Generating Random WMC Instances

An Empirical Analysis with Varying Primal Treewidth

Paulius Dilkas

University of Edinburgh, UK

Workshop on Counting and Sampling 2022

Weighted Model Counting (WMC)

- A generalisation of propositional model counting (#SAT)
- Applications:
  - graphical models
  - probabilistic programming
  - neural-symbolic artificial intelligence
- WMC algorithms use:
  - dynamic programming
  - knowledge compilation
  - SAT solvers

Example

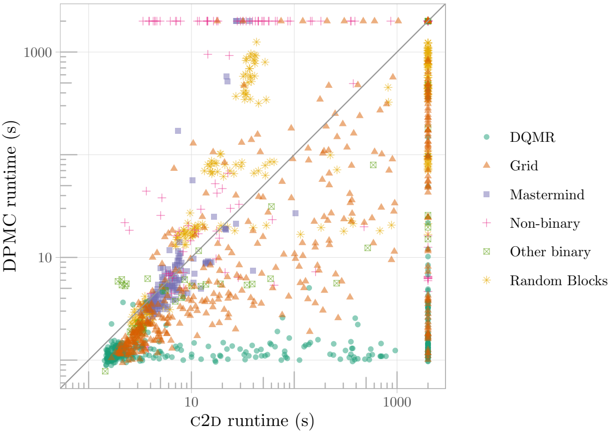
$w(x) = 0.3,\ w(\neg x) = 0.7,$
$w(y) = 0.2,\ w(\neg y) = 0.8$

$\mathrm{WMC}(x \lor y) = w(x)w(y) + w(x)w(\neg y) + w(\neg x)w(y) = 0.44$

(Some of the) WMC Algorithms

- CACHET (Sang et al. 2004)
  - component caching
- C2D (Darwiche 2004)
  - knowledge compilation to d-DNNF
- D4 (Lagniez and Marquis 2017)
  - knowledge compilation to decision-DNNF
- miniC2D (Oztok and Darwiche 2015)
  - knowledge compilation to decision sentential decision diagrams
- DPMC (Dudek, Phan and Vardi 2020)
  - dynamic programming with algebraic decision diagrams (ADDs) and tree decomposition based planning

The Performance Characteristics of WMC Algorithms



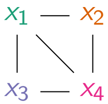
Data from Dilkas and Belle (2021): various Bayesian networks encoded using the method by Darwiche (2002)

A Note on Primal Treewidth

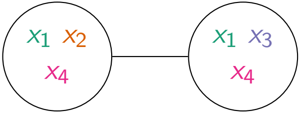
Formula in CNF:

$$\phi = (x_4 \vee \neg x_3 \vee x_1) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

Its primal graph:



Its minimum-width tree decomposition:



$\therefore$ the primal treewidth of ϕ is 2

The Parameterised Complexity of WMC Algorithms

Let n be the number of variables and m be the number of clauses.

- Component caching (used in CACHET) is $2^{\mathcal{O}(w)}n^{\mathcal{O}(1)}$, where w is the branchwidth of the underlying hypergraph (Bacchus, Dalmao and Pitassi 2009)
  - Branchwidth is within a constant factor of primal treewidth
- C2D is based on an algorithm, which is $\mathcal{O}(2^w mw)$, where w is at most primal treewidth (Darwiche 2001; Darwiche 2004)
- DPMC can be shown to be $\mathcal{O}(4^w mn)$, where w is an upper bound on primal treewidth

Generating Random WMC Instances: The Algorithm

$\phi \leftarrow$ empty CNF formula;
$G \leftarrow$ empty graph;
for $i \leftarrow 1$ **to** m **do**
    $X \leftarrow \emptyset$;
    for $j \leftarrow 1$ **to** k **do**
        $x \leftarrow$ `newVariable(`X, G`)`;
        $\mathcal{V}(G) \leftarrow \mathcal{V}(G) \cup \{ x \}$;
        $\mathcal{E}(G) \leftarrow \mathcal{E}(G) \cup \{ \{ x, y \} \mid y \in X \}$;
        $X \leftarrow X \cup \{ x \}$;
    $\phi \leftarrow \phi \cup \{ \{ l \leftsquigarrow \mathcal{U}\{ x, \neg x \} \mid x \in X \} \}$;

- the number of clauses
- clause width
- a function to pick a variable
- a (fair) coin flip

How to Pick a Variable

Parameter $\rho \in [0, 1]$ biases the probability distribution towards adding variables that would introduce fewer new edges.

Function `newVariable`(*set of variables* X, *primal graph* G)**:**

$N \leftarrow \{ e \in \mathcal{E}(G) \mid |e \cap X| = 1 \}$;

if $N = \emptyset$ **then return** $x \leftarrow\!\!\sim \mathcal{U}(\{ x_1, x_2, \dots, x_n \} \setminus X)$;

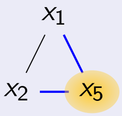
return

$$x \leftarrow\!\!\sim \left(\{ x_1, x_2, \dots, x_n \} \setminus X, y \mapsto \frac{1-\rho}{n-|X|} + \rho \frac{|\{ z \in X \mid \{ y, z \} \in \mathcal{E}(G) \}|}{|N|} \right);$$

An Example

Setup

Primal graph:



Let $n = 5$, $k = 3$, and $\rho = 0.3$.
Partial formula: $(\neg x_5 \vee x_2 \vee x_1) \wedge (x_5 \vee ? \vee ?)$.

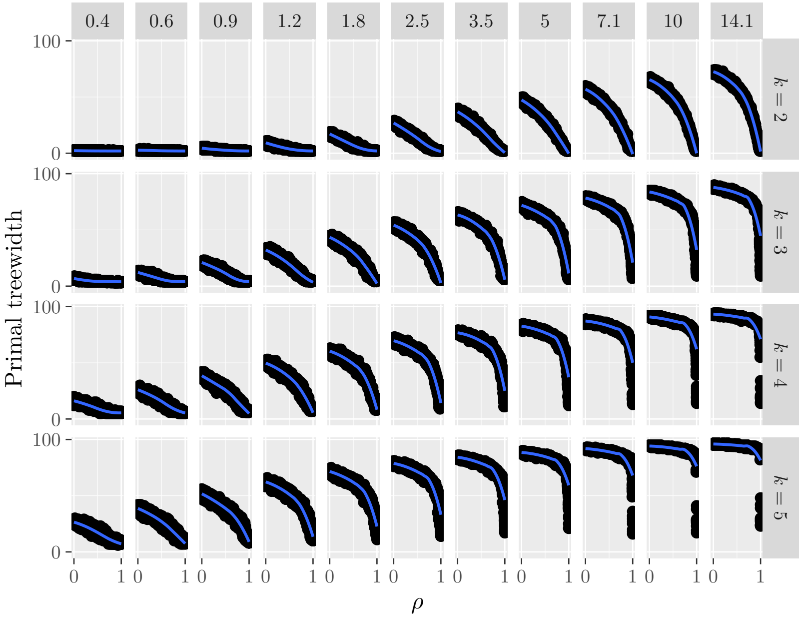
The probability distribution for the next variable

Base probability of each variable being chosen:

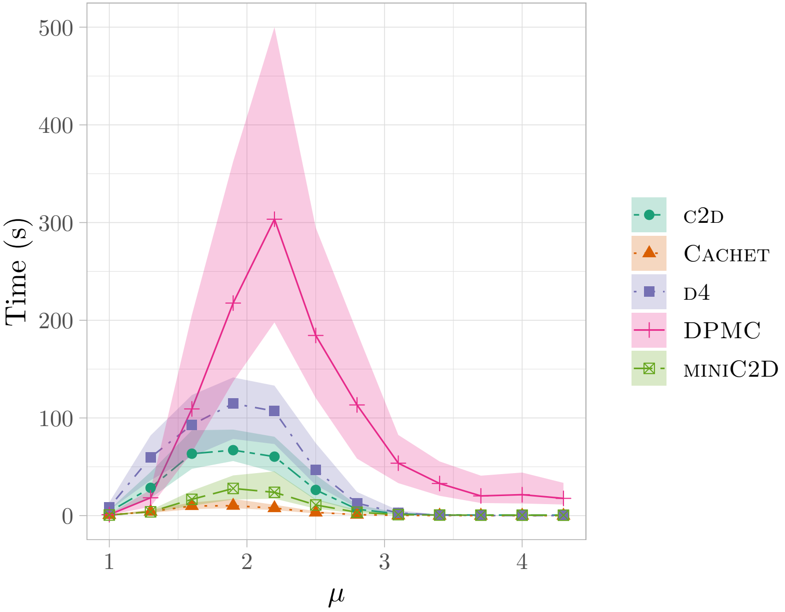
$$\frac{1-\rho}{n-|X|} = \frac{1-0.3}{5-1} = 0.175.$$

Both x_1 and x_2 get a bonus probability of $\rho/2$ for each being the endpoint of one out of the two neighbourhood edges.

The Relationship Between ρ and Primal Treewidth



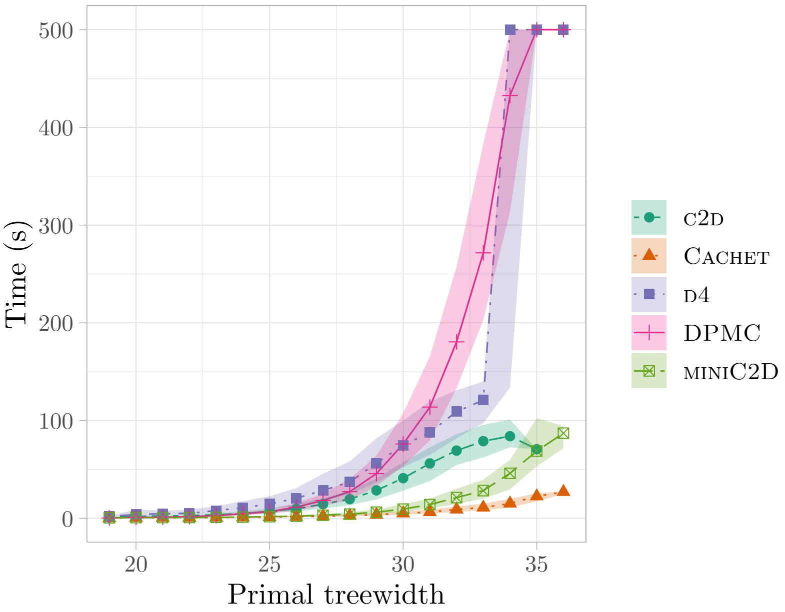
Hardness w.r.t. Density (when $\rho = 0$)



Peak Hardness w.r.t. Density

- Bayardo Jr. and Pehoushek (2000) show some #SAT algorithms to peak at $\mu = 1.2$ and $\mu = 1.9$
- Cachet is known to peak at $\mu = 1.8$ (Sang et al. 2004)
- In our experiments:
  - DPMC peaks at $\mu = 2.2$
  - all other algorithms peak at $\mu = 1.9$

Hardness w.r.t. Primal Treewidth (when $\mu = 1.9$)



Is The Relationship Exponential?

Let us fit the model $\ln t \sim \alpha w + \beta$, i.e., $t \sim e^{\beta}(e^{\alpha})^{w}$, where t is runtime, and w is primal treewidth

Is The Relationship Exponential?

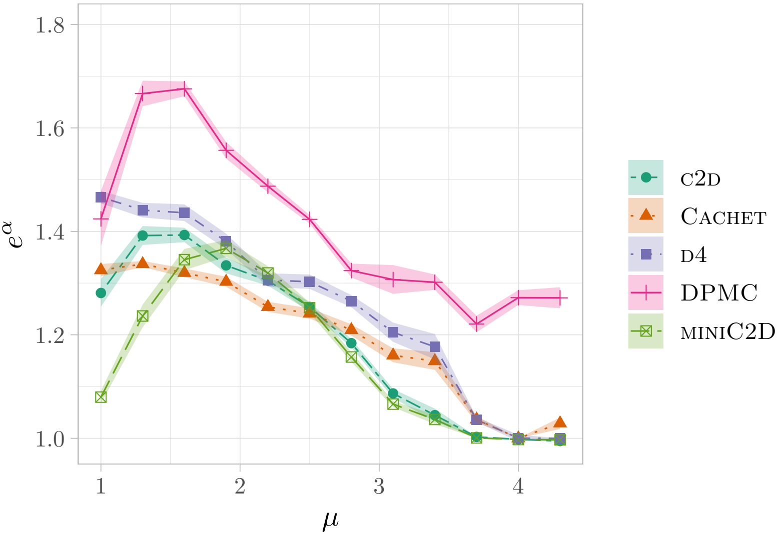
Let us fit the model $\ln t \sim \alpha w + \beta$, i.e., $t \sim e^{\beta}(e^{\alpha})^{w}$, where t is runtime, and w is primal treewidth

μ	c2d	Cachet	d4	DPMC	miniC2D
4.3	0.62	0.33	1	0.94	0.53
4	0.19	0.49	0	0.97	0.43
3.7	0.57	0.71	0.83	0.94	0.18
3.4	0.47	0.85	0.8	0.97	0.53
3.1	0.88	0.92	0.91	0.91	0.9
2.8	0.97	0.96	0.98	0.98	0.95
2.5	0.98	0.98	0.97	1	0.98
2.2	0.99	0.98	0.98	0.99	0.98
1.9	0.98	0.99	0.98	0.99	0.98
1.6	0.99	0.99	0.98	1	0.96
1.3	0.98	1	0.99	0.99	0.9
1	0.91	0.99	0.99	0.87	0.79

R^2 0.25 0.50 0.75 1.00

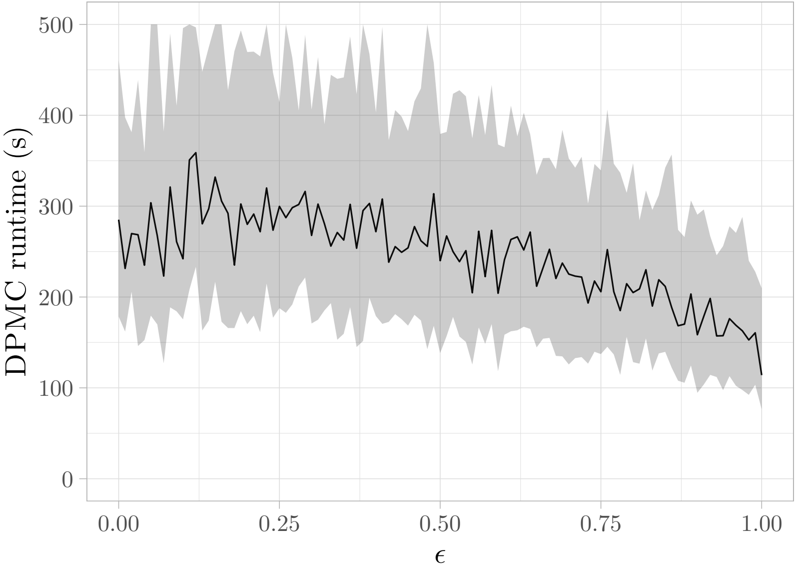
Is The Relationship Exponential?

Let us fit the model $\ln t \sim \alpha w + \beta$, i.e., $t \sim e^{\beta}(e^{\alpha})^{w}$, where t is runtime, and w is primal treewidth



Bonus: How DPMC Reacts to Redundancy in Weights

Let ϵ be the proportion of variables x s.t. $w(x) = w(\neg x) = 0.5$



Summary

Observations

- All algorithms scale exponentially w.r.t. primal treewidth
- The running time of DPMC peaks at a higher density and scales worse w.r.t. primal treewidth
  - Related to the complexity of ADD multiplication
- miniC2D seems to be best at low density high primal treewidth instances

Future work

- A theoretical relationship between ρ and primal treewidth
- Non-k-CNF instances
- Similar observations on real data
- SATzilla for WMC