Extending CDCL-based Model Enumeration with Weights

Giuseppe Spallitta Moshe Y. Vardi

Rice University

Pragmatics of SAT

Motivation: from weighted counting to weighted models

Weighted model counting considers a Boolean formula F together with a weight assigned to each literal, $w(A_i)$ and $w(\neg A_i)$.
Given a satisfying assignment μ, its weight is typically defined as

$$W(\mu) = \prod_{\ell \in \mu} w(\ell).$$

Weighted model counting

WMC computes the total weight of all satisfying assignments:

$$\mathrm{WMC}(F) = \sum_{\mu \models F} W(\mu).$$

Motivation: from weighted counting to weighted models

Weighted model counting considers a Boolean formula F together with a weight assigned to each literal, $w(A_i)$ and $w(\neg A_i)$.
Given a satisfying assignment μ, its weight is typically defined as

$$W(\mu) = \prod_{\ell \in \mu} w(\ell).$$

Weighted model counting

WMC computes the total weight of all satisfying assignments:

$$\mathrm{WMC}(F) = \sum_{\mu \models F} W(\mu).$$

Satisfying assignments contribute very differently to this total:

$$W(\top, \bot, \top) \; > \; W(\bot, \top, \top) \; > \; W(\top, \top, \top).$$

Weights naturally induce a ranking among assignments!

Research question

If weights distinguish satisfying assignments, why should we only aggregate them into a single number?
Some applications would benefit from explicit assignments and their "quality".

Can model enumeration be made weight-aware natively?

Weighted Model Enumeration

Let F be a CNF formula over variables $\mathcal{V}$, and let

$$w : \mathcal{L} \to \mathbb{Q}_{>0}$$

assign a positive weight to each literal.

Weighted Model Enumeration (WME)

$$\mathcal{M}_w(F) = \{(\eta, w(\eta)) \mid \eta \models F\}, \qquad w(\eta) = \prod_{\ell \in \eta} w(\ell).$$

WME: models are no longer equivalent

Formula

$$F = (A_1 \vee A_2) \wedge A_3.$$

	A_1	A_2	A_3
$w(A_i)$	0.6	0.8	0.5
$w(\neg A_i)$	0.4	0.2	0.5

A_1	A_2	A_3	$w(\eta)$
$\top$	$\top$	$\top$	$0.6 \cdot 0.8 \cdot 0.5 = \mathbf{0.24}$
$\bot$	$\top$	$\top$	$0.4 \cdot 0.8 \cdot 0.5 = \mathbf{0.16}$
$\top$	$\bot$	$\top$	$0.6 \cdot 0.2 \cdot 0.5 = \mathbf{0.06}$

WME

Three assignments satisfy F, but they are not equally "good". We have a natural ranking among them.

WME: some useful variants

Formula

$$F = (A_1 \vee A_2) \wedge A_3.$$

	A_1	A_2	A_3
$w(A_i)$	0.6	0.8	0.5
$w(\neg A_i)$	0.4	0.2	0.5

A_1	A_2	A_3	$w(\eta)$
$\top$	$\top$	$\top$	$0.6 \cdot 0.8 \cdot 0.5 = \mathbf{0.24}$
$\bot$	$\top$	$\top$	$0.4 \cdot 0.8 \cdot 0.5 = \mathbf{0.16}$
$\top$	$\bot$	$\top$	$0.6 \cdot 0.2 \cdot 0.5 = \mathbf{0.06}$

Top-k WME

Which are the best k solutions?

Fixed number of solutions

WME: some useful variants

Formula

$$F = (A_1 \vee A_2) \wedge A_3.$$

	A_1	A_2	A_3
$w(A_i)$	0.6	0.8	0.5
$w(\neg A_i)$	0.4	0.2	0.5

A_1	A_2	A_3	$w(\eta)$
$\top$	$\top$	$\top$	$0.6 \cdot 0.8 \cdot 0.5 = \mathbf{0.24}$
$\bot$	$\top$	$\top$	$0.4 \cdot 0.8 \cdot 0.5 = \mathbf{0.16}$
$\top$	$\bot$	$\top$	$0.6 \cdot 0.2 \cdot 0.5 = \mathbf{0.06}$

Threshold WME

Which solutions have a score greater than a threshold, i.e., $\theta = 0.10$?

Variable number of solutions

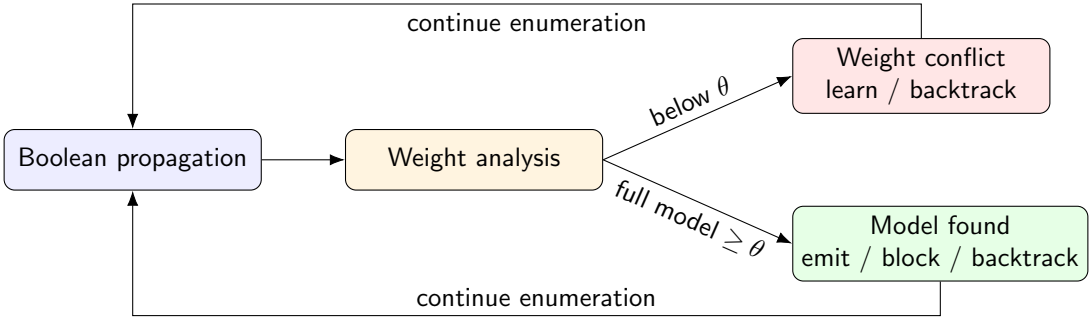
Weight-aware enumeration

A naive solution

- We could enumerate all satisfying assignments, compute their weights, and only afterward keep the models relevant to the query.
- But AllSAT is expensive and computationally demanding
- If I know I just need a narrow space of solutions, why waste computation?

We want **weight information to actively influence the enumeration** process, so that the solver can focus its effort on the models that matter.

Core idea: keep CDCL enumeration core, add weight reasoning



From ordinary AllSAT algorithms to weight-aware AllSAT

Detecting minimal changes on top of CDCL-based reasoning.

We identify four weight reasoning components that benefit enumeration.

Component 1: Weight pruning

Q1: should my partial assignment be expanded more or can it be *pruned early*?

For each variable we track the best choice among its two polarities:

$$best(A) = \max\{w(A), w(\neg A)\}.$$

For a trail μ, maintain both assignment weight and best optimistic bound:

$$w(\mu) = \prod_{\ell \in \mu} w(\ell), \qquad I_{\max}(\mu) = \prod_{A \in \mathcal{V} \setminus \mathrm{vars}(\mu)} best(A).$$

No completion of μ can reach θ if:

$$\underbrace{w(\mu)}_{\text{Current assignment weight}} \times \underbrace{I_{\max}(\mu)}_{\text{Best assignment extension}} < \underbrace{\theta}_{\text{Weight threshold}}$$

Component 2: Weight conflict analysis

Q2: If an assignment μ should not be extended further, *why*?

Extract a small set $S \subseteq \mu$ that already explains the conflict, and learn:

$$C_w = \bigvee_{\ell \in S} \neg \ell.$$

Toy example

Let $F = (A_1 \vee A_2 \vee \neg A_3)$, $\theta = 0.2$, and $\mu = \{\neg A_1, A_2, \neg A_3\}$. If $\neg A_3$ alone makes the optimistic bound fall below θ, the conflict set is:

$$S = \{\neg A_3\}.$$

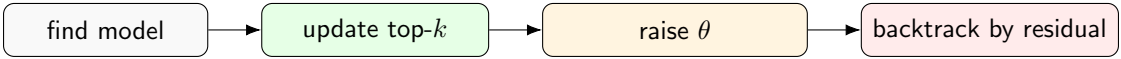
So the learned clause is:

$$C_w = A_3.$$

Component 3: Weight backtracking

Q3: If I want to get the best solutions, can I backtrack more aggressively?

In top-k enumeration, the active threshold θ is the current k-th best weight.



Residual-aware backtracking

After θ increases, keep popping the trail until

$$w(\mu) \cdot I_{\max}(\mu) > \theta.$$

Popping a literal replaces its actual weight with the best weight available for that variable:

$$best(A) \cdot w(\mu') \geq w(\ell) \cdot w(\mu').$$

The optimistic bound can therefore increase, opening the possibility of exploring better alternatives.

Weight backtracking example

Formula and weights

$$F = (A_1 \vee A_2) \wedge A_3.$$

	A_1	A_2	A_3
$w(A_i)$	0.6	0.8	0.7
$w(\neg A_i)$	0.4	0.2	0.3

Suppose the current top-1 model is

$$\eta = \{\neg A_1, A_2, A_3\}, \qquad w(\eta) = 0.4 \cdot 0.8 \cdot 0.7 = 0.224.$$

Therefore, $\theta = 0.224$.

With $\{\neg A_1, A_2\}$ fixed:

$$\max(A_3) = 0.7, \qquad 0.4 \cdot 0.8 \cdot 0.7 = 0.224.$$

No strict improvement is possible.

With only $\{\neg A_1\}$ fixed:

$$0.4 \cdot 0.8 \cdot 0.7 = 0.224.$$

The bound still cannot improve. But at the root,

$$0.6 \cdot 0.8 \cdot 0.7 = 0.336 > \theta$$

Component 4: Weight-aware branching

Q4: do all variables have the same importance in weighted environments?

Many CNF instances include auxiliary variables, especially after Tseitin or Plaisted–Greenbaum encodings. Some variables do not affect the score.

Partition variables

$$\mathcal{V} = W_r \cup W_i,$$

where W_r contains **weight-relevant** variables and W_i contains **weight-irrelevant** variables.

Weight Relevant

$$w(A) \neq w(\neg A)$$

Branch early: it changes the score and tightens bounds.

Weight Irrelevant

$$w(B) = w(\neg B)$$

Delay branching: it only completes the Boolean structure.

Two CDCL enumeration designs

These components still operate within an enumeration strategy, which may itself affect performance[1].

Non-chronological CDCL: `WME-NCB`

- explicit model blocking;
- restarts enabled;

Intuition: top-k, tight bounds, aggressive pruning.

Chronological CDCL[2]: `WME-CB`

- implicit blocking;
- sensitive to decision order.

Intuition: many models, dense feasible region.

Does one of these designs outperform the other?

[1] G. Spallitta, R. Sebastiani, and A. Biere (2024). *Disjoint partial enumeration without blocking clauses*. In: *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*

S. Möhle and A. Biere (2019). *Backing Backtracking*. In: *Theory and Applications of Satisfiability Testing – SAT 2019*

Experimental setup

Implemented solvers

Both `WME-CB` and `WME-NCB` variants have been implemented on top of `CaDiCaL`.

Tasks

- top-50 enumeration;
- threshold-based enumeration.

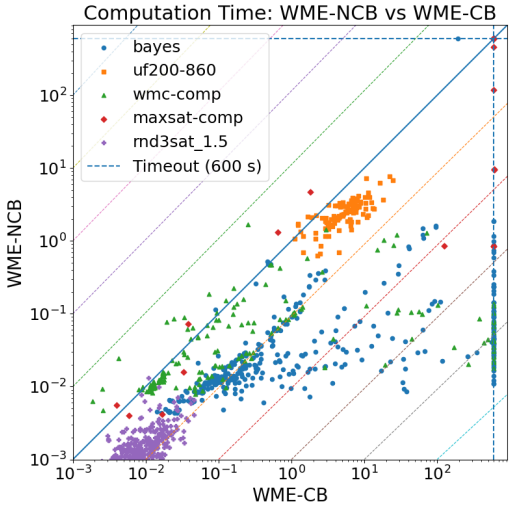
Baseline (from MaxSAT competition)

- top-k `RC2`;
- top-k `MaxHS`.

Benchmarks

- Random 3-SAT from SATLIB and from enumeration-inspired benchmarks, with uniformly distributed weights.
- Bayesian network instances
- WMC competition benchmarks
- MaxSAT top-k competition benchmark

Top-50 WME: WME_NCB vs WME_CB



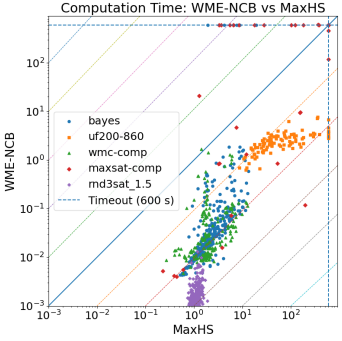
How to read the scatterplot

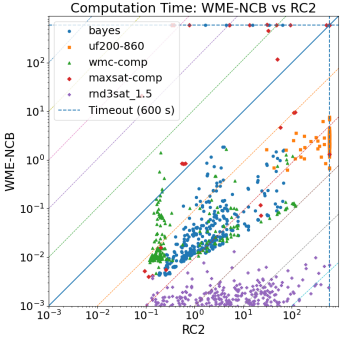
- The axes report the time needed to enumerate the top-50 models.
- Points below the diagonal indicate instances where WME_NCB is faster.

Takeaway

- In top-50 enumeration, the goal is to quickly focus on the highest-weight models.
- WME_NCB benefits restarts and learned weight conflicts, so explicit blocking has minor impact.
- This makes non-chronological search more suitable for focused top-k exploration.

Top-50 WME: comparison with MaxSAT solvers





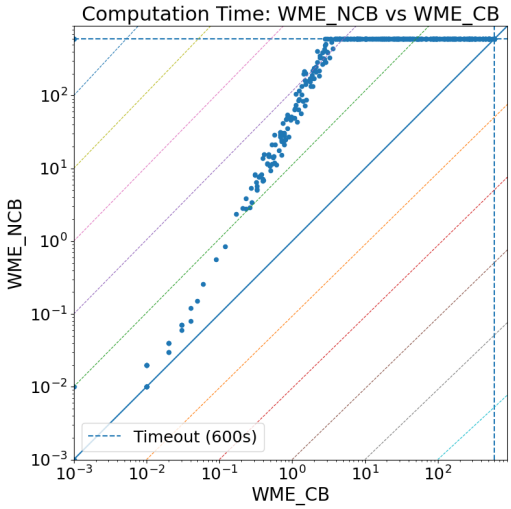
How to read the scatterplots

- The axes report the time needed to enumerate the top-50 models.
- Points below the diagonal indicate instances where `WME_NCB` is faster.

Takeaway

- `MaxHS` and `RC2` are strong optimization baselines, but WME-native search is often faster on literal-weighted instances.
- `WME_NCB` is fastest on most `bayes`, `wmc-comp`, `uf200-860`, and `rnd3sat_1.5` instances.
- The `maxsat-comp` family is more mixed: it comes from MaxSAT, and only unit-soft instances can be converted directly.

Threshold WME: WME_CB vs WME_NCB



Experimental setting

We consider the subset of `rnd3sat-1.5` instances with threshold

$$\theta = 0.5^n,$$

where n is the number of variables. This relatively permissive threshold typically produces a dense-output regime, with many models above θ.

Takeaway

- `WME_CB` avoids accumulating explicit blocking clauses.
- This makes chronological enumeration more robust in dense-output regimes.

Summary and future work

Summary

- Weighted enumeration under weight-based criteria can be performed efficiently.
- Weight pruning, weight conflicts, and weight backtracking are necessary for efficiency
- `WME-CB` and `WME-NCB` are complementary, depending on the task.

Future work

- Investigate more effective weight-based algorithms and data structures;
- Extend `WME-CB` and `WME-NCB` to SMT weighted enumeration;
- Investigate connections to Weighted Model Integration.

Take-home: WME is a native solver task, not a post-processing trick.

Thanks for listening!

Questions?

Extending CDCL-based Model Enumeration with Weights

Scan the QR code to access the paper.

Why a CDCL-based approach?

Knowledge compilation is a natural option for weighted reasoning: once a formula has been compiled, many counting and enumeration queries can be answered efficiently.

However, our goal is a general-purpose approach

We want Weighted Model Enumeration to extend beyond propositional SAT, including richer settings such as SMT.

For SMT formulas, knowledge compilation is less straightforward:

- Boolean paths may be propositionally consistent but inconsistent with the background theory;
- theory reasoning must therefore be incorporated into the compiled representation;
- obtaining compact or canonical representations becomes more difficult in the presence of theory constraints.

Why CDCL?

CDCL provides a mature search framework that has already been extended successfully to SMT through theory propagation, theory conflict analysis, and theory-lemma learning.

Backup: why not just repeat MaxSAT?

Repeated MaxSAT strategy

1. find an optimum model;
2. add a blocking clause;
3. re-optimize;
4. repeat until k models are found.

Problem

This treats ranked enumeration as repeated optimization. WME instead keeps one enumeration search with online bounds, learned clauses, and incremental pruning.

Backup: soundness intuition for pruning

For any completion $\eta \supseteq \mu$:

$$w(\eta) = \prod_{\ell \in \mu} w(\ell) \cdot \prod_{A \notin \mathrm{vars}(\mu)} w(\ell_A) \leq w(\mu) \cdot \prod_{A \notin \mathrm{vars}(\mu)} best(A).$$

Therefore:

$$w(\eta) \leq w(\mu) I_{\max}(\mu).$$

Conclusion

If the optimistic upper bound is below θ, then every real completion is below θ.

Weight pruning example

Formula

$$F = (A_1 \vee \phi) \wedge A_2,$$

where ϕ uses variables $B_1, \ldots, B_n$.

	A_1	A_2	B_i
$w(\ell)$	0.6	0.3	β_i
$w(\neg\ell)$	0.4	0.7	$1 - \beta_i$

Assume $\theta = 0.2$ and the trail is:

$$\mu = \{\neg A_1, A_2\}.$$

Then:

$$w(\mu) = 0.4 \cdot 0.3 = 0.12.$$

Let

$$\alpha = \prod_i best(B_i) \leq 1.$$

The optimistic upper bound is:

$$w(\mu) I_{\max}(\mu) = 0.12\alpha \leq 0.12 < 0.2.$$

Consequence

The solver prunes the branch without assigning the variables inside ϕ.