Modèles Probabilistes Complexes

Models et Algorithms

Pierre-Henri WUILLEMIN

Christophe Gonzales – Lionel Torti – Morgan Chopin

(prenom.nom@lip6.fr)

8 novembre 2011

Inference in Bayesian networks

Variable eliminitation

Input : a set of CPTs $\mathbf{P}$ and a set of variables $\mathbf{X}$

Output : $P(\mathbf{X})$

$\mathbf{W} \leftarrow$ all the variables of the CPTs of $\mathbf{P}$ except $\mathbf{X}$

while $\mathbf{W} \neq \emptyset$ **do**

  let X_j be a variable in $\mathbf{W}$; remove X_j from $\mathbf{W}$

  let $\mathbf{Q}$ be the set of tables in $\mathbf{P}$ containing X_j

  compute table $q = \sum_{X_j} \prod_{f \in \mathbf{Q}} f$

  $\mathbf{P} \leftarrow (\mathbf{P} \backslash \mathbf{Q}) \cup \{q\}$

return table $\prod_{f \in \mathbf{P}} f$

Dynamic Bayesian networks (dBNs)

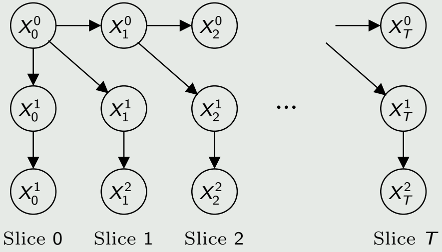
Definition

Bayesian network that models a dynamic system (or a repetition of patterns) :

Dynamic Bayesian networks (dBNs)

Definition

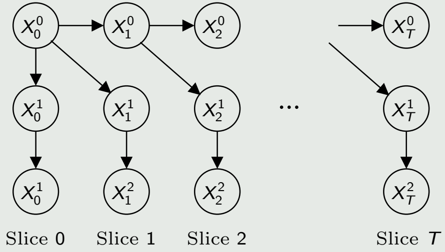
Bayesian network that models a dynamic system (or a repetition of patterns) :



Dynamic Bayesian networks (dBNs)

Definition

Bayesian network that models a dynamic system (or a repetition of patterns) :



- *Transition* model :

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

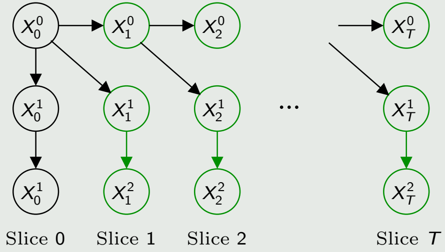
- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

Dynamic Bayesian networks (dBNs)

Definition

Bayesian network that models a dynamic system (or a repetition of patterns) :



- *Transition* model :

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

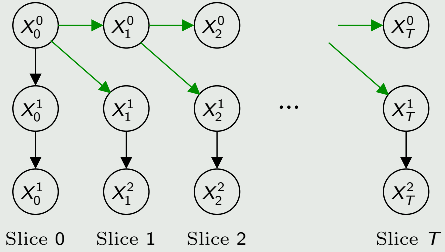
- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

Dynamic Bayesian networks (dBNs)

Definition

Bayesian network that models a dynamic system (or a repetition of patterns) :



- *Transition* model :

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

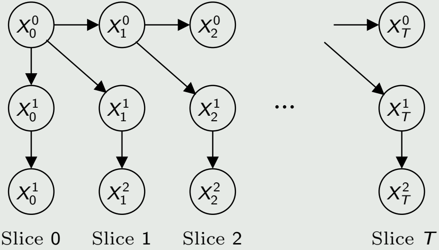
- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

Dynamic Bayesian networks (dBNs)

Limitations

Bayesian network proposes a simplification of the dynamic modelisation :



- *Transition* model :

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

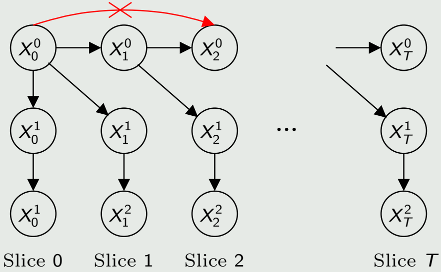
- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

Dynamic Bayesian networks (dBNs)

Limitations

Bayesian network proposes a simplification of the dynamic modelisation :



- *Transition* model :

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

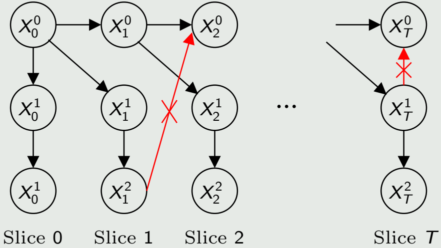
- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

Dynamic Bayesian networks (dBNs)

Limitations

Bayesian network proposes a simplification of the dynamic modelisation :



- *Transition* model :

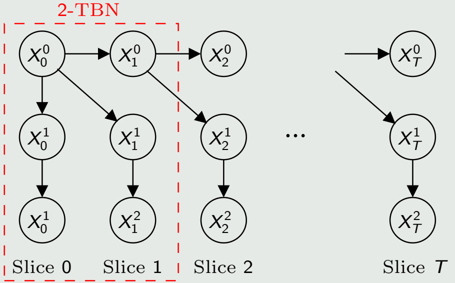
$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

Limitations

Bayesian network proposes a simplification of the dynamic modelisation :



- *Transition* model :

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

DBN triangulation issues

How to triangulate a dBN ?

A dBN is a Bayesian network $\Rightarrow$ triangulation based inference algorithm.

DBN triangulation issues

How to triangulate a dBN ?

A dBN is a Bayesian network $\Rightarrow$ triangulation based inference algorithm.

1. Unroll the 2-TBN to any desired length T and moralize.

DBN triangulation issues

How to triangulate a dBN ?

A dBN is a Bayesian network $\Rightarrow$ triangulation based inference algorithm.

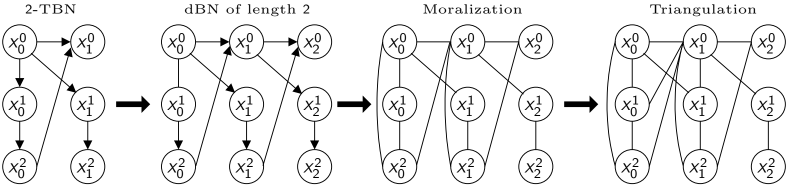
1. Unroll the 2-TBN to any desired length T and moralize.
2. Apply any classical triangulation algorithm.

DBN triangulation issues

How to triangulate a dBN ?

A dBN is a Bayesian network $\Rightarrow$ triangulation based inference algorithm.

1. Unroll the 2-TBN to any desired length T and moralize.
2. Apply any classical triangulation algorithm.

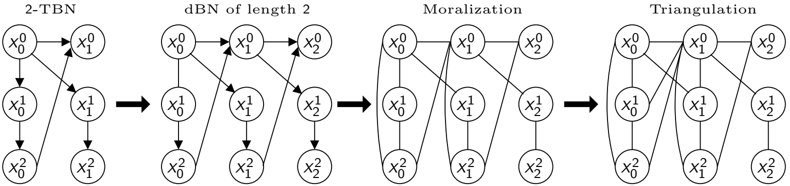


DBN triangulation issues

How to triangulate a dBN ?

A dBN is a Bayesian network $\Rightarrow$ triangulation based inference algorithm.

1. Unroll the 2-TBN to any desired length T and moralize.
2. Apply any classical triangulation algorithm.



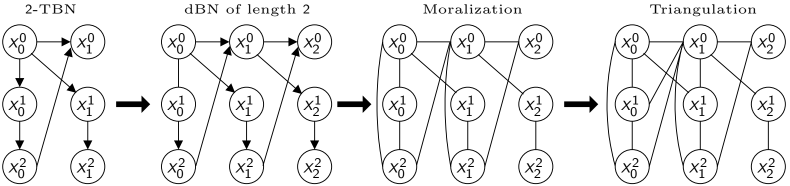
Problems to fix

DBN triangulation issues

How to triangulate a dBN ?

A dBN is a Bayesian network $\Rightarrow$ triangulation based inference algorithm.

1. Unroll the 2-TBN to any desired length T and moralize.
2. Apply any classical triangulation algorithm.



Problems to fix

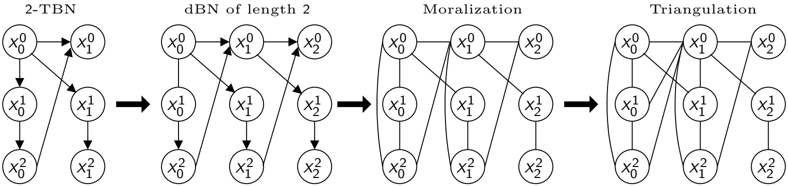
1. The dBN can be arbitrarily large,

DBN triangulation issues

How to triangulate a dBN ?

A dBN is a Bayesian network $\Rightarrow$ triangulation based inference algorithm.

1. Unroll the 2-TBN to any desired length T and moralize.
2. Apply any classical triangulation algorithm.



Problems to fix

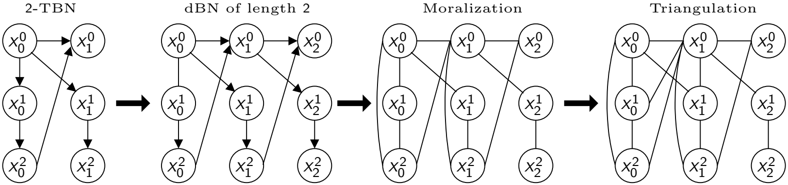
1. The dBN can be arbitrarily large,
2. Need to re-triangulate the dBN if its length changes,

DBN triangulation issues

How to triangulate a dBN ?

A dBN is a Bayesian network ⇒ triangulation based inference algorithm.

1. Unroll the 2-TBN to any desired length T and moralize.
2. Apply any classical triangulation algorithm.



Problems to fix

1. The dBN can be arbitrarily large,
2. Need to re-triangulate the dBN if its length changes,
3. The size of some cliques may be a function of T.

Constrained triangulation : forward

How to provide a way to avoid re-triangulating the dBN ?

Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

Constrained triangulation : forward

How to provide a way to avoid re-triangulating the dBN ?

Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.

Constrained triangulation : forward

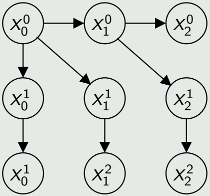
How to provide a way to avoid re-triangulating the dBN ?

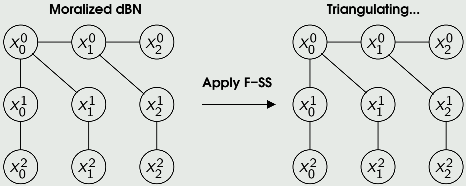
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.





Constrained triangulation : forward

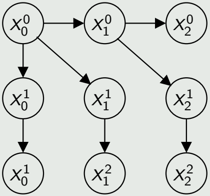
How to provide a way to avoid re-triangulating the dBN ?

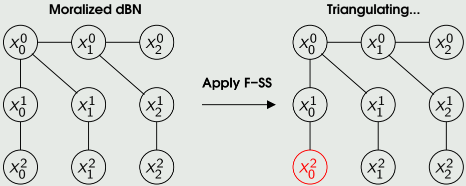
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.





Constrained triangulation : forward

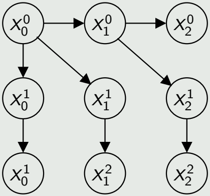
How to provide a way to avoid re-triangulating the dBN ?

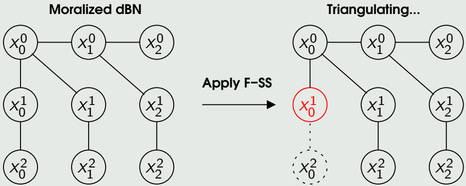
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.





Constrained triangulation : forward

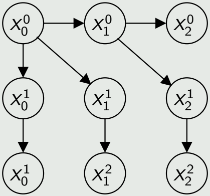
How to provide a way to avoid re-triangulating the dBN ?

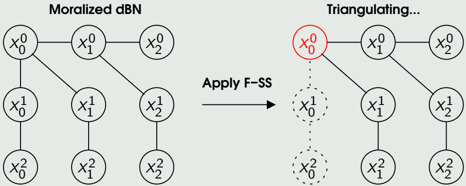
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.





Constrained triangulation : forward

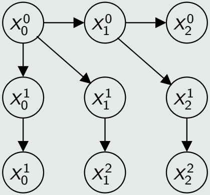
How to provide a way to avoid re-triangulating the dBN ?

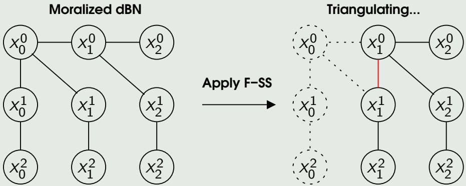
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.





Constrained triangulation : forward

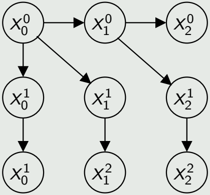
How to provide a way to avoid re-triangulating the dBN ?

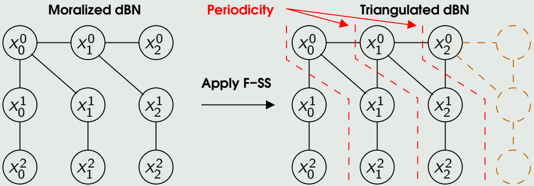
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.





Constrained triangulation : forward

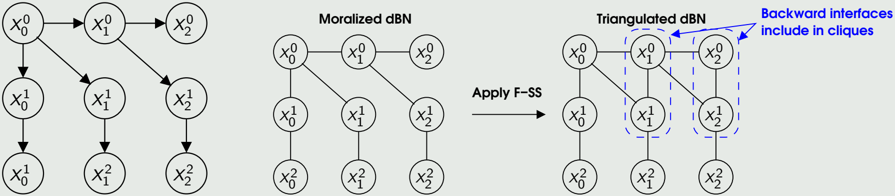
How to provide a way to avoid re-triangulating the dBN ?

Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.



Constrained triangulation : forward

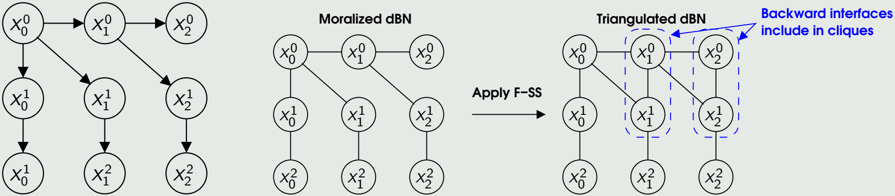
How to provide a way to avoid re-triangulating the dBN ?

Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.



Constrained triangulation : forward

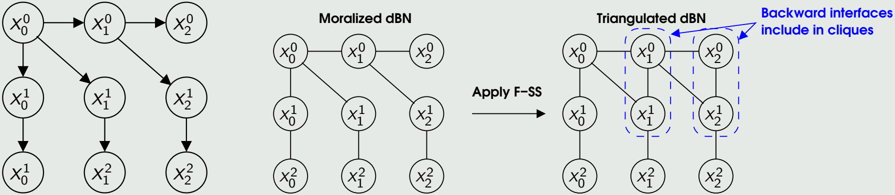
How to provide a way to avoid re-triangulating the dBN ?

Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The forward slice-by-slice elimination (F-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the past to the future.



$\hookrightarrow$ *Backward interface* : nodes with parents or spouses in the previous slice.

Constrained triangulation : backward

How to provide a way to avoid re-triangulating the dBN ?

Constrained triangulation : backward

How to provide a way to avoid re-triangulating the dBN ?

Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

Constrained triangulation : backward

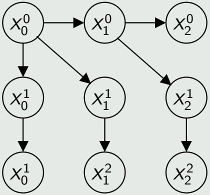
How to provide a way to avoid re-triangulating the dBN ?

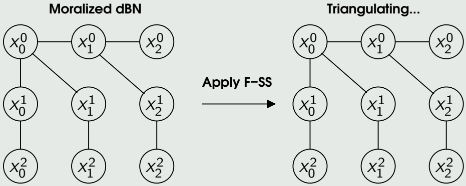
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The backward slice-by-slice elimination (B-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the future to the past :





Constrained triangulation : backward

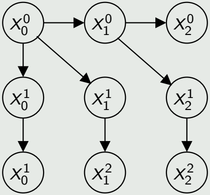
How to provide a way to avoid re-triangulating the dBN ?

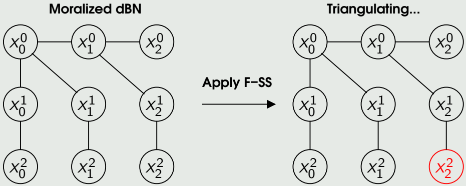
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The backward slice-by-slice elimination (B-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the future to the past :





Constrained triangulation : backward

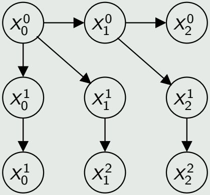
How to provide a way to avoid re-triangulating the dBN ?

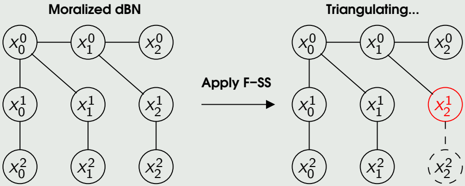
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The backward slice-by-slice elimination (B-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the future to the past :





Constrained triangulation : backward

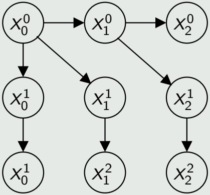
How to provide a way to avoid re-triangulating the dBN ?

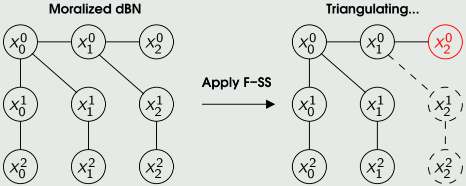
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The backward slice-by-slice elimination (B-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the future to the past :





Constrained triangulation : backward

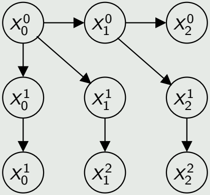
How to provide a way to avoid re-triangulating the dBN ?

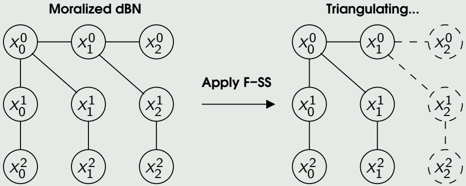
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The backward slice-by-slice elimination (B-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the future to the past :





Constrained triangulation : backward

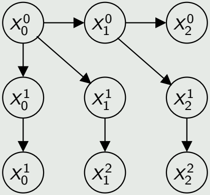
How to provide a way to avoid re-triangulating the dBN ?

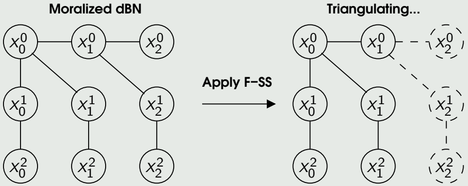
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The backward slice-by-slice elimination (B-SS) [darwiche, 2001]

Consists in eliminating a slice after an other from the future to the past :





Constrained triangulation : backward

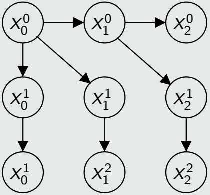
How to provide a way to avoid re-triangulating the dBN ?

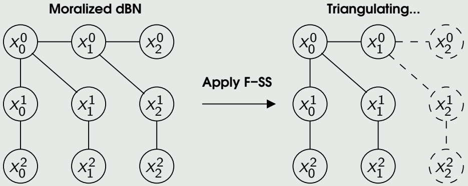
Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order.

The backward slice-by-slice elimination (B-SS) [darwiche, 2001]

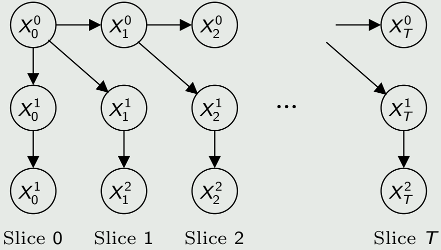
Consists in eliminating a slice after an other from the future to the past :





↪ *Forward interface* : nodes with children in the next slice.

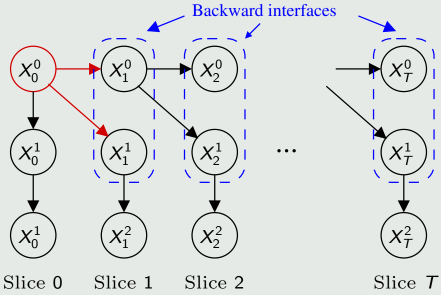
General interface



- *Transition* model :
$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

- *Prior* distribution :
$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

General interface



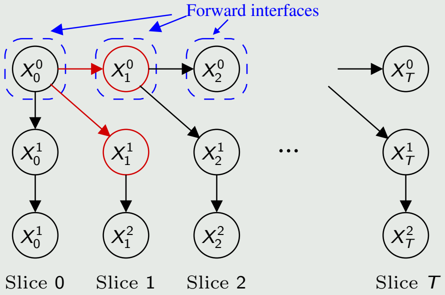
- *Transition* model :

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

General interface



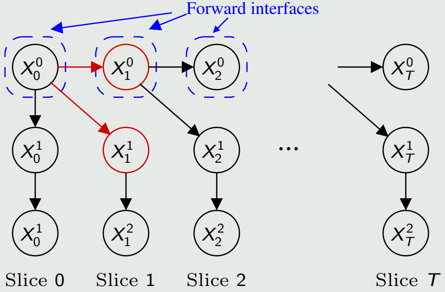
- *Transition* model :

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0)$$
$$= \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

General interface



- *Transition* model :

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{X}_1|\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_1^i|pa(X_1^i))$$

- *Prior* distribution :

$$p(\mathbf{X}_0) = \prod_{i=0}^{n-1} p(X_0^i|pa(X_0^i))$$

Interface

Subset of nodes such that if they were removed they will disconnect the past from the future in the moral graph of a dBN.

Complexity for interface based ceo

Interface based Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order given by an interface.

1. Provide a way to avoid re-triangulating the dBN,

Complexity for interface based ceo

Interface based Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order given by an interface.

1. Provide a way to avoid re-triangulating the dBN,
2. The size ω of the maximal clique is bounded :

Complexity for interface based ceo

Interface based Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order given by an interface.

1. Provide a way to avoid re-triangulating the dBN,
2. The size ω of the maximal clique is bounded :

Lower bound

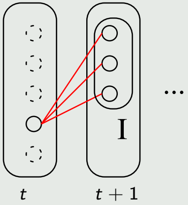
Complexity for interface based ceo

Interface based Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order given by an interface.

1. Provide a way to avoid re-triangulating the dBN,
2. The size ω of the maximal clique is bounded :

Lower bound



Best case : clique of size $|I|+1$

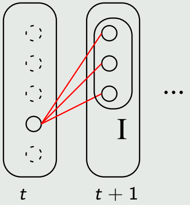
Complexity for interface based ceo

Interface based Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order given by an interface.

1. Provide a way to avoid re-triangulating the dBN,
2. The size ω of the maximal clique is bounded :

Lower bound



Best case : clique of size $|I|+1$

Upper bound

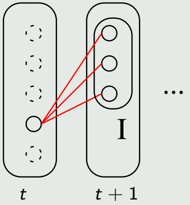
Complexity for interface based ceo

Interface based Constrained elimination order (ceo)

Elimination order such that we enforce nodes to be eliminated with respect to a partial order given by an interface.

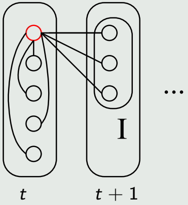
1. Provide a way to avoid re-triangulating the dBN,
2. The size ω of the maximal clique is bounded :

Lower bound



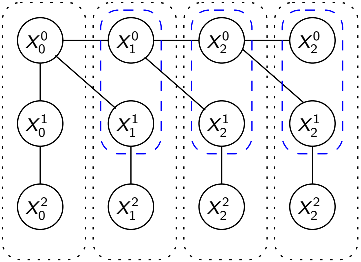
Best case : clique of size $|I| + 1$

Upper bound

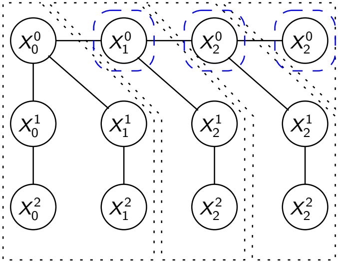


Worst case : clique of size $|I| + S$ (S : $\#$ nodes in a slice)

How to improve the bounds ?

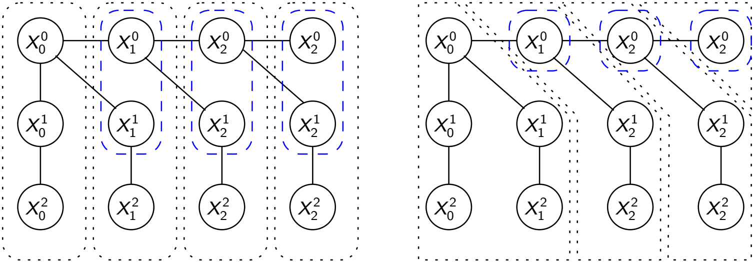


$|I| + 1 = 3 \leq \omega = 3 \leq |I| + N = 5$



$\omega = 2$

How to improve the bounds ?

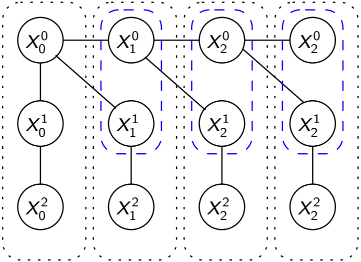


$|I| + 1 = 3 \leq \omega = 3 \leq |I| + N = 5$

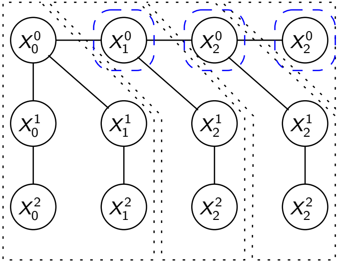
$\omega = 2$

$\hookrightarrow$ Looking for a smaller interface by extending the slice [bilmes et al, 2003]

How to improve the bounds ?



$|I| + 1 = 3 \leq \omega = 3 \leq |I| + N = 5$

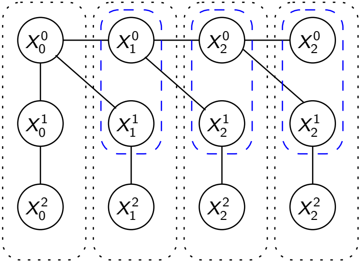


$\omega = 2$

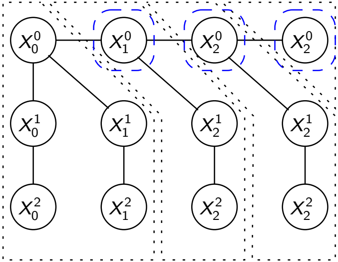
$\hookrightarrow$ Looking for a smaller interface by extending the slice [bilmes et al, 2003]

[Chopin and Wuillemin, 2010]

How to improve the bounds ?



$|I| + 1 = 3 \leq \omega = 3 \leq |I| + N = 5$



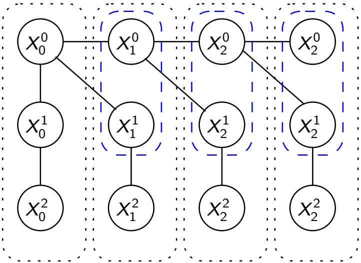
$\omega = 2$

$\hookrightarrow$ Looking for a smaller interface by extending the slice [bilmes et al, 2003]

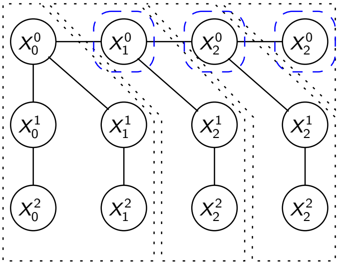
[Chopin and Wuillemin, 2010]

1. Algorithm to find the optimal interface in polynomial time,

How to improve the bounds ?



$|I| + 1 = 3 \leq \omega = 3 \leq |I| + N = 5$



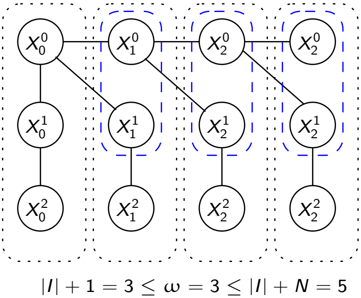
$\omega = 2$

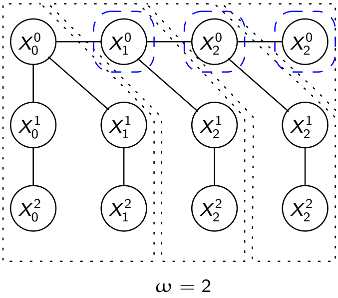
$\hookrightarrow$ Looking for a smaller interface by extending the slice [bilmes et al, 2003]

[Chopin and Wuillemin, 2010]

1. Algorithm to find the optimal interface in polynomial time,
2. Deduce a *ceo* with better theoretical guarantee on cliques size.

How to improve the bounds ?





$\hookrightarrow$ Looking for a smaller interface by extending the slice [bilmes et al, 2003]

[Chopin and Wuillemin, 2010]

1. Algorithm to find the optimal interface in polynomial time,
2. Deduce a *ceo* with better theoretical guarantee on cliques size.

Relation to the MINIMUM s–t CUT problem

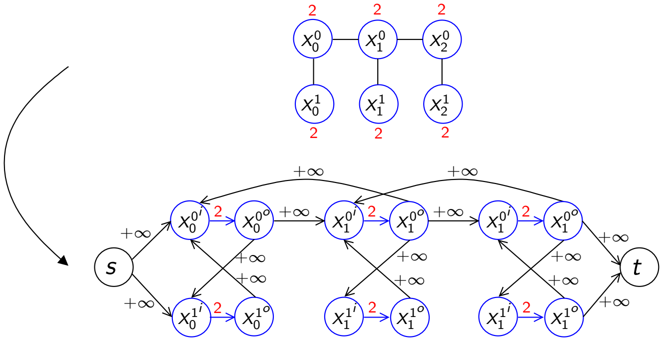
Theorem

Given a dBN of length T, finding a minimum interface is equivalent to solve the minimum s–t cut problem

Relation to the MINIMUM s–t CUT problem

Theorem

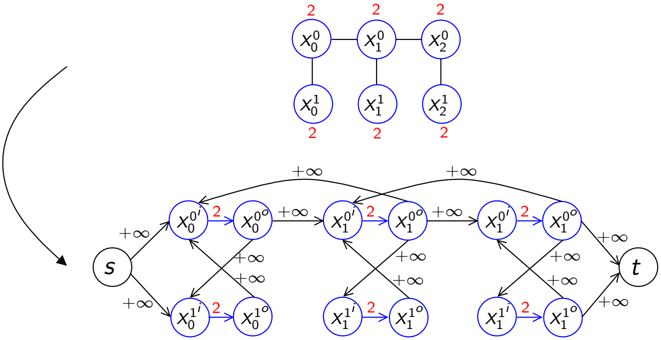
Given a dBN of length T, finding a minimum interface is equivalent to solve the minimum s–t cut problem



Relation to the MINIMUM s–t CUT problem

Theorem

Given a dBN of length T, finding a minimum interface is equivalent to solve the minimum s–t cut problem



$\Rightarrow$ Polynomial time algorithm to find the minimum interface in a dBN of length T

Building the optimal elimination order

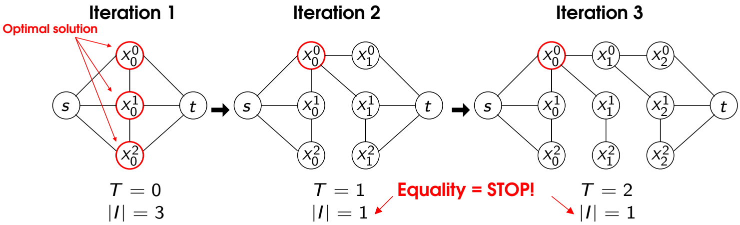
For wich value of T do we have the minimum interface ?

Building the optimal elimination order

For wich value of T do we have the minimum interface ?
$\hookrightarrow$ Proceed by iteratively solving the min s–t cut problem and increasing the length T

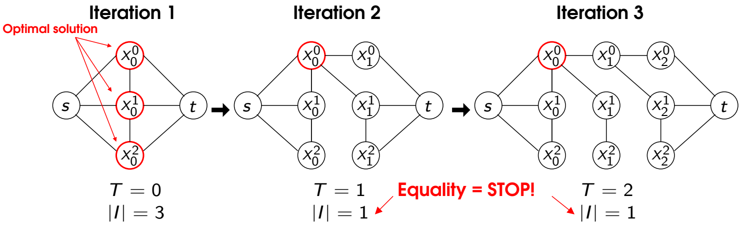
Building the optimal elimination order

For wich value of T do we have the minimum interface ?
$\hookrightarrow$ Proceed by iteratively solving the min s–t cut problem and increasing the length T



Building the optimal elimination order

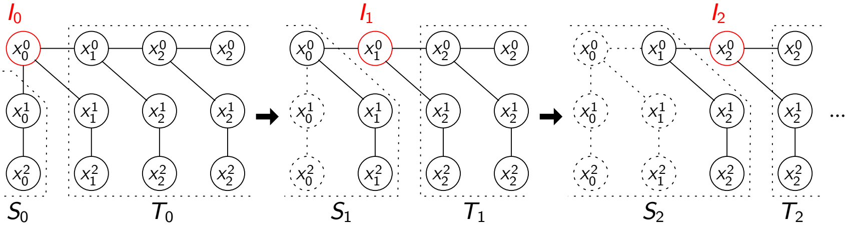
For wich value of T do we have the minimum interface ?
$\hookrightarrow$ Proceed by iteratively solving the min s–t cut problem and increasing the length T



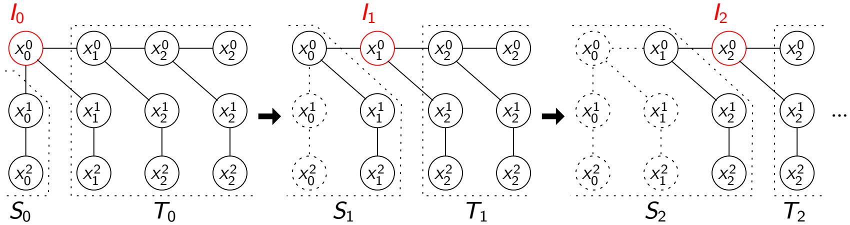
Theorem

At most $O(h)$ iterations where h is the number of nodes in a slice.

Building the elimination order

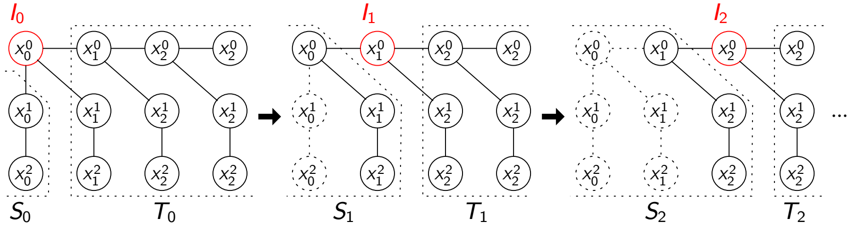


Building the elimination order



$$\textsc{Min-Elim} : \sigma = \mathbf{S}_0, \mathbf{S}_1 - \mathbf{S}_0, \mathbf{S}_2 - \mathbf{S}_1, ..., \mathbf{I}_{n-1} \cup \mathbf{T}_{n-1}$$

Building the elimination order



$$\textsc{Min-Elim} : \sigma = \mathbf{S}_0, \mathbf{S}_1 - \mathbf{S}_0, \mathbf{S}_2 - \mathbf{S}_1, ..., \mathbf{I}_{n-1} \cup \mathbf{T}_{n-1}$$

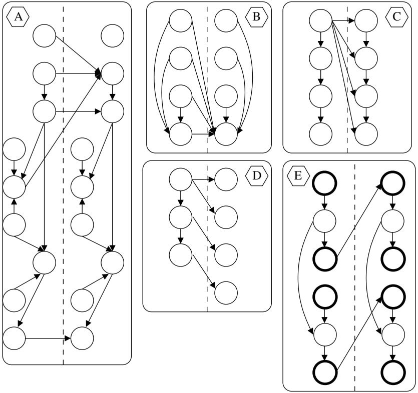
Better lower bound

Best case : clique of size $|I^*| + 1$

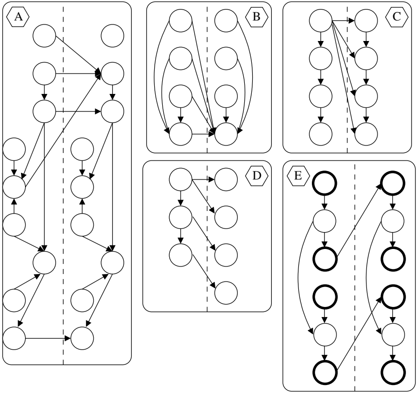
Better upper bound

Worst case : clique of size $|I^*| + S$
(S : # nodes in a slice)

Experimental results – Classical dBNs



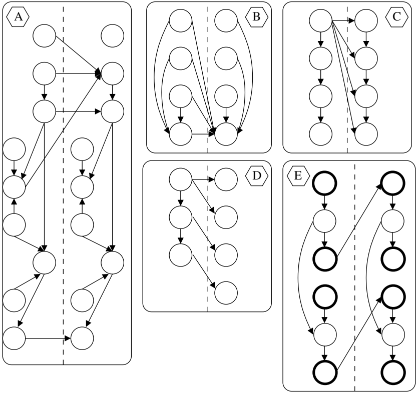
Experimental results – Classical dBNs



Maximum clique size :

Fig.	B-SS	F-SS	Min-Elim
C	3.16	**3.03**	**3.03**
B	**5.54**	**5.54**	**5.54**
D	**1.38**	3.00	**1.38**
A	**2.07**	3.45	**2.07**
E	3.46	3.23	**3.00**

Experimental results – Classical dBNs



Maximum clique size :

Fig.	B-SS	F-SS	Min-Elim
C	3.16	**3.03**	**3.03**
B	**5.54**	**5.54**	**5.54**
D	**1.38**	3.00	**1.38**
A	**2.07**	3.45	**2.07**
E	3.46	3.23	**3.00**

Mean clique size :

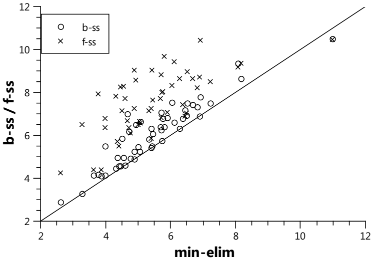
Fig.	B-SS	F-SS	Min-Elim
C	**3.46**	**3.46**	**3.46**
B	**5.54**	**5.54**	**5.54**
D	**1.39**	3.46	**1.39**
A	**2.08**	3.46	**2.08**
E	4.16	**3.46**	**3.46**

Experimental results – Random dBNs

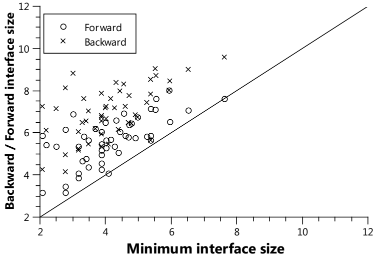
Results on randomly generated dBNs

1. 5 variables per slices
2. Cardinalities chosen uniformly between 2 and 8
3. DBNs unrolled 500 time steps

Mean clique size



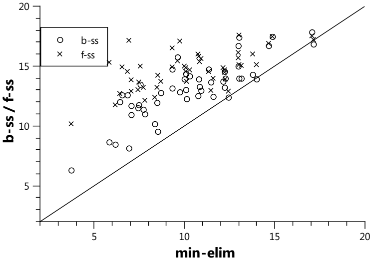
Interface size



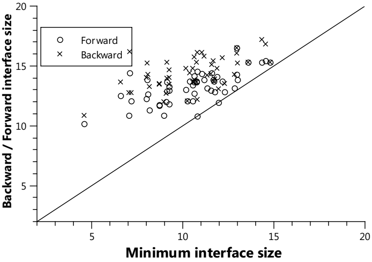
Results on randomly generated dBNs

1. $\mathbf{10}$ variables per slices
2. Cardinalities chosen uniformly between 2 and 8
3. DBNs unrolled 500 time steps

Mean clique size



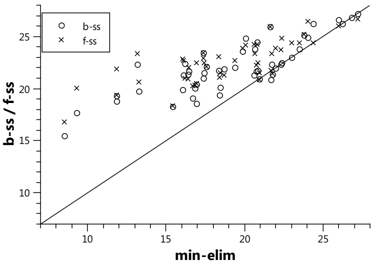
Interface size



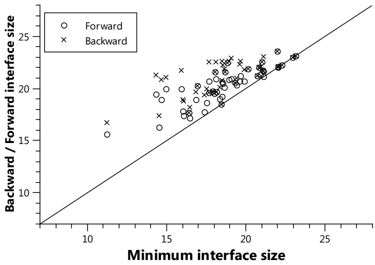
Results on randomly generated dBNs

1. $\mathbf{15}$ variables per slices
2. Cardinalities chosen uniformly between 2 and 8
3. DBNs unrolled 500 time steps

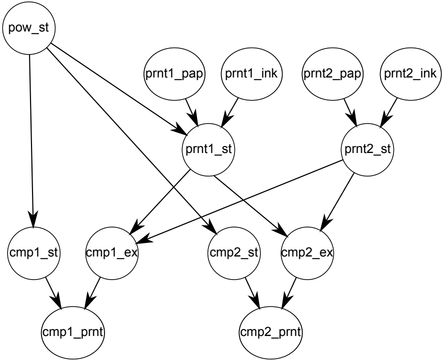
Mean clique size



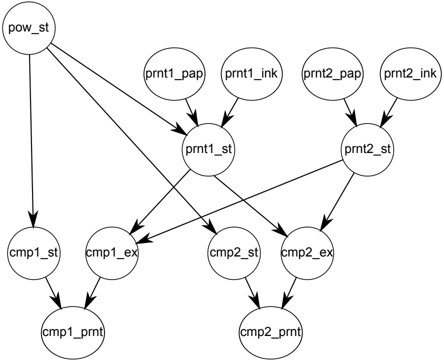
Interface size

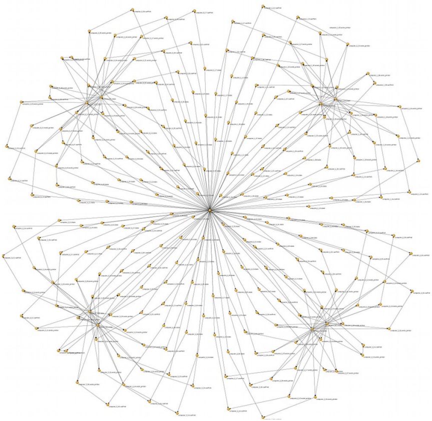


Motivation

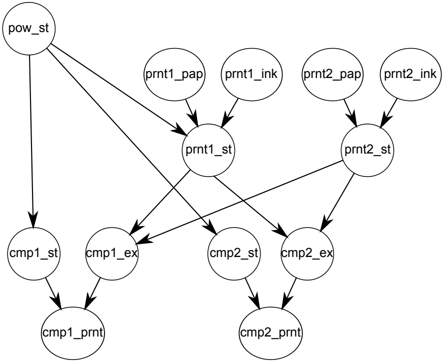

pow_st
prnt1_pap
prnt1_ink
prnt2_pap
prnt2_ink
prnt1_st
prnt2_st
cmp1_st
cmp1_ex
cmp2_st
cmp2_ex
cmp1_prnt
cmp2_prnt

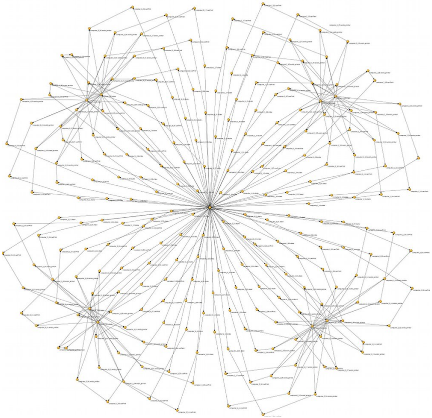
Motivation


pow_st
prnt1_pap
prnt1_ink
prnt2_pap
prnt2_ink
prnt1_st
prnt2_st
cmp1_st
cmp1_ex
cmp2_st
cmp2_ex
cmp1_prnt
cmp2_prnt



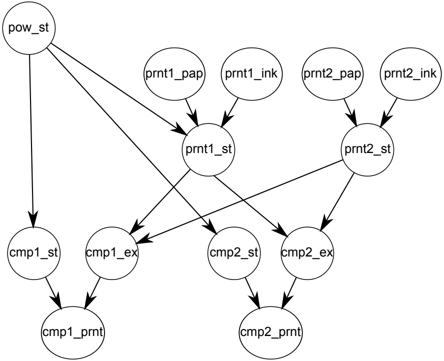
Motivation

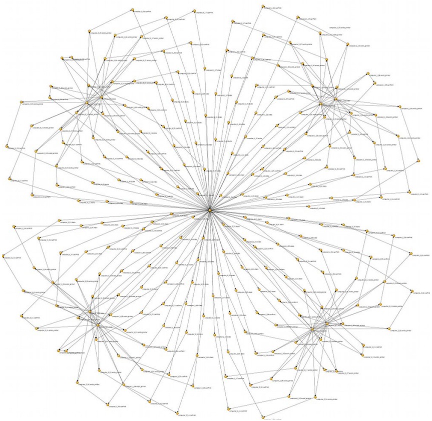




- Large-scale Bayesian networks "undesirable" features :

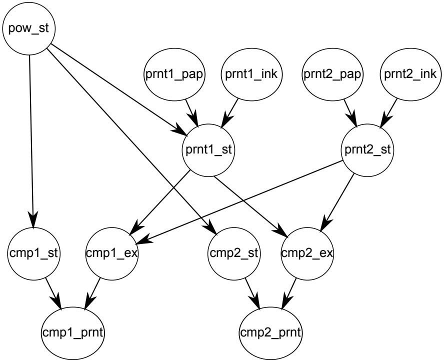
Motivation

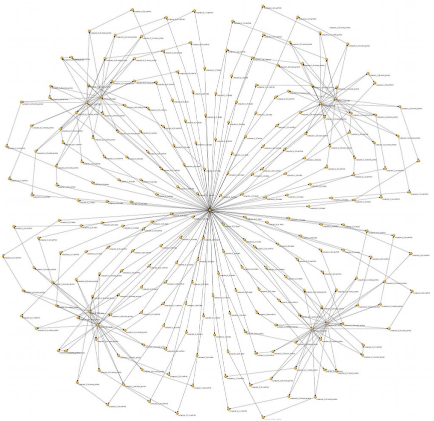




- Large-scale Bayesian networks “undesirable” features :
    - design cost

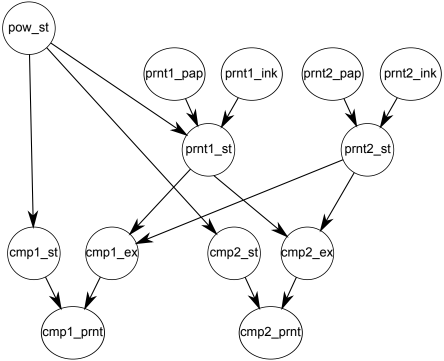
Motivation

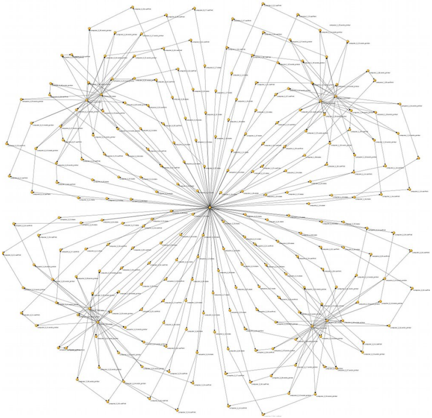




- Large-scale Bayesian networks “undesirable” features :
  - design cost
  - maintenance cost

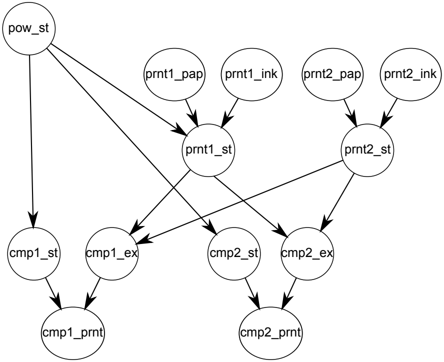
Motivation

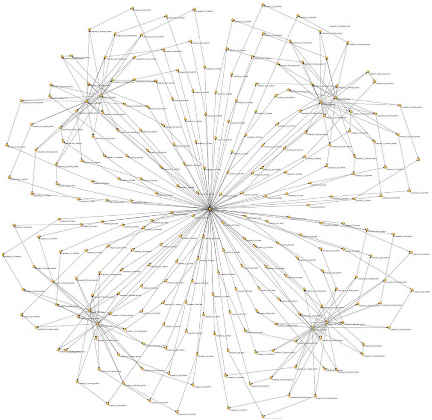




- Large-scale Bayesian networks “undesirable” features :
  - design cost
  - maintenance cost
  - inference times

Motivation



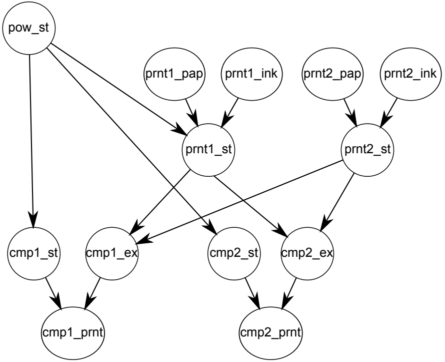


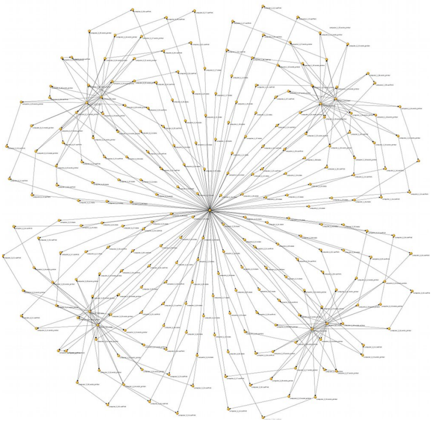
- Large-scale Bayesian networks "undesirable" features :
  - design cost
  - maintenance cost
  - inference times

$\Longrightarrow$ often inadequate for large scale applications

[Mahoney & Laskey, 96], [Pfeffer et al., 99]

Motivation





- Large-scale Bayesian networks “undesirable” features :
  - design cost
  - maintenance cost
  - inference times

$\Longrightarrow$ often inadequate for large scale applications

[Mahoney & Laskey, 96], [Pfeffer et al., 99]

Bayes net extensions

- First-order logic extensions (e.g., MLN)

[Jaeger, 97], [Kersting & De Raedt, 01]

Bayes net extensions

- First-order logic extensions (e.g., MLN)

  [Jaeger, 97], [Kersting & De Raedt, 01]

- Entity-Relationship extensions (e.g., MEBN)

  [Heckerman et al., 04], [Laskey, 08]

Bayes net extensions

- First-order logic extensions (e.g., MLN)

  [Jaeger, 97], [Kersting & De Raedt, 01]

- Entity-Relationship extensions (e.g., MEBN)

  [Heckerman et al., 04], [Laskey, 08]

- Object-Oriented extensions (e.g., PRM)

  [Koller & Pfeffer, 97], [Mahoney & Laskey, 96]

Bayes net extensions

- First-order logic extensions (e.g., MLN)

  [Jaeger, 97], [Kersting & De Raedt, 01]

- Entity-Relationship extensions (e.g., MEBN)

  [Heckerman et al., 04], [Laskey, 08]

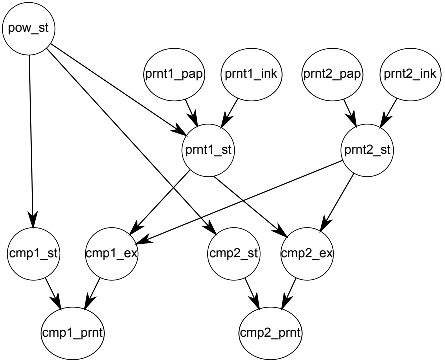
- Object-Oriented extensions (e.g., PRM)

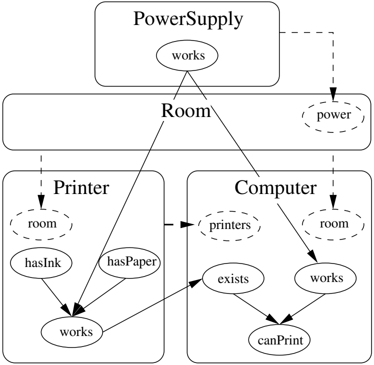
  [Koller & Pfeffer, 97], [Mahoney & Laskey, 96]

our goal : design complex PGM and speed-up inference in PRMs

Probabilistic Relational Models (Pfeffer, 2000)

PRMs are an extension of object-oriented Bayesian Networks (Pfeffer & Koller, 1997) and (Bangsø & Wuillemin, 2000).



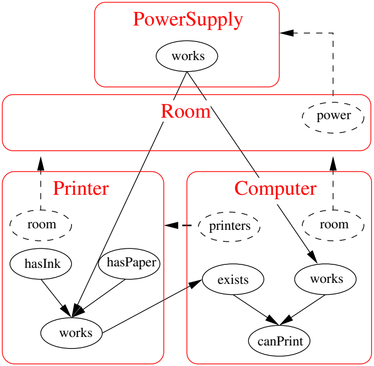


Probabilistic Relational Models (Pfeffer, 2000)

PRMs are an extension of object-oriented Bayesian Networks (Pfeffer & Koller, 1997) and (Bangsø & Wuillemin, 2000).

PRMs in 4 definitions

- Classes are BNs fragments.

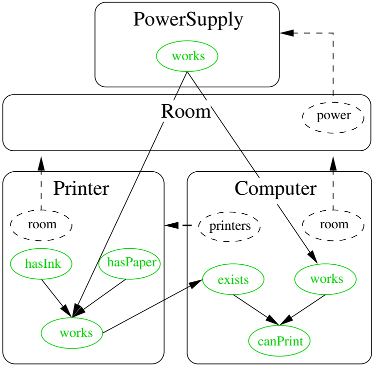


Probabilistic Relational Models (Pfeffer, 2000)

PRMs are an extension of object-oriented Bayesian Networks (Pfeffer & Koller, 1997) and (Bangsø & Wuillemin, 2000).

PRMs in 4 definitions

- Classes are BNs fragments.
- Nodes are called attributes.

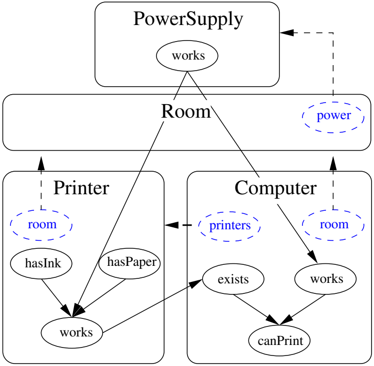


Probabilistic Relational Models (Pfeffer, 2000)

PRMs are an extension of object-oriented Bayesian Networks (Pfeffer & Koller, 1997) and (Bangsø & Wuillemin, 2000).

PRMs in 4 definitions

- Classes are BNs fragments.
- Nodes are called attributes.
- Reference between classes are used to define dependencies between different fragment.

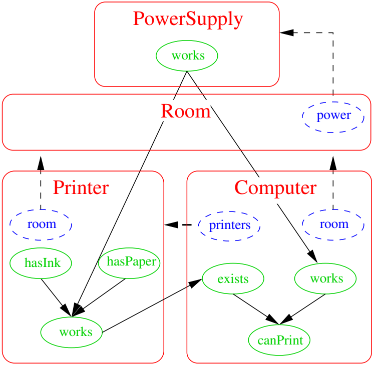


Probabilistic Relational Models (Pfeffer, 2000)

PRMs are an extension of object-oriented Bayesian Networks (Pfeffer & Koller, 1997) and (Bangsø & Wuillemin, 2000).

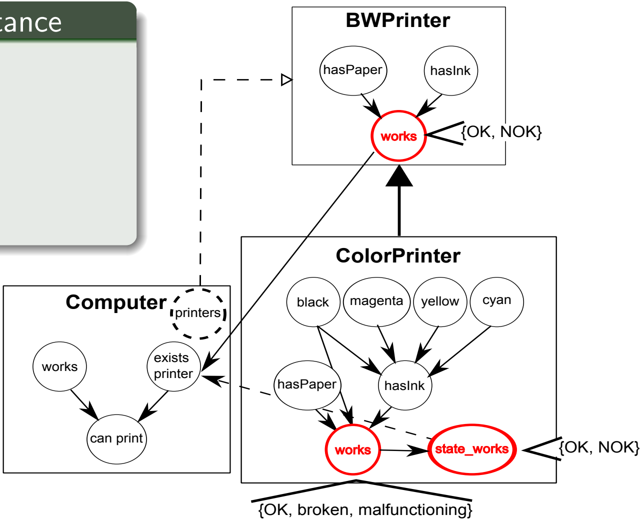
PRMs in 4 definitions

- Classes are BNs fragments.
- Nodes are called attributes.
- Reference between classes are used to define dependencies between different fragment.
- Classes are instantiated in a system.



(1) Reinforcing the object-oriented aspect of PRMs ([ANR SKOOB, Wuillemin, Torti), 07-10])

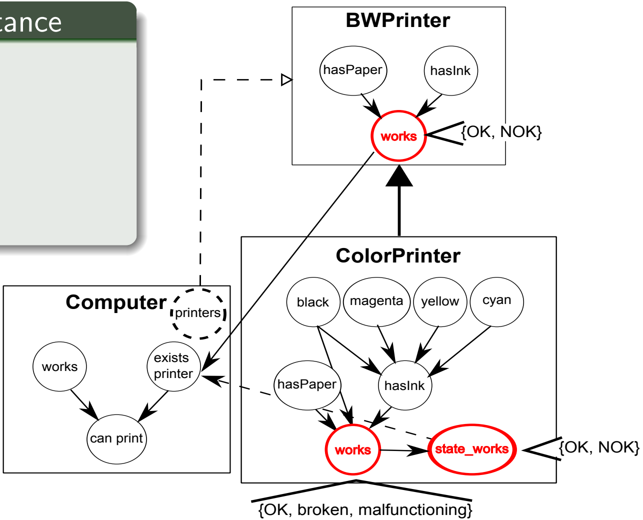
Enhanced class inheritance



(1) Reinforcing the object-oriented aspect of PRMs ([ANR SKOOB, Wuillemin, Torti), 07-10])

Enhanced class inheritance

A subclass can :

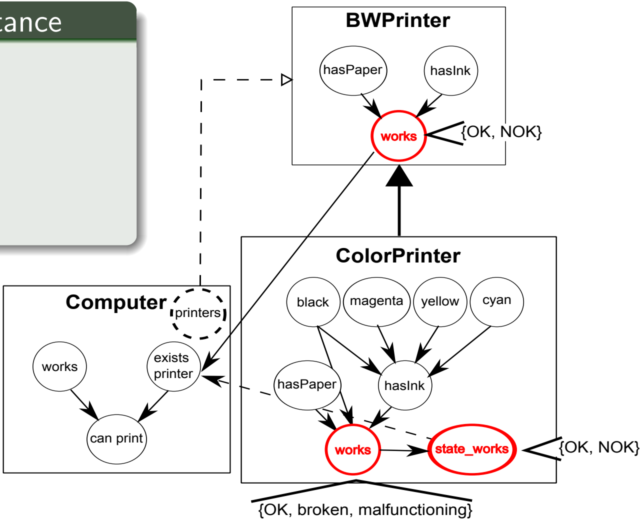


(1) Reinforcing the object-oriented aspect of PRMs ([ANR SKOOB, Wuillemin, Torti), 07-10])

Enhanced class inheritance

A subclass can :

- add attributes,

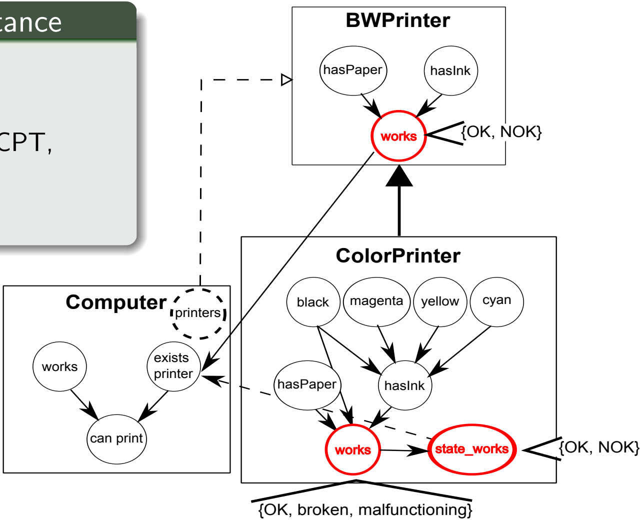


(1) Reinforcing the object-oriented aspect of PRMs ([ANR SKOOB, Wuillemin, Torti), 07-10])

Enhanced class inheritance

A subclass can :

- add attributes,
- change an attribute CPT,

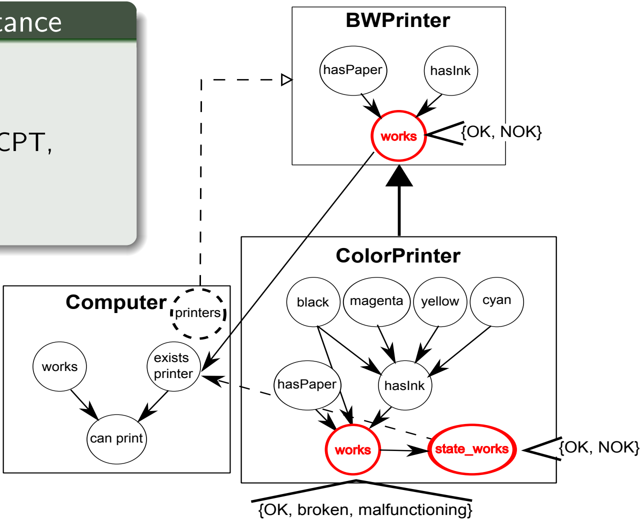


(1) Reinforcing the object-oriented aspect of PRMs ([ANR SKOOB, Wuillemin, Torti), 07-10])

Enhanced class inheritance

A subclass can :

- add attributes,
- change an attribute CPT,
- add references,

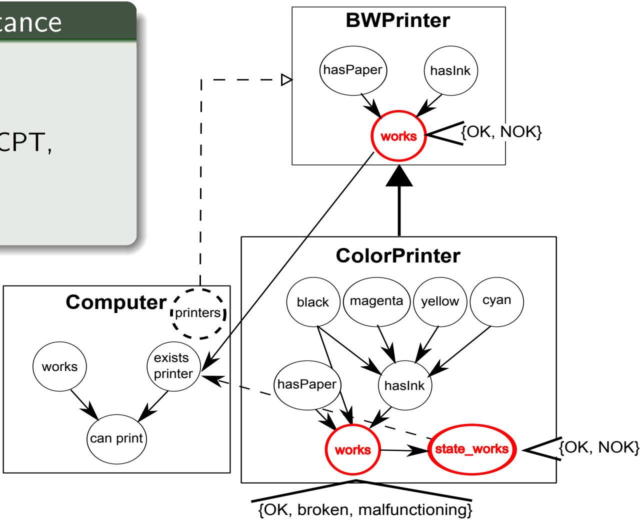


(1) Reinforcing the object-oriented aspect of PRMs ([ANR SKOOB, Wuillemin, Torti), 07-10])

Enhanced class inheritance

A subclass can :

- add attributes,
- change an attribute CPT,
- add references,
- Subtype attribute.



(2) Interface and multiple inheritance

Interface implementation

(2) Interface and multiple inheritance

Interface implementation

- An interface is only defined by a set of attributes types and references.

(2) Interface and multiple inheritance

Interface implementation

- An interface is only defined by a set of attributes types and references.
- This implies the absence of a DAG or attribute's CPT in the interface definition.

(2) Interface and multiple inheritance

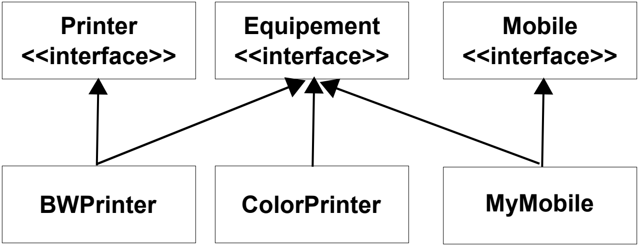
Interface implementation

- An interface is only defined by a set of attributes types and references.
- This implies the absence of a DAG or attribute's CPT in the interface definition.
- Any class implementing an interface guarantees the existence of the interface's attributes and references.

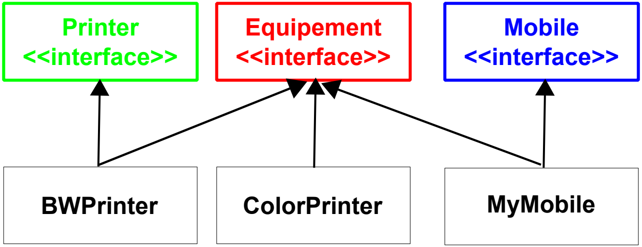
(2) Interface and multiple inheritance

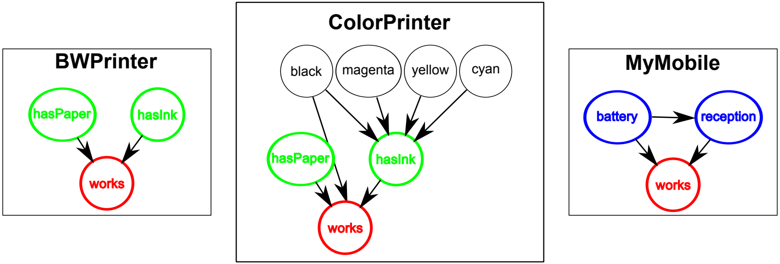
Interface implementation

- An interface is only defined by a set of attributes types and references.
- This implies the absence of a DAG or attribute's CPT in the interface definition.
- Any class implementing an interface guarantees the existence of the interface's attributes and references.



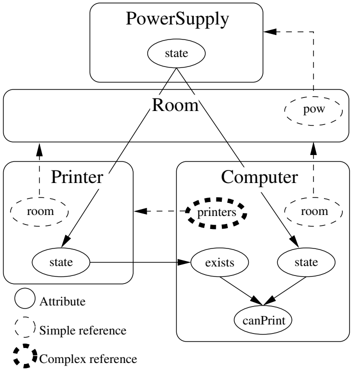
(2) Interface and multiple inheritance



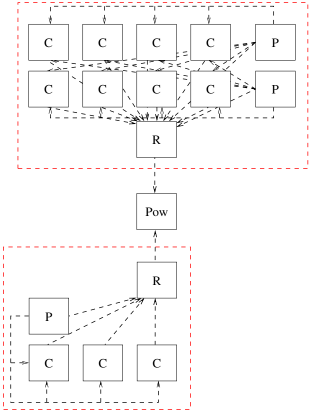


System : the printer example

A Class Dependency Graph.



A Relational Skeleton.



The SKOOB ANR project

Applications

The SKOOB ANR project

- SKOOB is a consortium of industrials, research laboratories and experts which focus on the use of probabilistic graphical models in reliability and risk management.

Applications

The SKOOB ANR project

- SKOOB is a consortium of industrials, research laboratories and experts which focus on the use of probabilistic graphical models in reliability and risk management.
- It has also lead to the definition of a declarative object oriented language for the specification of PRMs (the SKOOL language).

Applications

The SKOOB ANR project

- SKOOB is a consortium of industrials, research laboratories and experts which focus on the use of probabilistic graphical models in reliability and risk management.
- It has also lead to the definition of a declarative object oriented language for the specification of PRMs (the SKOOL language).
- The enhancement presented here have been implemented in two different prototypes.

Applications

The SKOOB ANR project

- SKOOB is a consortium of industrials, research laboratories and experts which focus on the use of probabilistic graphical models in reliability and risk management.
- It has also lead to the definition of a declarative object oriented language for the specification of PRMs (the SKOOL language).
- The enhancement presented here have been implemented in two different prototypes.
- All of which have been successfully used by experts to represent different complex systems.

Applications

The SKOOB ANR project

- SKOOB is a consortium of industrials, research laboratories and experts which focus on the use of probabilistic graphical models in reliability and risk management.
- It has also lead to the definition of a declarative object oriented language for the specification of PRMs (the SKOOL language).
- The enhancement presented here have been implemented in two different prototypes.
- All of which have been successfully used by experts to represent different complex systems.
- The models we created required the modification of the PRM framework presented here.

Inference in PRM

Grounded Bayesian Networks

Inference in PRM

Grounded Bayesian Networks

- Any system can be transformed in a Bayesian Network (called a grounded network).

Inference in PRM

Grounded Bayesian Networks

- Any system can be transformed in a Bayesian Network (called a grounded network).
- The generation algorithm is polynomial [Pfeffer, 2000].

Inference in PRM

Grounded Bayesian Networks

- Any system can be transformed in a Bayesian Network (called a grounded network).
- The generation algorithm is polynomial [Pfeffer, 2000].
- It is possible to use Bayesian Networks state-of-the-art inference algorithms.

Inference in PRM

Grounded Bayesian Networks

- Any system can be transformed in a Bayesian Network (called a grounded network).
- The generation algorithm is polynomial [Pfeffer, 2000].
- It is possible to use Bayesian Networks state-of-the-art inference algorithms.

Why do we want to infer in PRM ?

Inference in PRM

Grounded Bayesian Networks

- Any system can be transformed in a Bayesian Network (called a grounded network).
- The generation algorithm is polynomial [Pfeffer, 2000].
- It is possible to use Bayesian Networks state-of-the-art inference algorithms.

Why do we want to infer in PRM ?

- It is easy to produce very large scale systems :
  - *naive* grounded networks use too much memory ;

Inference in PRM

Grounded Bayesian Networks

- Any system can be transformed in a Bayesian Network (called a grounded network).
- The generation algorithm is polynomial [Pfeffer, 2000].
- It is possible to use Bayesian Networks state-of-the-art inference algorithms.

Why do we want to infer in PRM ?

- It is easy to produce very large scale systems :
  - *naive* grounded networks use too much memory ;
  - *intelligent* grounded networks would be redundant.

Inference in PRM

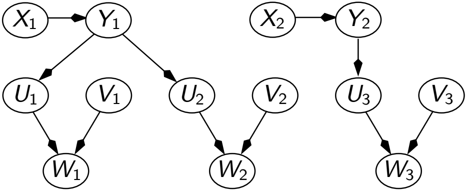
Grounded Bayesian Networks

- Any system can be transformed in a Bayesian Network (called a grounded network).
- The generation algorithm is polynomial [Pfeffer, 2000].
- It is possible to use Bayesian Networks state-of-the-art inference algorithms.

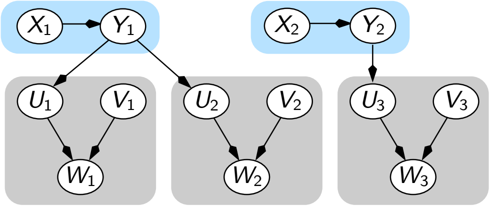
Why do we want to infer in PRM ?

- It is easy to produce very large scale systems :
  - *naive* grounded networks use too much memory ;
  - *intelligent* grounded networks would be redundant.
- Faster inference !

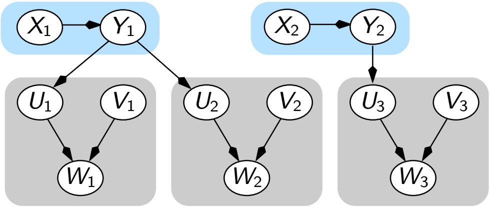
Overview of PRMs

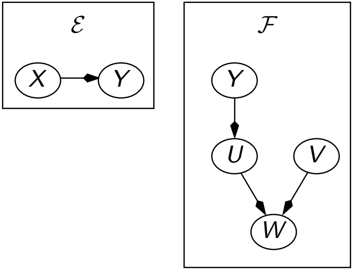


Overview of PRMs

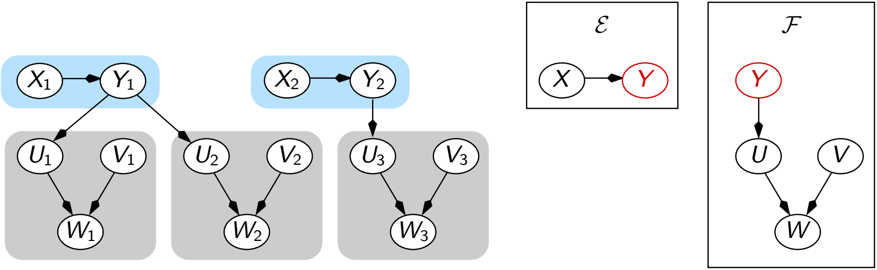


Overview of PRMs

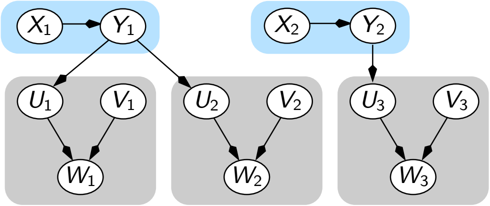


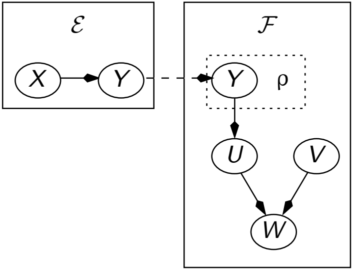


Overview of PRMs

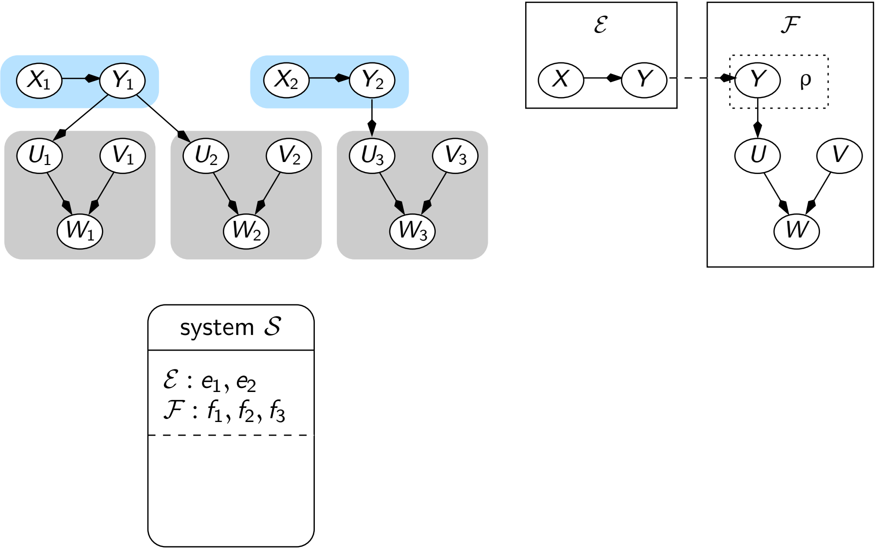


Overview of PRMs

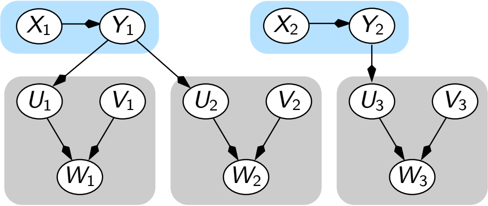


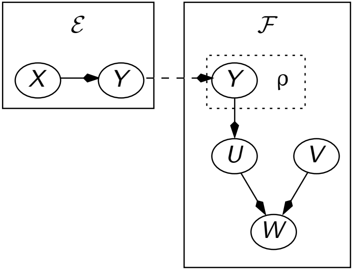


Overview of PRMs



Overview of PRMs



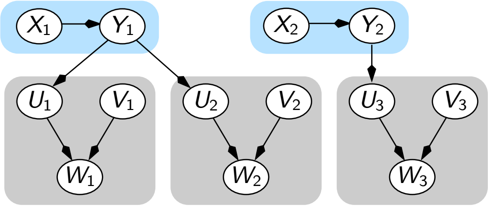


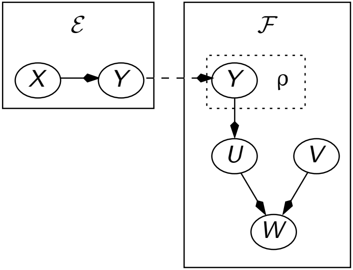
system $\mathcal{S}$

$\mathcal{E} : e_1, e_2$
$\mathcal{F} : f_1, f_2, f_3$

$f_1.\rho = e_1$
$f_2.\rho = e_1$
$f_3.\rho = e_2$

Overview of PRMs

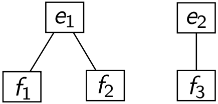




system $\mathcal{S}$

$\mathcal{E}$: e_1, e_2
$\mathcal{F}$: f_1, f_2, f_3

$f_1.\rho = e_1$
$f_2.\rho = e_1$
$f_3.\rho = e_2$



Relational skeleton

Structured Value Elimination [Pfeffer, 2000]

What is structural information ?

- Each class in a PRM defines a pattern repeated in each of its instances.

Structured Value Elimination [Pfeffer, 2000]

What is structural information ?

- Each class in a PRM defines a pattern repeated in each of its instances.
- In each class, some attributes can be eliminated before others.

Structured Value Elimination [Pfeffer, 2000]

What is structural information ?

- Each class in a PRM defines a pattern repeated in each of its instances.
- In each class, some attributes can be eliminated before others.
- Eliminating such attributes creates an identical potential in each instance.

Structured Value Elimination [Pfeffer, 2000]

What is structural information ?

- Each class in a PRM defines a pattern repeated in each of its instances.
- In each class, some attributes can be eliminated before others.
- Eliminating such attributes creates an identical potential in each instance.
- SVE uses this principle to prevent redundant computations.

Structured Value Elimination [Pfeffer, 2000]

What is structural information ?

- Each class in a PRM defines a pattern repeated in each of its instances.
- In each class, some attributes can be eliminated before others.
- Eliminating such attributes creates an identical potential in each instance.
- SVE uses this principle to prevent redundant computations.
- It does not require the generation of a grounded network.

Structured Value Elimination [Pfeffer, 2000]

What is structural information ?

- Each class in a PRM defines a pattern repeated in each of its instances.
- In each class, some attributes can be eliminated before others.
- Eliminating such attributes creates an identical potential in each instance.
- SVE uses this principle to prevent redundant computations.
- It does not require the generation of a grounded network.
- However, it works only if the inner attributes are not observed.

Structured Value Elimination [Pfeffer, 2000]

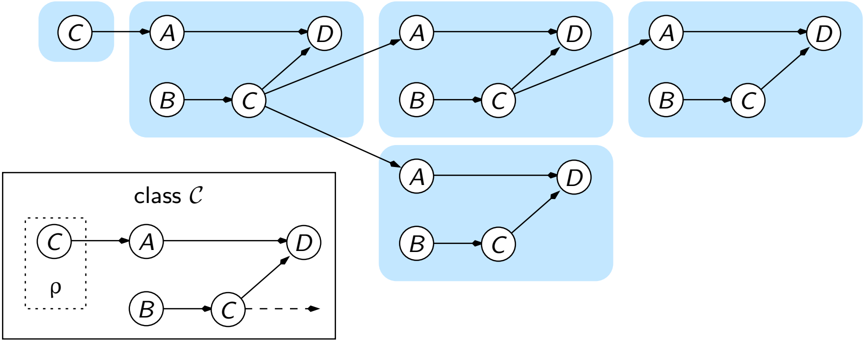
What is structural information ?

- Each class in a PRM defines a pattern repeated in each of its instances.
- In each class, some attributes can be eliminated before others.
- Eliminating such attributes creates an identical potential in each instance.
- SVE uses this principle to prevent redundant computations.
- It does not require the generation of a grounded network.
- However, it works only if the inner attributes are not observed.

Once inner attributes are eliminated, SVE proceeds with a **bottom-up** elimination of each instance.

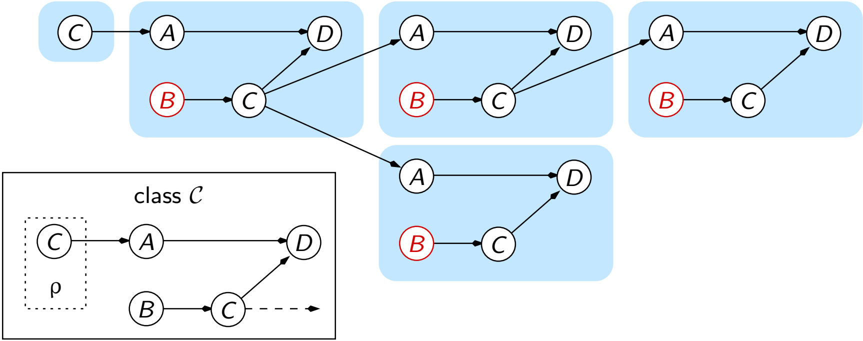
Structured Inference

Probability $P(C)$?



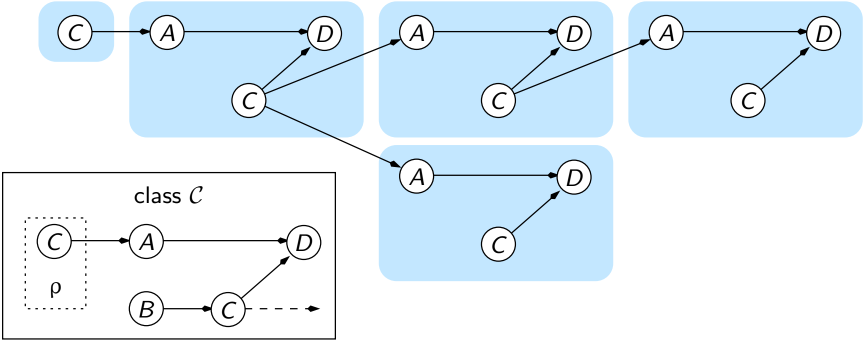
Structured Inference

Probability $P(C)$?

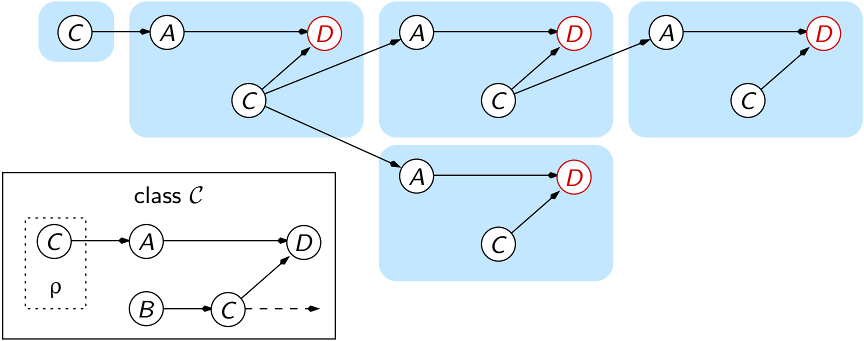


Structured Inference

Probability $P(C)$?

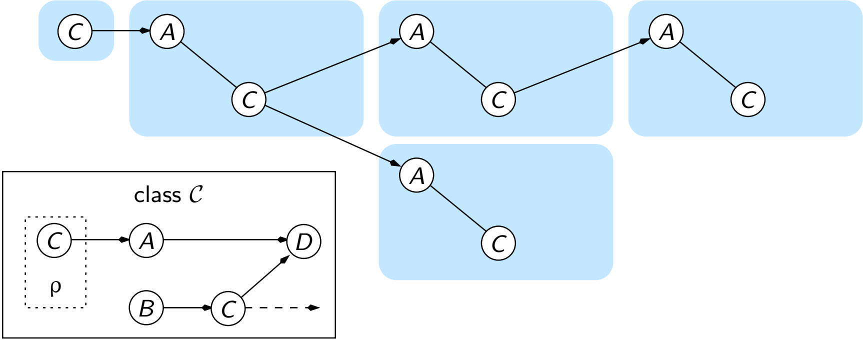


Probability $P(C)$?



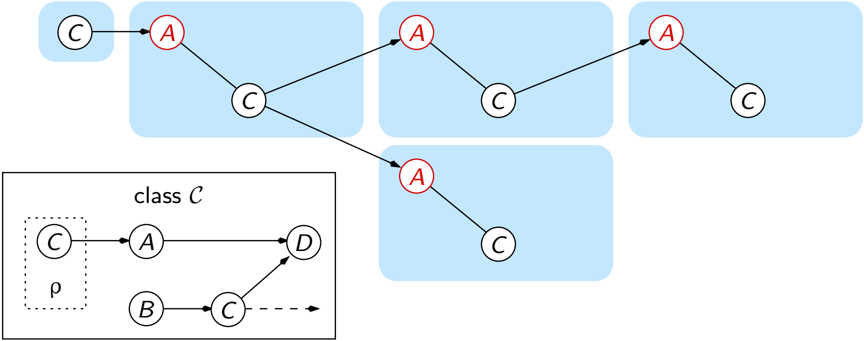
Structured Inference

Probability $P(C)$?



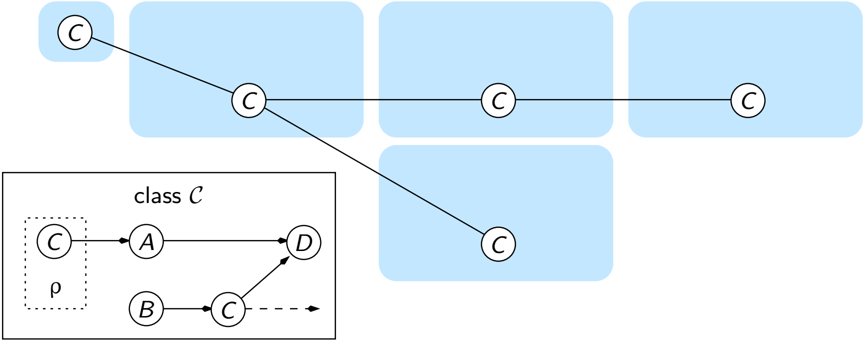
Structured Inference

Probability $P(C)$?



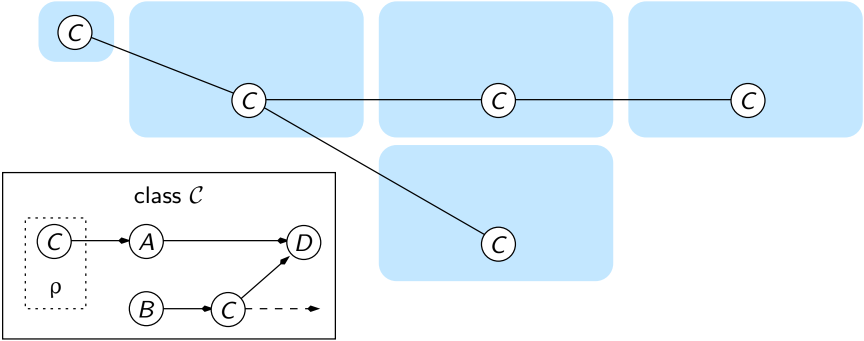
Structured Inference

Probability $P(C)$?



Structured Inference

Probability $P(C)$?



Elimination at class level : performed only once !

It works only if the inner attributes are not observed.
Only internal variables can be eliminated at class level !

D-Separation in PRMs ([Torti and Wuillemin, 2010])

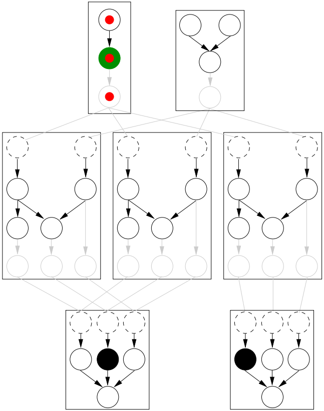


FIGURE: Different configurations due to evidence.

Integrating D-Separation in SVE

Following SVE bottom-up elimination order :

1. Recursive calls are made on each reverse slot chain.
2. If at least one reverse slot chain is active :
   1. Activate any required slot chains.
   2. Either compute the inner node elimination or retrieve any existing computation.

D-Separation in PRMs ([Torti and Wuillemin, 2010])

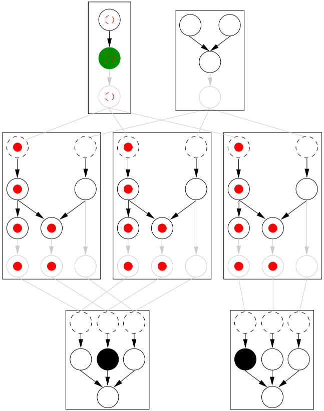


FIGURE: Different configurations due to evidence.

Integrating D-Separation in SVE

Following SVE bottom-up elimination order :

1. Recursive calls are made on each reverse slot chain.
2. If at least one reverse slot chain is active :
    1. Activate any required slot chains.
    2. Either compute the inner node elimination or retrieve any existing computation.

D-Separation in PRMs ([Torti and Wuillemin, 2010])

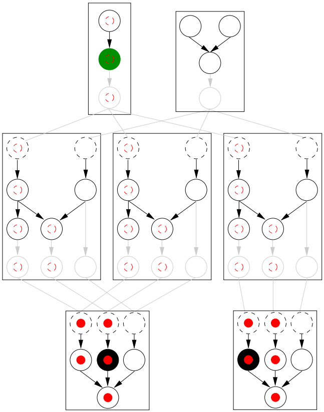


FIGURE: Different configurations due to evidence.

Integrating D-Separation in SVE

Following SVE bottom-up elimination order :

1. Recursive calls are made on each reverse slot chain.
2. If at least one reverse slot chain is active :
   1. Activate any required slot chains.
   2. Either compute the inner node elimination or retrieve any existing computation.

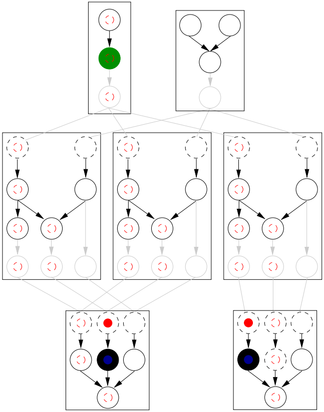


FIGURE: Different configurations due to evidence.

Integrating D-Separation in SVE

Following SVE bottom-up elimination order :

1. Recursive calls are made on each reverse slot chain.
2. If at least one reverse slot chain is active :
    1. Activate any required slot chains.
    2. Either compute the inner node elimination or retrieve any existing computation.

D-Separation in PRMs ([Torti and Wuillemin, 2010])

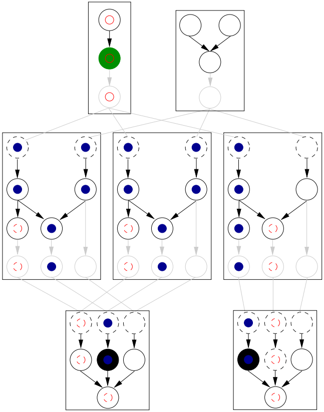


FIGURE: Different configurations due to evidence.

Integrating D-Separation in SVE

Following SVE bottom-up elimination order :

1. Recursive calls are made on each reverse slot chain.
2. If at least one reverse slot chain is active :
   1. Activate any required slot chains.
   2. Either compute the inner node elimination or retrieve any existing computation.

D-Separation in PRMs ([Torti and Wuillemin, 2010])

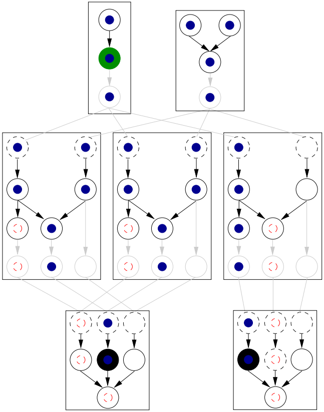


FIGURE: Different configurations due to evidence.

Integrating D-Separation in SVE

Following SVE bottom-up elimination order :

1. Recursive calls are made on each reverse slot chain.
2. If at least one reverse slot chain is active :
   1. Activate any required slot chains.
   2. Either compute the inner node elimination or retrieve any existing computation.

D-Separation in PRMs ([Torti and Wuillemin, 2010])

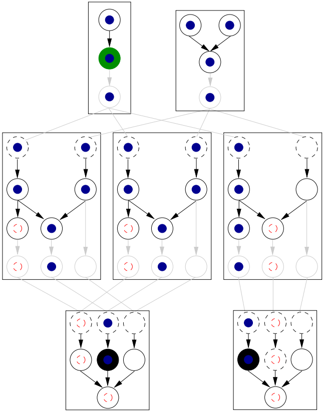


FIGURE: Different configurations due to evidence.

Integrating D-Separation in SVE

Following SVE bottom-up elimination order :

1. Recursive calls are made on each reverse slot chain.
2. If at least one reverse slot chain is active :
    1. Activate any required slot chains.
    2. Either compute the inner node elimination or retrieve any existing computation.

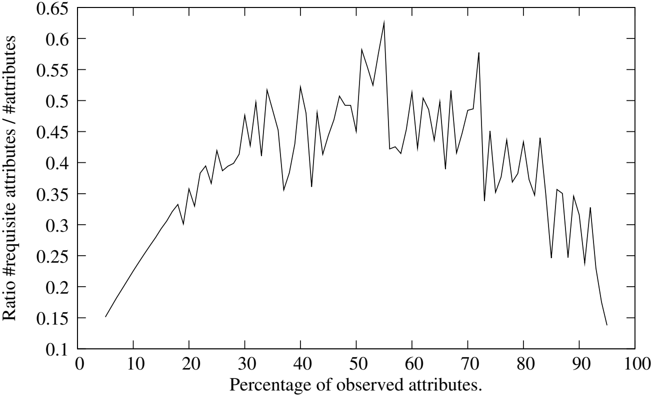
Generalized BayesBall rule

Experimental results

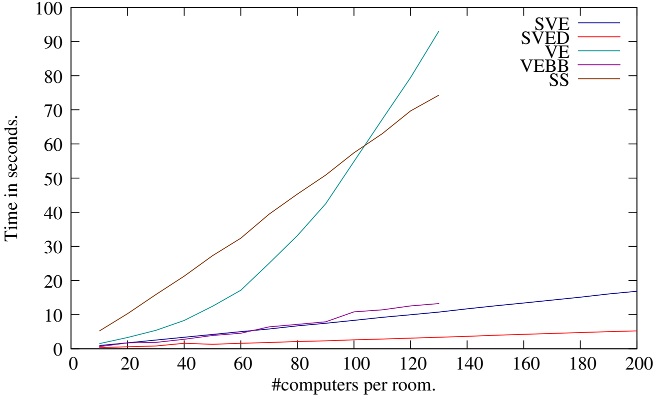
The example used for our experimentations

- We used the printer example with the following parameters :
  - different number of computers per room ;
  - different number of printers per room ;
  - different number of evidences.
- The examples we used have a 1400 to 24000 attributes.

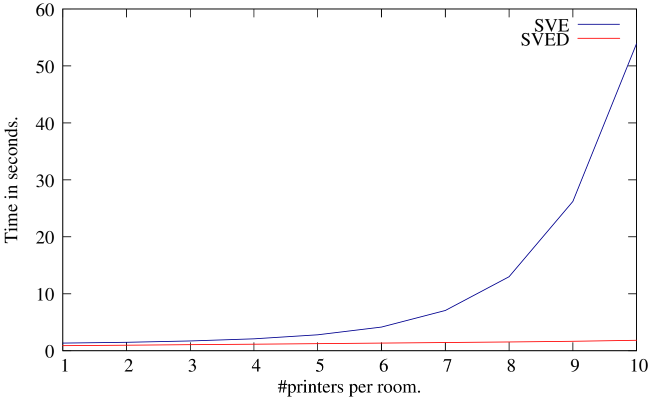
Efficiency of D-Separation



Increasing the number of computers



Increasing the number of printers



Only internal variables can be eliminated at class level

- Construction by experts $\Longrightarrow$ genericity and design patterns.

Only internal variables can be eliminated at class level

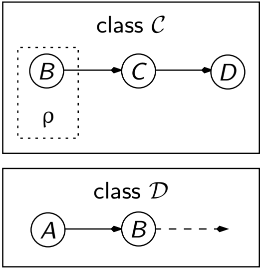
- Construction by experts $\Longrightarrow$ genericity and design patterns.

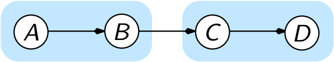
$\Longrightarrow$ For instance : in genealogy, relations only between parents and children.

Only internal variables can be eliminated at class level

- Construction by experts $\Longrightarrow$ genericity and design patterns.

$\Longrightarrow$ For instance : in genealogy, relations only between parents and children.

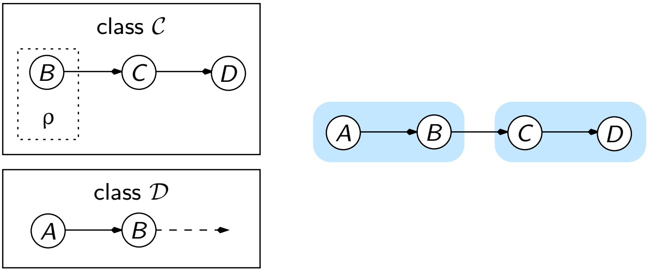




Only internal variables can be eliminated at class level

- Construction by experts $\Longrightarrow$ genericity and design patterns.

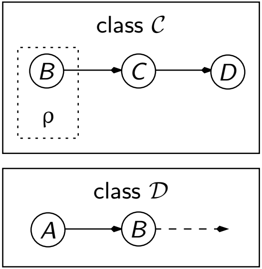
$\Longrightarrow$ For instance : in genealogy, relations only between parents and children.

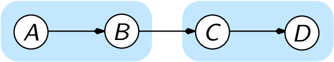


Only internal variables can be eliminated at class level

- Construction by experts $\Longrightarrow$ genericity and design patterns.

$\Longrightarrow$ For instance : in genealogy, relations only between parents and children.



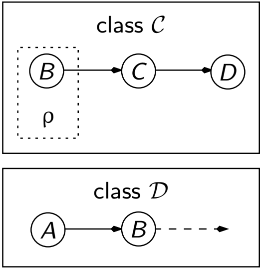


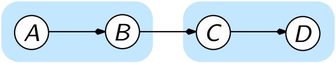
- Solution : create compound classes : dynamic classes

Only internal variables can be eliminated at class level

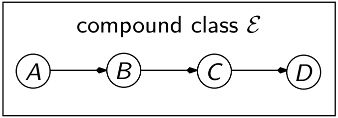
- Construction by experts $\Longrightarrow$ genericity and design patterns.

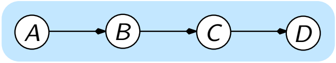
$\Longrightarrow$ For instance : in genealogy, relations only between parents and children.



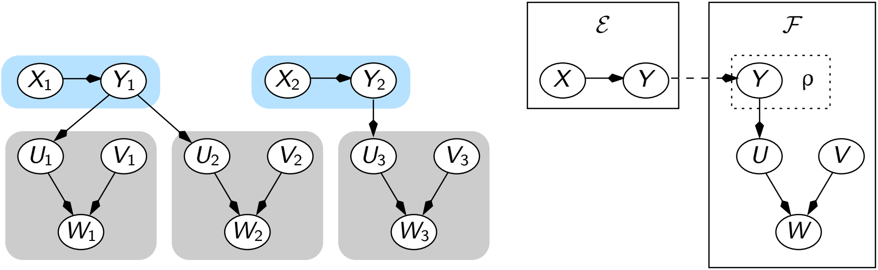


- Solution : create compound classes : dynamic classes





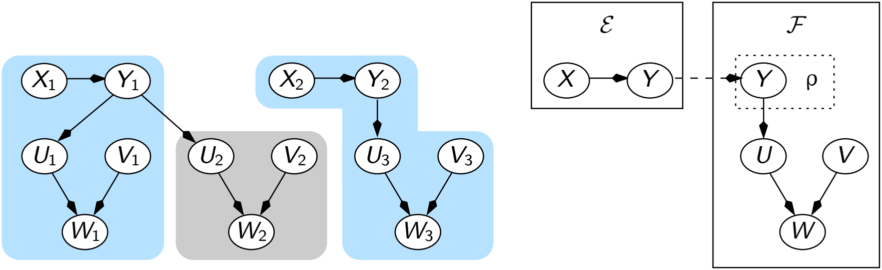
Substitution rule



Rule

One random variable can only belong to one instance.

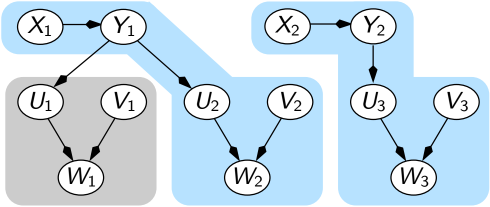
Substitution rule

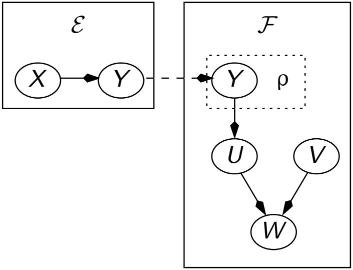


Rule

One random variable can only belong to one instance.

Substitution rule





Rule

One random variable can only belong to one instance.

Optimal dynamic classes [Torti, Gonzales and Wuillemin, 2011]

Proposition

The following problem is NP-hard :

Instance : a PRM, an integer $K \geq 0$

Question : does there exist a set of dynamic classes / substitutions s.t. the number of operations performed by structured inference is $\leq K$?

Optimal dynamic classes [Torti, Gonzales and Wuillemin, 2011]

Proposition

The following problem is NP-hard :

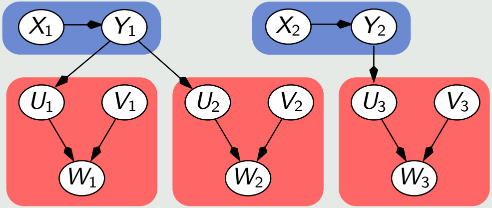
Instance : a PRM, an integer $K \geq 0$

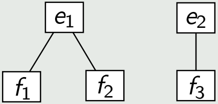
Question : does there exist a set of dynamic classes / substitutions s.t. the number of operations performed by structured inference is $\leq K$?

remains NP-hard even when classical inference is polynomial

Approximate algorithm

Boundary graph

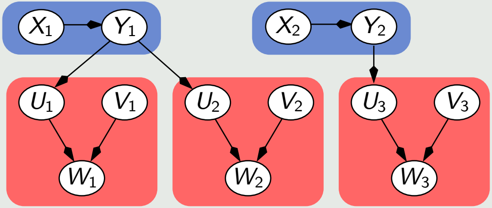


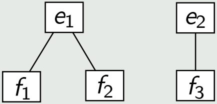


Relational skeleton

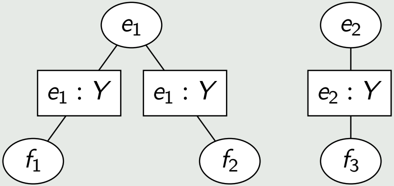
Approximate algorithm

Boundary graph





Relational skeleton



Boundary graph

$\Longrightarrow$ find frequent patterns in the boundary graph

Search tree

- Mining frequent subgraphs

  [Inokuchi et al, 05], [Kuramochi and Karypis, 01], [Yan and Han, 02]

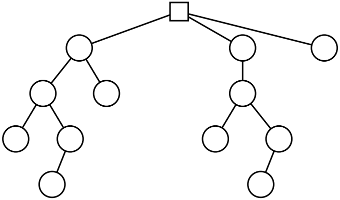
Search tree

- Mining frequent subgraphs

  [Inokuchi et al, 05], [Kuramochi and Karypis, 01], [Yan and Han, 02]

- Variant of the gSpan algorithm :

1 edge
2 edges
3 edges
4 edges



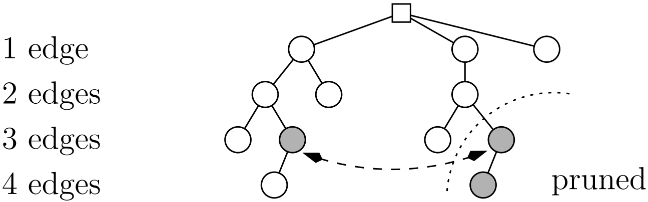
- Node $= (\text{dynamic class}, \text{set of instances})$

Search tree

- Mining frequent subgraphs

[Inokuchi et al, 05], [Kuramochi and Karypis, 01], [Yan and Han, 02]

- Variant of the gSpan algorithm :



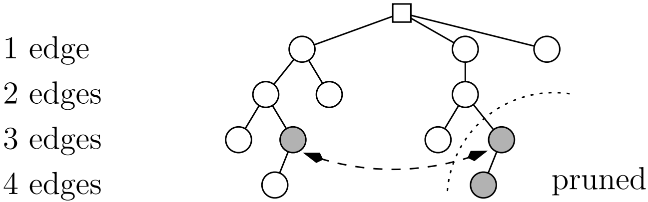
- Node $=$ (dynamic class, set of instances)

Search tree

- Mining frequent subgraphs

  [Inokuchi et al, 05], [Kuramochi and Karypis, 01], [Yan and Han, 02]

- Variant of the gSpan algorithm :



- Node $=$ (dynamic class, set of instances)

- Possible substitutions (Rule 1) : Max Independence Set

Another pruning rule

Efficiency Pruning Rule

Prune nodes/classes whose substitutions do not speed-up inference.

Another pruning rule

Efficiency Pruning Rule

Prune nodes/classes whose substitutions do not speed-up inference.

- Gain estimation by dynamic programming

  $\Longrightarrow$ α-value : $\alpha > 0 \Longleftrightarrow$ class unattractive

search tree not monotonically α-decreasing !

- Rule applied : prune subtree whenever $\alpha > 0$

Experiments