsirelos, and Simon 2010) or cutting planes in SAT solvers (Nordström 2015) have been so far largely unsuccessful. Furthermore, a key issue from a practical perspective is whether stronger proof systems are automatizable (Bonet, Pitassi, and Raz 2000), and unfortunately, most results regarding automatizability are negative (Bonet, Pitassi, and Raz 2000).

Recent work (Ignatiev, Morgado, and Marques-Silva 2017) proposed a different take on SAT solving. Propositional formulas can be re-encoded, using a variant of the well-known dual rail encoding (Bryant et al. 1987; Palopoli, Pirri, and Pizzuti 1999) and then refuted with a MaxSAT solver, e.g. MaxSAT resolution (Larrosa and Heras 2005; Bonet, Levy, and Manyà 2007) or core-guided MaxSAT (Morgado et al. 2013). The re-encoded formulas are polynomial size (in fact linear). Somewhat surprisingly, the propositional encoding of the pigeonhole principle (PHP), if reencoded with the modified dual rail encoding, can be refuted in polynomial time, both with MaxSAT resolution and with core-guided MaxSAT (Ignatiev, Morgado, and Marques-Silva 2017)[1]. In contrast, the ordinary (non-dual-rail encoded) PHP formulas have exponential lower bounds for RES and CDCL (Haken 1985; Beame, Kautz, and Sabharwal 2004), but also for MaxSAT resolution (Bonet, Levy, and Manyà 2007).

This paper begins the process of charting the relative efficiency of the dual rail MaxSAT (DRMaxSAT) proof system. The first result is that DRMaxSAT p-simulates general resolution, and thus since DRMaxSAT has short proofs of the PHP, it is a strictly stronger proof system than either CDCL or RES. The paper also shows that DRMaxSAT cannot p-simulate either of AC^0-Frege+PHP or cutting planes. Finally, the paper investigates a variant of the pigeonhole principle (Biere 2013a). In practice, this variant is much harder than plain PHP formula. Experimental results on formulas encoding this variant of PHP confirm that the practical implementation of DRMaxSAT outperforms modern CDCL SAT solvers.

Preliminaries

MaxSAT and Weighted MaxSAT. *MaxSAT* is the problem of finding an assignment that minimizes the number of falsified clauses of a CNF formula. MaxSAT has several generalizations. To define them, we need to give weights to clauses, with the weight indicating the "cost" of falsifying the clause. A *weighted* clause is written (A, w) where A is a clause and $w \in \{1, 2, 3, \ldots\} \cup \{\top\}$. The value $\top$ is viewed

*This work was financially supported in part by TIN2016-76573-C2-2-P and by FCT funding of post-doctoral grants SFRH/BPD/103609/2014, SFRH/BPD/120315/2016, and LASIGE Research Unit, ref. UID/CEC/00408/2013.

[1]Earlier work (Sabharwal 2005) showed that problem-specific symmetry-breaking can serve to reduce refutations of PHP from exponential to polynomial. In contrast, the results in (Ignatiev, Morgado, and Marques-Silva 2017) exploit a general reduction to Horn MaxSAT.

as equaling infinity, but we write "$\top$" instead of "∞". A typical use of weighted clauses is for *Partial MaxSAT*, where the clauses of Γ are partitioned into *soft* clauses and *hard* clauses. Soft clauses may be falsified and have weight 1; hard clauses may not be falsified and have weight $\top$. So *Partial MaxSAT* is the problem of finding an assignment that satisfies all the hard clauses and minimizes the number of falsified soft clauses. In *Weighted Partial MaxSAT*, the soft clauses may have any (finite) weight ≥ 1. *Weighted Partial MaxSAT* is the problem of finding an assignment that satisfies all the hard clauses and minimizes the sum of the weights of falsified soft clauses.

The *MaxSAT resolution* calculus is a sound and complete calculus for MaxSAT based on resolution. This system was first defined by (Larrosa and Heras 2005), and proven complete by (Bonet, Levy, and Manyà 2007). A similar calculus can also be defined for Partial MaxSAT and Weighted Partial MaxSAT. Like classical resolution, (Weighted) (Partial) MaxSAT resolution is based on a unique inference rule. In classical resolution, every application of the resolution rule adds a new clause to the system. The inference rule for (Weighted) (Partial) MaxSAT, however, *replaces* two clauses by a different set of clauses. In other words, a clause may be used only once as a hypothesis of a (Weighted) (Partial) MaxSAT resolution inference.

The inference rule for clauses with finite weights is:

$$\frac{\begin{array}{c}(x \vee A, w_1)\\ (\overline{x} \vee B, w_2)\end{array}}{\begin{array}{c}(A \vee B, min(w_1, w_2))\\ (x \vee A, w_1 - min(w_1, w_2))\\ (\overline{x} \vee B, w_2 - min(w_1, w_2))\\ (x \vee A \vee \overline{B}, min(w_1, w_2))\\ (\overline{x} \vee \overline{A} \vee B, min(w_1, w_2))\end{array}} \tag{1}$$

The notation $x \vee A \vee \overline{B}$, where $A = a_1 \vee \cdots \vee a_s$ and $B = b_1 \vee \cdots \vee b_t$ with $t > 0$, is the abbreviation of the set of clauses

$$\begin{array}{l}x \vee a_1 \vee \cdots \vee a_s \vee \overline{b}_1\\ x \vee a_1 \vee \cdots \vee a_s \vee b_1 \vee \overline{b}_2\\ \ldots\\ x \vee a_1 \vee \cdots \vee a_s \vee b_1 \vee \cdots \vee b_{t-1} \vee \overline{b_t}\end{array} \tag{2}$$

When $t = 0$, $\overline{B}$ is the constant true, so $x \vee A \vee \overline{B}$ denotes the empty set of clauses. $\overline{x} \vee \overline{A} \vee B$ is defined similarly.

In the rule, conclusion clauses with weight 0 are omitted; e.g., at least one of the second or third conclusions is omitted; both are omitted if $w_1 = w_2$. If one or both weights are $\top$, the following rules apply

$$\frac{\begin{array}{c}(x \vee A, w)\\ (\overline{x} \vee B, \top)\end{array}}{\begin{array}{c}(A \vee B, w)\\ (x \vee A \vee \overline{B}, w)\\ (\overline{x} \vee B, \top)\end{array}} \quad \text{and} \quad \frac{\begin{array}{c}(x \vee A, \top)\\ (\overline{x} \vee B, \top)\end{array}}{\begin{array}{c}(A \vee B, \top)\\ (x \vee A, \top)\\ (\overline{x} \vee B, \top)\end{array}}$$

for finite w. The second rule is just the ordinary resolution inference, as the premises are still available as conclusions.

After applying the rule, we remove tautologies, and collapse repeated occurrences of variables in clauses. As noted, for MaxSAT inferences the premises are *replaced* with the conclusions. Note that these inferences depend on the orderings of the literals $a_1, \ldots, a_s$ and the literals $b_1, \ldots, b_t$. This means that, in general, there are multiple ways to apply the rule to a given pair of clauses.

It is easy to check that if a truth assignment τ falsifies the *formula* $x \vee A \vee \overline{B}$, then it falsifies exactly one of the clauses in (2), and similarly for $\overline{x} \vee \overline{A} \vee B$. Also, if τ makes one of the premises of (1) with weight w false, then the sum of the weights of the falsified conclusions is w. Likewise, if τ satisfies both premises of (1), then it satisfies all the conclusions. Similar considerations apply to inferences on clauses with weight $\top$. The soundness of the Weighted MaxSAT rule follows immediately.

A (Weighted) (Partial) MaxSAT refutation starts with a multiset Γ of clauses. After each inference, the multiset of clauses is updated by removing the rule's premises and adding its conclusions. The MaxSAT refutation ends with a multiset containing $k > 0$ occurrences of the empty clause $\perp$, possibly with weights.

The rules give a sound and complete system for Weighted Partial MaxSAT (Bonet, Levy, and Manyà 2007). Given a set Γ of weighted clauses and a truth assignment τ, the cost of τ is the sum of weights of the clauses that τ falsifies; the cost is infinite if some hard clause is falsified. Soundness means that if there is a derivation from Γ of empty clauses with weights summing up to k, then there is no assignment of cost $< k$. Completeness means that if k is the minimum cost of an assignment for Γ, then there is a derivation from Γ of empty clauses with weights adding up to k.

It is useful to also have the following two rules when dealing with soft clauses with weights bigger than 1.

$$\text{Extraction: } \frac{(A, w_1{+}w_2)}{(A, w_1) \qquad (A, w_2)}$$

$$\text{Contraction: } \frac{(A, w_1) \qquad (A, w_2)}{(A, w_1{+}w_2)}$$

The MaxSAT system is unusual in that its rules have multiple conclusions. This can have unexpected consequences. For example, one might expect that since soft clauses cannot be reused, this means that the portion of a MaxSAT refutation that uses soft clauses is tree-like. This is not true however, because an inference may have multiple soft clauses among its conclusions, which can be used at different points in the refutation.

Dual Rail MaxSAT. We now define the *dual rail MaxSAT* system (Ignatiev, Morgado, and Marques-Silva 2017) for refuting a set of clauses Γ. The dual rail MaxSAT system is based on MaxSAT resolution, but as already mentioned is strictly stronger than resolution.

Let Γ be a set of clauses (viewed as hard clauses) over the variables $\{x_1, \ldots, x_s\}$. The dual rail encoding Γ^{dr} of Γ, uses $2s$ variables $n_1, \ldots, n_s$ and $p_1, \ldots, p_s$ in place of the s variables x_i. The intent is that p_i is true if x_i is true, and that n_i is true if x_i is false. The dual rail encoding C^{dr} of a clause C is defined by replacing each (unnegated) variable x_i in C with $\overline{n_i}$, and replacing each (negated) literal $\overline{x_i}$

in C with $\overline{p_i}$. For example, if C is $\{x_1, \overline{x_3}, x_4\}$, then $C^{\rm dr}$ is $\{\overline{n_1}, \overline{p_3}, \overline{n_4}\}$. Note that every literal in $C^{\rm dr}$ is negated.

The dual rail encoding $\Gamma^{\rm dr}$ of Γ contains the following clauses: (1) the hard clause $C^{\rm dr}$ for each $C \in \Gamma$; (2) the hard clauses $\overline{p_i} \vee \overline{n_i}$ for $1 \le i \le s$; and (3) the soft clauses p_i and n_i for $1 \le i \le s$. Note that all clauses of Γ are Horn: the hard clauses contain only negated literals and the soft clauses are unit clauses. A dual rail MaxSAT refutation of Γ is defined as a MaxSAT derivation of a multiset of clauses containing $\ge s{+}1$ many copies of the empty clause $\bot$ from $\Gamma^{\rm dr}$. This is based on the fact that Γ is satisfiable if and only if there is a truth assignment τ which makes all the hard clauses of $\Gamma^{\rm dr}$ true, and only s of the soft clauses false (Ignatiev, Morgado, and Marques-Silva 2017).

$\Gamma^{\rm dr}$ is equivalently represented as a set of weighted clauses:

$$\begin{array}{cl} (C^{\rm dr}, \top) & \text{for } C \in \Gamma \\ (\overline{p_i} \vee \overline{n_i}, \top) & \text{for } 1 \le i \le s \\ (p_i, 1) & \text{for } 1 \le i \le s \\ (n_i, 1) & \text{for } 1 \le i \le s. \end{array}$$

More generally, given a set of finite positive weights $w_1, \ldots, w_s$, the *weighted dual rail encoding* $\Gamma^{\rm wdr}$ of Γ is defined as the set of clauses

$$\begin{array}{cl} (C^{\rm dr}, \top) & \text{for } C \in \Gamma \\ (\overline{p_i} \vee \overline{n_i}, \top) & \text{for } 1 \le i \le s \\ (p_i, w_i) & \text{for } 1 \le i \le s \\ (n_i, w_i) & \text{for } 1 \le i \le s. \end{array}$$

Letting $k = \sum_i w_i$, a *weighted dual rail MaxSAT* refutation is a MaxSAT derivation of a set of empty clauses with total weight $k{+}1$, from $\Gamma^{\rm wdr}$.

Note that each choice of weights $w_1, \ldots, w_s$ gives a different weighted dual rail encoding. What is important for the refutations is that the weights are chosen sufficiently large: any "extra" weight can be handled by using the fact that an empty clause $\bot$ can be derived from the hard clause $\overline{p_i} \vee \overline{n_i}$ and the two soft clauses p_i and n_i.

When the weights w_i are all small (i.e., polynomially bounded), then it is convenient to work with the *multiple dual rail MaxSAT* system. In this system, instead of including the clauses (p_i, w_i) and (n_i, w_i) with weights w_i possibly larger than 1, we introduce w_i many copies of the soft clauses p_i and n_i, each of weight 1. The resulting set of clauses is denoted by $\Gamma^{\rm mdr}$. Any MaxSAT derivation from $\Gamma^{\rm mdr}$ is readily converted into a MaxSAT derivation from $\Gamma^{\rm wdr}$. Conversely, if there is polynomial upper bound on the values w_i, then the size of a MaxSAT derivation from $\Gamma^{\rm wdr}$ can be converted into a MaxSAT derivation from $\Gamma^{\rm mdr}$ with size only polynomially bigger. This means that the weighted dual rail MaxSAT system is a strengthening of the multiple dual rail MaxSAT system. For the present paper, the main advantage of working with the multiple dual rail MaxSAT system instead of with the weighted dual rail MaxSAT system is that it simplifies notation for the proof of Theorem 1 by letting us discuss soft and hard clauses without explicitly writing their weights.

An example. We present a very simple example of a DRMaxSAT refutation which refutes the three clauses $\overline{x_1} \vee x_2$, x_1 and $\overline{x_2}$. This is almost the simplest possible example, but still reveals interesting aspects. The dual rail encoding has the five hard clauses

$$\overline{p_1} \vee \overline{n_2} \qquad \overline{n_1} \qquad \overline{p_2} \qquad \overline{p_1} \vee \overline{n_1} \qquad \overline{p_2} \vee \overline{n_2},$$

plus the four soft unit clauses

$$p_1 \qquad n_1 \qquad p_2 \qquad n_2.$$

Since there are two variables, a DRMaxSAT refutation must derive a multiset containing three copies of the empty clause $\bot$. The following four inferences will be used to form the refutation (the weights 1 and $\top$ are used for soft and hard clauses, respectively):

$$\frac{\begin{array}{c}(\overline{n_1}, 1)\\(n_1, \top)\end{array}}{\begin{array}{c}(\bot, 1)\\(n_1, \top)\end{array}} \qquad \frac{\begin{array}{c}(\overline{p_2}, 1)\\(p_2, \top)\end{array}}{\begin{array}{c}(\bot, 1)\\(p_2, \top)\end{array}}$$

$$\frac{\begin{array}{c}(p_1, 1)\\(\overline{p_1} \vee \overline{n_2}, \top)\end{array}}{\begin{array}{c}(\overline{n_2}, 1)\\(p_1 \vee n_2, 1)\\(\overline{p_1} \vee \overline{n_2}, \top)\end{array}} \qquad \frac{\begin{array}{c}(\overline{n_2}, 1)\\(n_2, 1)\end{array}}{(\bot, 1)}$$

We describe a DRMaxSAT refutation using these four inferences; its "lines" consist of five multisets of clauses $\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$. The initial multiset Γ_0 contains the nine clauses given above. Since the set of hard clauses never changes, each Γ_i has the form $\Gamma_i = S_i \cup H$ where H is the set of five hard clauses above, and S_i is a multiset of soft (weight 1) clauses. Namely,

$$\begin{array}{rcl} S_0 & = & \{p_1,\ n_1,\ p_2,\ n_2\} \\ S_1 & = & \{p_1,\ \bot,\ p_2,\ n_2\} \\ S_2 & = & \{p_1,\ \bot,\ \bot,\ n_2\} \\ S_3 & = & \{\overline{n_2},\ p_1 \vee n_2,\ \bot,\ \bot,\ n_2\} \\ S_3 & = & \{\bot,\ p_1 \vee n_2,\ \bot,\ \bot\}. \end{array}$$

Here S_0 is the four initial soft clauses; and S_3 contains three copies of $\bot$ as needed for a valid DRMaxSAT refutation.

There is a couple interesting observations about even such a simple derivation. First, it splits neatly into three independent parts: one that uses n_1 and $\overline{n_1}$ to derive $\bot$, one that uses p_2 and $\overline{p_2}$ to derive $\bot$, and one that uses the other clauses to derive a third copy of $\bot$. This splitting is part of the reason that DRMaxSAT can give simpler proofs than ordinary resolution, say for PHP. Second, there is an extra soft clause $p_1 \vee n_2$ that is derived but not used; this is a common feature of DRMaxSAT refutations.

The Parity principle and Dual Rail encoding of Doubled Pigeonhole principles. The present paper uses (unweighted) dual rail encodings of two combinatorial principles. The first is the *Parity Principle*, expressing a kind of mod 2 counting, which states that no graph on $2m + 1$ nodes consists of a complete perfect matching (Ajtai 1990; Beame et al. 1996; Beame and Pitassi 1996). The propositional version of the Parity Principle, for $m \ge 1$, uses

$\binom{2m+1}{2}$ variables $x_{i,j}$, where $i \neq j$ and $x_{i,j}$ is identified with $x_{j,i}$. The intuitive meaning of $x_{i,j}$ is that there is an edge between vertex i and vertex j. The Parity Principle, Parity^{2m+1}, has the following sets of clauses:

$\bigvee_{j\neq i} x_{i,j}$ for $i \in [2m+1]$

$\overline{x_{i,j}} \vee \overline{x_{k,j}}$ for i, j, k distinct members of $[2m+1]$.

These clauses state that each vertex has degree one.

The second combinatorial principle is the *Doubled Pigeonhole Principle*, also called the "Two Pigeons Per Hole Principle", which states that if $2m+1$ pigeons are mapped to m holes then some hole contains at least three pigeons (Biere 2013b). This is encoded with the following clauses $2\mathrm{PHP}^{2m+1}_m$:

$\bigvee_{j=1}^m x_{i,j}$ for $i \in [2m+1]$

$\overline{x_{i,j}} \vee \overline{x_{k,j}} \vee \overline{x_{\ell,j}}$ for distinct $i, k, \ell \in [2m+1]$.

The dual rail encoding, $(2\mathrm{PHP}^{2m+1})^{\mathrm{dr}}$, of $2\mathrm{PHP}^{2m+1}_m$ contains the hard clauses

$\bigvee_{j=1}^m \overline{n_{i,j}}$ for $i \in [2m+1]$

$\overline{p_{i,j}} \vee \overline{p_{k,j}} \vee \overline{p_{\ell,j}}$ for $j \in [m]$ and distinct $i, k, \ell \in [2m+1]$.

The soft clauses are the unit clauses $n_{i,j}$ and $p_{i,j}$ for all $i{\in}[2m+1]$ and $j{\in}[m]$. There are $(2m+1)m$ positive variables $p_{i,j}$ and likewise $(2m+1)m$ negative variables $n_{i,j}$, for a total of $2(2m+1)m$ many soft clauses. A dual rail MaxSAT refutation for $2\mathrm{PHP}^{2m+1}_m$ must produce $(2m+1)m + 1$ many empty clauses ($\perp$'s) from $(2\mathrm{PHP}^{2m+1})^{\mathrm{dr}}$.

AC^0-Frege and Cutting Planes proof systems. To be able to compare dual rail MaxSAT with resolution, AC^0-Frege and Cutting Planes, we need the following terminology. Proof length is measured in terms of the total number of symbols appearing in the proof. A proof system $\mathcal{P}$ is said to *simulate* another proof system $\mathcal{Q}$ provided that there is a polynomial $p(n)$ so that any $\mathcal{Q}$ proof of size N can be transformed (by a polynomial time construction) into a $\mathcal{P}$-proof of size $\leq p(N)$ of the same formula. For more information on proof complexity, see e.g. the surveys (Buss 2012; Pudlák 1999).

A *Frege* system is a textbook-style proof system, usually defined to have modus ponens as its only rule of inference (Cook and Reckhow 1979). For convenience in defining the depth of formulas, we can treat an implication $A \to B$ as being an abbreviate for $\neg A \vee B$. The depth of propositional formula is measured in terms of alternations: assume a formula φ uses only the connectives $\vee$, $\wedge$ and $\neg$. Using deMorgan's rules, there is a canonical transformation of φ into a formula φ' in "negation normal form", i.e., with negations applied only to variables. Viewing φ' as a tree, the *depth* of φ is the maximum number of blocks of adjacent $\vee$'s and adjacent $\wedge$'s along any branch in the tree φ'. A depth d Frege proof is a Frege proof in which every formula has depth $\leq d$. An AC^0-Frege proof is a proof with a constant upper bound on the depth of formulas appearing in the proof.

The cutting planes system is a pseudo-Boolean propositional proof system. It uses variables x_i which take on $0/1$ values, indicating Boolean values *False* and *True*. The lines of a cutting planes proof are inequalities of the form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq a_{n+1},$$

where the a_i's are integers. Logical axioms include $x_i \geq 0$ and $-x_i \geq -1$; inference rules include addition, multiplication by a integer, and a special division rule. A cutting planes proof refuting a set Γ of clauses has axioms expressing the truth of the clauses in Γ, and has $0 \geq 1$ as its last line. The cutting planes system CP uses integers a_i written in binary; the system CP^* uses the integers a_i written in unary notation. The *size* of a CP or CP^* proof is the total number of symbols in the proof, including the bits used for representing the values of the coefficients a_i. For more on cutting planes, see e.g. (Pudlák 1997; Buss and Clote 1996).

Simulations of Resolution

Tree-like resolution

Theorem 1. *Multiple dual rail MaxSAT simulates tree-like resolution.*

We start with a useful observation. The dual rail encodings include soft unit clauses p_i and n_i and hard clauses $\overline{p_i} \vee \overline{n_i}$. Applying a MaxSAT inference to p_i and $\overline{p_i} \vee \overline{n_i}$ yields the two soft clauses $\overline{n_i}$ and $p_i \vee n_i$. Combining $\overline{n_i}$ and n_i with a MaxSAT inference yields the clause $\perp$. Thus, we have used up the soft clauses p_i and n_i and obtained one instance of $\perp$ plus the clause $p_i \vee n_i$. As shown next, the soft clause $p_i \vee n_i$ will let MaxSAT simulate a resolution step.

Proof. (Sketch) Let $\mathcal{R}$ be a tree-like refutation of Γ over the variables $x_1, \ldots, x_n$. Let k_i be the number of times that x_i is resolved on in $\mathcal{R}$. We form Γ^{mdr} by adding the soft clauses p_i and n_i with multiplicity k_i, and the hard clauses $\overline{p_i} \vee \overline{n_i}$. (This is permitted as the values k_i correspond to the weights w_i of a weighted DRMaxSAT refutation.) Set $K = \sum_i k_i$. By the above observation, from these clauses there is a MaxSAT derivation of K many instances of $\perp$, plus the clauses $p_i \vee n_i$ with multiplicity k_i.

We modify the derivation $\mathcal{R}$. For each clause A in Γ, let A^{dr} be the result of replacing members x_i with $\overline{n_i}$, and members $\overline{x_i}$ with $\overline{p_i}$. An inference in Γ resolving $x_i \vee A$ and $\overline{x_i} \vee B$ to obtain $A \vee B$ becomes

$$\frac{\overline{n_i} \vee A^{\mathrm{dr}} \qquad \overline{p_i} \vee B^{\mathrm{dr}}}{A^{\mathrm{dr}} \vee B^{\mathrm{dr}}}$$

To make this a valid MaxSAT inference, first resolve $\overline{n_i} \vee A^{\mathrm{dr}}$ against an available soft clause $p_i \vee n_i$ to obtain the soft clause $p_i \vee A$ plus some additional clauses. A further MaxSAT inference resolves this against $\overline{p_i} \vee B^{\mathrm{dr}}$ to obtain $A^{\mathrm{dr}} \vee B^{\mathrm{dr}}$ plus some additional clauses. Continuing this process yields a valid MaxSAT refutation of $\perp^{\mathrm{dr}}$, i.e. of $\perp$. This gives a total of $K + 1$ clauses $\perp$ as desired. $\square$

Note the proof works as long as k_i is greater than or equal to the number of times x_i is resolved on. For applications, this means it is only needed to have an upper bound on the number of resolutions on x_i; for instance, taking k_i equal to the total number of inferences in $\mathcal{R}$ certainly works.

General resolution

Theorem 2. *Weighted dual rail MaxSAT simulates general resolution.*

Proof. (Sketch) Let $\mathcal{R}$ be a resolution refutation of Γ containing clauses $C_1, \ldots, C_m$. Each C_i is either an initial clause from Γ or is derived from two clauses C_{j_1} and C_{j_2}, where $j_1 < j_2 < i$. We define a directed graph $G = ([m], E)$ encoding the dependencies in the derivation. The set of vertices of G is $\{1, \ldots, m\}$ corresponding to the m clauses of $\mathcal{R}$. The edges are based on inference rules; E is the set of directed edges (j, i) such that C_j is a hypothesis of the resolution inference introducing C_i. Thus, the vertex m (corresponding to C_m) is a sink of G. The sources in G correspond to initial clauses in Γ. All other vertices in G have in-degree two. Since $\mathcal{R}$ is not assumed to be tree-like, the out-degrees can be greater than one.

We must assign to each clause $C_i \in \mathcal{R}$ a weight $w_i \in \mathbb{N}$. These weights give the weights k_i needed for the soft clauses n_i and p_i when we construct a weighted dual rail MaxSAT refutation of Γ. The last (m-th) clause is the final $\perp$ derived for the MaxSAT refutation: this clause has weight one, $w_m = 1$. For all $j < m$, define

$$w_j \;=\; \textstyle\sum_{(j,i) \in E} w_i.$$

This is the same as defining w_j to be the sum of the weights of the clauses which are inferred directly from C_j.

Recall the Fibonacci numbers $F_1 = F_2 = 1$ and $F_i = F_{i-1} + F_{i-2}$ for $i > 2$. The next lemma depends only on the fact that $G = ([m], E)$ has indegree 0 or 2 at every node, and that the directed edges respect the usual ordering of $[m]$.

Lemma 3. *$w_i \le F_{m+1-i}$. Thus $w_i < \phi^m/\sqrt{5}$ where ϕ is the golden ratio.*

(The proof of the lemma is simple and is omitted here for space reasons.) To finish the proof of Theorem 2, we also need to fix weights k_i for the variables x_i. Set k_i to be equal (or be greater than) the sum of the weights w_j of clauses C_j which are introduced by a resolution on x_i. By Lemma 3, $k_i \le \sum_{i=1}^{m-2} \phi^i < \phi^{m-1}$, so $k_i = 2^m$ is always sufficient. Now Theorem 2 can be proved with the essentially the same construction as Theorem 1. A clause C_ℓ in $\mathcal{R}$ becomes the weighted clause $(C_\ell^{\mathrm{dr}}, w_\ell)$ in $\mathcal{R}^{\mathrm{wdr}}$. If C_ℓ is equal to $A \vee B$ and is derived from $x_i \vee A$ and $\overline{x_i} \vee B$, then in $\mathcal{R}^{\mathrm{wdr}}$, it becomes the (not-yet-valid) inference

$$\frac{(\overline{n_i} \vee A^{\mathrm{dr}}, w_\ell) \qquad (\overline{p_i} \vee B^{\mathrm{dr}}, w_\ell)}{(A^{\mathrm{dr}} \vee B^{\mathrm{dr}}, w_\ell)} \tag{3}$$

Note the weights of all three clauses are equal to w_ℓ. As described below, this is arranged for the two hypotheses by earlier extraction inferences. In $\mathcal{R}^{\mathrm{wdr}}$, the "inference" (3) is replaced by two MaxSAT resolution inferences which resolve against the weighted soft clauses (n_i, w_ℓ) and (p_i, w_ℓ) and the hard clauses $(\overline{n_i} \vee \overline{p_i}, \top)$.

$\mathcal{R}^{\mathrm{wdr}}$ needs inferences to fix up the weights. For $i \le n$, let $C_{\ell_1}, \ldots, C_{\ell_s}$ be the clauses which are inferred by resolving on x_i, so $k_i \ge \sum_j w_{\ell_j}$. At the start of $\mathcal{R}^{\mathrm{wdr}}$, from the initial soft clauses (n_i, k_i) and (p_i, k_i), extraction rules are used to derive all the clauses (n_i, w_{ℓ_j}) and (p_i, w_{ℓ_j}). Similarly, let $C_{\ell_1}, \ldots, C_{\ell_s}$ now denote clauses which are derived by resolution using C_ℓ, so $w_\ell = \sum_j w_{\ell_j}$. Extraction inferences are used to derive all of the clauses (C_ℓ, w_{ℓ_j}) from (C_ℓ, w_ℓ). These clauses are used as hypotheses of later inferences similarly as was done for (3). $\square$

AC^0-Frege+PHP simulates dual rail MaxSAT

This section proves that constant depth Frege augmented with the schematic pigeonhole principle PHP_n^{n+1} can polynomially simulate the dual rail MaxSAT proof system.

Theorem 4. *AC^0-Frege+PHP p-simulates the dual rail MaxSAT system. More precisely, there is a constant d_0 and a polynomial $p(s)$ so that the following holds. If Γ is a set of clauses and Γ^{dr} has a MaxSAT refutation of size s, then Γ has a depth d_0 Frege refutation from instances of the PHP_n^{n+1} of size $p(s)$.*

The value of d_0 depends on the exact definitions of the Frege system (e.g., with modus ponens, or with the sequent calculus, etc.) and of depth; however, d_0 is small, approximately equal to 3. In particular, the Frege proof uses instances of PHP which are obtained by substituting depth one formulas (either conjunctions or disjunctions of literals) for the variables $z_{i,j}$ of a pigeonhole formula.

It is open whether the theorem holds for the dual rail MaxSAT system generalized to allow arbitrary (binary-encoded) weights.

Corollary 5. *MaxSAT refutations of the dual rail encoded Parity Principle require exponential size 2^{n^ϵ} for some $\epsilon > 0$.*

Corollary 5 follows from Theorem 4 since (Beame and Pitassi 1996) and (Riis 1993), building on (Ajtai 1990), showed that AC^0-Frege+PHP refutations of Parity_n require size 2^{n^ϵ} for some $\epsilon > 0$.

Corollary 6. *The dual rail MaxSAT proof system does not polynomially simulate* CP *or even* CP^*.

Corollary 6 follows from Corollary 5 since it is easy to give polynomial size CP^* proofs of the parity principle.

Proof. We now prove Theorem 4. Let Γ be an unsatisfiable set of clauses in the variables $x_1, \ldots, x_N$. Its dual rail encoding Γ^{dr} uses the variables n_i and p_i for $i \in [N]$. By hypothesis, there is a MaxSAT derivation $\mathcal{D}$ of $N+1$ many empty clauses $\perp$ from Γ^{dr}. Our goal is to give a AC^0-Frege+PHP refutation of Γ; this refutation involves only the variables x_i. The intuition for forming the AC^0-Frege proof is that we assume that Γ is satisfied by $x_1, \ldots, x_N$, and use the refutation $\mathcal{D}$ to define a contradiction to the pigeonhole principle.

The MaxSAT refutation $\mathcal{D}$ has size s and contains $m < s$ inferences. The j-th inference of $\mathcal{D}$ has the form

$$\frac{l \vee A \qquad \bar{l} \vee B}{A \vee B \qquad l \vee A \vee \overline{B} \qquad \bar{l} \vee \overline{A} \vee B} \tag{4}$$

for l a literal. Here, $l \vee A \vee \overline{B}$ and $\bar{l} \vee \overline{A} \vee B$ denote a sets of zero or more clauses, which depend on orderings of the literals in A and in B.

Let $\mathcal{D}_j$ be the multiset of clauses which are available for use in $\mathcal{D}$ after the j-th inference. Thus, $\mathcal{D}_0$ is the same as Γ^{dr}. The multiset D_{j+1} is obtained from $\mathcal{D}_j$ by removing the hypotheses of the j-th inference (4) and adding its conclusions. Since $\mathcal{D}$ is a valid MaxSAT refutation, the final set $\mathcal{D}_m$ contains $N+1$ many empty clauses $\bot$. Two extra sets $\mathcal{D}_{-1}$ and $\mathcal{D}_{m+1}$ are defined by letting $\mathcal{D}_{-1}$ contain the N clauses $x_1,\ldots,x_N$ and letting $\mathcal{D}_{m+1}$ be the multiset containing $N+1$ copies of the empty clause $\bot$.

Let $\mathcal{D}_*$ denote the disjoint union of the multisets D_j for $-1 \le j \le m+1$. Members of the multiset $\mathcal{D}_*$ are denoted (C,j) indicating that C is a member of $\mathcal{D}_j$. If there are multiple occurrences of C in $\mathcal{D}_j$, then there are multiple occurrences of (C,j) in $\mathcal{D}_*$. We will assume that multiple occurrences are correctly tracked with each "C" labelled as to which occurrence it is, but suppress this in the notation.

Let S be the cardinality of $\mathcal{D}_*$, so $S = s^{O(1)}$. Define

$$T = \bigcup_{0\le i\le m+1} \mathcal{D}_i \qquad \text{and} \qquad U = \bigcup_{-1\le i\le m} \mathcal{D}_i.$$

We have $|T| = S-N$ and $|U| = S-N-1$, so $|T| = |U|+1$. We wish to define a total and injective function $f : T \to U$, based on the assumption that $x_1,\ldots,x_N$ specify a satisfying assignment for Γ: this will contradict the pigeonhole principle. For this, it is necessary to define formulas $P_{\alpha,\beta}$ for each $\alpha = (C,j) \in T$ and each $\beta = (C',j') \in U$ which define the condition that $f(\alpha) = \beta$. These formulas $P_{\alpha,\beta}$ will involve the variables $x_1,\ldots,x_N$.

If $(C,j) \in U$, then C is a clause (possibly empty) involving only the variables n_i and p_i. We wish to identify p_i and n_i with x_i and $\overline{x_i}$ to evaluate the truth of C. Accordingly, define $X(C)$ to the the formula obtained by replacing the literals p_i and $\overline{n_i}$ with x_i, and the literals $\overline{p_i}$ and n_i with $\overline{x_i}$. If C contains both p_i and n_i (or both $\overline{p_i}$ and $\overline{n_1}$) for some i, then $X(C)$ becomes a tautologous clause and can be treated as the constant $\top$. Each C is a fixed clause in $\mathcal{D}$, thus each $X(C)$ is also a fixed clause.

We next give the definition of the function f and define the formulas $P_{\alpha,\beta}$. Let α be (C,j) and β be (C',j'). The intuition is that if C is true, then $f(\alpha) = \alpha$; and if C is false then $f(\alpha) = \beta$ exactly when $j' = j-1$ and C' is the false formula in $\mathcal{D}_j$ which corresponds to C under the application of the j-th inference of $\mathcal{D}$. More formally:

1. Suppose $j = m+1$, so C is an empty clause $\bot$ in the "extra" set $\mathcal{D}_{m+1}$. We arbitrarily order the members $\bot$ of $\mathcal{D}_{m+1}$ and $\mathcal{D}_m$. Suppose C is the ℓ-th member of $\mathcal{D}_{m+1}$. We wish to assign $f(\alpha)$ to equal the ℓ-th $\bot$ in $\mathcal{D}_m$. Accordingly, $P_{\alpha,\beta}$ is the constant $\top$ (true) if and only if $j' = j-1 = m$ and C' is the ℓ-th $\bot$ in $\mathcal{D}_m$. Otherwise, $P_{\alpha,\beta}$ is the constant $\bot$ (false).
2. Suppose $j \ge 1$, and that C, as a member of $\mathcal{D}_j$, is not a clause in the conclusion of the j-th inference (4). The idea is that if C is true, then $f(\alpha) = \alpha$, and if C is false, then $f(\alpha) = \beta$ provided $j' = j-1$ and C' is the same formula as C, namely the occurrence of the clause in $\mathcal{D}_{j-1}$ which corresponds to C. More formally, $P_{\alpha,\alpha}$ is the formula $X(C)$. And, if $j' = j-1$ and $C' \in \mathcal{D}_{j-1}$ is the corresponding occurrence of the clause C in $\mathcal{D}_{j-1}$, then $P_{\alpha,\beta}$ is the formula $\neg X(C)$. In all other cases, $P_{\alpha,\beta}$ is $\bot$.
3. Suppose $j \ge 1$, and C is one of the conclusions of the j-th inference (4). The idea is that if C is true, then $f(\alpha) = \alpha$, and if C is false, then $f(\alpha) = \beta$ provided $j' = j-1$ and C' is the false hypothesis of (4). More formally, $P_{\alpha,\alpha}$ is the formula $X(C)$. And, if $j' = j-1$ and $C' \in \mathcal{D}_{j-1}$ is one of the hypotheses of (4), then $P_{\alpha,\beta}$ is the formula $\neg X(C) \wedge \neg X(C')$, which is a conjunction of literals. (This can make $P_{\alpha,\beta}$ false by virtue of containing both ℓ and $\overline{\ell}$.) In all other cases, $P_{\alpha,\beta}$ is $\bot$.
4. Suppose $j = 0$ and C is a hard clause of Γ^{dr} in $\mathcal{D}_0$. Assuming Γ is satisfied by $x_1,\ldots,x_N$, C is true; the idea is that $f(\alpha) = \alpha$. Accordingly, $P_{\alpha,\alpha}$ is the clause $X(C)$. For all other β, $P_{\alpha,\beta}$ is $\bot$.
5. Finally suppose $j = 0$ and C is a soft unit clause in Γ^{dr}, i.e. either p_i or n_i. The intuition is again that $f(\alpha) = \alpha$ if C is true. Otherwise $f(\alpha) = (x_i,-1)$. Formally, $P_{\alpha,\alpha}$ is $X(C)$. And, for $\beta = (x_i,-1)$, $P_{\alpha,\beta}$ is $\neg X(C)$. For all other β, $P_{\alpha,\beta}$ is $\bot$.

The formulas $P_{\alpha,\beta}$ are linear size and depth one, either conjunctions or disjunctions of literals. We must argue there are constant depth Frege proofs of the injectivity conditions

$$\neg P_{\alpha,\beta} \vee \neg P_{\alpha',\beta} \qquad \text{for all } \alpha \ne \alpha' \in T \text{ and all } \beta$$

and of the totality conditions

$$\bigvee_{\beta\in U} P_{\alpha,\beta} \qquad \text{for all } \alpha \in T.$$

The injectivity conditions are easy to check since so many $P_{\alpha,\beta}$'s are the constant $\bot$. First, suppose that $\alpha = (C,j)$ and $\alpha' = (C',j)$ where C and C' are two of the conclusions of the j-th inference (4). By inspection, C and C' contain a clashing literal; thus they cannot both be false. It follows that at least one of $P_{\alpha,\beta}$ or $P_{\alpha',\beta}$ is false. A similar, even simpler, argument works when $\alpha = (p_i,0)$ and $\alpha' = (n_i,0)$. The injectivity conditions for all other α,α',β are trivial.

There are only a couple non-trivial cases to check for the provability of the totality conditions. The first case is when $\alpha = (C,j)$ is the conclusion of the j-th inference (4). For this, we must argue that if $X(C)$ is false, then (4) has a hypothesis C' that has $X(C')$ false. This is completely trivial to prove with a constant depth Frege proof, since either (a) one of the hypotheses is a subclause C' of C so $X(C')$ is a subclause of $X(C)$ and thus $X(C')$ is false, or (b) C is $A \vee B$ in (4) and since $X(\ell)$ is either false or true and C' can be taken to be the first or second hypothesis (respectively). The second non-trivial case to check for totality is the case where $\alpha = (C,0)$ with C one of the hard clauses in Γ^{dr}. In this case, $P_{\alpha,\alpha}$ holds only if $X(C)$ is true. However, $X(C)$ is a member of Γ, and hence $X(C)$ holds under the assumption that $x_1,\ldots,x_N$ satisfy the clauses of Γ.

The above obtained a contradiction to the pigeonhole principle from the assumption that the clauses of Γ are true. The argument can be formalized in constant depth Frege; hence AC^0-Frege+PHP refutes Γ. By construction, the AC^0-Frege+PHP refutation is polynomial size in s. $\square$

Upper bound for the doubled PHP

This section discusses the "doubled" pigeonhole principle which states that if $2m+1$ pigeons are mapped to m holes then some hole contains at least three pigeons (Biere 2013b).

Theorem 7. *There are polynomial size MaxSAT refutations of the dual rail encoding of the* $2\mathrm{PHP}_m^{2m+1}$ *clauses.*

Proof sketch. The MaxSAT refutation first derives $2m{+}1$ clauses $\perp$, one for each pigeon $i \in [2m{+}1]$, by resolving the hard clause $\bigvee_{j=1}^{m} \overline{n_{i,j}}$ against the soft unit clauses $n_{i,j}$ to obtain the clause $\perp$. These inferences derive other clauses as well, but they are not needed for the refutation, so we just ignore them. The remainder of the MaxSAT refutation is more complex and derives $2m-1$ empty clauses for each hole $j \in [m]$. This gives a total of $(2m{-}1)m$ additional $\perp$'s and, since $2m{+}1{+}(2m{-}1)m$ is equal to $(2m{+}1)m{+}1$, suffices to complete the MaxSAT refutation.

Fix a hole j. We inductively construct MaxSAT derivations of $2m-1$ empty clauses from the clauses involving literals $p_{i,j}$. The construction is to be repeated (independently) for each $j \in [m]$. The general idea is to derive $I-2$ many $\perp$'s from the first I pigeons, namely using only the literals $p_{i,j}$ for $i \leq I$.

The construction proceeds in stages, one for each value $I = 3, 4, \ldots, 2m{+}1$. The proof is involved, and we omit it from this version for lack of space. $\square$

Experiments. Two sets of "doubled" pigeonhole formulas were considered encoding $AtMost2$ constraints by (1) *triplewise* encoding as studied earlier in the paper and (2) sequential counters (Sinz 2005), i.e. with the use of auxiliary variables (Tseitin 1968). The latter set contains PHP_m^{2m+1} formulas for $m \in \{5, \ldots, 100\}$ while the largest triplewise-encoded formula is constructed for $m = 25$ (due to the formula's growth as m^3 if triplewise-encoded). Our evaluation targets SAT and MaxSAT solvers. We tested two CDCL SAT solvers: Glucose 3 (Audemard, Lagniez, and Simon 2013) and lingeling[2] (Biere 2013a; 2014). Also, the MaxSAT solvers used are: MaxHS (Davies and Bacchus 2011; 2013a; 2013b), LMHS (Saikko, Berg, and Järvisalo 2016), Eva500a (Narodytska and Bacchus 2014), OpenWBO16 (Martins, Manquinho, and Lynce 2014), and MSCG (Morgado, Ignatiev, and Marques-Silva 2015).

Figure 1 depicts the performance of the considered competitors. As expected, SAT solvers can only deal with PHP_m^{2m+1} for $m \leq 7$ given 1800s timeout, while MaxSAT solvers do not perform *much* better being able to deal with $m \leq 15$. This can be attributed to the clauses of the dual rail encoding, more precisely to clauses $(\overline{p_i} \vee \overline{n_i}, \top)$, $(p_i, 1)$ and $(n_i, 1)$ introduced for every variable x_i of the original CNF formula. Obviously, these clauses comprise unsatisfiable cores of the dual rail MaxSAT formula and these cores are known to potentially confuse a MaxSAT solver and, thus, be harmful, as observed in earlier work (Ignatiev, Morgado, and Marques-Silva 2017). Indeed, our results confirm this conjecture as the performance of all MaxSAT solvers gets tremendously increased when clauses $(\overline{p_i} \vee \overline{n_i}, \top)$ are discarded[3] (the corresponding configurations of MaxSAT solvers in Figure 1 are marked with additional symbol '$'$').

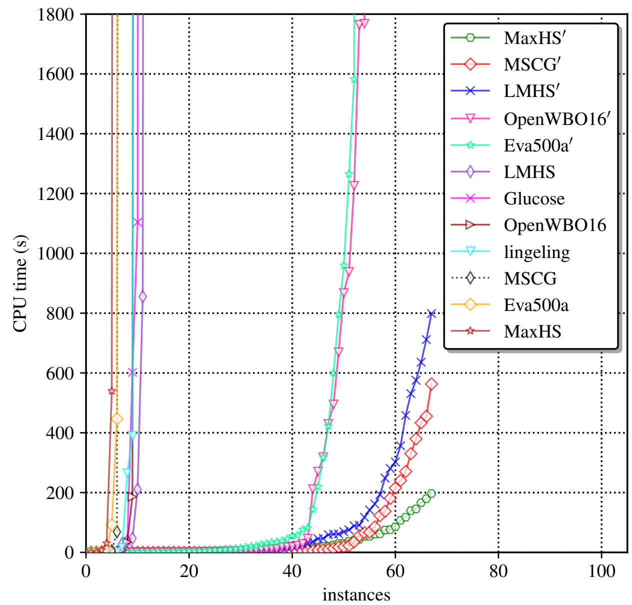

Figure 1: Performance of SAT and MaxSAT solvers on "doubled" pigeonhole formulas.

In particular, MaxHS, LMHS, as well as MSCG can solve all the considered instances (for m up to 100) with the "harmful" clauses being discarded while Eva500a and OpenWBO16 are a few instances behind. This suggests that removing these clauses enables MaxSAT solvers to produce a short proof when dealing with $2\mathrm{PHP}_m^{2m+1}$.

Conclusions

This paper investigates the relative efficiency of the DRMaxSAT proof system (Ignatiev, Morgado, and Marques-Silva 2017). The paper shows that DRMaxSAT p-simulates general resolution. Given as earlier result of polynomial time refutations of PHP formulas (Ignatiev, Morgado, and Marques-Silva 2017), we conclude that DRMaxSAT is a stronger proof system than either general resolution or conflict-driven clause learning. The paper also compares DRMaxSAT with AC^0-Frege+PHP, and proves that AC^0-Frege+PHP p-simulates DRMaxSAT. Moreover, the paper also proves that DRMaxSAT does not p-simulate AC^0-Frege+PHP or the cutting plane based proof systems CP and CP*. Finally, the paper investigates the formulas encoding the doubled PHP principle, and derives polynomial size refutations with the DRMaxSAT proof system.

The results in this paper motivate a number of research lines. The first is to understand whether CP p-simulates DRMaxSAT. Another research line is to investigate whether the weighted version is stronger than plain DRMaxSAT. One line of research is to extend the results in this paper to the case of core-guided algorithms (Ignatiev, Morgado, and Marques-Silva 2017).

[2]Cardinality reasoning (Biere 2013a), which helps lingeling deal with PHP formulas, was disabled.

[3]Note that the short proof for $2\mathrm{PHP}_m^{2m+1}$ provided in the previous section does not use clauses $(\overline{p_i} \vee \overline{n_i}, \top)$.

References

Ajtai, M. 1990. Parity and the pigeonhole principle. In *Feasible Mathematics*, 1–24. Birkhäuser.

Audemard, G.; Katsirelos, G.; and Simon, L. 2010. A restriction of extended resolution for clause learning SAT solvers. In *AAAI*.

Audemard, G.; Lagniez, J.; and Simon, L. 2013. Improving Glucose for incremental SAT solving with assumptions: Application to MUS extraction. In *SAT*, 309–317.

Beame, P., and Pitassi, T. 1996. An exponential separation between the parity principle and the pigeonhole principle. *Annals of Pure and Applied Logic* 80(3):195–228.

Beame, P., and Pitassi, T. 2001. Propositional proof complexity: Past, present, and future. In *Current Trends in Theoretical Computer Science*. 42–70.

Beame, P.; Impagliazzo, R.; Krajíček, J.; Pitassi, T.; and Pudlák, P. 1996. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society* 73(3):1–26.

Beame, P.; Kautz, H. A.; and Sabharwal, A. 2004. Towards understanding and harnessing the potential of clause learning. *J. Artif. Intell. Res.* 22:319–351.

Biere, A. 2013a. Lingeling, plingeling and treengeling entering the SAT competition 2013. In *Proceedings of SAT Competition 2013*, 51–52.

Biere, A. 2013b. Two pigeons per hole problem. In *Proceedings of SAT Competition 2013*, 103–103.

Biere, A. 2014. Lingeling essentials, A tutorial on design and implementation aspects of the SAT solver lingeling. In *Pragmatics of SAT workshop*, 88.

Bonet, M. L.; Levy, J.; and Manyà, F. 2007. Resolution for Max-SAT. *Artif. Intell.* 171(8-9):606–618.

Bonet, M. L.; Pitassi, T.; and Raz, R. 2000. On interpolation and automatization for frege systems. *SIAM J. Comput.* 29(6):1939–1967.

Bryant, R. E.; Beatty, D. L.; Brace, K. S.; Cho, K.; and Sheffler, T. J. 1987. COSMOS: A compiled simulator for MOS circuits. In *DAC*, 9–16.

Buss, S. R., and Clote, P. 1996. Cutting planes, connectivity and threshold logic. *Archive for Mathematical Logic* 35:33–62.

Buss, S. R. 2012. Towards NP-P via proof complexity and proof search. *Annals of Pure and Applied Logic* 163(9):1163–1182.

Cook, S. A., and Reckhow, R. A. 1979. The relative efficiency of propositional proof systems. *J. Symb. Log.* 44(1):36–50.

Davies, J., and Bacchus, F. 2011. Solving MAXSAT by solving a sequence of simpler SAT instances. In *CP*, 225–239.

Davies, J., and Bacchus, F. 2013a. Exploiting the power of mip solvers in maxsat. In *SAT*, 166–181.

Davies, J., and Bacchus, F. 2013b. Postponing optimization to speed up MAXSAT solving. In *CP*, 247–262.

Haken, A. 1985. The intractability of resolution. *Theor. Comput. Sci.* 39:297–308.

Huang, J. 2010. Extended clause learning. *Artif. Intell.* 174(15):1277–1284.

Ignatiev, A.; Morgado, A.; and Marques-Silva, J. 2017. On tackling the limits of resolution in SAT solving. In *SAT*, 164–183.

Larrosa, J., and Heras, F. 2005. Resolution in Max-SAT and its relation to local consistency in weighted CSPs. In *IJCAI*, 193–198.

Martins, R.; Manquinho, V. M.; and Lynce, I. 2014. Open-WBO: A modular MaxSAT solver,. In *SAT*, 438–445.

Morgado, A.; Heras, F.; Liffiton, M. H.; Planes, J.; and Marques-Silva, J. 2013. Iterative and core-guided MaxSAT solving: A survey and assessment. *Constraints* 18(4):478–534.

Morgado, A.; Ignatiev, A.; and Marques-Silva, J. 2015. MSCG: Robust core-guided MaxSAT solving. *JSAT* 9:129–134.

Narodytska, N., and Bacchus, F. 2014. Maximum satisfiability using core-guided MaxSAT resolution. In *AAAI*, 2717–2723.

Nordström, J. 2015. On the interplay between proof complexity and SAT solving. *SIGLOG News* 2(3):19–44.

Palopoli, L.; Pirri, F.; and Pizzuti, C. 1999. Algorithms for selective enumeration of prime implicants. *Artif. Intell.* 111(1-2):41–72.

Pudlák, P. 1997. Lower bounds for resolution and cutting planes proofs and monotone computations. *Journal of Symbolic Logic* 62:981–998.

Pudlák, P. 1999. On the complexity of propositional calculus, Sets and proofs. In *Logic Colloquium '97*, 197–218. Cambridge University Press.

Riis, S. 1993. *Independence in Bounded Arithmetic*. Ph.D. Dissertation, Oxford University.

Sabharwal, A. 2005. *Algorithmic applications of propositional proof complexity*. Ph.D. Dissertation, University of Washington.

Saikko, P.; Berg, J.; and Järvisalo, M. 2016. LMHS: A SAT-IP hybrid MaxSAT solver. In *SAT*, 539–546.

Sinz, C. 2005. Towards an optimal CNF encoding of Boolean cardinality constraints. In *CP*, 827–831.

Tseitin, G. S. 1968. On the complexity of derivation in propositional calculus. *Studies in constructive mathematics and mathematical logic* 2(115-125):10–13.