Combining Stochastic Constraint Optimization and Probabilistic Programming

From Knowledge Compilation to Constraint Solving

Anna Latour, LIACS, Leiden University.

Universiteit Leiden The Netherlands

KU LEUVEN

UCL Université catholique de Louvain

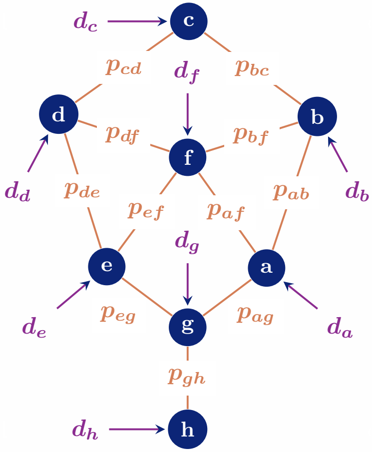
CP2017, Melbourne

Authors

Anna Latour, Leiden University
Behrouz Babaki, KU Leuven
Anton Dries, KU Leuven
Angelika Kimmig, KU Leuven
Guy Van den Broeck, UCLA
Siegfried Nijssen, UC Louvain

Examples of Stochastic Constraint Optimization Problems

Example problem: Viral Marketing



Targeting budget = θ

Decision to target person i directly = $d_i \in \{0, 1\}$

Expected number of people buying = $\mathbb{E}$

SCOP: **who** do we **target directly** such that $\mathbb{E}$ is **maximized** and $\sum_i d_i \leq \theta$?

D. Kempe, J. Kleinberg, É. Tardos. *"Maximizing the Spread of Influence Through a Social Network."* ACM KDD 2003.

Example problem: Theory Compression

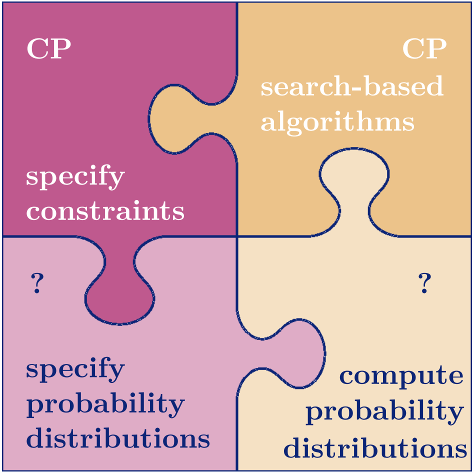
Ourfali et al., *"SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments."* Bioinformatics, 2007

Using CP to solve Stochastic Constraint Optimization Problems

Why use CP for these problems?

- They are **discrete constraint optimization** problems
- They represent a whole **class** of **similar problems**, obtainable by changing constraints and optimization criteria
- CP allows separations of the **modeling** and **solving** of these problems.

General SCOP solving method



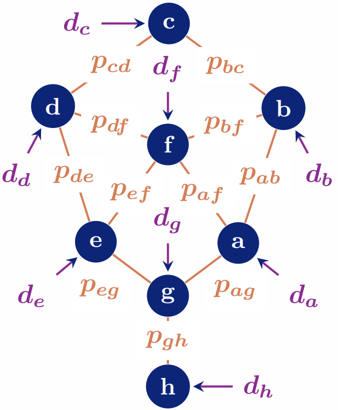
Probabilistic Logic Programming

(modeling)

L. De Raedt et al. *"ProbLog: A probabilistic Prolog and its application in link discovery." IJCAI, 2007*

G. Van den Broeck et al. *"DTProbLog: A decision-theoretic probabilistic Prolog."* AAAI, 2010

Modeling Viral Marketing with ProbLog

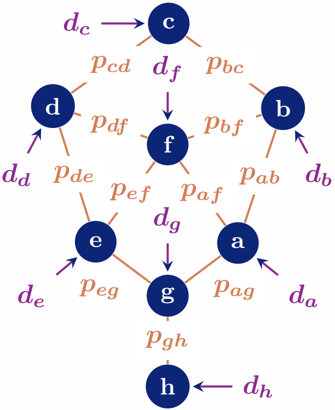



```
% Background knowledge
person(a). person(c).
person(b). person(d).   ...

% Probabilistic facts
0.7::directed(a,b).
0.4::directed(d,f).   ...

% Decision variables
?::marketed(P) :- person(P).

% Relations
trusts(X,Y) :- directed(X,Y).
trusts(Y,X) :- directed(X,Y).
buys(X) :- marketed(X).
buys(X) :- trusts(X,Y), buys(Y).

% Queries
query(buys(a)). query(buys(c)).
query(buys(b)). query(buys(d)).   ...
```

Stochastic Constraint

Probabilistic Logic Programming

CP + ProbLog = SC-ProbLog




```
% Background knowledge
person(a). person(c).
person(b). person(d).   ...

% Probabilistic facts
0.7::directed(a,b).
0.4::directed(d,f).   ...

% Decision variables
?::marketed(P) :- person(P).

% Relations
trusts(X,Y) :- directed(X,Y).
trusts(Y,X) :- directed(X,Y).
buys(X) :- marketed(X).
buys(X) :- trusts(X,Y), buys(Y).

% SCOP
{marketed(P) => 1 :- person(P).} 4.
#maximize{buys(P) => 1 :- person(P).}
```

CONTRIBUTION 1

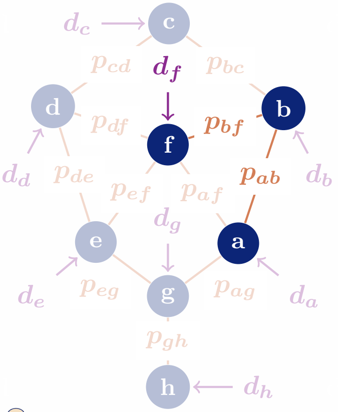
Solving with SC-ProbLog

Part A: computing probabilities
Part B: solving SCOPs the naive way
Part C: solving SCOPs with CP

L. De Raedt et al. *"ProbLog: A probabilistic Prolog and its application in link discovery." IJCAI, 2007*

G. Van den Broeck et al. *"DTProbLog: A decision-theoretic probabilistic Prolog."* AAAI, 2010

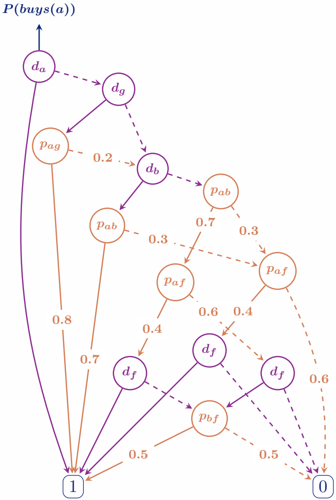
Part A: Ground program for each query

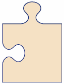


$$
\begin{aligned}
&P(buys(a)) = \\
&P(d_a \lor \\
&\quad (d_b \land p_{ab}) \lor \\
&\quad (d_f \land p_{af}) \lor \\
&\quad (d_g \land p_{ag}) \lor \\
&\quad (d_b \land p_{bf} \land p_{af}) \lor \\
&\quad (d_f \land p_{bf} \land p_{ab}) \lor \ldots)
\end{aligned}
$$

Part A: Compile to Decision Diagram

$$
\begin{aligned}
&P(buys(a)) = \\
&P(d_a \lor \\
&\quad (d_b \land p_{ab}) \lor \\
&\quad (d_f \land p_{af}) \lor \\
&\quad (d_g \land p_{ag}) \lor \\
&\quad (d_b \land p_{bf} \land p_{af}) \lor \\
&\quad (d_f \land p_{bf} \land p_{ab}) \lor \ldots)
\end{aligned}
$$

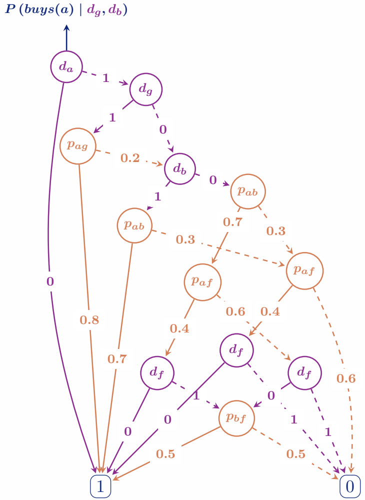




Part A: enumerate all strategies

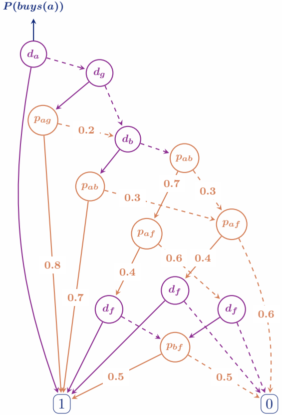
Weighted Model Counting:

$$
\begin{aligned}
&\boldsymbol{P(buys(a) \mid d_g, d_b)}\\
&\boldsymbol{P}(\bot \lor\\
&\quad (\top \land p_{ab}) \lor\\
&\quad (\bot \land p_{af}) \lor\\
&\quad (\top \land p_{ag}) \lor\\
&\quad (\top \land p_{bf} \land p_{af}) \lor\\
&\quad (\bot \land p_{bf} \land p_{ab}) \lor \ldots)
\end{aligned}
$$

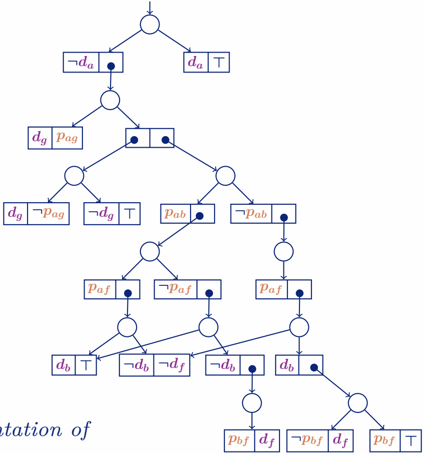


Part A: Sentential Decision Diagrams (SDDs)

Binary Decision Diagram



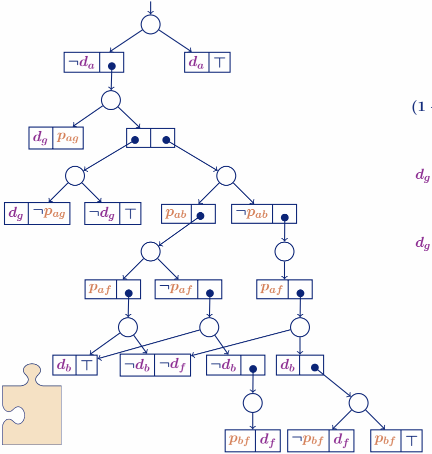
Sentential Decision Diagram



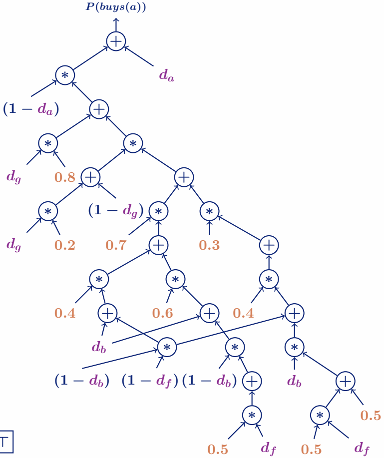
A. Darwiche. *“SDD: A New Canonical Representation of Propositional Knowledge Bases.”* AAAI 2011.

Part A: From SDD to Arithmetic Circuit

Sentential Decision Diagram



Arithmetic Circuit



Part B: Naive Solving

For each strategy $\boldsymbol{\sigma}$, the objective value for the Viral Marketing problem evaluates to:

$$\sum_{i} P(buys(i) \mid \sigma)$$

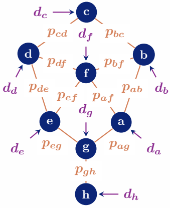
and the constraint is

$$|\sigma| \leq \theta$$

Simply **enumerate** and evaluate **all strategies** to solve the problem

Remark: ProbLog does not support this

Part B: Naive method summary

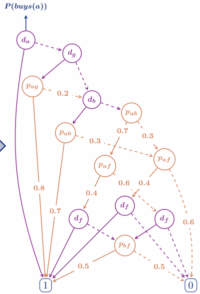


model

```
% Background knowledge
person(a). person(c).
person(b). person(d). ...

% Probabilistic facts
0.7::directed(a,b).
0.4::directed(d,f). ...

% Decision variables
?::marketed(P) :- person(P).

% Relations
trusts(X,Y) :- directed(X,Y).
trusts(Y,X) :- directed(X,Y).
buys(X) :- marketed(X).
buys(X) :- trusts(X,Y), buys(Y).

% Queries
query(buys(a)). query(buys(c)).
query(buys(b)). query(buys(d)). ...
```

ground

(for each query $\boldsymbol{q_i}$)



enumerate

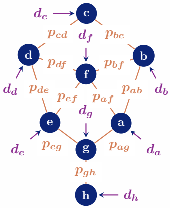
For each strategy $\boldsymbol{\sigma}$

compute $\boldsymbol{P(q_i \mid \sigma)}$ for each $\boldsymbol{q_i}$

and evaluate $\sum_i \boldsymbol{P(q_i \mid \sigma)}$

if $|\boldsymbol{\sigma}| \leq \boldsymbol{\theta}$

Part C: CP + ProbLog = SC-ProbLog

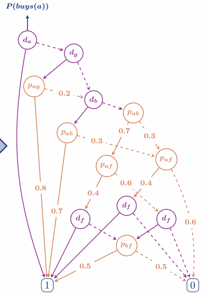


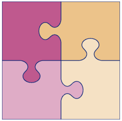
model

```
% Background knowledge
person(a). person(c).
person(b). person(d). ...

% Probabilistic facts
0.7::directed(a,b).
0.4::directed(d,f). ...

% Decision variables
?::marketed(P) :- person(P).

% Relations
trusts(X,Y) :- directed(X,Y).
trusts(Y,X) :- directed(X,Y).
buys(X) :- marketed(X).
buys(X) :- trusts(X,Y), buys(Y).

% Queries
query(buys(a)). query(buys(c)).
query(buys(b)). query(buys(d)). ...
```

ground

(for each query q_i)





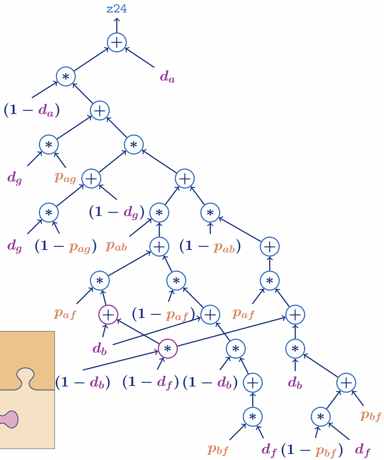
encode

Mixed-Integer Problem

Solve with CP/MIP solver

Part C: AC to Mixed Integer Problem

Arithmetic Circuit



MIP

```
% Objective Function:
maximize(z24 + ... )

% Constraint:
da + ... + dg <= theta

z24 = z23 + da
z23 = (1-da) * z22
z22 = z19 * z20
z21 = dg * pag
z20 = z14 + z15
z19 = z16 + (1-dg)
z18 = (1-pab) * z15
z17 = pab * z14
z16 = dg * (1-pag)
z15 = z13
z14 = z11 + z12
z13 = paf * z10

z12 = (1-paf) * z9
z11 = paf * z8
z10 = z5 + z7
z9 = db + z6
z8 = db + z5
z7 = (1-db) * z4
z6 = (1-db) * z3
z5 = (1-db) * (1-df)
z4 = z2 + pbf
z3 = z1
z2 = (1-pbf) * df
z1 = pbf * df
```

And solve with off-the-shelf solver…

Part C: Linearizability of SDDs

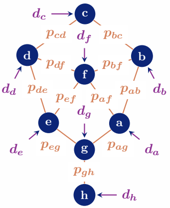
Typically, SDDs yield **quadratic** MIPs
SDDs with **special property** yield **linear** MIPs

Smaller SDDs yield smaller MIPs
Minimization typically **destroys** the special property in SDDs that makes them **linear**

Solution: custom minimization algorithm that **preserves linearity**

A. Choi, A. Darwiche. *"Dynamic Minimization of Sentential Decision Diagrams."* AAAI 2013.

Part C: SC-ProbLog Summary

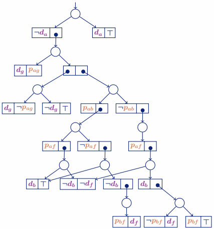


model

```
% Background knowledge
person(a). person(c).
person(b). person(d). ...

% Probabilistic facts
0.7::directed(a,b).
0.4::directed(d,f). ...

% Decision variables
?::marketed(P) :- person(P).

% Relations
trusts(X,Y) :- directed(X,Y).
trusts(Y,X) :- directed(X,Y).
buys(X) :- marketed(X).
buys(X) :- trusts(X,Y), buys(Y).

% Queries
query(buys(a)). query(buys(c)).
query(buys(b)). query(buys(d)). ...
```

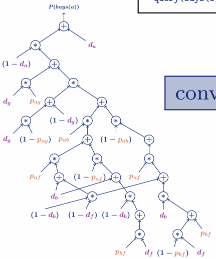
ground

(for each query q_i)



get AC

(for each query q_i)

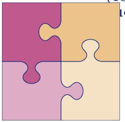


convert

```
% Objective Function:
maximize(z24 + ... )

% Constraint:
da + ... + dg <= theta

z24 = z23 + da
z23 = (1-da) * z22
z22 = z19 * z20
z21 = dg * pag
z20 = z14 + z15
z19 = z16 + (1-dg)
z18 = (1-pab) * z15
z17 = pab * z14
z16 = dg * (1-pag)
z15 = z13
z14 = z11 + z12
z13 = paf * z10
z12 = (1-paf) * z9
z11 = paf * z8
z10 = z5 + z7
z9 = db + z6
z8 = db + z5
z7 = (1-db) * z4
z6 = (1-db) * z3
z5 = (1-db) * (1-df)
z4 = z2 + pbf
z3 = z1
z2 = (1-pbf) * df
z1 = pbf * df
```

SOLVE

(with off-the-shelf CP solver)



Experiments & results

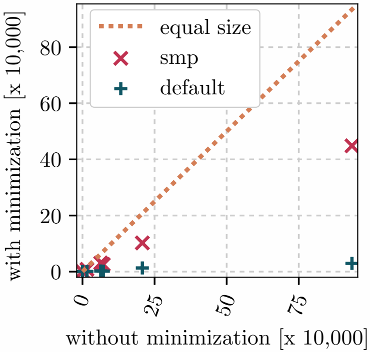
Experiments

For the experiments, we evaluate performance for:

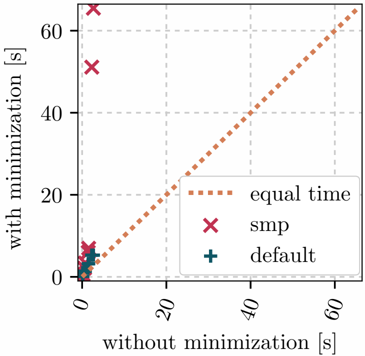
- **Sentential Decision Diagrams** (SDDs) **only**, no Ordered Binary Decision Diagrams (OBDDs)
- `Gurobi` (MIP solver) and `Gecode` (CP solver)
- **custom** minimization algorithm and **default** minimization algorithm

Results I

How do the **sizes** of the SDDs obtained compare to each other?



How does minimization influence **compilation time**?



Results II

When looking at the **total solving time** (in seconds), what is the best strategy?

instance		*size*		Gurobi		Gecode	
		n_d	n_q	no mini	smp mini	no mini	default mini
viral marketing	**setting 1**	20	20	545.8	412.7	t/o	**130.9**
viral marketing	**setting 2**	20	20	188.6	163.8	2859.9	**6.9**
viral marketing	**setting 1**	33	10	2076.8	**1185.7**	t/o	t/o
viral marketing	**setting 2**	33	10	364.6	**346.4**	t/o	t/o
theory compression	**setting 1**	36	23	3.9	**3.4**	1389.5	591.4
theory compression	**setting 2**	36	23	4.1	**3.9**	70.9	31.4
theory compression	**setting 1**	76	13	5.9	**5.6**	t/o	t/o
theory compression	**setting 2**	76	13	**4.7**	5.7	t/o	1878.2
theory compression	**setting 3**	86	26	**443.2**	471.3	t/o	t/o
theory compression	**setting 4**	71	13	23.3	21.9	222.9	**8.6**

Contributions

1. Extension of ProbLog to **SC-ProbLog**
2. **Custom SDD minimization** algorithm for producing linear MIPs
3. Proposal and implementation **SCOP solving toolchain**

Conclusion & Future work

While results are encouraging, it remains a **challenge** to solve these SCOPs on **larger networks**.

We believe that our **custom SDD minimization** algorithm can also be **applied** in **other contexts**.

Interested? Find **code** at

`https://bitbucket.org/antondries/problog/branch/sc-problog`,

or **e-mail** us at

`a.l.d.latour@liacs.leidenuniv.nl`

Acknowledgements

We thank **Luc De Raedt** for his support, for his advice and for the numerous other ways in which he contributed to this work.

This research was supported by the Netherlands Organisation for Scientific Research (**NWO**) and **NSF** grant #IIS-1657613.

References I

D. Kempe, J. Kleinberg, É. Tardos.
"Maximizing the Spread of Influence Through a Social Network."
ACM KDD 2003

Oved Ourfali, Tomer Shlomi, Trey ideker, Eytan Ruppin, Roded Sharan.
"SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments."
Bioinformatics, 2007

Luc De Raedt, Angelika Kimmig and Hannu Toivonen.
"ProbLog: A Probabilistic Prolog and Its Application in Link Discovery."
IJCAI, 2007

L. De Raedt, K. Kersting, A. Kimmig, K. Revoredo, H. Toivonen.
"Compressing probabilistic Prolog programs"
Machine Learning, 2008

References II

Guy Van den Broeck, Ingo Thon, Martijn Van Otterlo, Luc De Raedt.
"DTProbLog: A decision-theoretic probabilistic Prolog."
Proceedings of AAAI, 2010

Adnan Darwiche.
"SDD: A new canonical representation of propositional knowledge bases."
IJCAI Proceedings, 2011

Arthur Choi, Adnan Darwiche.
"Dynamic minimization of sentential decision diagrams"
AAAI Proceedings, 2013

M.E.J. Newman.
"The Structure of Scientific Collaboration Networks."
Proceedings of the National Academy of Sciences, 2001

Theme by Joost Schalken. Updated by Pepijn van Heiningen & Anna Latour.