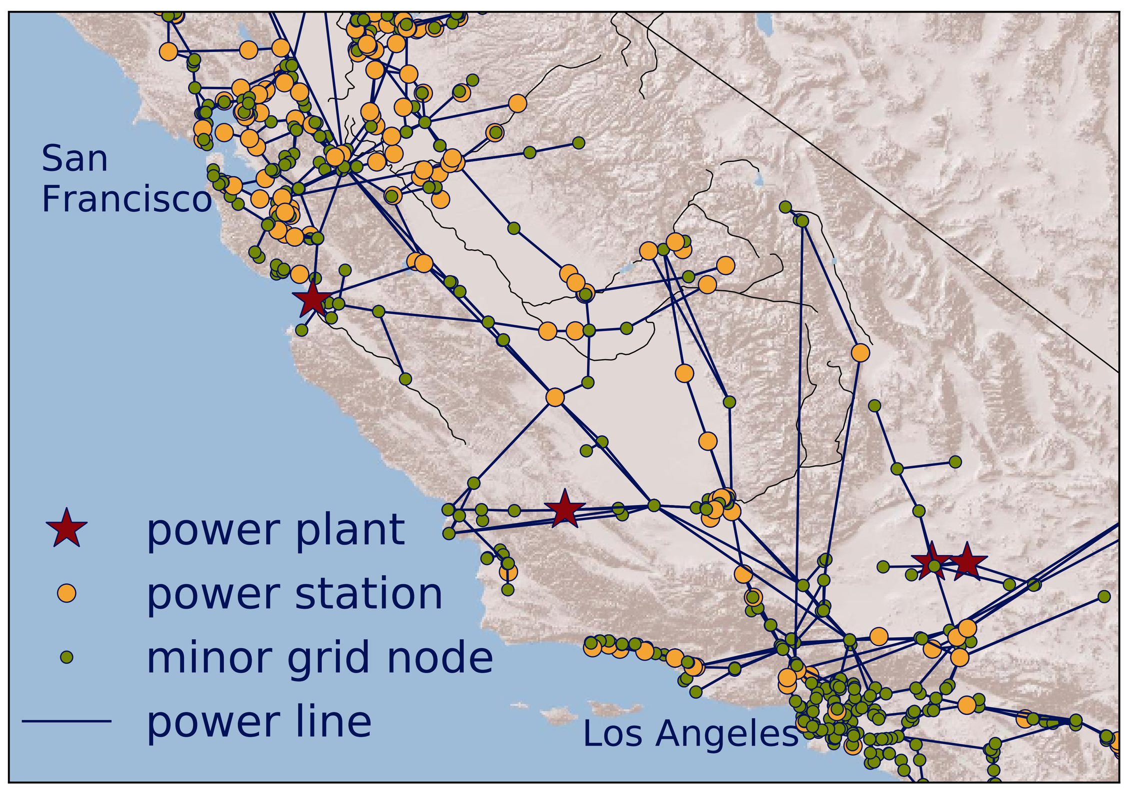
Optimal decision making under constraints and uncertainty

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

The powergrid reliability problem (PRP)

Natural disasters may cause power lines to break. Reinforcing power lines increases the probability that they remain intact.



example of a **Stochastic Constraint Optimisation Problem** (**SCOP**):

probability: chance determines if a line remains intact;
constraint: guarantees on expectation;
optimal decision making: minimise costs.

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

which satisfies $\sum_{i \in \text{buildings}} P(\phi_i \mid \sigma) \geq k,$

while minimising $\sum_{(u,v) \in \text{lines}} c_{uv} d_{uv}.$

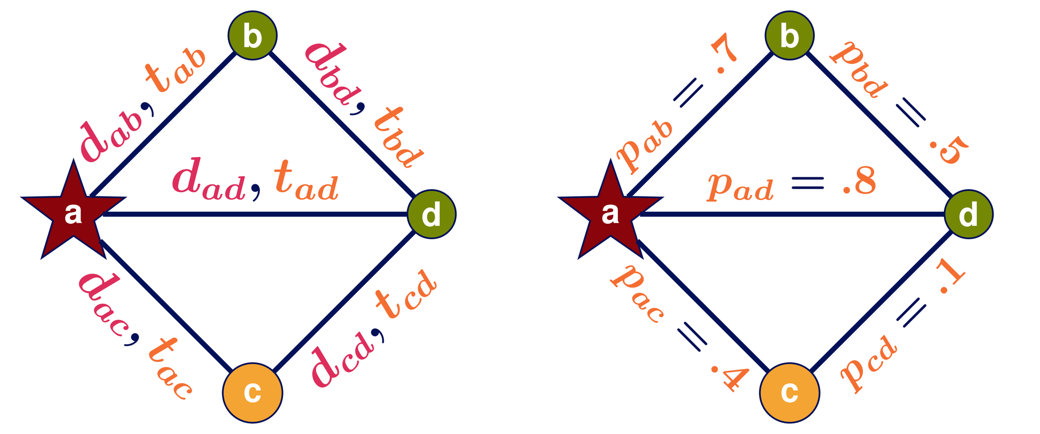
Question: Which power lines do we **choose** to reinforce, to **guarantee** that the **expected number of buildings** still connected to a power plant after a disaster is **at least** k, and **minimise** the **costs**?

- ϕ_i building i is connected to at least one power plant
- k minimum expected number of connected buildings
- c_{uv} **cost** of reinforcing power line (u, v)
- $d_{uv} = 1$ if we reinforce line (u, v) and 0 otherwise

Step 1: create logical model of PRP

Two simplifications for this poster:

- $c_{uv} = 1$ for any line (u, v)
- the **probability** that line (u, v) remains intact is $\pi_{uv} \begin{cases} p_{uv} & \text{if reinforced} \\ 0 & \text{otherwise} \end{cases}$



For line (u, v), introduce two **variables**:

$d_{uv} \in \{0, 1\}$ (by **decision**),
$t_{uv} \in \{0, 1\}$ (by **chance** p_{uv}).

(u, v) is intact post-disaster iff $d_{uv} \wedge t_{uv} = \top$ holds.

Step 2: define stochastic constraint

For simplicity of this poster, suppose we just want to solve

$$P(\phi_{ac} \mid \sigma) \geq \theta, \quad (1)$$

where $0 < \theta \leq 1$, c is connected to a iff $\phi_{ac} = \top$ holds and where

$$\phi_{ac} = (d_{ac} \wedge t_{ac}) \vee (d_{ad} \wedge t_{ad} \wedge d_{cd} \wedge t_{cd}) \vee (d_{ab} \wedge t_{ab} \wedge d_{bd} \wedge t_{bd} \wedge d_{cd} \wedge t_{cd}) \quad (2)$$

Exact solving eq. (1) requires

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

Background: Weighted Model Counting

Given a **strategy** $\sigma = \{d_{ab}, d_{ad}, d_{bd}, d_{cd}\}$ $(d_{ab} = d_{ad} = d_{bd} = d_{cd} = \top, d_{ac} = \bot)$.

Sum the weights of the models (solutions) of $\phi_{ac} \mid \sigma = (t_{ad} \wedge t_{cd}) \vee (t_{ab} \wedge t_{bd} \wedge t_{cd})$:

model	weight
$\{t_{ad}, t_{cd}\}$	$.3 \cdot .6 \cdot .8 \cdot .5 \cdot .1 = .0072$
$\{t_{ac}, t_{ad}, t_{cd}\}$	$.3 \cdot .4 \cdot .8 \cdot .5 \cdot .1 = .0048$
$\{t_{ab}, t_{ad}, t_{bd}, t_{cd}\}$	$.7 \cdot .4 \cdot .8 \cdot .5 \cdot .1 = .0112$
⋮	⋮
	$P(\phi_{ac} \mid \sigma) = .087$

Background: SCOPs are hard

Problem:

- ▲ **WMC** is **#P-complete** (at least as hard as NP);
- ■ **Exponential** number of possible **strategies**.

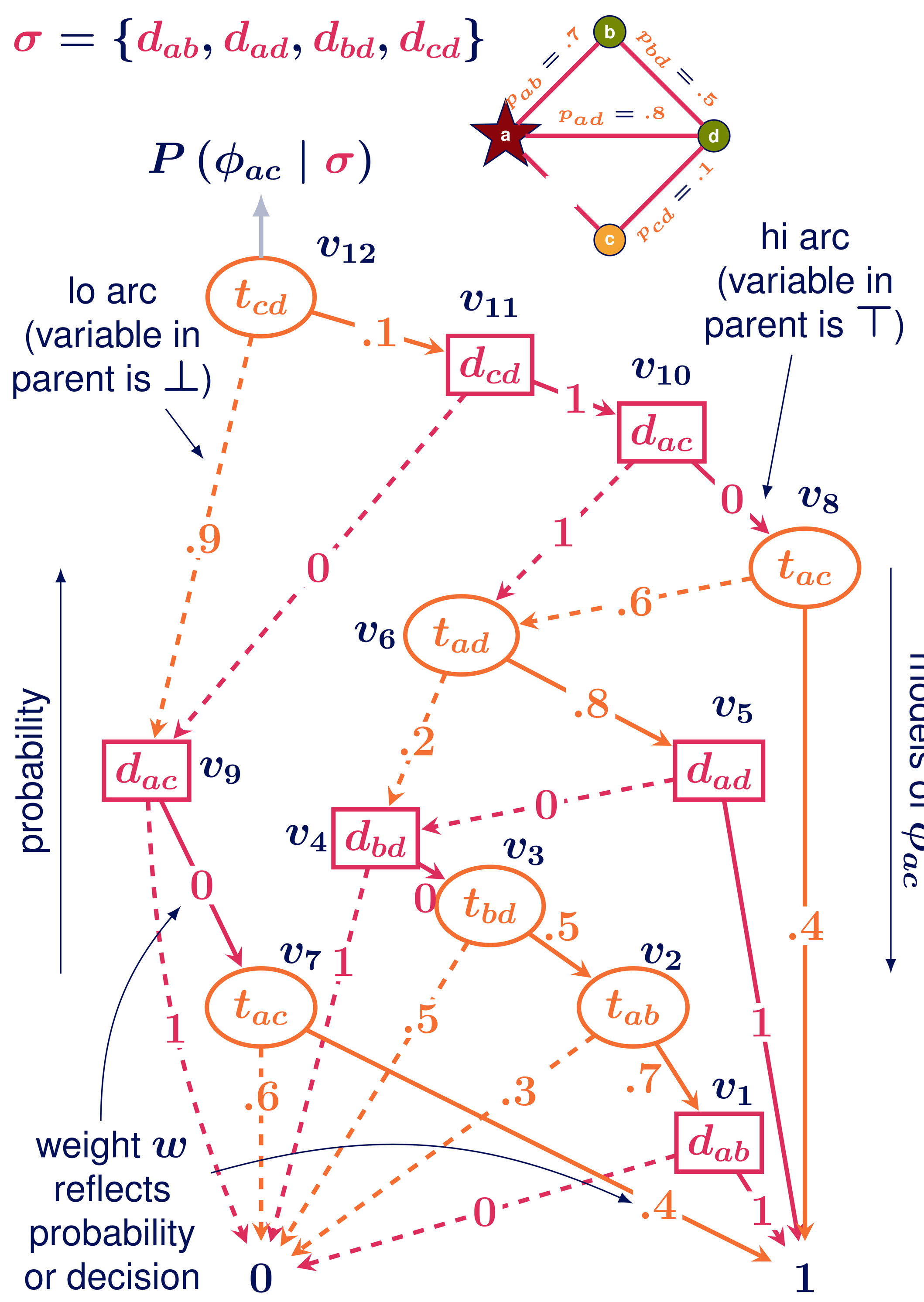
Naïve enumeration and evaluation does not scale.

Our approach:

- ▲ **Compile** ϕ to Ordered Binary Decision Diagram (**OBDD**) for tractable **WMC**;
- ■ Use Constraint Programming (**CP**) technology to efficiently traverse **search space**.

Step 3: use OBDD to evaluate strategy

$\sigma = \{d_{ab}, d_{ad}, d_{bd}, d_{cd}\}$



OBDD is summary of truth table of eq. (2). Paths from root to leaf 1 represent **models** of ϕ_{ac}.

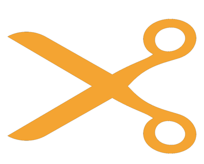
Upward sweep: for each OBDD node r, compute score:

$$v_r = w \cdot v_{hi} + (1 - w) \cdot v_{lo}. \quad (3)$$

Computing $v_{12} = P(\phi \mid \sigma)$ is $O(|\text{OBDD}|)$ (**linear** instead of **exponential**).

Step 4: decompose OBDD and solve

Equation (1) is a **global constraint** on OBDD. Cut it up into **local constraints** using eq. (3). **Solve** with CP solver.



$\theta \leq .1 \cdot v_{11} + .9 \cdot v_9$
$v_{11} = d_{cd} \cdot v_{10} + (1 - d_{cd}) \cdot v_9 \qquad 0 \leq v_{11} \leq .93$
⋮ ⋮
$v_2 = .7 \cdot v_1 \qquad 0 \leq v_2 \leq .7$
$v_1 = d_{ab}$
$d_{ab}, d_{ac}, d_{ad}, d_{bd}, d_{cd} \in \{0, 1\}$

Problem: solver does not always detect when a **decision variable must be** $\top$.
Local constraints are **not Domain Consistent**.

our methods allow us to **solve** Stochastic Constraint Optimisation Problems **exactly**, with increasing **efficiency**

Alternative step 4: domain consistent global constraint on OBDD

Goal: Create **global constraint** that is domain consistent.

Method: Detect which power lines are **crucial** and **must be** reinforced to satisfy eq. (1).

Observe: reinforcing more lines cannot decrease $P(\phi_{ac})$. Use this for **global constraint propagation algorithm** that **guarantees domain consistency**.

Approach: in each node of search tree, **propagate** current **partial strategy** σ':

- select **unbound** d;
- construct **optimistic strategy** $\sigma'_\top$: extend σ' with $d' = \top$ for each **unbound** d';
- if $P(\phi \mid \sigma'_\top \setminus \{d\}, d = \bot) \not\geq \theta$: update $\sigma' \leftarrow \sigma' \cup \{d = \top\}$.

Optimal search space **pruning**, contrary to decomposition.

Problem: complexity is $O(mn)$ with $m = |\text{OBDD}|$ and n the number of **unbound decision variables**.

Solution: Incrementally compute **global** change in score for a change in **decision variable** d, using **local** information:

$$\frac{\partial v_{root}(d)}{\partial d} = \sum_{r_d \in \text{OBDD}_d} \rho(r_d) \left[v(\text{hi child of } r_d) - v(\text{lo child of } r_d) \right] = P(\phi \mid \sigma'_\top) - P(\phi \mid \sigma'_\top \setminus \{d\}, d = \bot) \quad (4)$$

- OBDD_d all OBDD nodes labelled with d
- $\rho(r_d)$ computed with **downward sweep** of OBDD
- $v(r_d)$ computed with eq. (3) **upward sweep** of OBDD.

Only two sweeps of OBDD needed to compute $\rho(r_d)$ and $v(r_d)$ **for each unbound** d.

Two sweeps are enough to compute eq. (4) for each d.

Complexity is $O(m + n)$.

Universiteit Leiden
The Netherlands

[1]Universiteit Leiden
[2]Polytechnique Montréal
[3]Université catholique de Louvain

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

A.L.D. Latour, B. Babaki, S. Nijssen, *Stochastic Constraint Optimization using Propagation on Ordered Binary Decision Diagrams.* StarAI, 2018

GridKit by SciGrid. github.com/bdw/GridKit, scigrid.de, 2016

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

This research was supported by NWO.