ADDMC: Weighted Model Counting with Algebraic Decision Diagrams

Jeffrey M. Dudek, **Vu H. N. Phan** (presenter), **Moshe Y. Vardi**
Computer Science Department, Rice University

AAAI 2020/02/11, New York City

Abstract

- Algebraic decision diagrams (ADDs): efficient data structure for pseudo-Boolean functions
- ADDMC: ADD-based framework for computing exact weighted model counts of Boolean formulas

Overview: Model Counting

Model counting (#SAT): computing number of satisfying assignments of Boolean formula

- Complexity: #P-complete [Valiant, 1979]
- Numerous applications, especially in probabilistic reasoning
  Examples:
  - Medical diagnosis [Shwe et al., 1991]
  - Reliability analysis of power transmission [Duenas-Osorio et al., 2017]

Progress

1. Boolean Model Counting Problem (#SAT)
2. Algebraic Decision Diagrams (ADDs)
3. Factored Representation and Dynamic Programming
4. Experimental Evaluation

Background: Boolean Logic

$\mathbb{B} = \{0, 1\}$ (**Boolean set**)

Variable $x \in \mathbb{B}$	**Negation** $\neg x$
0	1
1	0

x_1	x_2	**Disjunction** $x_1 \vee x_2$
0	0	0
0	1	1
1	0	1
1	1	1

x_1	x_2	**Conjunction** $x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1

Problem: Unweighted Model Counting

Formula: $F = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_3)$
Variable set: $V = \{x_1, x_2, x_3\}$
Assignment set: power set 2^V

Assignment $\alpha \in 2^V$			$F(\alpha) : 2^V \to \mathbb{B}$	Is α a **model** of F?
x_1	x_2	x_3		
0	0	0	0	Yes iff $F(\alpha) = 1$
0	0	1	0	
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	0	1	1	
1	1	0	1	
1	1	1	1	

Unweighted model count: $\#F = \sum_{\alpha \in 2^V} F(\alpha) = 5$

Problem: Weighted Model Counting

Weight function: $W : 2^V \to \mathbb{R}$ (**real-number set**)

Assignment $\alpha \in 2^V$			$W(\alpha)$
x_1	x_2	x_3	
0	0	0	2.0
0	0	1	3.0
0	1	0	2.0
0	1	1	3.0
1	0	0	3.0
1	0	1	3.0
1	1	0	4.0
1	1	1	4.0

Problem: Weighted Model Counting

Formula-weight product: $F \cdot W : 2^V \to \mathbb{R}$

Assignment $\alpha \in 2^V$			$F(\alpha)$	$W(\alpha)$	$(F \cdot W)(\alpha)$
x_1	x_2	x_3			
0	0	0	0	2.0	0.0
0	0	1	0	3.0	0.0
0	1	0	1	2.0	2.0
0	1	1	0	3.0	0.0
1	0	0	1	3.0	3.0
1	0	1	1	3.0	3.0
1	1	0	1	4.0	4.0
1	1	1	1	4.0	4.0

Weighted model count: $\#(F, W) = \sum_{\alpha \in 2^V} (F \cdot W)(\alpha) = 16.0$

Related Work: Weighted Model Counting

Existing approaches and tools:

1. Search: DPLL-based exploration of solution space
    - `Cachet` [Sang et al., 2004]
2. Knowledge compilation: efficient data structure – *exponential blowup in worst case*
    - `c2d` [Darwiche, 2004]
    - `miniC2D` [Oztok and Darwiche, 2015]
    - `d4` [Lagniez and Marquis, 2017]

Contribution: `ADDMC`

- Efficient data structure: algebraic decision diagrams (ADDs)
- Dynamic programming for combining ADDs – *mitigating exponential blowup*

Progress

1. Boolean Model Counting Problem (#SAT)
2. Algebraic Decision Diagrams (ADDs)
3. Factored Representation and Dynamic Programming
4. Experimental Evaluation

Data Structure: Binary Decision Diagram (BDD) [Bryant, 1986]

Formula $F : 2^V \to \mathbb{B}$ with variable count $n = |V|$

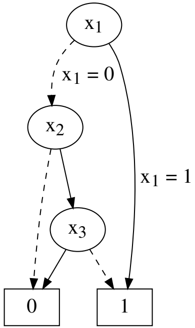
Full table

Inefficient data structure: $\Theta(2^n)$

Assignment $\alpha \in 2^V$			$F(\alpha)$
x_1	x_2	x_3	
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Binary decision diagram (BDD)

More efficient data structure: $O(2^n)$



Data Structure: Algebraic Decision Diagram (ADD) [Bahar et al., 1997]

Weight function $W : 2^V \to \mathbb{R}$ with variable count $n = |V|$

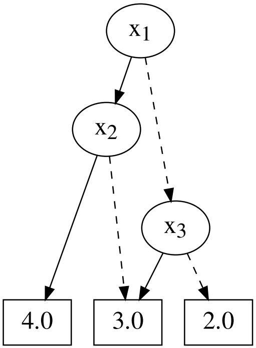
Full table

Inefficient data structure: $\Theta(2^n)$

Assignment $\alpha \in 2^V$			$W(\alpha)$
x_1	x_2	x_3	
0	0	0	2.0
0	0	1	3.0
0	1	0	2.0
0	1	1	3.0
1	0	0	3.0
1	0	1	3.0
1	1	0	4.0
1	1	1	4.0

Algebraic decision diagram (ADD)

More efficient data structure: $O(2^n)$



Projection: Unweighted Model Counting Problem

- Formula $F : 2^{\{x_1,\ldots,x_n\}} \to \mathbb{B}$ as function $2^{\{x_1,\ldots,x_n\}} \to \mathbb{N}$ (**natural-number set** $\{0, 1, 2, \ldots\}$)
- **Projection** of F w.r.t. variable x_1:

$$\left(\sum_{x_1} F\right)(x_2, \ldots, x_n) = F(0, x_2, \ldots, x_n) + F(1, x_2, \ldots, x_n)$$

- Exhaustive projection:

$$\sum_{x_n} \ldots \sum_{x_2} \sum_{x_1} F = F(0, 0, \ldots, 0) + F(0, 0, \ldots, 1) + \ldots + F(1, 1, \ldots, 1)$$

Remark 1 (Unweighted Model Count via Projection)

$$\#F = \sum_{x_n} \ldots \sum_{x_2} \sum_{x_1} F$$

Projection: Weighted Model Counting Problem

- Formula-weight product $F \cdot W : 2^{\{x_1,\dots,x_n\}} \to \mathbb{R}$
- **Projection** of $F \cdot W$ w.r.t. variable x_1:

$$\left(\sum_{x_1}(F \cdot W)\right)(x_2, \dots, x_n) = (F \cdot W)(0, x_2, \dots, x_n) + (F \cdot W)(1, x_2, \dots, x_n)$$

- Exhaustive projection:

$$\sum_{x_n} \dots \sum_{x_2} \sum_{x_1}(F \cdot W) = (F \cdot W)(0, 0, \dots, 0) + \dots + (F \cdot W)(1, 1, \dots, 1)$$

Theorem 1 (Weighted Model Count via Projection)

$$\#(F, W) = \sum_{x_n} \dots \sum_{x_2} \sum_{x_1}(F \cdot W)$$

Monolithic Representation versus Factored Representation

Naive approach: using *monolithic representation* of formula F and weight function W

- Constructs big ADDs for F and W with n variables
- Scales poorly for large instances: ADDs are $O(2^n)$

Contribution: algorithm that exploits *factored representation* of F and W

- Constructs small ADDs for factors of F and W
- Combines ADDs with dynamic programming

Progress

Factored Representation: Conjunctive Normal Form (CNF) Formula

Formula:

$$F = (x_1 \vee x_3) \wedge (\neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3) \wedge x_3$$

- **Positive literals** are non-negated variables: x_1, x_2, x_3
- **Negative literals** are negated variables: $\neg x_2, \neg x_3$

Factored Representation: Conjunctive Normal Form (CNF) Formula

Formula:

$$F = (x_1 \vee x_3) \wedge (\neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3) \wedge x_3$$

- **Positive literals** are non-negated variables: x_1, x_2, x_3
- **Negative literals** are negated variables: $\neg x_2, \neg x_3$
- **Clauses** are disjunctions of literals:

$$x_1 \vee x_3 : 2^{\{x_1, x_3\}} \to \mathbb{B} \qquad \neg x_2 \vee x_3 : 2^{\{x_2, x_3\}} \to \mathbb{B}$$

$$x_2 \vee \neg x_3 : 2^{\{x_2, x_3\}} \to \mathbb{B} \qquad x_3 : 2^{\{x_3\}} \to \mathbb{B}$$

- **Conjunctive Normal Form (CNF) formula** is conjunction of clauses: $F : 2^{\{x_1, x_2, x_3\}} \to \mathbb{B}$

Factorization:

$$F = (x_1 \vee x_3) \cdot (\neg x_2 \vee x_3) \cdot (x_2 \vee \neg x_3) \cdot x_3$$

Factored Representation: Literal-Weight Function

Each variable gets two **literal weights**:

$\texttt{weight}(x_1) \in \mathbb{R}$	$\texttt{weight}(x_2) \in \mathbb{R}$
$\texttt{weight}(\neg x_1) \in \mathbb{R}$	$\texttt{weight}(\neg x_2) \in \mathbb{R}$

Equivalently, each variable gets a **unit-weight function**:

$W_{x_1} : 2^{\{x_1\}} \to \mathbb{R}$	$W_{x_2} : 2^{\{x_2\}} \to \mathbb{R}$

Literal-weight function:

$$W : 2^{\{x_1, x_2\}} \to \mathbb{R}$$

Factorization:

$$W = W_{x_1} \cdot W_{x_2}$$

Factored Representation: Literal-Weighted Model Count of CNF Formula

Construct factors of:

- Conjunctive Normal Form (CNF) formula F with clauses C:

$$F = \prod_{C \in F} C$$

- Literal-weight function W with variable set V:

$$W = \prod_{x \in V} W_x$$

Compute weighted model count:

$$\#(F, W) = \sum_{x_n} \cdots \sum_{x_2} \sum_{x_1} (F \cdot W) = \sum_{x_n} \cdots \sum_{x_2} \sum_{x_1} \left(\prod_{C \in F} C \cdot \prod_{x \in V} W_x \right)$$

Push projection ($\sum$) inward: early projection

Early Projection

Theorem 2

If we have:

- *Variable sets* Y *and* Z
- *Functions* $g : 2^Y \to \mathbb{R}$ *and* $h : 2^Z \to \mathbb{R}$
- *Variable* $x \in Y \setminus Z$

Then:

$$\sum_x (g \cdot h) = \left(\sum_x g \right) \cdot h$$

Early projection can reduce size of intermediate computation

- Database join-query optimization [McMahan et al., 2004]
- Boolean satisfiability [Pan and Vardi, 2005]

Early Projection: Unweighted Model Counting

CNF formula $F = (x_1 \vee x_3) \wedge (\neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3) \wedge x_3$

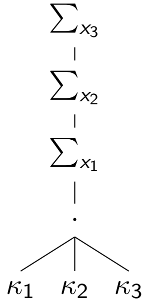
Clusters (partition of clauses)

$\kappa_1 = \{x_1 \vee x_3\}$

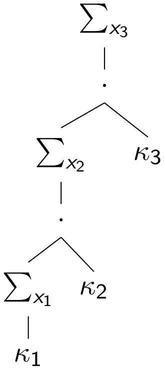
$\kappa_2 = \{\neg x_2 \vee x_3, x_2 \vee \neg x_3\}$

$\kappa_3 = \{x_3\}$

Late projection



Early projection

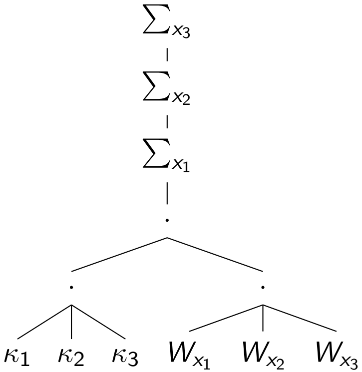


Heuristic: **bucket elimination** (of variable x_i from cluster κ_i) [Dechter, 1999]

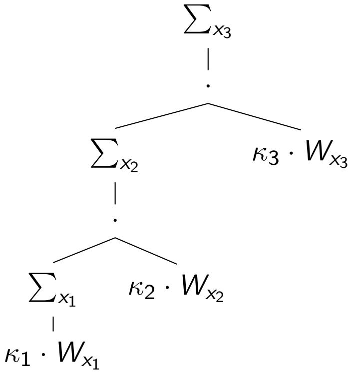
Early Projection: Weighted Model Counting

CNF Formula $F = \kappa_1 \wedge \ \kappa_2 \wedge \kappa_3$ and literal-weight function $W = W_{x_1} \cdot W_{x_2} \cdot W_{x_3}$

Late projection



Early projection



Heuristic: bucket elimination (of variable x_i from cluster κ_i and unit-weight function W_{x_i})

Contributions: Theoretical Framework and Experimental Evaluation

Contributions:

1. Algorithm for weighted model counting using algebraic decision diagrams (ADDs)
   - Constructing small ADDs for factors of formula and weight function
   - Combining ADDs with dynamic programming and early projection
2. Tool: `Algebraic Decision Diagram Model Counter` (`ADDMC`)
   - Comparison of `ADDMC` to state-of-the-art weighted model counters

   Public GitHub repository:

   `https://github.com/vardigroup/ADDMC`

Progress

Benchmarks

1914 **benchmarks**: CNF model counting problem instances

1091 benchmarks from the **Bayes** class [Sang et al., 2005]

- *Deterministic Quick Medical Reference*
- *Grid Networks*
- *Plan Recognition*

https://www.cs.rochester.edu/u/kautz/

823 benchmarks from the **Non-Bayes** class [Clarke et al., 2001; Sinz et al., 2003; Palacios and Geffner, 2009; Klebanov et al., 2013]

- *Planning*
- *Bounded Model Checking*
- *Circuit*
- *Configuration*
- *Quantitative Information Flow*
- *Scheduling*
- *Handmade*
- *Random*

http://www.cril.univ-artois.fr/KC/

Experiment: Comparing Weighted Model Counters

High-performance computing cluster at Rice University (NOTS):

- Hardware: Xeon E5-2650v2 CPU (2.60-GHz)
- Memory limit: 24 GB
- Time limit: 1000 seconds

Experiment: Comparing Weighted Model Counters

Table 1: Performance of state-of-the-art weighted model counters

Weight model counters		Benchmarks solved (of 1914)		
		Unique solver	Fastest solver	Total
Virtual best solvers (VBS)	VBS1: with ADDMC	–	–	1771
	VBS0: without ADDMC	–	–	1647
Actual solvers	d4	12	283	1587
	c2d	0	13	1417
	miniC2D	8	61	1407
	ADDMC – our tool	**124**	**763**	1404
	Cachet	14	651	1383

Experiment: Comparing Weighted Model Counters

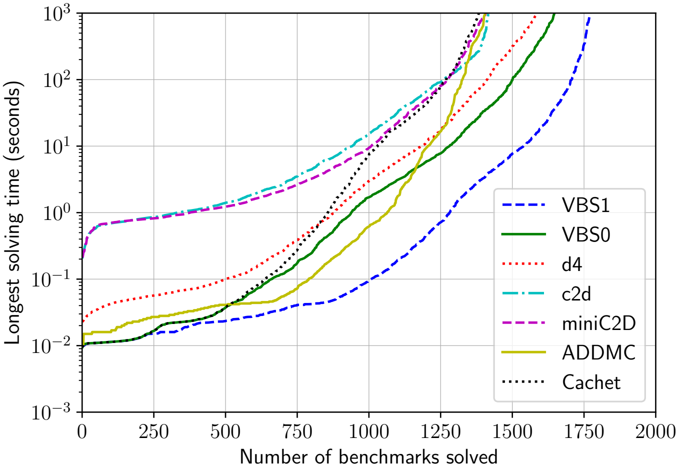


Figure 1: Cactus plot of virtual best solvers (VBS1 with ADDMC; VBS0 without ADDMC) and actual solvers

Conclusion

Summary:

- Problem: Boolean model counting (#SAT)
  - Complexity: #P-complete
  - Numerous applications, especially in probabilistic reasoning
- Techniques:
  - Efficient data structure: algebraic decision diagrams (ADDs)
  - Dynamic programming
- Experimental result: `ADDMC` improves virtual best solver

Future work:

- Other efficient data structures
  - Affine algebraic decision diagrams (AADDs) [Sanner and McAllester, 2005]
  - AND/OR multi-valued decision diagrams (AOMDDs) [Mateescu et al., 2008]
- Graph decomposition for Conjunctive Normal Form (CNF) clause clustering
  - Model counting with tensor-network contraction [Dudek et al., 2019]
  - Model counting with database technology [Dresden, 2020]

References I

R Iris Bahar, Erica A Frohm, Charles M Gaona, Gary D Hachtel, Enrico Macii, Abelardo Pardo, and Fabio Somenzi. Algebraic decision diagrams and their applications. *Form Method Syst Des*, 10(2-3):171–206, 1997.

Randal E Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE TC*, 35(8), 1986.

Edmund Clarke, Armin Biere, Richard Raimi, and Yunshan Zhu. Bounded model checking using satisfiability solving. *Form Method Syst Des*, 19(1):7–34, 2001.

Adnan Darwiche. New advances in compiling CNF into decomposable negation normal form. In *ECAI*, pages 328–332, 2004.

Rina Dechter. Bucket elimination: a unifying framework for reasoning. *AI*, 113(1-2):41–85, 1999.

TU Dresden. Exploiting database management systems and treewidth for counting. *Practical Aspects of Declarative Languages*, page 151, 2020.

References II

Jeffrey M Dudek, Leonardo Dueñas-Osorio, and Moshe Y Vardi. Efficient contraction of large tensor networks for weighted model counting through graph decompositions. *arXiv preprint arXiv:1908.04381*, 2019.

Leonardo Duenas-Osorio, Kuldeep S Meel, Roger Paredes, and Moshe Y Vardi. Counting-based reliability estimation for power-transmission grids. In *AAAI*, 2017.

Vladimir Klebanov, Norbert Manthey, and Christian Muise. SAT-based analysis and quantification of information flow in programs. In *QEST*, pages 177–192, 2013.

Jean-Marie Lagniez and Pierre Marquis. An improved decision-DNNF compiler. In *IJCAI*, pages 667–673, 2017.

Robert Mateescu, Rina Dechter, and Radu Marinescu. AND/OR multi-valued decision diagrams for graphical models. *JAIR*, 33:465–519, 2008.

Benjamin J McMahan, Guoqiang Pan, Patrick Porter, and Moshe Y Vardi. Projection pushing revisited. In *EDBT*, pages 441–458, 2004.

References III

Umut Oztok and Adnan Darwiche. A top-down compiler for sentential decision diagrams. In *IJCAI*, pages 3141–3148, 2015.

Hector Palacios and Hector Geffner. Compiling uncertainty away in conformant planning problems with bounded width. *JAIR*, 35:623–675, 2009.

G. Pan and M.Y. Vardi. Symbolic techniques in satisfiability solving. *J Autom Reason*, 35 (1-3):25–50, 2005.

Tian Sang, Fahiem Bacchus, Paul Beame, Henry A Kautz, and Toniann Pitassi. Combining component caching and clause learning for effective model counting. *SAT*, pages 20–28, 2004.

Tian Sang, Paul Beame, and Henry Kautz. Solving Bayesian networks by weighted model counting. In *AAAI*, volume 1, pages 475–482. AAAI Press, 2005.

Scott Sanner and David McAllester. Affine algebraic decision diagrams and their application to structured probabilistic inference. In *IJCAI*, pages 1384–1390, 2005.

References IV

Michael A Shwe, Blackford Middleton, David E Heckerman, Max Henrion, Eric J Horvitz, Harold P Lehmann, and Gregory F Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. *Methods of Information in Medicine*, 1991.

Carsten Sinz, Andreas Kaiser, and Wolfgang Küchlin. Formal methods for the validation of automotive product configuration data. *AI EDAM*, 17(1):75–97, 2003.

Leslie G Valiant. The complexity of enumeration and reliability problems. *SICOMP*, 8(3): 410–421, 1979.