# Algorithms for Satisfiability beyond Resolution.

**Maria Luisa Bonet**

UPC, Barcelona, Spain

Oaxaca, August, 2018

**Co-Authors:** Sam Buss, Alexey Ignatiev, Joao Marques-Silva, Antonio Morgado.

# Motivation.

- Satisfiability (SAT) is the problem of determining if there is an interpretation that satisfies a given boolean formula in conjunctive normal form.
- SAT is an NP-Complete problem, therefore we don't expect to have polynomial algorithms for it.
- SAT is very important because many other problems can be encoded as satisfiability.
- Even though SAT is NP-Complete, we can solve efficiently many hard real life problems.
- Even though an unsatisfiable formula may have a short refutation, finding it might be hard.

# Motivation.

- Conflict Driven Clause Learning (CDCL) is the main technique for solving SAT
- When formulas are unsatisfiable, CDCL is equivalent to Resolution.
- Some basic problems, like pigeon-hole principle, cannot have short Resolution Refutations.
- Research on stronger proof systems, like Extended Resolution or Cutting Planes, for refuting some formulas efficiently, has failed.
- Ideas for improvements of SAT solving procedures for some hard crafted instances.

# Dual-Rail Approach

- Encode the principle as a partial MaxSAT problem using the dual-rail encoding;
- then use MaxSAT.
- Advantages:
  Polynomial size encodings.
  We can use MaxSAT algorithms, like core-guided or minimum hitting set.
  Method efficiently solves some hard problems for Resolution, like pigeon-hole.
- Topic of present work: what is the real power of dual-rail MaxSAT technique compared with other proof systems?

# MaxSAT and Partial MaxSAT

- Need to give weights to clauses, weight indicating the "cost" of falsifying the clause.
- Clauses 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 $\top$.

### Definition

So *Partial MaxSAT* is the problem of finding an assignment that satisfies all the hard clauses and minimizes the number of falsified soft clauses.

# Dual-Rail MaxSAT [Ignatiev-Morgado-MarquesSilva].

- $\Gamma$ a set of hard clauses over the variables $\{x_1, \ldots, x_N\}$.
- The dual-rail encoding $\Gamma^{dr}$ of $\Gamma$, uses $2N$ variables $n_1, \ldots, n_N$ and $p_1, \ldots, p_N$ in place of variables $x_i$.
- $p_i$ is true if $x_i$ is true, and that $n_i$ is true if $x_i$ is false.
- $C^{dr}$ of a clause $C$:
  - replace (unnegated) $x_i$ with $\overline{n_i}$, and (negated) $\overline{x_i}$ with $\overline{p_i}$.
  - **Example:** if $C$ is $\{x_1, \overline{x_3}, x_4\}$, then $C^{dr}$ is $\{\overline{n_1}, \overline{p_3}, \overline{n_4}\}$.
  - Every literal in $C^{dr}$ is negated.
- dual rail encoding $\Gamma^{\mathrm{dr}}$ of $\Gamma$ contains:
  1. The hard clause $C^{\mathrm{dr}}$ for each $C \in \Gamma$.
  2. The hard clauses $\overline{p_i} \vee \overline{n_i}$ for $1 \leq i \leq N$.
  3. The soft clauses $p_i$ and $n_i$ for $1 \leq i \leq N$.

# Dual-Rail MaxSAT approach

### Lemma (Ignatiev-Morgado-Marques-Silva)

*$\Gamma$ is satisfiable if and only if there is an assignment that satisfies all the hard clauses of $\Gamma^{\mathrm{dr}}$, and $N$ of the soft ones.*

### Corollary

*$\Gamma$ is unsatisfiable iff every assignment that satisfies all hard clauses of $\Gamma^{\mathrm{dr}}$, must falsify at least $N+1$ soft clauses.*

In the context of proof systems:
$\Gamma$ is unsatisfiable, if using a proof system for Partical MaxSAT, we can obtain at least $N+1$ empty clauses ($\perp$).

# MaxSAT Inference Rule. [Larrosa-Heras, Bonet-Levy-Manya]

(Partial) MaxSAT rule, **replaces** two clauses by a different set of clauses.
A clause may be used only once as a hypothesis of an inference.

$$\frac{\begin{array}{c}(x \vee A, 1)\\(\overline{x} \vee B, \top)\end{array}}{\begin{array}{c}(A \vee B, 1)\\(x \vee A \vee \overline{B}, 1)\\(\overline{x} \vee B, \top)\end{array}} \qquad \frac{\begin{array}{c}(x \vee A, 1)\\(\overline{x} \vee B, 1)\end{array}}{\begin{array}{c}(A \vee B, 1)\\(x \vee A \vee \overline{B}, 1)\\(\overline{x} \vee \overline{A} \vee B, 1)\end{array}} \qquad \frac{\begin{array}{c}(x \vee A, \top)\\(\overline{x} \vee B, \top)\end{array}}{\begin{array}{c}(A \vee B, \top)\\(x \vee A, \top)\\(\overline{x} \vee B, \top)\end{array}}$$

$x \vee A \vee \overline{B}$, where $A = a_1 \vee \cdots \vee a_s$, $B = b_1 \vee \cdots \vee b_t$ and $t > 0$, is

$$\begin{array}{l}x \vee a_1 \vee \cdots \vee a_s \vee \overline{b_1}\\x \vee a_1 \vee \cdots \vee a_s \vee b_1 \vee \overline{b_2}\\\cdots\\x \vee a_1 \vee \cdots \vee a_s \vee b_1 \vee \cdots \vee b_{t-1} \vee \overline{b_t}\end{array} \tag{1}$$

# Example

Consider the unsatisfiable set of clauses: $\overline{x_1} \vee x_2$, $x_1$ and $\overline{x_2}$.

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 dual-rail MaxSAT refutation must derive a multiset containing three copies of the empty clause $\perp$.

$$\frac{\begin{array}{c}(\overline{n_1}, \top) \\ (n_1, 1)\end{array}}{\begin{array}{c}(\perp, 1) \\ (\overline{n_1}, \top)\end{array}} \qquad \frac{\begin{array}{c}(\overline{p_2}, \top) \\ (p_2, 1)\end{array}}{\begin{array}{c}(\perp, 1) \\ (\overline{p_2}, \top)\end{array}} \qquad \frac{\begin{array}{c}(p_1, 1) \\ (\overline{p_1} \vee \overline{n_2}, \top)\end{array}}{\begin{array}{c}(\overline{n_2}, 1) \\ (p_1 \vee n_2, 1) \\ (\overline{p_1} \vee \overline{n_2}, \top)\end{array}} \qquad \frac{\begin{array}{c}(\overline{n_2}, 1) \\ (n_2, 1)\end{array}}{(\perp, 1)}$$

# Core-guided Algorithm for MaxSAT

1. **Input**: $F = S \cup H$, soft clauses $S$ and hard clauses $H$
2. $(R, F_W, \lambda) \to (\emptyset, S \cup H, 0)$
3. **while true do**
4. $\quad (st, C, A) \to SAT(F_W)$
5. $\quad$ if $st$ then return $\lambda, A$
6. $\quad \lambda \to \lambda + 1$
7. $\quad$ **for** $c \in C \cap S$ **do**
8. $\quad\quad R \to R \cup \{r\}$ // r is a fresh variable
9. $\quad\quad S \to S \setminus \{c\}$
10. $\quad\quad H \to H \cup \{c \cup \{r\}\}$
11. $\quad F_W \to S \cup H \cup CNF(\sum_{r \in R} r \leq \lambda)$

# Relevant Proof Systems

A **Frege** system is a textbook-style proof system, usually defined to have modus ponens as its only rule of inference.

An $AC^0$-**Frege** proof is a Frege proof with a constant upper bound on the depth of formulas appearing in the Frege proof.

$AC^0$-**Frege+PHP** is constant depth Frege augmented with the schematic pigeonhole principle.

The **Cutting Planes** system is a pseudo-Boolean propositional proof system, with variables taking on 0/1 values.
The lines of a cutting planes proof are inequalities of the form

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

where the $a_i$'s are integers.
Logical axioms are $x_i \geq 0$ and $-x_i \geq -1$;
rules are addition, multiplication by a integer, and a special division rule.

# The Pigeonhole principle

There is no $1 - 1$ function from $[n+1]$ to $[n]$.

Set of clauses:

$$\bigvee_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{x_{i,j}} \qquad \text{for } i \in [n{+}1]$$

$$\overline{\mathbf{x_{i,j}}} \vee \overline{\mathbf{x_{k,j}}} \qquad \text{for distinct } i, k \in [n{+}1].$$

[Cook-Reckhow] Polynomial size **extended Frege** proofs of $PHP_n^{n+1}$.

[Buss'87] Polynomial size **Frege** proofs of $PHP_n^{n+1}$.

[Haken'85] **Resolution** requires exponential size refutations of $PHP_n^{n+1}$.

Polynomial size **Cutting Planes** refutations of $PHP_n^{n+1}$.

# Translation of the PHP to the dual-rail Language.

The dual-rail encoding, $(PHP_n^{n+1})^{dr}$ of $PHP_n^{n+1}$.
Hard clauses:

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

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

Soft clauses are:
Unit clauses $\mathbf{n_{i,j}}$ and $\mathbf{p_{i,j}}$ for all $i{\in}[n{+}1]$ and $j{\in}[n]$.

[Ignatiev-Morgado-MarquesSilva] Polynonial sequence of Partial MaxSAT resolution steps to obtain $(n+1)n+1$ soft empty clauses $\bot$.

[Bonet-Levy-Manya] MaxSAT rule requires exponential number of steps to show one clause cannot be satisfied, when using usual encoding.

# Relationship of dual-rail MaxSAT and Resolution

**Theorem**

*The core-guided MaxSAT algorithm with the dual-rail encoding simulates Resolution.*

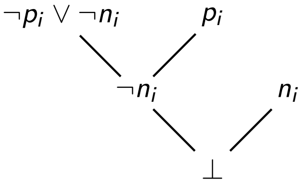
**Theorem**

*Multiple dual rail MaxSAT simulates tree-like Resolution.*

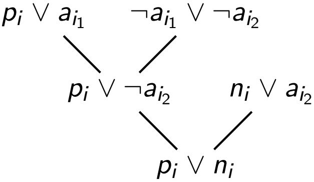
**Theorem**

*Weighted dual rail MaxSAT simulates general Resolution.*

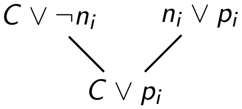
# Dual-rail Core-guided MaxSAT simulation of Resolution




Substitute $\{p_i, n_i\}$ soft, by $\{p_i \vee a_{i_1}, n_i \vee a_{i_2}, a_{i_1} + a_{i_2} \leq 1)\}$ hard, $a_{i_1}$ and $a_{i_2}$ new variables.




For every $i$, we have $p_i \vee n_i$.




Now we have all clauses with $p_i$ vars.
Follow resolution refutation.

# Dual-rail MaxSAT simulation of Resolution

$$\frac{(p_i, w_i) \quad (\overline{p_i} \vee \overline{n_i}, \top)}{(\overline{n_i}, w_i) \quad (p_i \vee n_i, w_i) \quad (\overline{p_i} \vee \overline{n_i}, \top)}$$

$$\frac{(\overline{n_i}, w_i) \quad (n_i, w_i)}{(\bot, w_i)}$$

$$\frac{(C \vee \overline{n_i}, \top) \quad (p_i \vee n_i, w_i)}{(C \vee p_i, w_i) \quad \text{other clauses}}$$

We used soft clauses $n_i$ and $p_i$, and obtained soft $\bot$ and $p_i \vee n_i$.
Soft clauses $n_i$ and $p_i$ will have considerable weight initially,
$p_i \vee n_i$ will have weight to eliminate $n_i$ variables,
weights will be used to account for several uses of a clause in the refutation.

# The Parity Principle.

Given a graph with an odd number of vertices, it is not posible to have every vertex with degree one.

The propositional version of the Parity Principle, for $m \geq 1$, uses $\binom{2m+1}{2}$ variables $x_{i,j}$, where $i \neq j$ and $x_{i,j}$ is identified with $x_{j,i}$.
Meaning of $x_{i,j}$: there is an edge between vertex $i$ and vertex $j$.

The **Parity Principle**, $\mathrm{Parity}^{2m+1}$,

| | |
|---|---|
| $\bigvee_{\mathbf{j \neq i}} \mathbf{x_{i,j}}$ | for $i \in [2m+1]$ |
| $\overline{\mathbf{x_{i,j}}} \vee \overline{\mathbf{x_{k,j}}}$ | for $i, j, k$ distinct members of $[2m+1]$. |

# Results using the Parity Principle

**Theorem**

*$AC^0$-Frege+PHP p-simulates the dual-rail MaxSAT system.*

**Theorem (Beame-Pitassi)**

*$AC^0$-Frege+PHP refutations of Parity require exponential size.*

**Corollary**

*MaxSAT refutations of the dual-rail encoded Parity Principle require exponential size.*

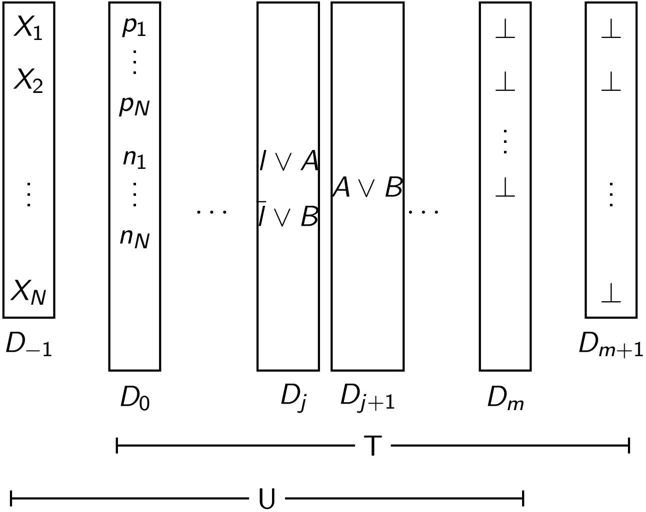
**Corollary**

*The dual rail MaxSAT proof system does not polynomially simulate CP.*

**Fact**

*Dual-rail minimum hitting set algorithm has short proofs of the Parity principle.*

# $AC^0$-Frege+PHP p-simulation the dual-rail MaxSAT



# The Double Pigeonhole Principle

if $2m+1$ pigeons are mapped to $m$ holes then some hole contains at least three pigeons.

Set of clauses of $2\mathrm{PHP}_m^{2m+1}$:

| | |
|---|---|
| $\bigvee_{\mathbf{j=1}}^{\mathbf{m}} \mathbf{x_{i,j}}$ | for $i \in [2m+1]$ |
| $\overline{\mathbf{x_{i,j}}} \vee \overline{\mathbf{x_{k,j}}} \vee \overline{\mathbf{x_{\ell,j}}}$ | for distinct $i, k, \ell \in [2m+1]$. |

# Translation of the Double PHP to dual-rail

The dual-rail encoding, $(2\mathrm{PHP}^{2m+1})^{dr}$, of $2\mathrm{PHP}^{2m+1}_{m}$.
Hard clauses:

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

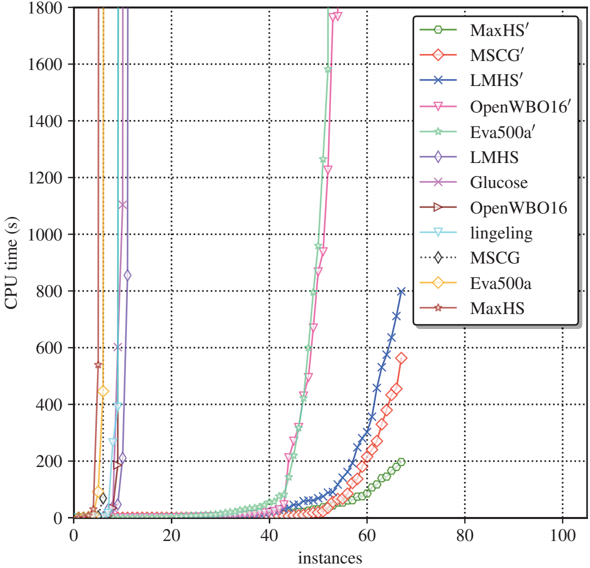
$\overline{\mathbf{p_{i,j}}} \vee \overline{\mathbf{p_{k,j}}} \vee \overline{\mathbf{p_{\ell,j}}}$ for $j \in [m]$ and distinct $i, k, \ell \in [2m+1]$.

Soft clauses are:
$\mathbf{n_{i,j}}$ and $\mathbf{p_{i,j}}$ for all $i \in [2m+1]$ and $j \in [m]$.

### Theorem

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

# Experimentation




Performance of SAT and MaxSAT solvers on $2\mathrm{PHP}_m^{2m+1}$.

# Summary of Results

- dual-rail MaxSAT is strictly stronger than Resolution.
- A stronger pigeon-hole principle also has polynomial-size proofs in dual-rail MaxSAT, but requires exponential size in Resolution.
- We did experimentation with such pigeon-hole principle to back up the theoretical results.
- dual-rail MaxSAT does not simulate Cutting Planes.