Who gets free ice cream?

Stochastic Constraint Propagation for Mining Probabilistic Networks

Anna Louise D. Latour[1], Behrouz Babaki[2], Siegfried Nijssen[3]

The Viral Marketing problem

running example of a

Stochastic Constraint Optimization Problem (SCOP):

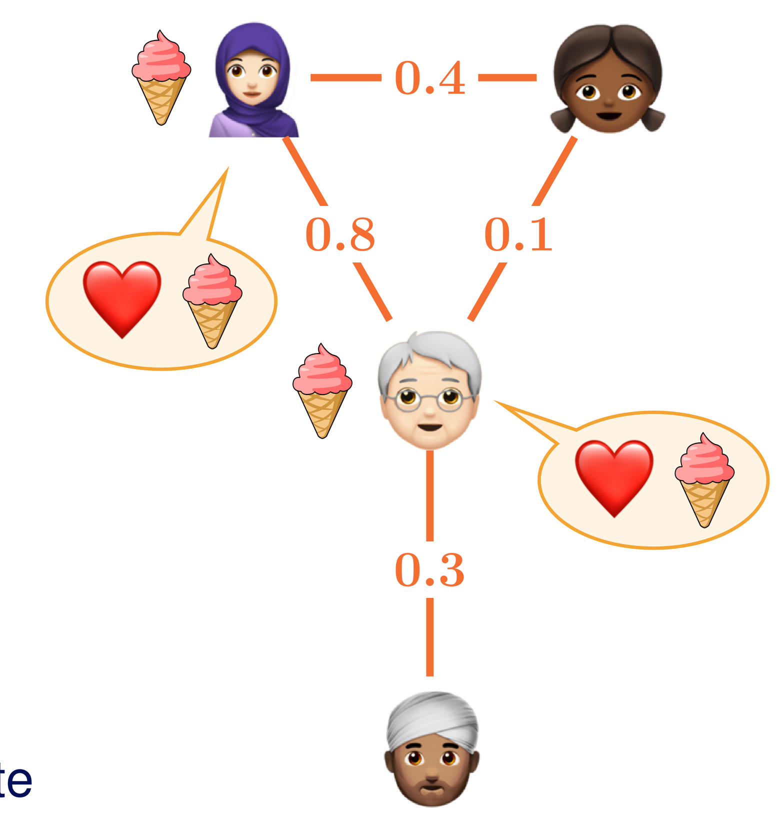
probabilistic spread of influence (word-of-mouth);
constraint on number of free ice cream samples we distribute;
optimal decision making: maximize expected # people who buy ice cream.

Problem: Find a **strategy** σ (a set of **decisions**)

which satisfies $\sum_{i \in \text{people}} d_i \leq k$,

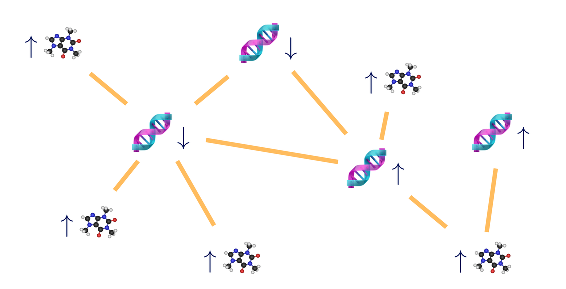
while maximizing $\sum_{i \in \text{people}} P(\phi_i \mid \sigma)$

$d_i = 1$ if person i gets a free ice cream sample and 0 otherwise
k maximum number of free ice cream samples we can distribute
ϕ_i person i buys ice cream

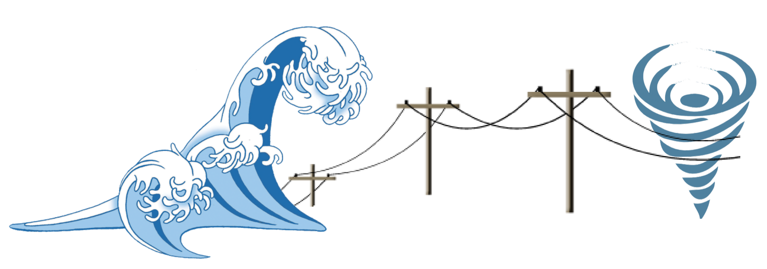


Other examples of Stochastic Constraint Optimization Problems (SCOPs):

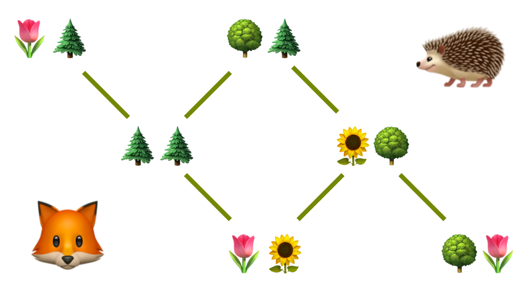
Signaling Regulatory Pathways



Powergrid Reliability



Landscape Connectivity



Common property: probability distributions are **monotonic**, which we exploit in a new constraint propagation algorithm for a **Stochastic Constraint on Monotonic Distributions** (SCMD).

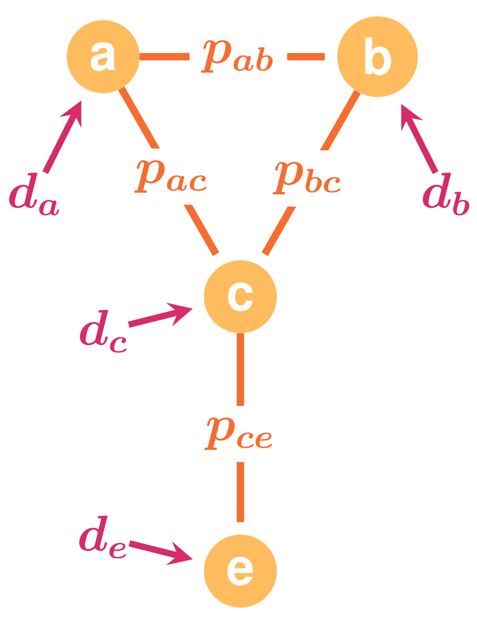
Background: SCOPs are hard

Problem:
I Weighted Model Counting (NP-hard);
II Exponential number of possible **strategies**.

Naïve enumeration and evaluation does not scale.

Our approach:
I Compile ϕ to Ordered Binary Decision Diagram (**OBDD**) for tractable **WMC**;
II Use Constraint Programming (**CP**) technology to efficiently traverse **search space**.

Step 1: logical model of problem



(Simplifying) assumptions for this poster:

- If a person receives a free sample, they will buy the ice cream in the future;
- A person i influences a person j with probability $p_{ij} = p_{ji}$;
- If a person i buys the ice cream and influences person j, then j will buy the ice cream.

Two types of Boolean variables:

$d_i \in \{0, 1\}$ (by **decision**),
$t_{ij} \in \{0, 1\}$ (by **chance** p_{ij}),

where $P(t_{ij} = 1) = p_{ij}$ and $P(t_{ij} = 0) = 1 - p_{ij}$.

Step 2: define stochastic constraint

Observe: maximization is repeated constraint solving, increasing the lower bound θ that we have to meet.
For simplicity of this poster, suppose we just want to solve

$$P(\phi_e \mid \sigma) \geq \theta \quad \text{(with } 0 < \theta \leq 1\text{)}, \tag{2}$$

where person e buys ice cream iff $\phi_e = \top$:

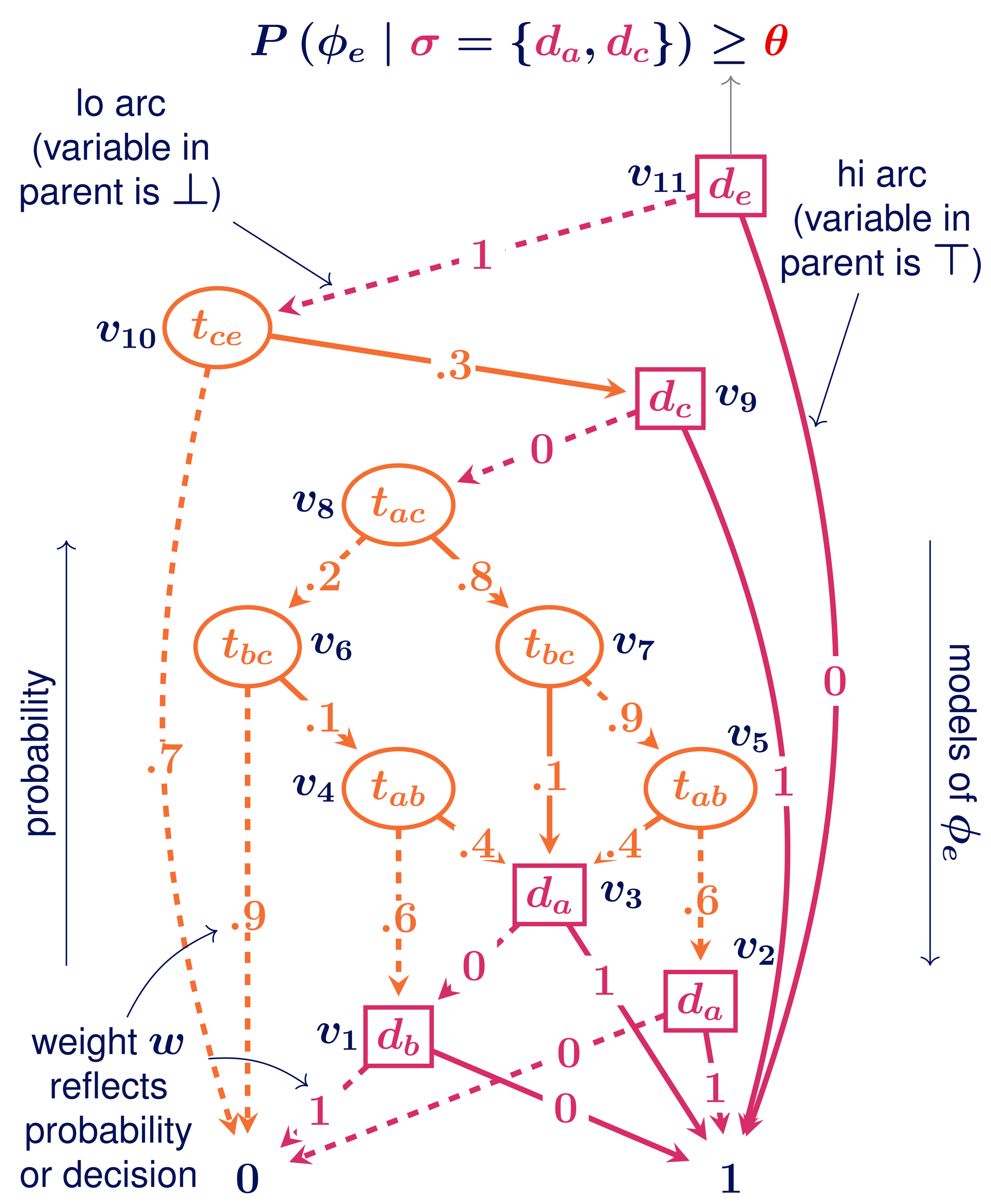
$$\begin{aligned}\phi_e = \; & d_e \vee (d_c \wedge t_{ce}) \vee (d_a \wedge t_{ac} \wedge t_{ce}) \vee \\ & (d_b \wedge t_{bc} \wedge t_{ce}) \vee \\ & (d_a \wedge t_{ab} \wedge t_{bc} \wedge t_{ce}) \vee \\ & (d_b \wedge t_{ab} \wedge t_{ac} \wedge t_{ce})\end{aligned} \tag{3}$$

Exact solving of eq. (2) requires

- Weighted Model Counting (**WMC**);
- evaluating quality of (all) **strategies**.

Existing methods not Generalized Arc Consistent (GAC).

Step 3: use OBDD to evaluate strategy



OBDD is a summary of truth table of eq. (3) and **encodes the probability distribution** (*not* the solutions to the constraint). Paths from root to leaf 1 represent **models** of ϕ_e.

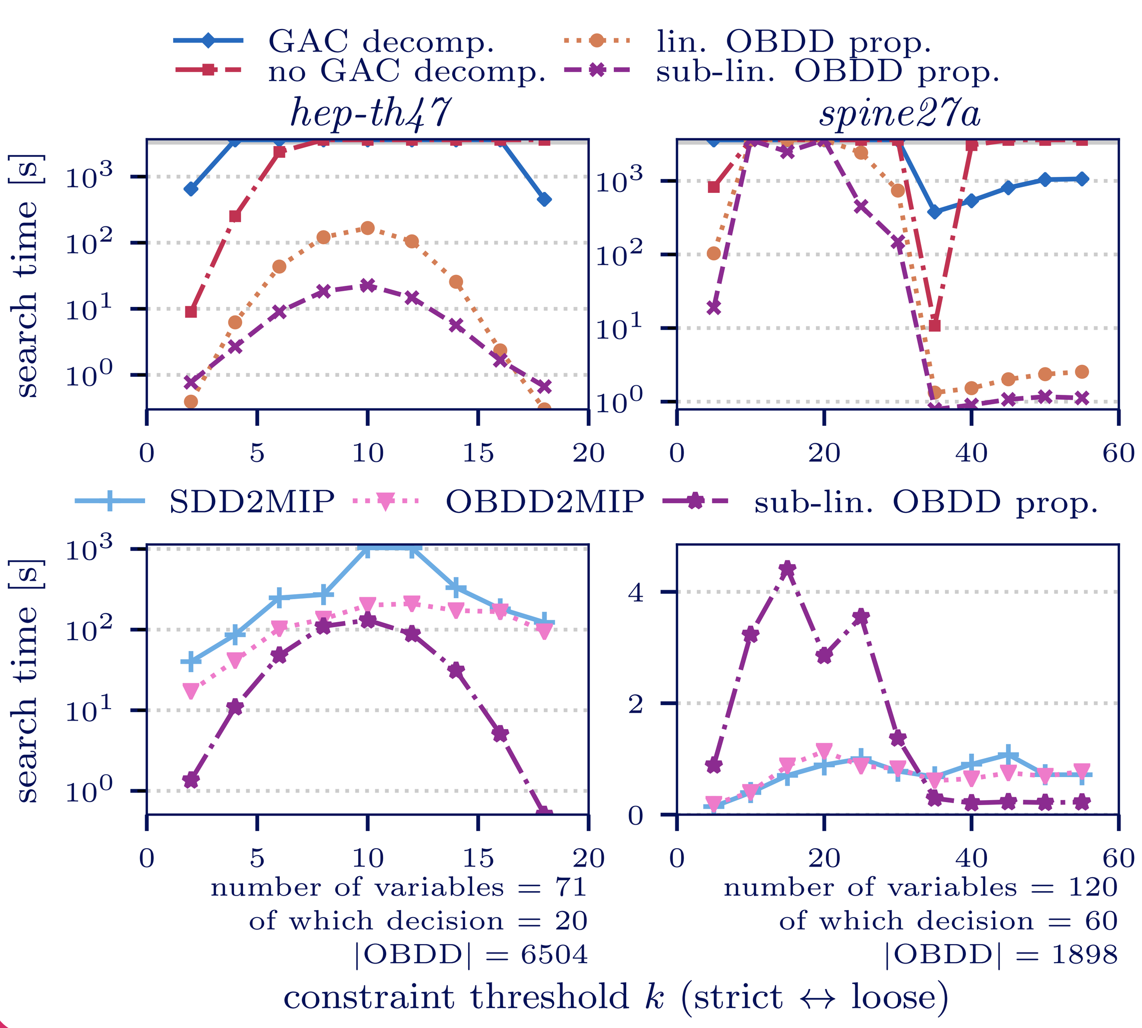
Upward sweep: for each OBDD node r, compute score:

$$v_r = w \cdot v_{hi} + (1 - w) \cdot v_{lo}. \tag{1}$$

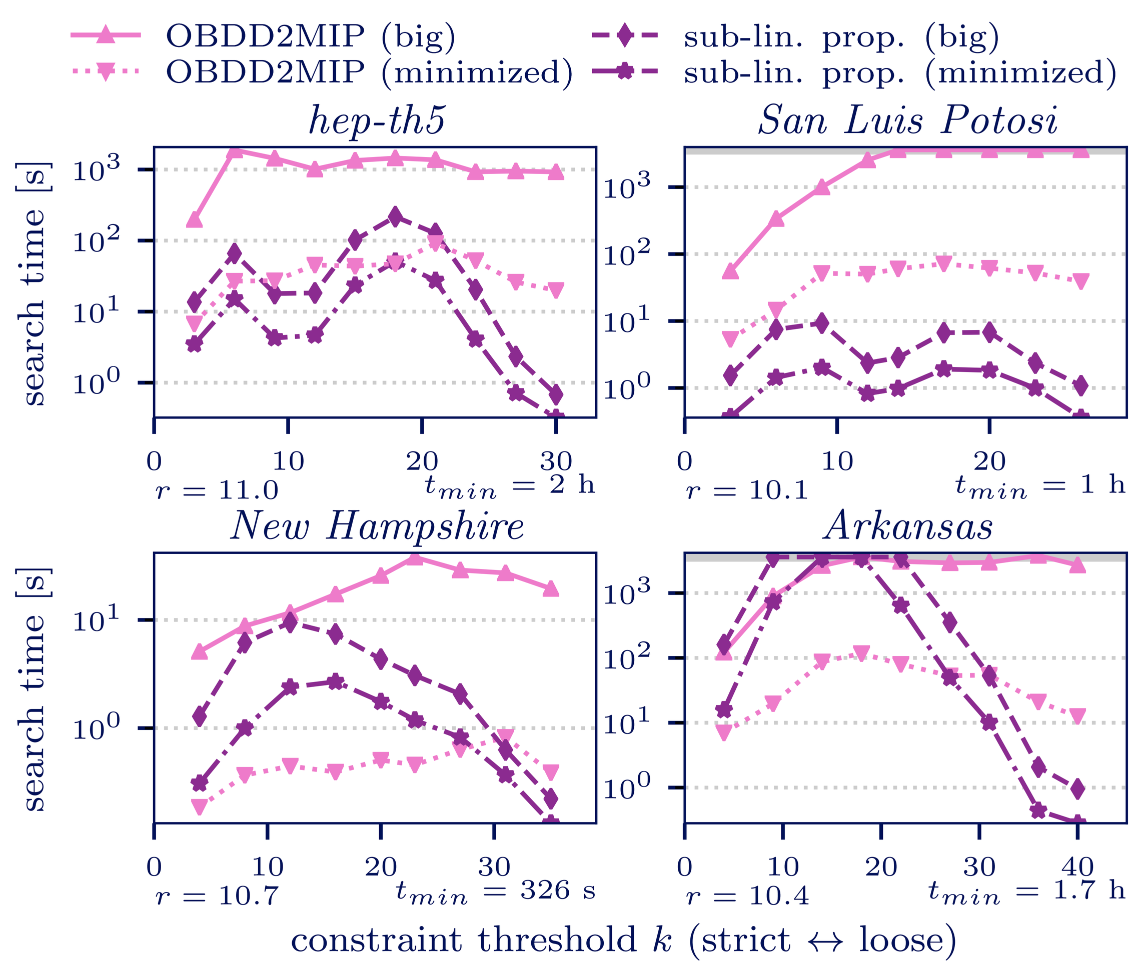
Computing $v_{11} = P(\phi \mid \sigma)$ is $O(|\text{OBDD}|)$. This is prohibitive if we have to do it an exponential number of times. We evaluate $P(\phi \mid \sigma) \geq \theta$ for many different σs.

Experimental evaluation

We compare the performance of our new **linear** and **sub-linear** global propagators to existing CP-based (one that guarantees GAC and one that does not) and MIP-based decomposition methods on various benchmarks.



We investigate how well our **sub-linear** global propagator scales with increasing OBDD size and compare this to how well an existing MIP-based decomposition method scales with OBDD size. In this figure 'big' means that the OBDD is not minimized.



Step 4: solve with SCMD propagator

Idea: exploit monotonicity to determine which decision variables **must be** $\top$ in order to satisfy eq. (2) (**contribution**).

Naïve SCMD propagator on OBDD (**contribution**):

- Define σ' as an optimistic extension of partial strategy σ, where all unbound decision variables are $\top$;
- For each unbound decision variable d, evaluate $P(\phi \mid \sigma' \setminus d, d = \bot)$ with **one sweep** of the OBDD **per free decision variable**;
- If $P(\phi \mid \sigma' \setminus d, d = \bot) < \theta$, fix d to $\top$ (this guarantees GAC).

This algorithm has time complexity $O(nm)$, with n the number of unbound decision variables and $m = |\text{OBDD}|$.

Smarter SCMD propagator on OBDD (**contribution**):

- Use **derivatives** of unbound decision variables to evaluate $P(\phi \mid \sigma' \setminus d, d = \bot)$;
- Compute these derivatives for **all** unbound decision variables **simultaneously** with **two sweeps** of the OBDD.

Resulting algorithm has (**linear**) time complexity $O(n + m)$. **Partial sweeps** can make the algorithm more efficient in practice. Space complexity is lower than GAC-guaranteeing version of decomposition-based method.

[1] Universiteit Leiden The Netherlands
[2] Polytechnique Montréal
[3] UCLouvain

References

A. Darwiche, *On the Tractable Counting of Theory Models and its Application to Belief Revision and Truth Maintenance.* Journal of Applied Non-Classical Logics, 2001

A.L.D. Latour, B. Babaki, A. Dries, A. Kimmig, G. Van den Broeck, S. Nijssen, *Combining Stochastic Constraint Optimization and Probabilistic Programming.* CP, 2017

This work was supported by the Netherlands Organisation for Scientific Research (NWO). Behrouz Babaki is supported by a postdoctoral scholarship from IVADO through the Canada First Research Excellence Fund (CFREF) grant.

code & info
github.com/latower/SCMD
ada.liacs.nl/scop-solver

contact
Anna Louise Latour
www.annalatour.nl
a.l.d.latour@liacs.leidenuniv.nl