EXTENDING COMPACT-DIAGRAM TO BASIC SMART MULTI-VALUED VARIABLE DIAGRAMS

CPAIOR19

Hélène Verhaeghe[1], Christophe Lecoutre[2], Pierre Schaus[1]

5 June 2019

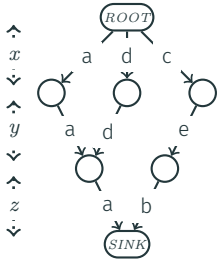
[1] ICTEAM, UCLouvain, Place Sainte Barbe 2, 1348 Louvain-la-Neuve, Belgium, $\{firstname.lastname\}@uclouvain.be$
[2] CRIL-CNRS UMR 8188, Université d'Artois, F-62307 Lens, France, $lecoutre@cril.fr$

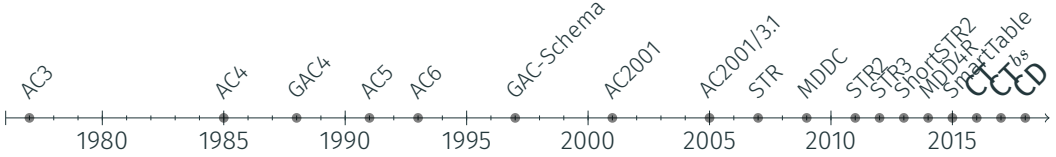
UCLouvain

	x	y	z
τ_1	a	a	a
τ_2	d	d	a
τ_3	c	e	b
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Tables are the oldest most used CP constraints

MDDs/MVDs are equivalent to tables



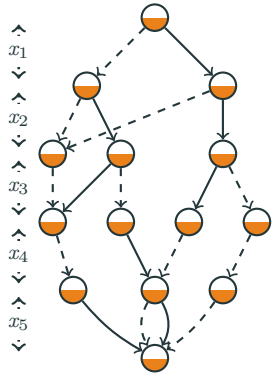


2016 : New Algorithm! Compact-Table [CP2016], based on bitwise operations, completely outperformed existing algorithms

THE BASIC SMART MULTI-VALUED VARIABLE DIAGRAM

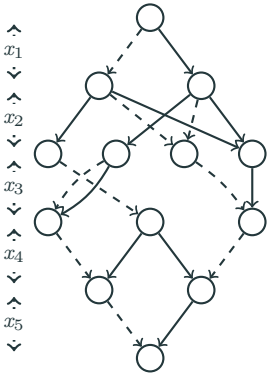
UCLouvain

Multi-valued Decision Diagram



Decision nodes

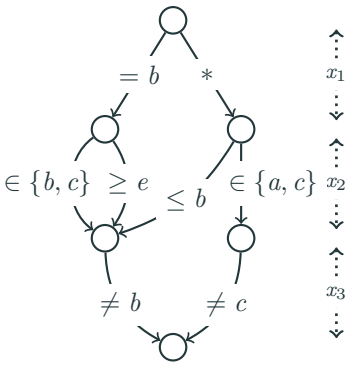
Multi-valued Variable Diagram



Non-decision nodes

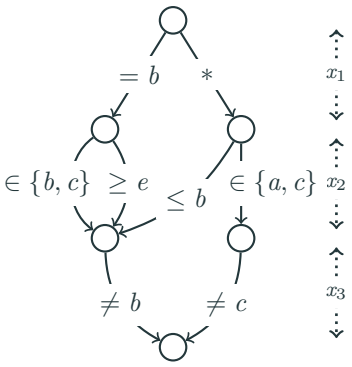
UCLouvain

A basic smart MVD

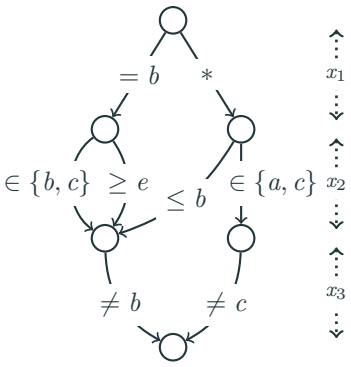


A basic smart MVD labeled with Smart Elements representing multiples values

a b c d e f



A basic smart MVD



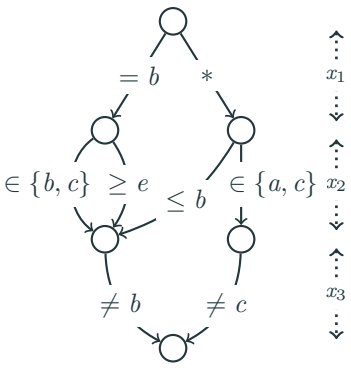
labeled with Smart Elements

single value: $= e$

representing multiples values

a	b	c	d	e	f
✗	✗	✗	✗	✔	✗

A basic smart MVD



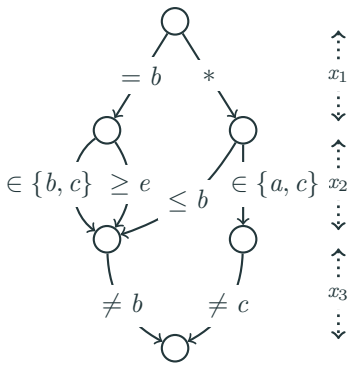
labeled with Smart Elements

representing multiples values

	a	b	c	d	e	f
single value: $= e$	✗	✗	✗	✗	✔	✗
universal value: $*$	✔	✔	✔	✔	✔	✔

UCLouvain

A basic smart MVD



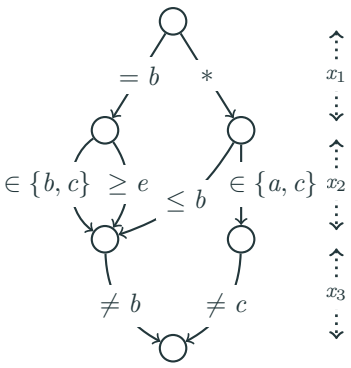
labeled with Smart Elements

representing multiples values

	a	b	c	d	e	f
single value: $= e$	✗	✗	✗	✗	✔	✗
universal value: $*$	✔	✔	✔	✔	✔	✔
exclusion: $\neq e$	✔	✔	✔	✔	✗	✔

UCLouvain

A basic smart MVD



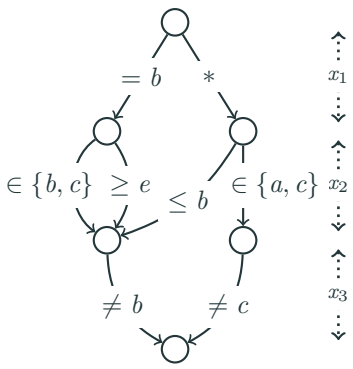
labeled with Smart Elements

representing multiples values

	a	b	c	d	e	f
single value: $= e$	✗	✗	✗	✗	✔	✗
universal value: $*$	✔	✔	✔	✔	✔	✔
exclusion: $\neq e$	✔	✔	✔	✔	✗	✔
upper bound: $\leq c$	✔	✔	✔	✗	✗	✗

UCLouvain

A basic smart MVD



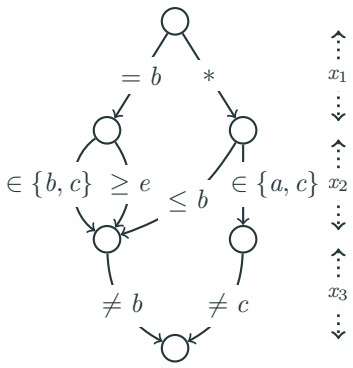
labeled with Smart Elements

representing multiples values

	a	b	c	d	e	f
single value: $= e$	✗	✗	✗	✗	✔	✗
universal value: $*$	✔	✔	✔	✔	✔	✔
exclusion: $\neq e$	✔	✔	✔	✔	✗	✔
upper bound: $\leq c$	✔	✔	✔	✗	✗	✗
lower bound: $\geq c$	✗	✗	✔	✔	✔	✔

UCLouvain

A basic smart MVD

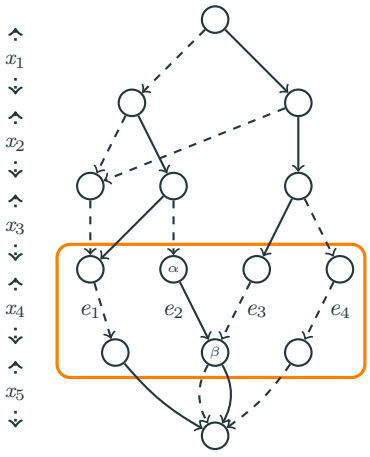


labeled with Smart Elements

representing multiples values

	a	b	c	d	e	f
single value: $= e$	✗	✗	✗	✗	✔	✗
universal value: $*$	✔	✔	✔	✔	✔	✔
exclusion: $\neq e$	✔	✔	✔	✔	✗	✔
upper bound: $\leq c$	✔	✔	✔	✗	✗	✗
lower bound: $\geq c$	✗	✗	✔	✔	✔	✔
set: $\in \{a, c, d\}$	✔	✗	✔	✔	✗	✗

THE COMPACT-DIAGRAM ALGORITHM, FOR MVDS



Name	Set	Bit-set
`currArcs`$[x_4]$ (1)	$\{e_1, e_2, e_3, e_4\}$	$[\,1\,1\,1\,1\,]$
`supports`$[x_4,0]$ (2)	$\{e_1, \cancel{e_2}, e_3, e_4\}$	$[\,1\,0\,1\,1\,]$
`arcsT`$[\alpha,x_4]$ (2)	$\{\cancel{e_1}, e_2, \cancel{e_3}, \cancel{e_4}\}$	$[\,0\,1\,0\,0\,]$
`arcsH`$[x_4,\beta]$ (2)	$\{\cancel{e_1}, e_2, e_3, \cancel{e_4}\}$	$[\,0\,1\,1\,0\,]$

(1) mutable (2) immutable and precomputed

1. Which edges are still valid?

2. Which values are no more supported?

1. Which edges are still valid?

   Update phase

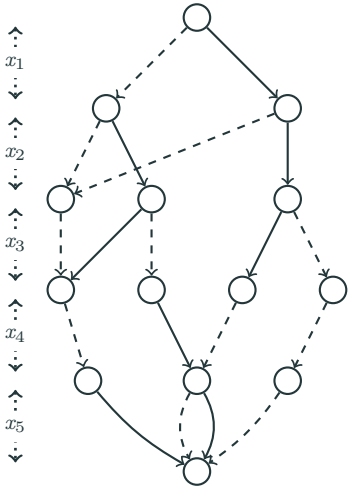
2. Which values are no more supported?

1. Which edges are still valid?

   Update phase

2. Which values are no more supported?

   Propagation phase



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

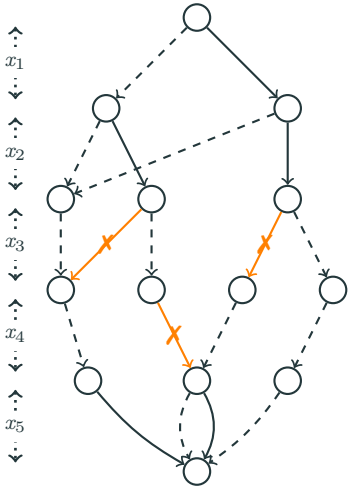
currArcs[x_3]
[1 1 1 1 1]

currArcs[x_4]
[1 1 1 1]

currArcs[x_5]
[1 1 1 1]

UCLouvain

Direct removal



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

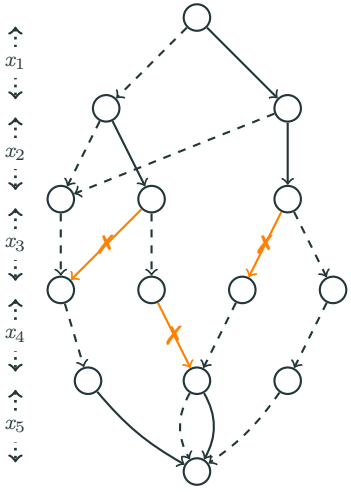
currArcs[x_3]
[1 1 1 1 1]

currArcs[x_4]
[1 1 1 1]

currArcs[x_5]
[1 1 1 1]

UCLouvain

Direct removal



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

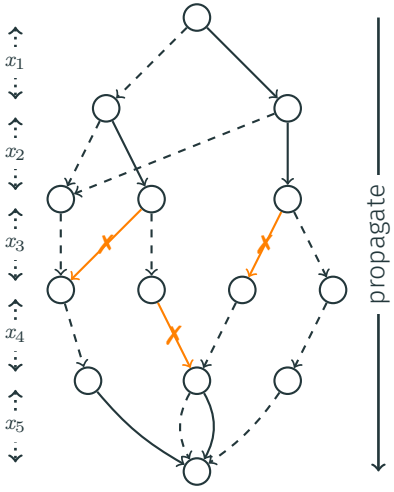
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 1 1]

currArcs[x_5]
[1 1 1 1]

UCLouvain

Top down



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

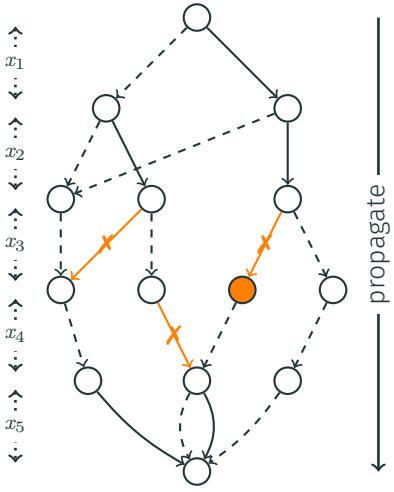
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 1 1]

currArcs[x_5]
[1 1 1 1]

UCLouvain

Top down



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

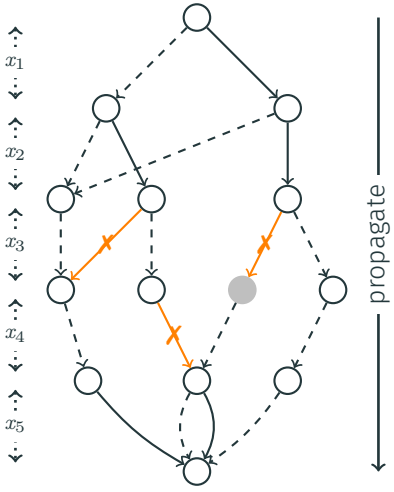
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 1 1]

currArcs[x_5]
[1 1 1 1]

UCLouvain

Top down



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

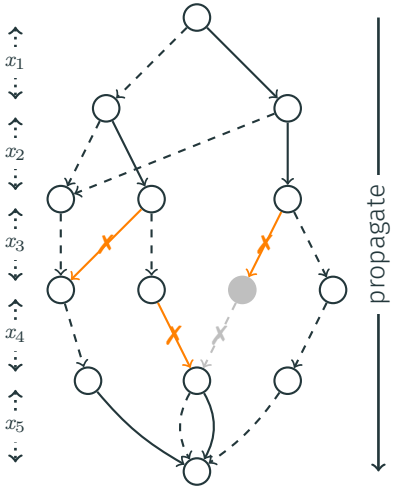
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 1 1]

currArcs[x_5]
[1 1 1 1]

UCLouvain

Top down



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

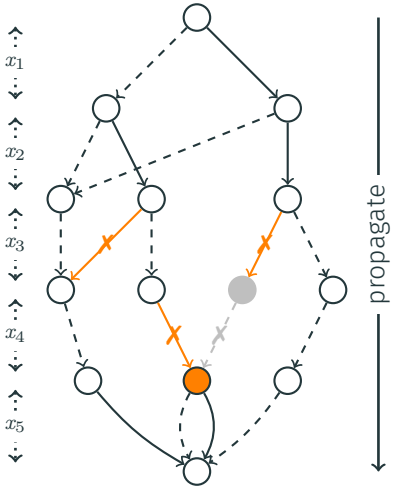
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 1 1 1]

UCLouvain

Top down



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

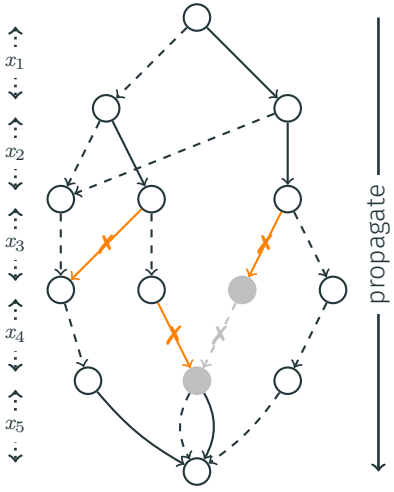
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 1 1 1]

UCLouvain

Top down



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

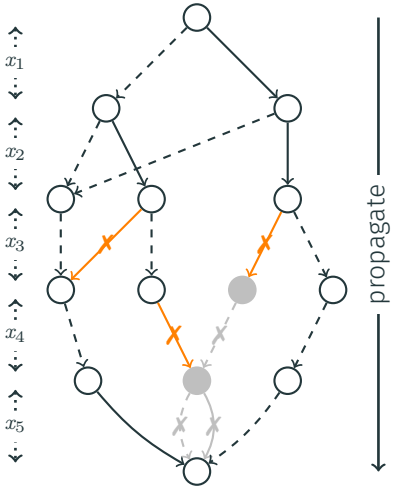
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 1 1 1]

UCLouvain

Top down



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

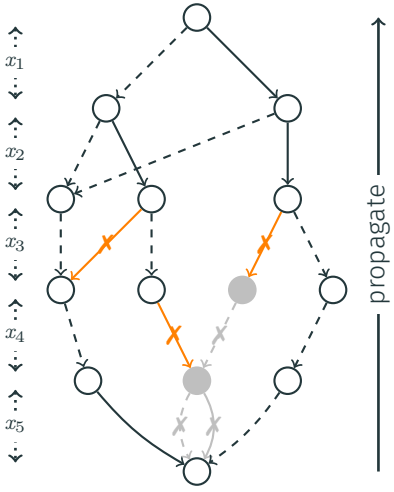
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 0 0 1]

UCLouvain

Bottom up



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

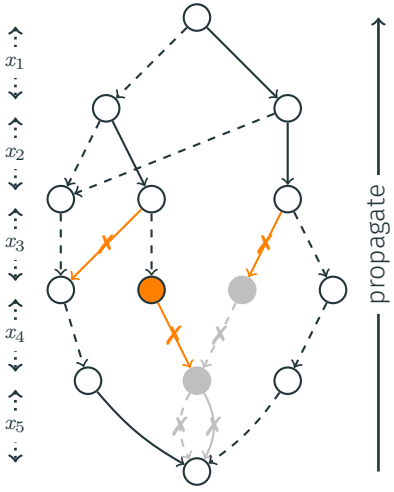
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 0 0 1]

UCLouvain

Bottom up



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

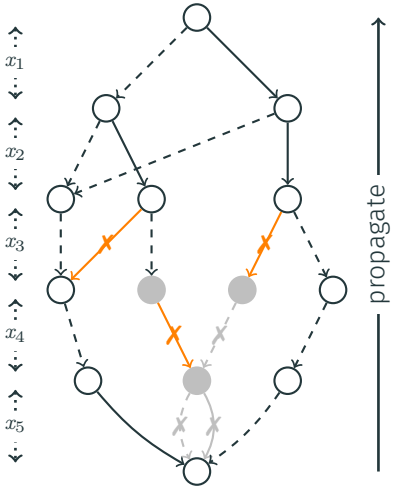
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 0 0 1]

UCLouvain

Bottom up



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

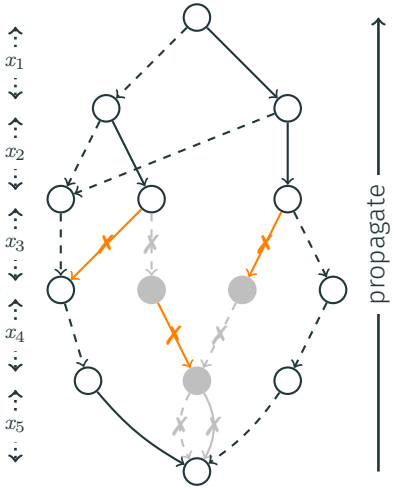
currArcs[x_3]
[1 0 1 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 0 0 1]

UCLouvain

Bottom up



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

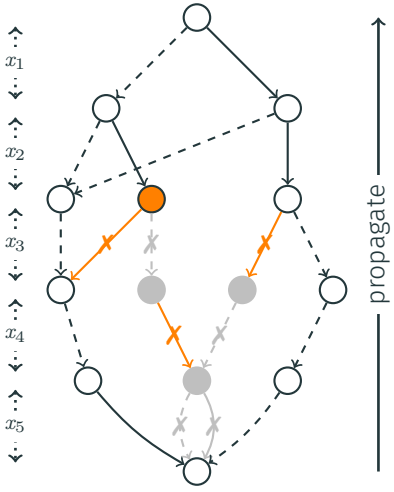
currArcs[x_3]
[1 0 0 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 0 0 1]

UCLouvain

Bottom up



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

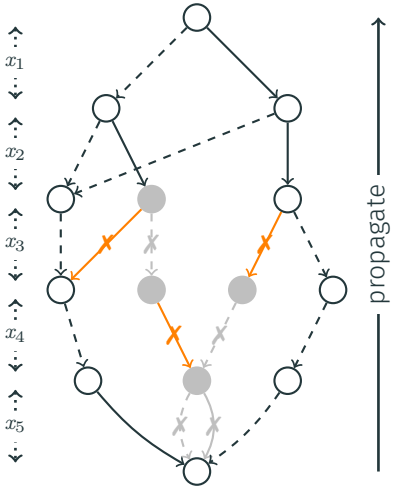
currArcs[x_3]
[1 0 0 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 0 0 1]

UCLouvain

Bottom up



currArcs[x_1]
[1 1]

currArcs[x_2]
[1 1 1 1]

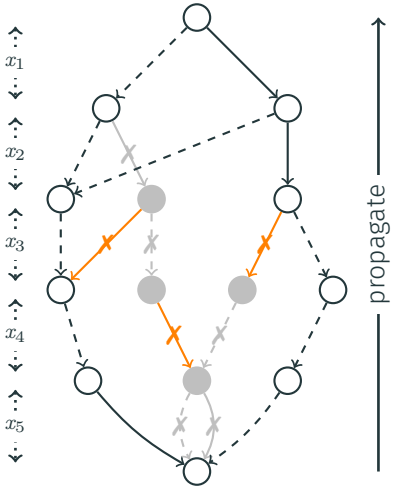
currArcs[x_3]
[1 0 0 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 0 0 1]

UCLouvain

Bottom up



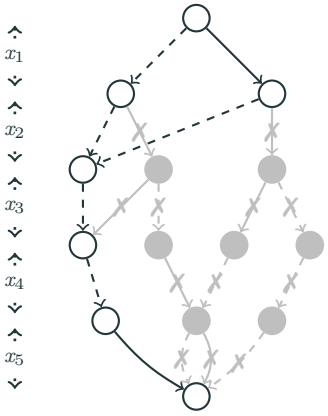
currArcs[x_1]
[1 1]

currArcs[x_2]
[1 0 1 1]

currArcs[x_3]
[1 0 0 0 1]

currArcs[x_4]
[1 0 0 1]

currArcs[x_5]
[1 0 0 1]

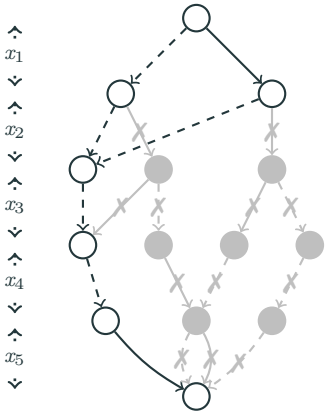


$\Delta_{x_2} = \{1\}$

x_1	x_2	x_3	x_4	x_5
$\{0,1\}$	$\{0\}$	$\{0,1\}$	$\{0,1\}$	$\{0,1\}$

(x,v)	currArcs[x]	supports[x,v]	$\cap$
$(x_1,0)$	11	10	10
$(x_1,1)$	11	01	01
$(x_3,0)$	10000	10101	10000
$(x_3,1)$	10000	01010	00000
$(x_4,0)$	1000	1011	1000
$(x_4,1)$	1000	0100	0000
$(x_5,0)$	1000	0101	0000
$(x_5,1)$	1000	1010	1000

UCLouvain



$\Delta_{x_2} = \{1\}$

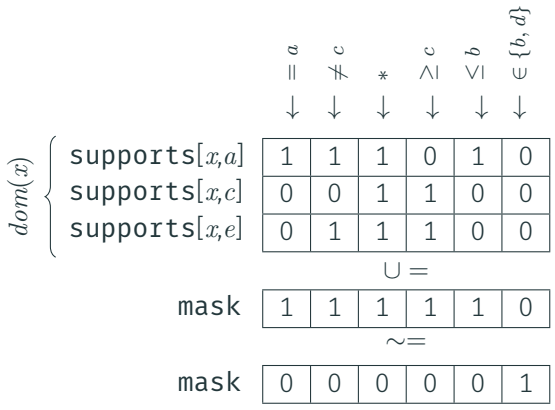
x_1	x_2	x_3	x_4	x_5
$\{0,1\}$	$\{0\}$	$\{0,\not{1}\}$	$\{0,\not{1}\}$	$\{\not{0},1\}$

(x,v)	currArcs[x]	supports[x,v]	$\cap$
$(x_1,0)$	11	10	10
$(x_1,1)$	11	01	01
$(x_3,0)$	10000	10101	10000
$(x_3,1)$	10000	01010	00000
$(x_4,0)$	1000	1011	1000
$(x_4,1)$	1000	0100	0000
$(x_5,0)$	1000	0101	0000
$(x_5,1)$	1000	1010	1000

The compact-diagram^{bs} algorithm, for bs-mvds

Algorithm: Direct removal part of the update

```
if layer without ∈ then
    if |Δ(x)| < |dom(x)| then
        Incremental update (=, ≠, ∗);
        Lower bound update (≤);
        Upper bound update (≥);
    else
        Reset update (=, ≠, ∗, ≤, ≥, ∈);
else
    Reset update (=, ≠, ∗, ≤, ≥, ∈);
```



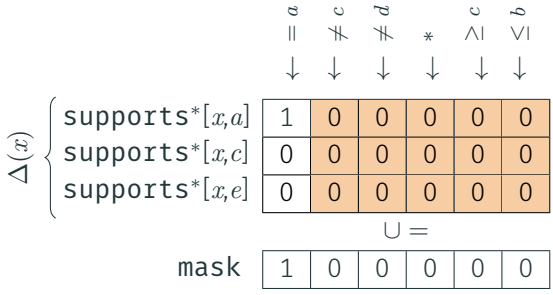
Algorithm: Reset update

foreach value $a \in dom(x)$ **do**
  mask$[x] \leftarrow$ mask$[x]$ |
    supports$[x, a]$;

mask$[x] \leftarrow \sim$ mask$[x]$;

A bit of supports$[x,a]$ is set to 1 if the label allows a.

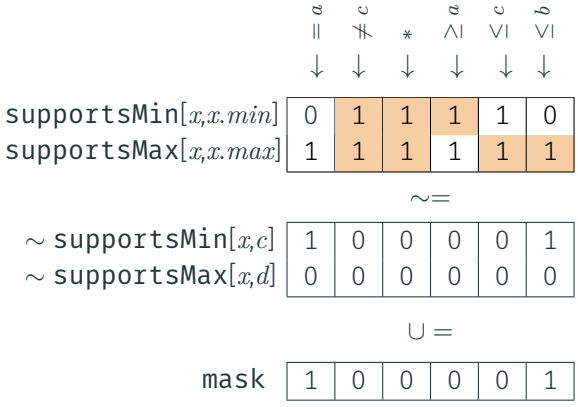
UCLouvain



Algorithm: Incremental update

foreach $value\ a \in \Delta_x$ **do**
  `mask`$[x] \leftarrow$ `mask`$[x]$ |
    `supports`$^*[x, a]$;

A bit of `supports`$^*[x,a]$ is set to 1 if the label allows **only** a.



Algorithm: Lower and upper bound updates

if $dom(x)$.minChanged() **then**
  mask$[x] \leftarrow$ mask$[x] \mid \sim$ supportsMin$[x, x.min]$;

if $dom(x)$.maxChanged() **then**
  mask$[x] \leftarrow$ mask$[x] \mid \sim$ supportsMax$[x, x.max]$;

A bit of supportsMin$[x,a]$ is set to 1 if the label allows at least a value $\geq a$.

A bit of supportsMax$[x,a]$ is set to 1 if the label allows at least a value $\leq a$.

UCLouvain

word 0				word 1				word 2			
w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	-	-
$=$	$\leq$	$\geq$	$\in$	$\neq$	$>$	$\notin$	$<$	$\neq$	$*$		

UCLouvain

word 0				word 1				word 2			
w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	-	-
$=$	$\leq$	$\geq$	$\in$	$\neq$	$>$	$\notin$	$<$	$\neq$	$*$		

$\Downarrow$

word 0				word 1				word 2				word 3			
w_0	w_4	w_8	w_9	w_3	w_6	-	-	w_1	w_7	-	-	w_2	w_5	-	-
$=$	$\neq$	$\neq$	$*$	$\in$	$\notin$			$\leq$	$<$			$\geq$	$>$		

UCLouvain

word 0				word 1				word 2				word 3			
w_0	w_4	w_8	w_9	w_3	w_6	-	-	w_1	w_7	-	-	w_2	w_5	-	-
$=$	$\neq$	$\neq$	$*$	$\in$	$\notin$			$\leq$	$<$			$\geq$	$>$		

Incremental or Reset update (depending if $|\Delta(x)| < |dom(x)|$)

Reset update

Lower bound update

Upper bound update

CD^{bs}: RESULTS

UCLouvain

A Table

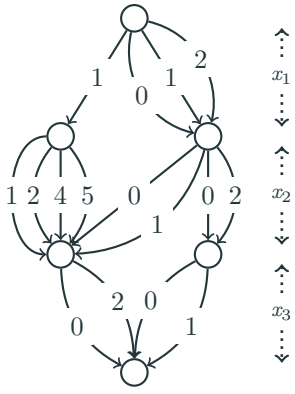
x_1	x_2	x_3
0	0	0
0	0	1
0	0	2
0	1	0
0	1	2
0	2	0
0	2	1
1	0	0
1	0	1
1	0	2
$\vdots$	$\vdots$	$\vdots$

UCLouvain

A Table

x_1	x_2	x_3
0	0	0
0	0	1
0	0	2
0	1	0
0	1	2
0	2	0
0	2	1
1	0	0
1	0	1
1	0	2
⋮	⋮	⋮

into an MVD

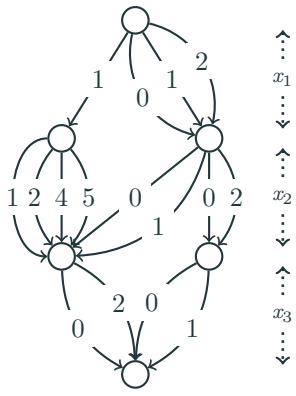


UCLouvain

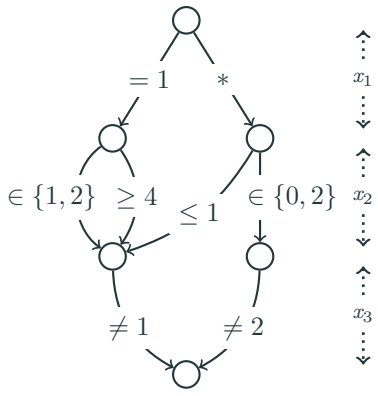
A Table

x_1	x_2	x_3
0	0	0
0	0	1
0	0	2
0	1	0
0	1	2
0	2	0
0	2	1
1	0	0
1	0	1
1	0	2
⋮	⋮	⋮

into an MVD



into a bs-MVD



A Table

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
0	2	1
0	3	0
0	3	1
1	0	0
1	0	1
1	1	0
$\vdots$	$\vdots$	$\vdots$

A Table

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
0	2	1
0	3	0
0	3	1
1	0	0
1	0	1
1	1	0
$\vdots$	$\vdots$	$\vdots$

into a bs-Table

x_1	x_2	x_3
$=1$	$=2$	≤ 1
$*$	$\neq 2$	≤ 1
$*$	≤ 2	$=1$

UCLouvain

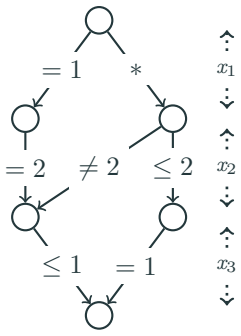
A Table

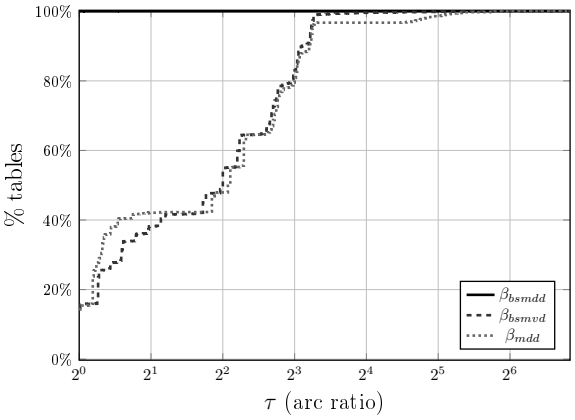
x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
0	2	1
0	3	0
0	3	1
1	0	0
1	0	1
1	1	0
⋮	⋮	⋮

into a bs-Table

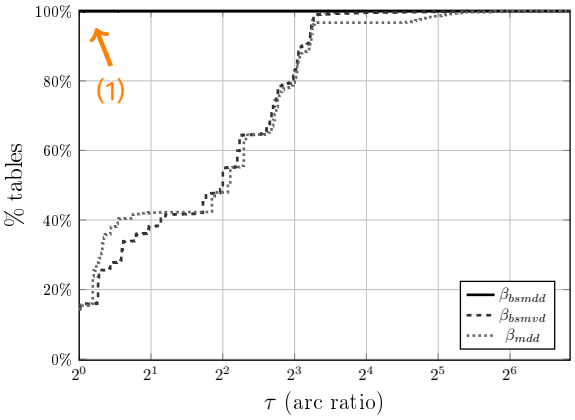
x_1	x_2	x_3
$=1$	$=2$	≤ 1
$*$	$\neq 2$	≤ 1
$*$	≤ 2	$=1$

into a bs-MVD



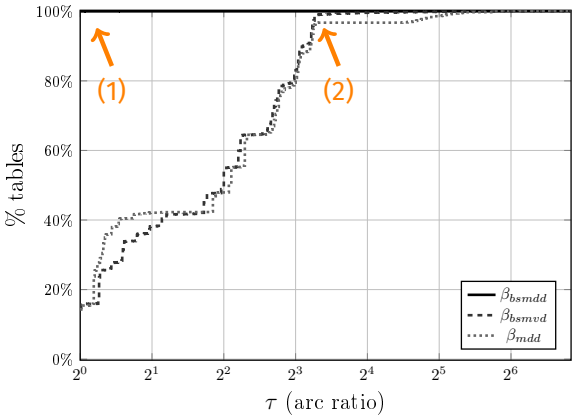

100%
80%
60%
40%
20%
0%
% tables
2^0
2^1
2^2
2^3
2^4
2^5
2^6
τ (arc ratio)
β_{bsmdd}
β_{bsmvd}
β_{mdd}

Arcs:



Arcs:

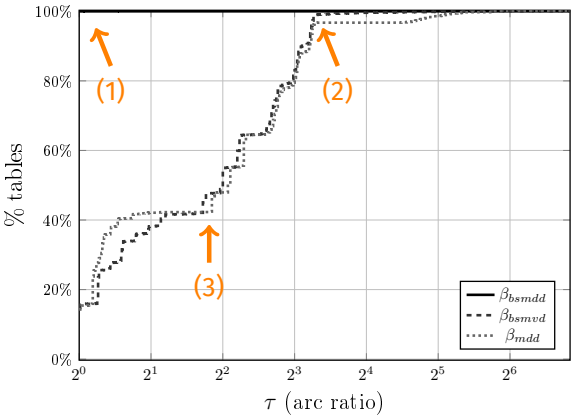
(1) bs-MDDs always less arcs



Arcs:

(1) bs-MDDs always less arcs

(2) bs-MDDs up to 10 times less arcs

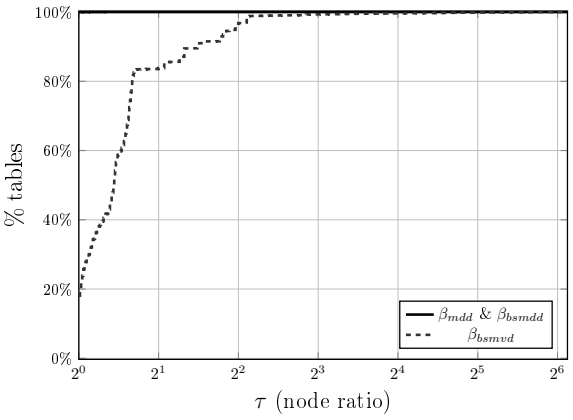


Arcs:

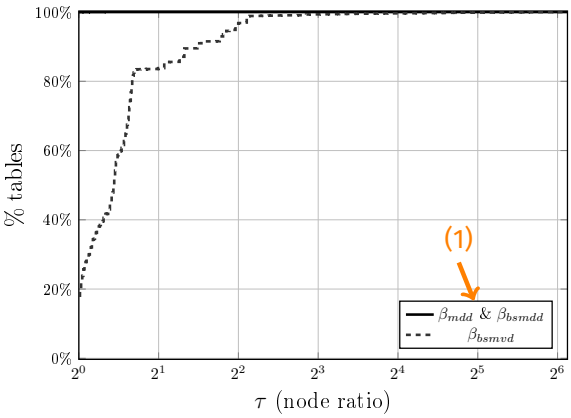
(1) bs-MDDs always less arcs

(2) bs-MDDs up to 10 times less arcs

(3) bs-MVDs and MDDs similar # arcs

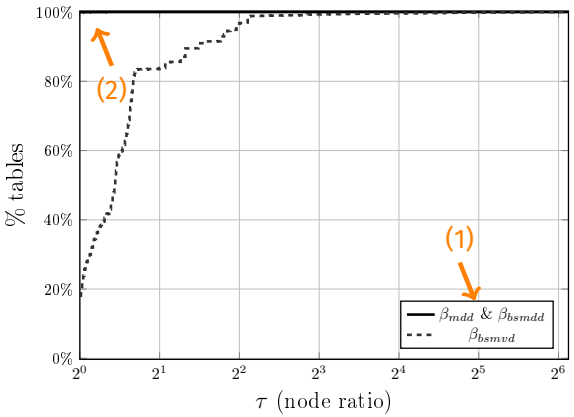

100%
80%
60%
40%
20%
0%
% tables
2^0
2^1
2^2
2^3
2^4
2^5
2^6
τ (node ratio)
β_{mdd} & β_{bsmdd}
β_{bsmvd}

Nodes:



Nodes:

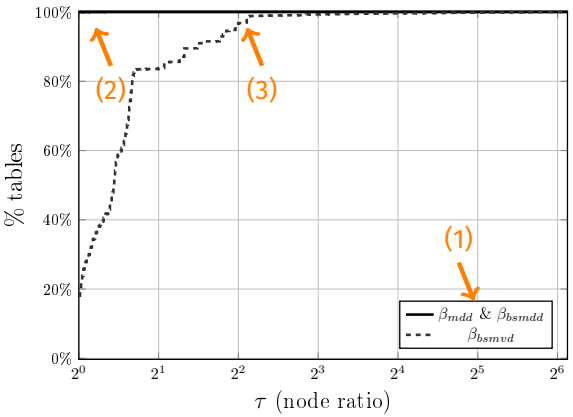
(1) bs-MDDs and MDDs same # nodes (by construction)



Nodes:

(1) bs-MDDs and MDDs same # nodes (by construction)

(2) bs-MDDs and MDDs always less nodes



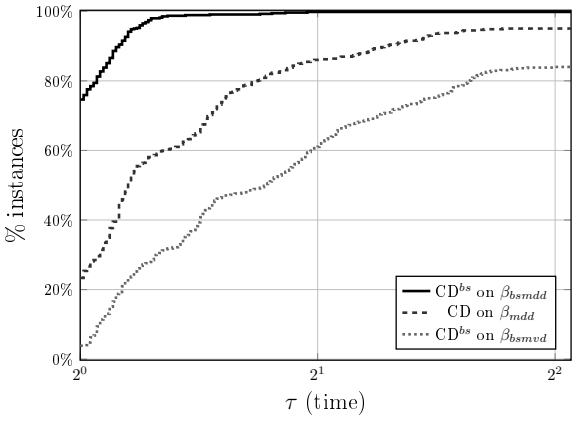
Nodes:

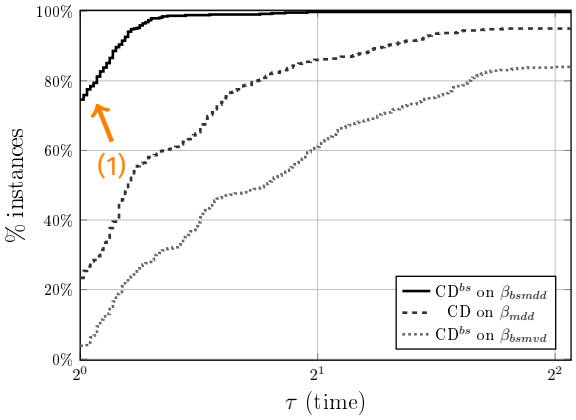
(1) bs-MDDs and MDDs same # nodes (by construction)

(2) bs-MDDs and MDDs always less nodes

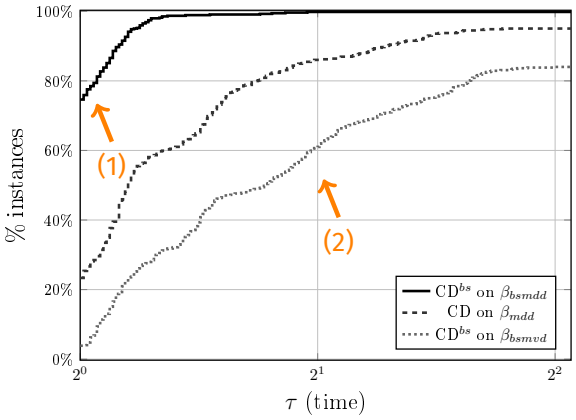
(3) bs-MVDs up to 4 times more nodes

RESULTS (4000 INSTANCES)





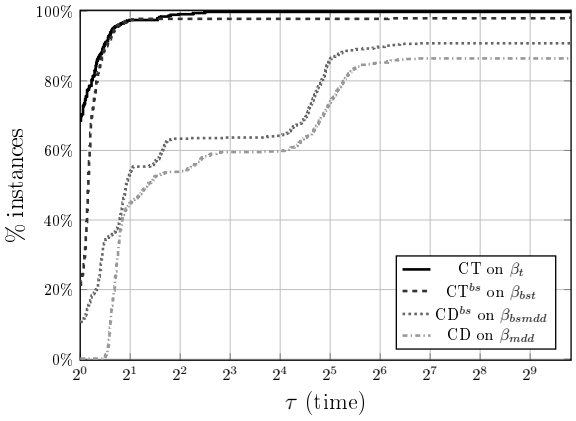
(1) CD^{bs} on bs-MDDs (fewer arcs) best 80% of the time

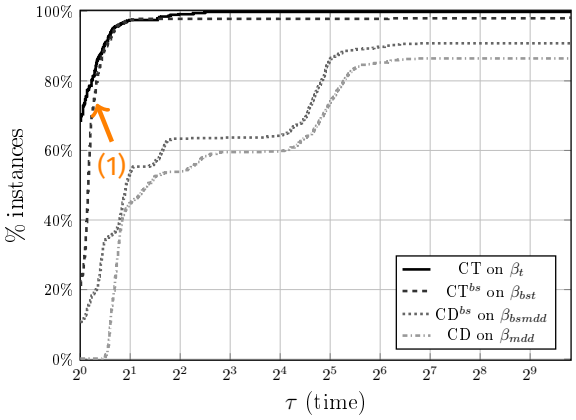


(1) CD^{bs} on bs-MDDs (fewer arcs) best 80% of the time

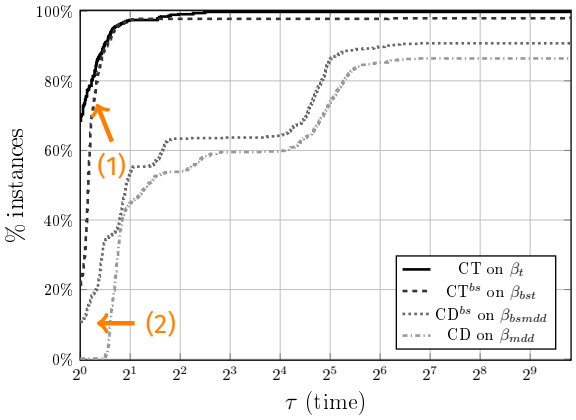
(2) CD^{bs} on bs-MVDs (more nodes) worst

RESULTS (4000 INSTANCES)



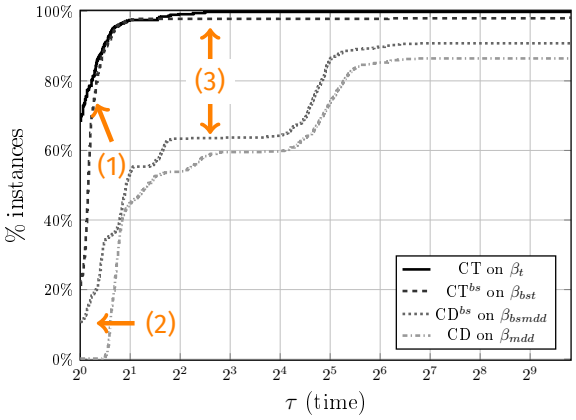


(1) CT and CT^{bs} still dominating



(1) CT and CT^{bs} still dominating

(2) CD^{bs} becomes efficient when compression is high

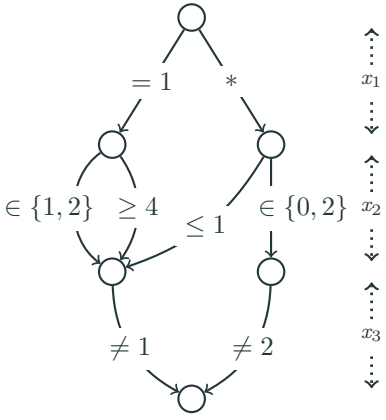


(1) CT and CT^{bs} still dominating

(2) CD^{bs} becomes efficient when compression is high

(3) gap reduced

- New type of layered graph (basic smart MVD) allowing **less edges**
- **Dedicated** propagator (CD^{bs})
- **Gap reduction** between table based (CT) and layered graph based (CD^{bs}) propagator



Thank you for listening!

Any questions?