Generating Random WMC Instances

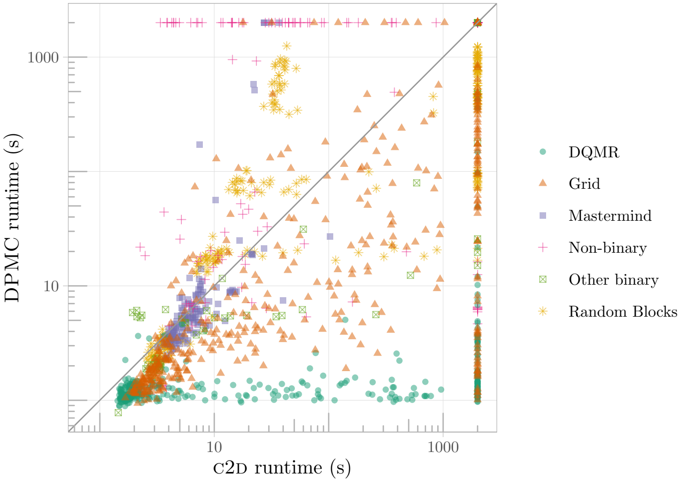
An Empirical Analysis with Varying Primal Treewidth

Paulius Dilkas

National University of Singapore

30th May 2024

Which Algorithm Is Better? It Depends on the Data



The runtime data is from Dilkas and Belle (2021): various Bayesian networks encoded using the approach by Darwiche (2002)

The Problem: Weighted Model Counting (WMC)

- A generalisation of propositional model counting (#SAT)
- Applications:
  - graphical models
  - probabilistic programming
  - neuro-symbolic AI
- 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)
  - a SAT solver with clause learning and 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 and tree decomposition based planning

Frequently Asked Questions

Frequently Asked Questions

Q: Why isn't SharpSAT-TD included in the experiments?

A: Because I started setting up these experiments eight days after the SharpSAT-TD paper came out

Frequently Asked Questions

Q: Why isn't SharpSAT-TD included in the experiments?

A: Because I started setting up these experiments eight days after the SharpSAT-TD paper came out

Q: Why isn't GANAK included in the experiments?

A: Because it's easy to argue that probabilistic algorithms are out of scope (and I had lots of algorithms already)

Frequently Asked Questions

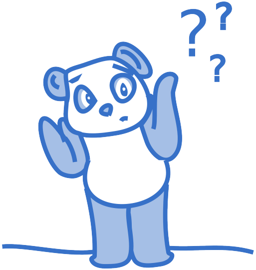
Q: Why isn't SharpSAT-TD included in the experiments?

A: Because I started setting up these experiments eight days after the SharpSAT-TD paper came out

Q: Why isn't GANAK included in the experiments?

A: Because it's easy to argue that probabilistic algorithms are out of scope (and I had lots of algorithms already)

Q: Why am I giving a talk about this now?

A: 

Tree Decompositions and Primal Treewidth

Formula in CNF:

$$\phi = (x_4 \lor \neg x_3 \lor x_1) \land (\neg x_2 \lor x_4) \land (\neg x_1 \lor x_2 \lor x_4)$$

Tree Decompositions and Primal Treewidth

Formula in CNF:

$$\phi = (x_4 \lor \neg x_3 \lor x_1) \land (\neg x_2 \lor x_4) \land (\neg x_1 \lor x_2 \lor x_4)$$

Its primal graph:

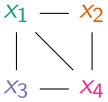
$$\begin{tikzcd} x_1 \arrow[r, dash] \arrow[d, dash] \arrow[dr, dash] & x_2 \arrow[d, dash] \\ x_3 \arrow[r, dash] & x_4 \end{tikzcd}$$

Tree Decompositions and Primal Treewidth

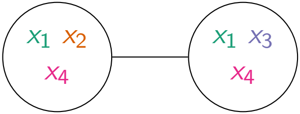
Formula in CNF:

$$\phi = (x_4 \lor \neg x_3 \lor x_1) \land (\neg x_2 \lor x_4) \land (\neg x_1 \lor x_2 \lor x_4)$$

Its primal graph:



Its minimum-width tree decomposition:

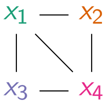


Tree Decompositions and Primal Treewidth

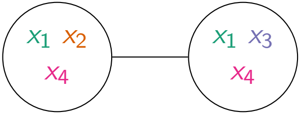
Formula in CNF:

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

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** $x \leftsquigarrow \mathcal{U}(\{\, x_1, x_2, \dots, x_n \,\} \setminus X)$;

return

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

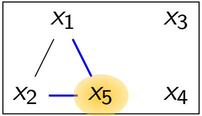
From Random SAT to Random WMC

We introduce parameter $\rho \in [0, 1]$ that biases the probability distribution towards adding variables that would introduce fewer new edges to the primal graph.

Example partially-filled formula:

$(\neg x_5 \lor x_2 \lor x_1) \land (x_5 \lor ? \lor ?)$

Its primal graph:



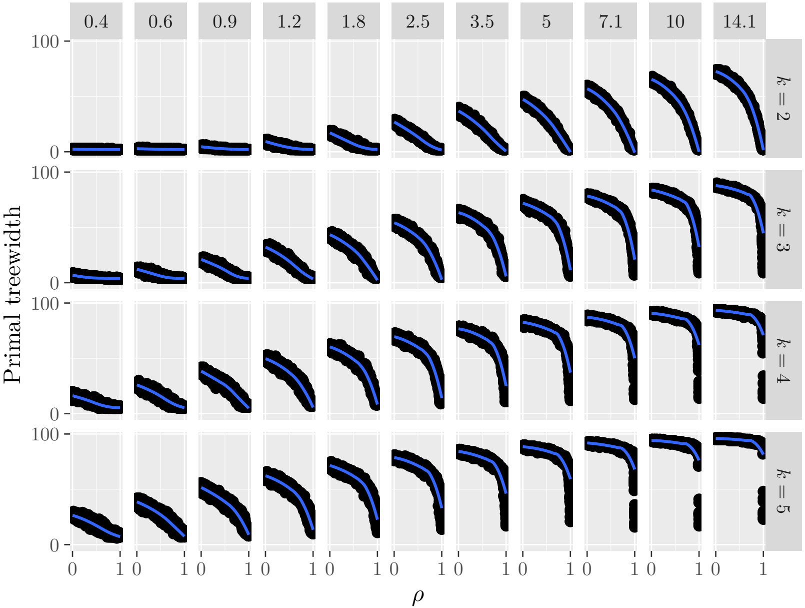
The probability distribution for the next variable

Base probability of each variable being chosen:

$$\frac{1-\rho}{4}.$$

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



Peak Hardness w.r.t. Density

Let μ denote the density, i.e., the number of clauses divided by the number of variables.

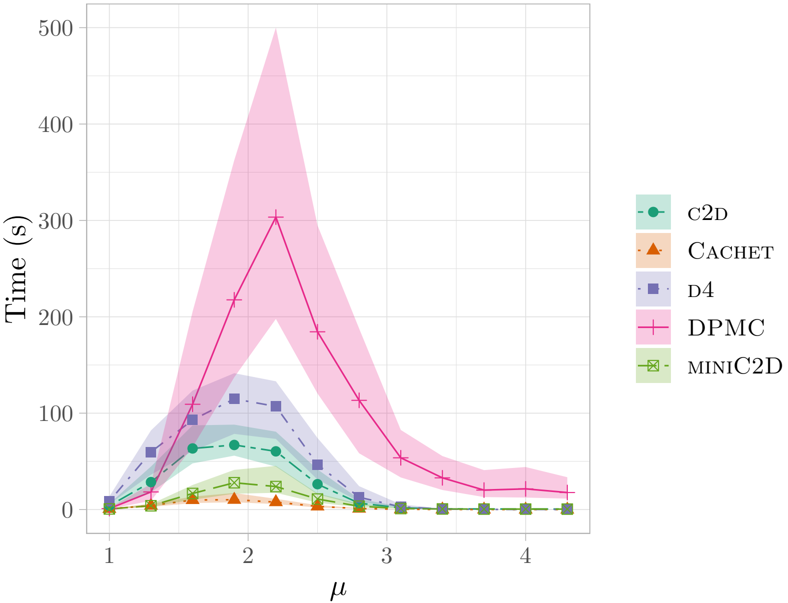
- CACHET is known to peak at $\mu = 1.8$ (Sang et al. 2004)
- Bayardo Jr. and Pehoushek (2000) show some #SAT algorithms to peak at $\mu = 1.2$ and $\mu = 1.9$

Peak Hardness w.r.t. Density

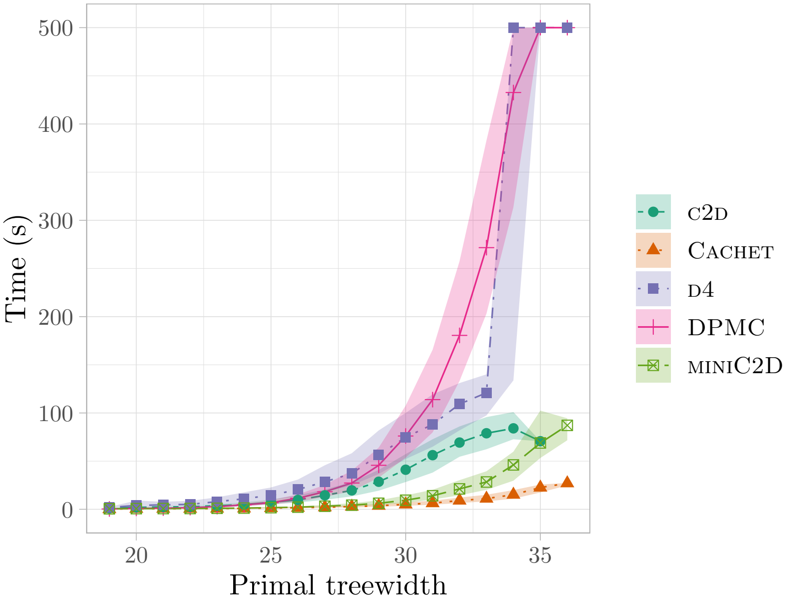
Let μ denote the density, i.e., the number of clauses divided by the number of variables.

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

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



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

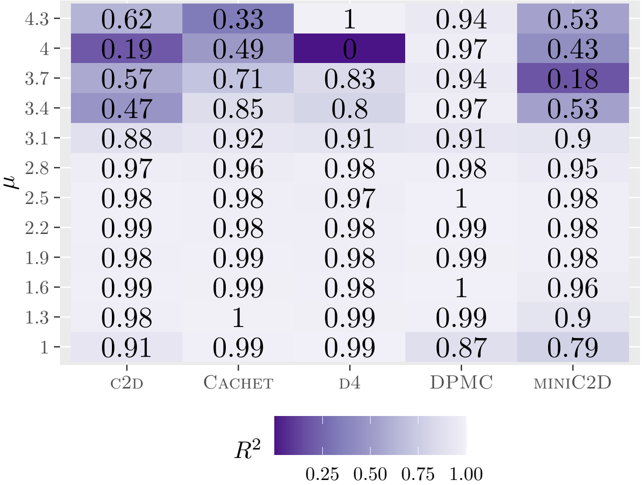


Is The Relationship Exponential?

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

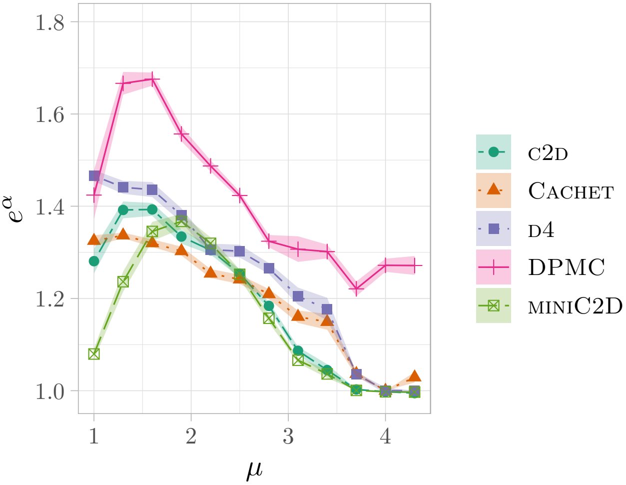
Is The Relationship Exponential?

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

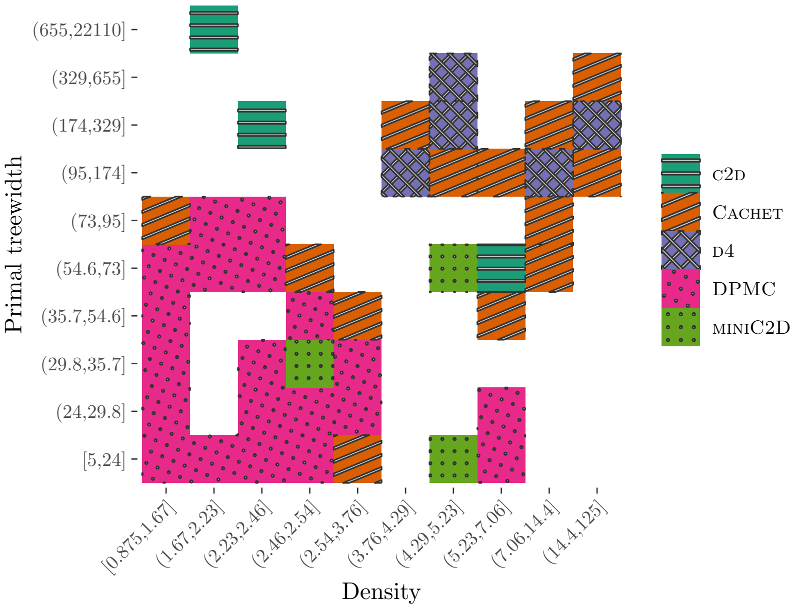


Is The Relationship Exponential?

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

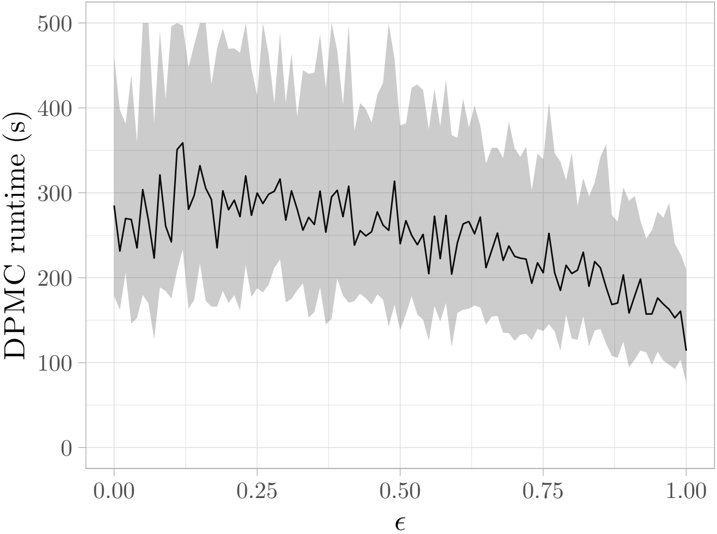


Does Real Data Confirm Our Observations?

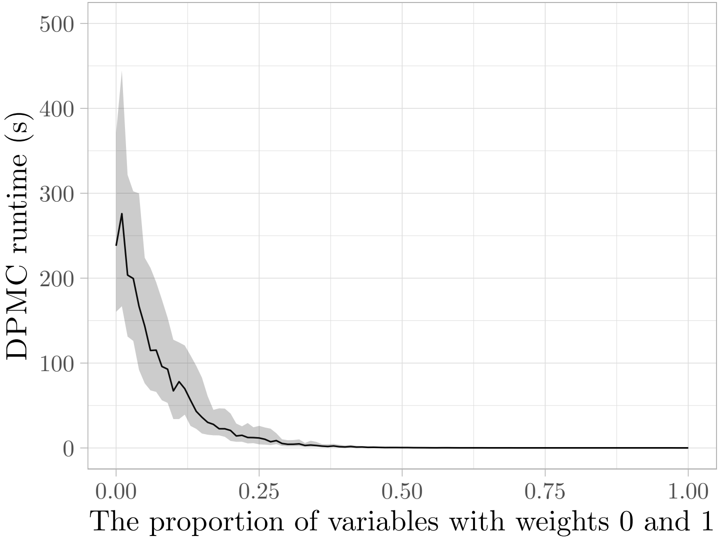


Bonus 1: How DPMC Reacts to Redundancy in Weights

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



Bonus 2: 0/1 Weights Make Counting Easy



Summary

- This work introduced a random model for WMC instances with a parameter that indirectly controls primal treewidth
- 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
- Future work:
  - A theoretical relationship between ρ and primal treewidth
  - Non-k-CNF instances
  - Algorithm portfolios for WMC