SOLVING COMPLEX PROBLEMS: GRAPHS, CONSTRAINTS, AND MACHINE LEARNING IN ACTION

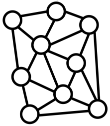
ModRef 2023

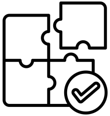
Hélène Verhaeghe

27 August 2023

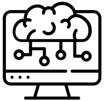
KULeuven, Leuven, Belgium, *helene.verhaeghe@kuleuven.be*

Graphs



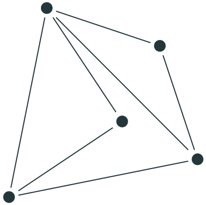


CP



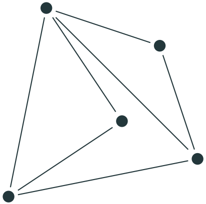
ML

Source of the icons: https://thenounproject.com/, designer Abd Majd, Smashing Stocks, and HAZHA undirected

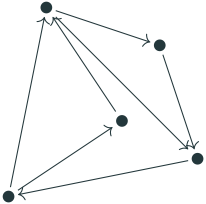


GRAPHS AS MODELING TOOL

undirected



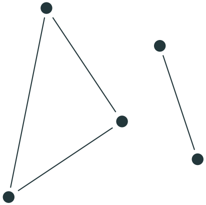
directed



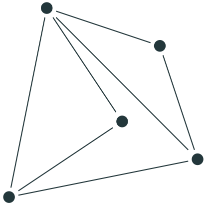
GRAPHS AS MODELING TOOL

KU LEUVEN

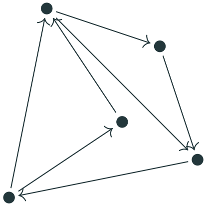
disconnected



undirected

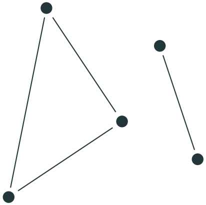


directed

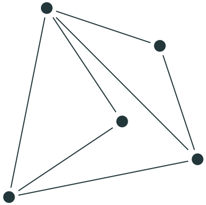


GRAPHS AS MODELING TOOL

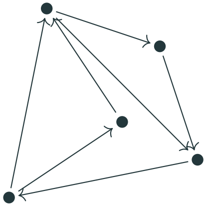
disconnected



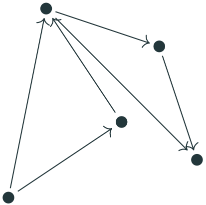
undirected



directed

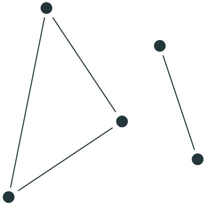


acyclic

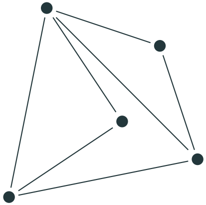


GRAPHS AS MODELING TOOL

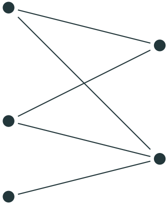
disconnected



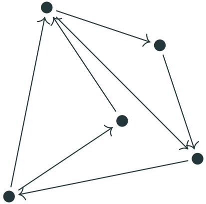
undirected



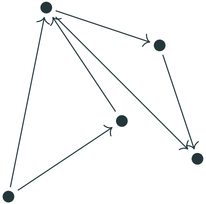
bipartite



directed

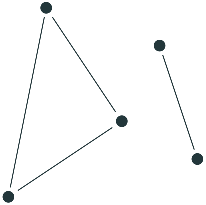


acyclic

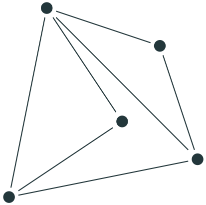


GRAPHS AS MODELING TOOL

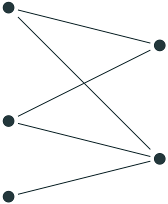
disconnected



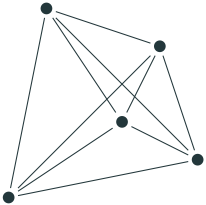
undirected



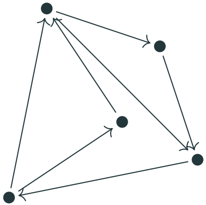
bipartite



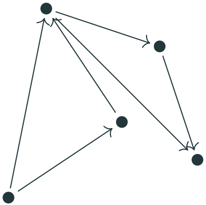
complete



directed



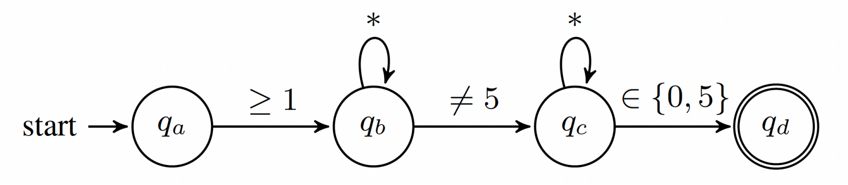
acyclic



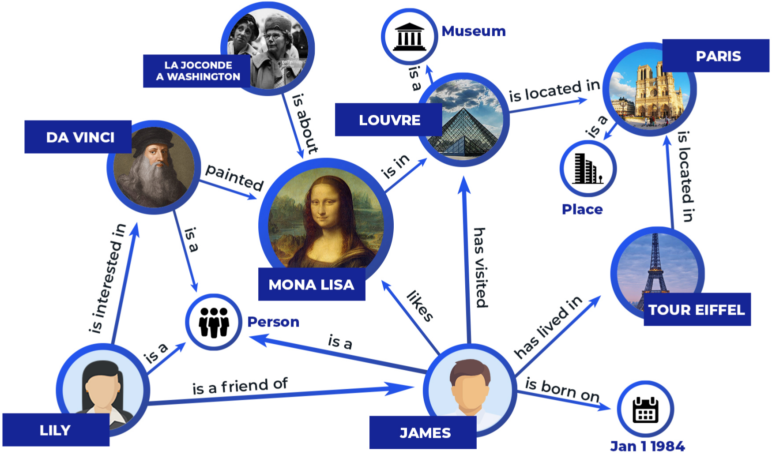
undirected
directed
disconnected
acyclic
bipartite
regular
complete
homogeneous
tree
planar
heterogeneous
Eulerian
weighted
⋮

GRAPHS AS MODELING TOOL: EXAMPLES

Source: https://www.huffpost.com/archive/qc/entry/le-roadtrip-ultime-aux-tats-unis-voici-le-trajet-ideal-pour-50_n_6852036


start
q_a
≥ 1
q_b
$*$
$\neq 5$
q_c
$*$
$\in \{0, 5\}$
q_d

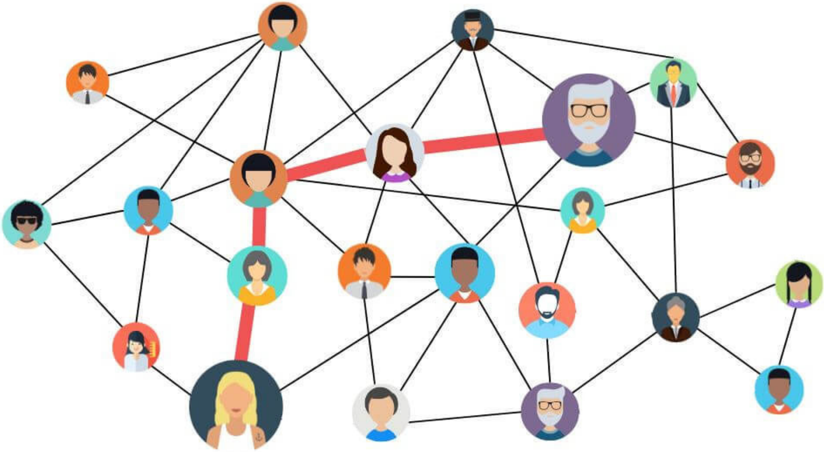
GRAPHS AS MODELING TOOL: EXAMPLES



Source: https://yashuseth.wordpress.com/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/

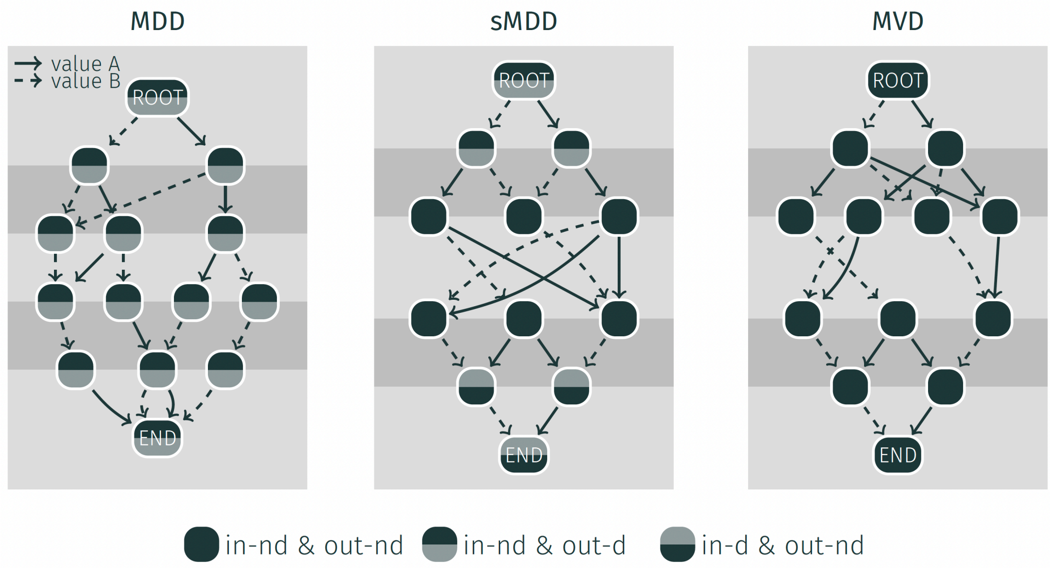
GRAPHS AS MODELING TOOL: EXAMPLES

KU LEUVEN

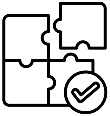


Source: https://greatpeopleinside.com/networking-particularities-men-women/

KU LEUVEN


MDD
value A
value B
ROOT
END
sMDD
ROOT
END
MVD
ROOT
END
in-nd & out-nd
in-nd & out-d
in-d & out-nd

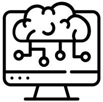
Model + Search



- Goal: Find (optimal) solution wrt some constraints
- Pro: Exact method
- Con: Difficulties in dealing with huge inputs

KU LEUVEN

(Big) Data + algorithms



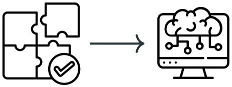
- Goal: Learn from examples
- Pro: Good with huge quantities of data
- Con: Difficulties to satisfy (hard) constraints in outputs

Can we get the best of both worlds?

Yes, by combining them!

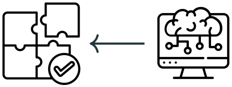
HOW TO COMBINE THEM?

KU LEUVEN



CP for ML

- Modeling ML problems (e.g., clustering using CP)
- Joint inference on NN output (e.g., visual sudoku problem)
- Improving the learning of NN (e.g., PLS experiment)

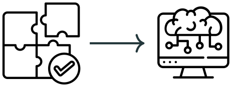


ML for CP

- Algorithm configuration (e.g., Sunny-CP solver)
- Learning to branch (e.g., SeaPearl project)
- Constraint acquisition (e.g., ClassAcq approach)

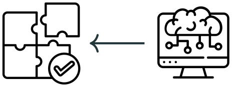
And many many other examples …

MY OWN PROJECTS … RELATED TO GRAPHS!



CP for ML

- Optimal decision trees
- CP-BP for learning



ML for CP

- Solving RCPSP using GNNs

WHEN CP HELPS ML: OPTIMAL DECISION TREES

Database

f_1	f_2	f_3	$\dots$	f_n	c
1	0	1	$\dots$	1	$+$
0	1	0	$\dots$	1	$-$
1	1	0	$\dots$	0	$+$
0	0	0	$\dots$	0	$+$
1	0	0	$\dots$	0	$+$
0	1	1	$\dots$	1	$-$
1	1	1	$\dots$	0	$-$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
1	1	1	$\dots$	1	$+$

- already a binary database

is green	produce gum	has flowers	poisonous?
yes	yes	no	$+$
no	yes	yes	$-$

- binarization required

height	age	F	sick?
134	34	1.45	$+$
178	23	3.66	$-$

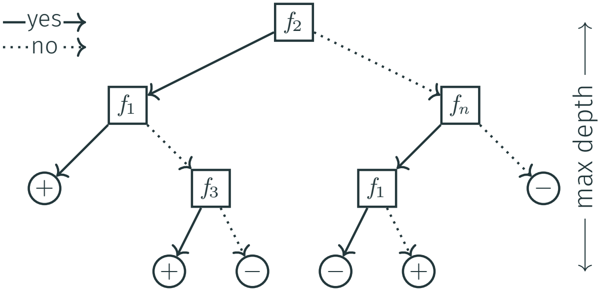
height< 150	height< 180	F< 1	$\dots$	sick?
yes	yes	no	$\dots$	$+$
no	yes	no	$\dots$	$-$

Database

f_1	f_2	f_3	$\dots$	f_n	c
1	0	1	$\dots$	1	$+$
0	1	0	$\dots$	1	$-$
1	1	0	$\dots$	0	$+$
0	0	0	$\dots$	0	$+$
1	0	0	$\dots$	0	$+$
0	1	1	$\dots$	1	$-$
1	1	1	$\dots$	0	$-$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
1	1	1	$\dots$	1	$+$

Database

f_1	f_2	f_3	$\dots$	f_n	c
1	0	1	$\dots$	1	+
0	1	0	$\dots$	1	−
1	1	0	$\dots$	0	+
0	0	0	$\dots$	0	+
1	0	0	$\dots$	0	+
0	1	1	$\dots$	1	−
1	1	1	$\dots$	0	−
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
1	1	1	$\dots$	1	+



$$\min \sum (pred(i) - c(i))$$

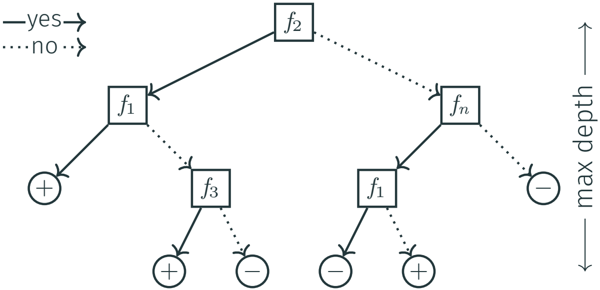
THE PROBLEM: LEARNING OPTIMAL DECISION TREES

Database

f_1	f_2	f_3	$\dots$	f_n	c
1	0	1	$\dots$	1	+
0	1	0	$\dots$	1	−
1	1	0	$\dots$	0	+
0	0	0	$\dots$	0	+
1	0	0	$\dots$	0	+
0	1	1	$\dots$	1	−
1	1	1	$\dots$	0	−
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
1	1	1	$\dots$	1	+

New sample

0	0	1	$\dots$	0	?



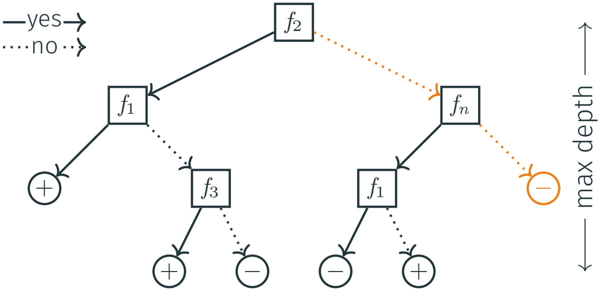
$$\min \sum (pred(i) - c(i))$$

Database

f_1	f_2	f_3	$\dots$	f_n	c
1	0	1	$\dots$	1	+
0	1	0	$\dots$	1	−
1	1	0	$\dots$	0	+
0	0	0	$\dots$	0	+
1	0	0	$\dots$	0	+
0	1	1	$\dots$	1	−
1	1	1	$\dots$	0	−
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
1	1	1	$\dots$	1	+

New sample

0	0	1	$\dots$	0	−



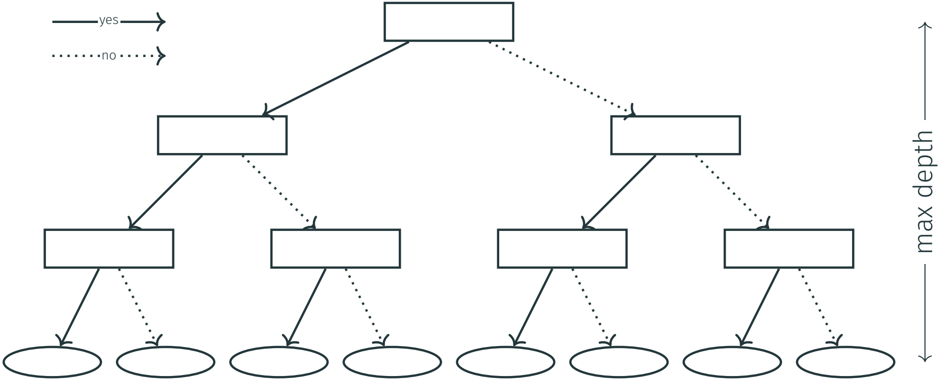
$$\min \sum (pred(i) - c(i))$$

Greedy methods:

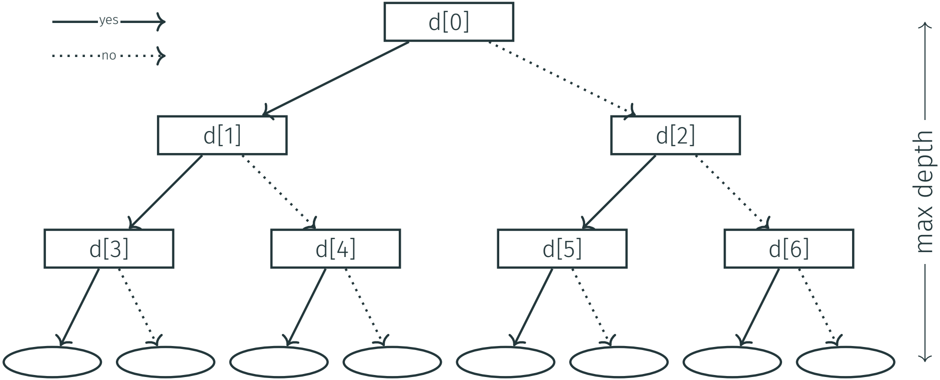
- ✔ easy construction
- ✗ hard to impose additional constraints
- ✗ potentially unnecessarily complex tree

- Mining optimal decision trees from itemset lattices, Nijssen, S., Fromont, E., 2007
- Minimising decision tree size as combinatorial optimisation, Bessiere, C., Hebrard, E., O'Sullivan, B., 2009
- Optimal constraint-based decision tree induction from itemset lattices,Nijssen, S., Fromont, É., 2010
- **Optimal classification trees**, Bertsimas, D., Dunn, J., 2017
- Learning optimal decision trees with sat, Narodytska, N., Ignatiev, A., Pereira, F., Marques-Silva, J., RAS, I., 2018
- Learning optimal and fair decision trees for non-discriminative decision-making, Aghaei, S., Azizi, M.J., Vayanos, P., 2019
- Learning optimal classification trees using a binary linear program formulation, Verwer, S., Zhang, Y., 2019

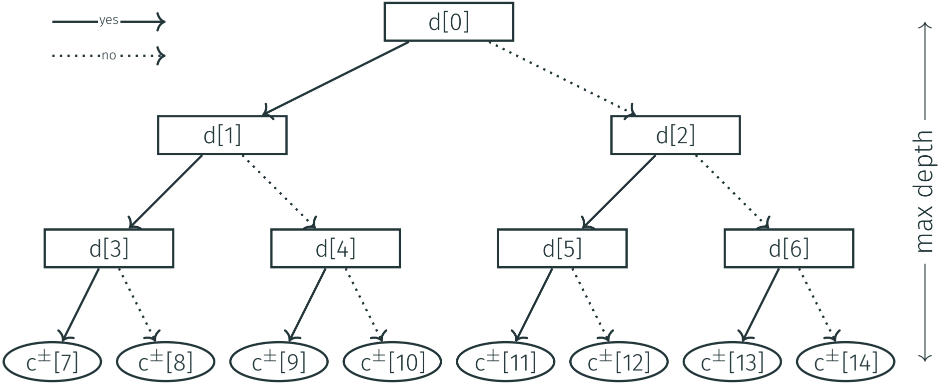
MODEL - DECISION VARIABLES



MODEL - DECISION VARIABLES

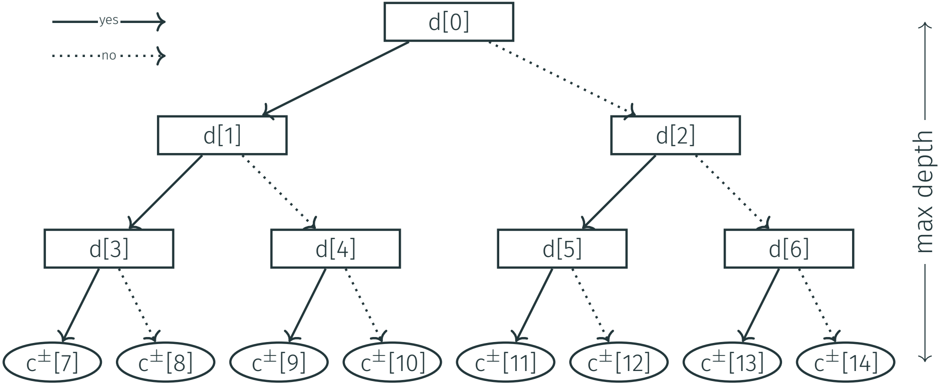


$$dom(d[i]) = \{1, ..., n\}$$



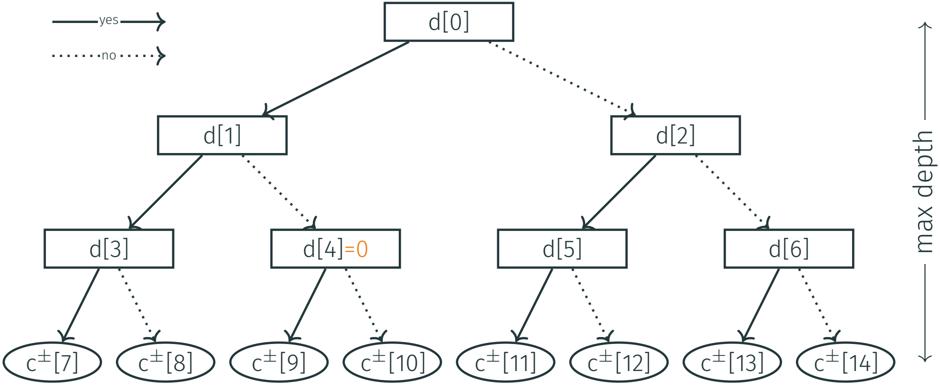
$$dom(d[i]) = \{1, ..., n\}$$

$$dom(c[i]) = \{0, ..., N\}$$



$dom(d[i]) = \{0, 1, ..., n\}$

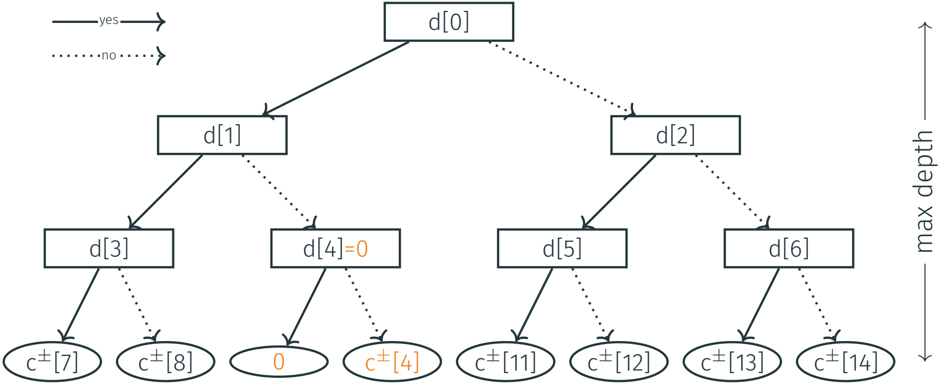
$dom(c[i]) = \{0, ..., N\}$



$dom(d[i]) = \{0, 1, ..., n\}$

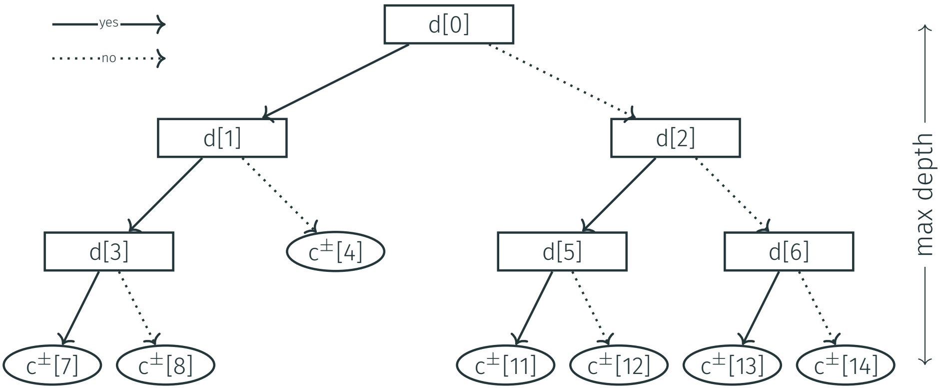
$dom(c[i]) = \{0, ..., N\}$

KU LEUVEN



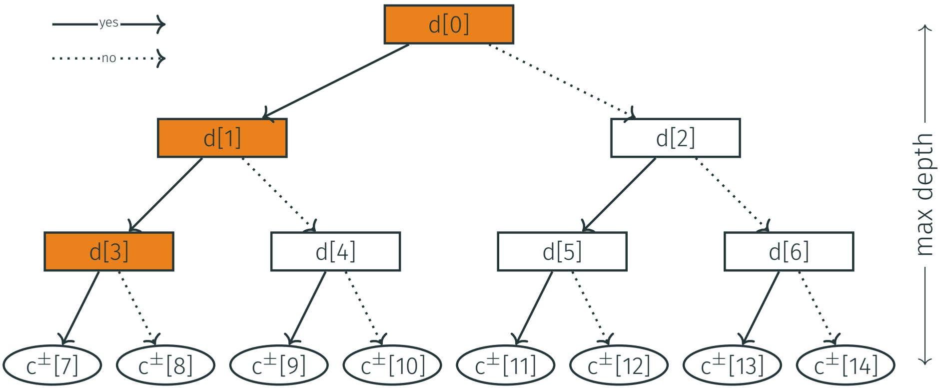
$dom(d[i]) = \{0, 1, ..., n\}$

$dom(c[i]) = \{0, ..., N\}$



$dom(d[i]) = \{0, 1, ..., n\}$

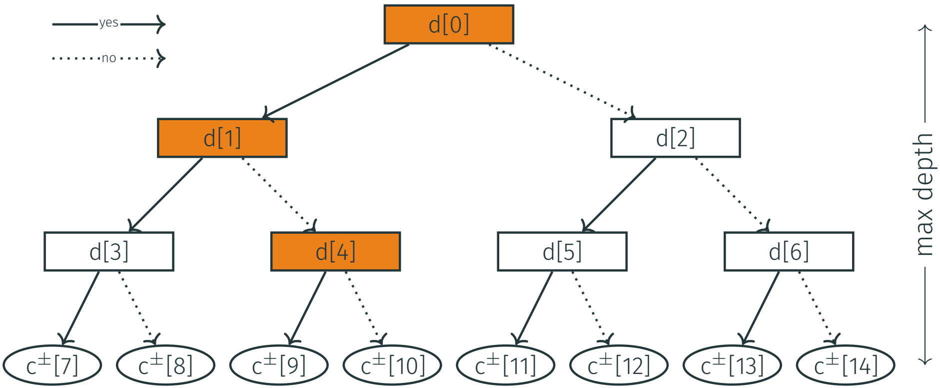
$dom(c[i]) = \{0, ..., N\}$



$dom(d[i]) = \{0, 1, ..., n\}$

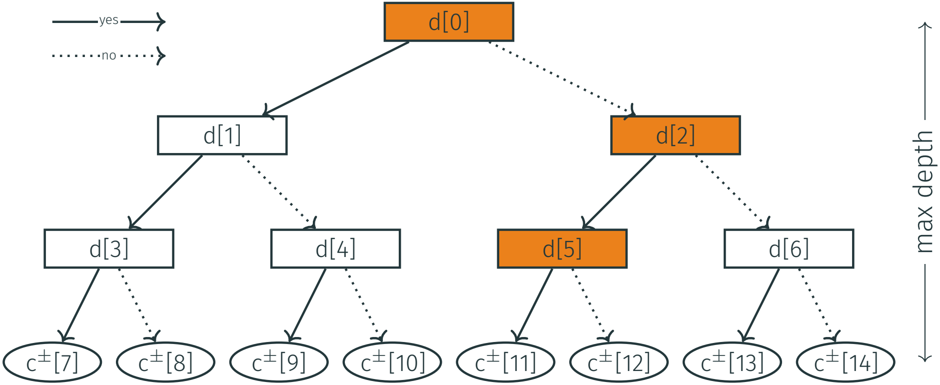
$dom(c[i]) = \{0, ..., N\}$

KU LEUVEN



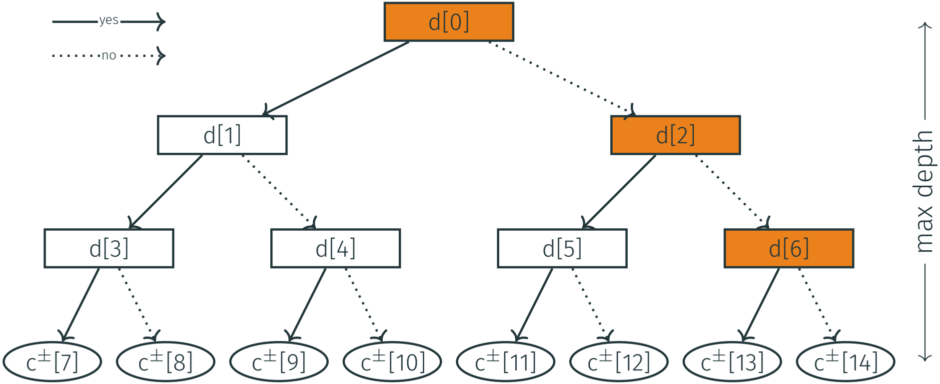
$dom(d[i]) = \{0, 1, ..., n\}$

$dom(c[i]) = \{0, ..., N\}$



$dom(d[i]) = \{0, 1, ..., n\}$

$dom(c[i]) = \{0, ..., N\}$



$dom(d[i]) = \{0, 1, ..., n\}$

$dom(c[i]) = \{0, ..., N\}$

f_1	f_2	f_3	f_4
1	0	1	1
0	1	0	1
1	1	0	0
0	0	0	0
1	0	0	0
0	1	1	1
1	1	1	0
1	1	1	1

Features (Dense)				Counter
x_1	x_2	x_3	x_4	

P. Schaus, J. Aoga, and T. Guns. "Coversize: A global constraint for frequency-based itemset mining". In CP 2017.

f_1	f_2	f_3	f_4
1	0	1	1
0	1	0	1
1	1	0	0
0	0	0	0
1	0	0	0
0	1	1	1
1	1	1	0
1	1	1	1

Features (Dense)				Counter
x_1	x_2	x_3	x_4	
0	1	0	1	

P. Schaus, J. Aoga, and T. Guns. "Coversize: A global constraint for frequency-based itemset mining". In CP 2017.

f_1	f_2	f_3	f_4
1	0	1	1
0	1	0	1
1	1	0	0
0	0	0	0
1	0	0	0
0	1	1	1
1	1	1	0
1	1	1	1

Features (Dense)				Counter
x_1	x_2	x_3	x_4	
0	1	0	1	3

P. Schaus, J. Aoga, and T. Guns. "Coversize: A global constraint for frequency-based itemset mining". In CP 2017.

f_1	f_2	f_3	f_4
1	0	1	1
0	1	0	1
1	1	0	0
0	0	0	0
1	0	0	0
0	1	1	1
1	1	1	0
1	1	1	1

Features (Dense)				Counter
x_1	x_2	x_3	x_4	
0	1	0	1	3

- Dense representation
- No feature rejection

KU LEUVEN

f_1	f_2	f_3	f_4
1	0	1	1
0	1	0	1
1	1	0	0
0	0	0	0
1	0	0	0
0	1	1	1
1	1	1	0
1	1	1	1

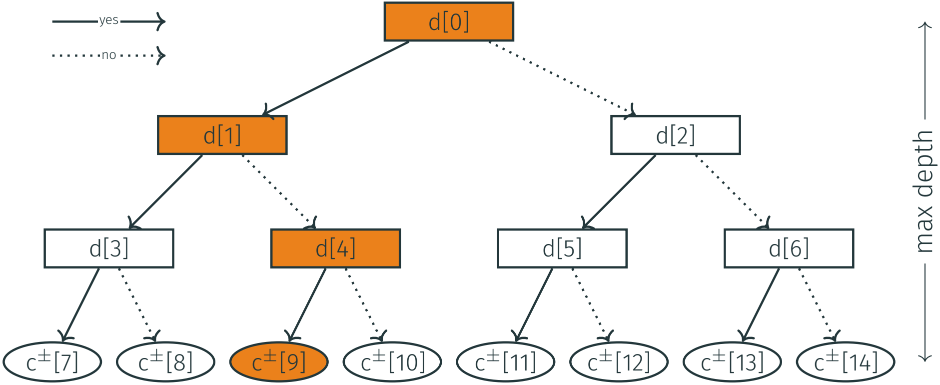
Features (Sparse)		Counter
y_1	y_2	
2	4	3

- ~~Dense representation~~
- No feature rejection

f_1	f_2	f_3	f_4
1	0	1	1
0	1	0	1
1	1	0	0
0	0	0	0
1	0	0	0
0	1	1	1
1	1	1	0
1	1	1	1

✔Features (Sparse)		✗Features (Sparse)	Counter
y_1	y_2	z_1	
2	4	3	1

- ~~Dense representation~~
- ~~No feature rejection~~

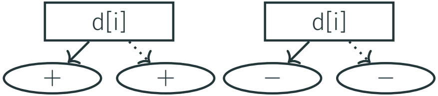


$Coversize(\{d[0], d[4]\}, \{d[1]\}, c^{+}[9])$ $\qquad$ $Coversize(\{d[0], d[4]\}, \{d[1]\}, c^{-}[9])$

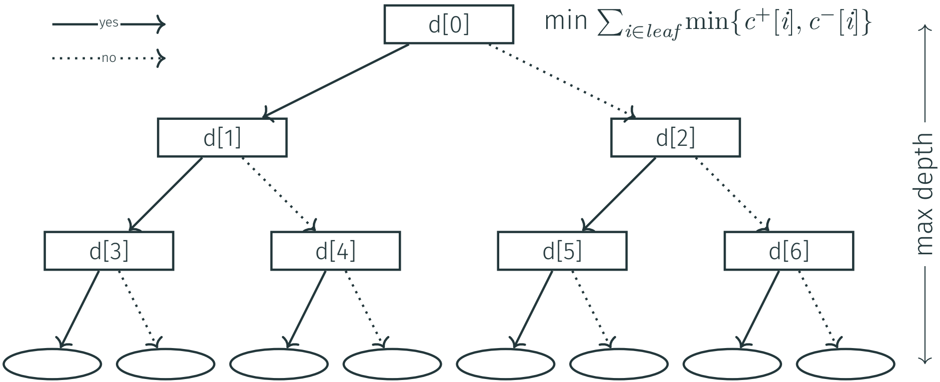
- constraints imposing minimum at leaf

$$c^{+}[i] + c^{-}[i] \geq N_{min}$$

- constraints avoiding useless decisions



- redundant constraints improving speed

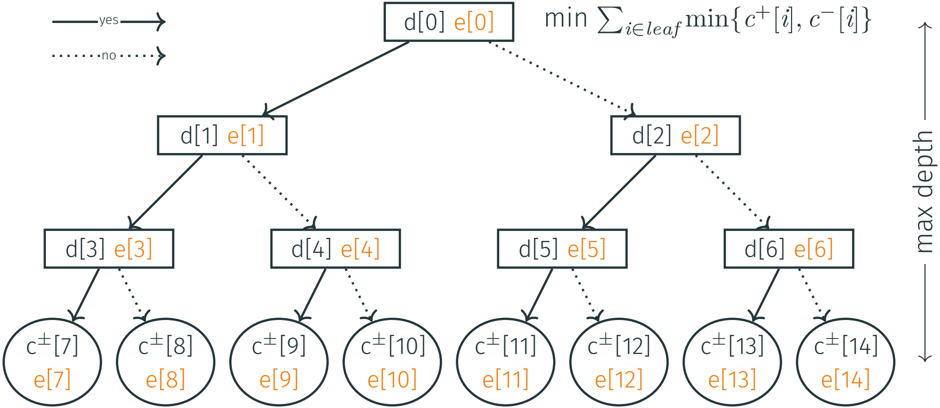


$\min \sum_{i \in leaf} \min\{c^+[i], c^-[i]\}$

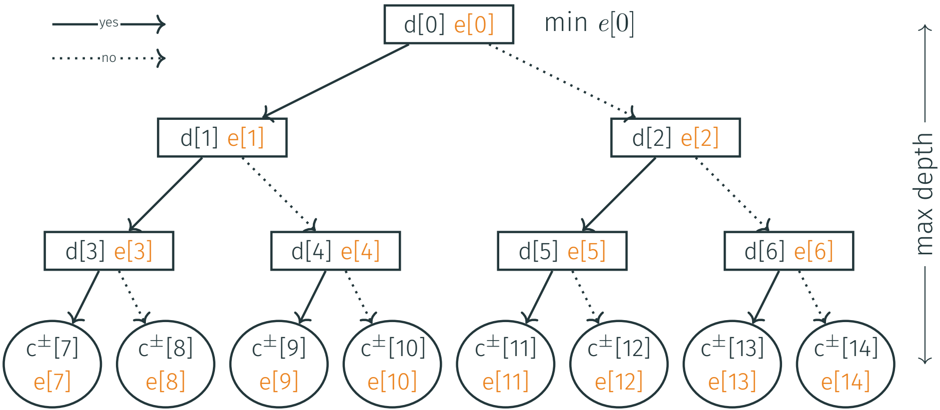
$dom(d[i]) = \{0, 1, ..., n\}$

$dom(c[i]) = \{0, ..., N\}$

$$\min \sum_{i \in leaf} \min\{c^+[i], c^-[i]\}$$

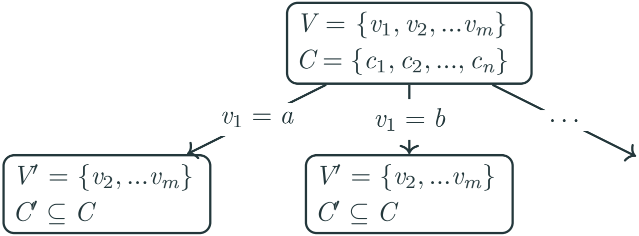


$dom(d[i]) = \{0, 1, ..., n\}$ $\quad dom(c[i]) = \{0, ..., N\}$ $\quad dom(e[i]) = \{0, ..., N\}$



$dom(d[i]) = \{0, 1, ..., n\}$ $dom(c[i]) = \{0, ..., N\}$ $dom(e[i]) = \{0, ..., N\}$

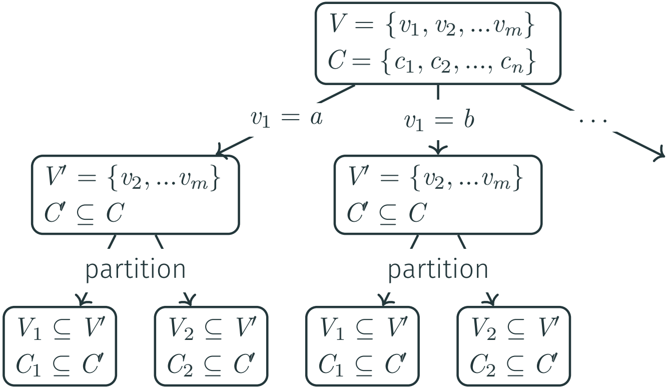
AND/OR SEARCH TREE



OR nodes

SOL = SOL_1 or SOL_2 or ...

AND/OR SEARCH TREE

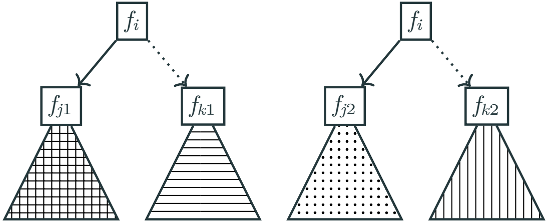


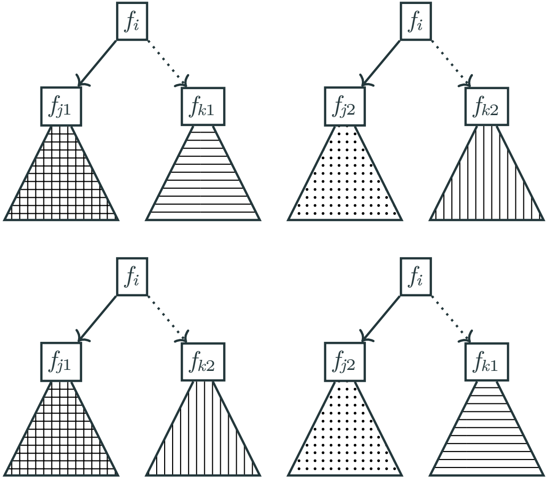
OR nodes

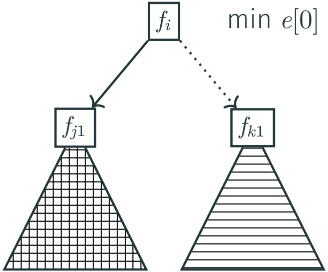
SOL = SOL_1 or SOL_2 or ...

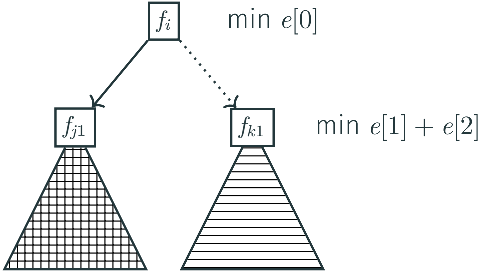
AND nodes

SOL = SOL_1 and SOL_2 and ...

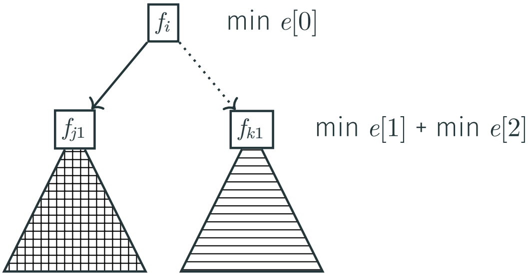

f_i
f_{j1}
f_{k1}
f_i
f_{j2}
f_{k2}

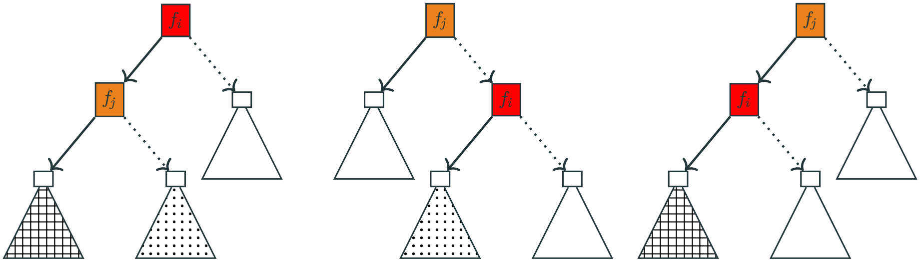

f_i
f_{j1}
f_{k1}
f_i
f_{j2}
f_{k2}
f_i
f_{j1}
f_{k2}
f_i
f_{j2}
f_{k1}


f_i
min $e[0]$
f_{j1}
f_{k1}

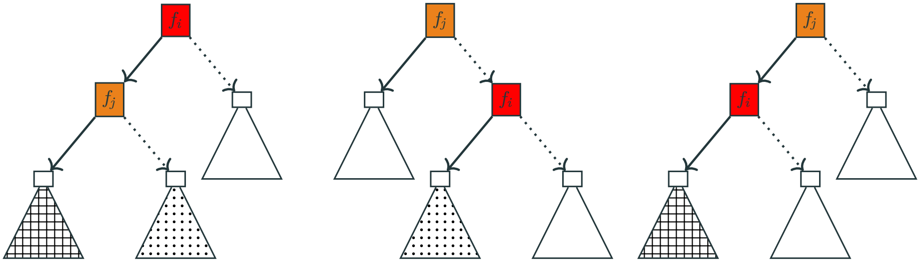


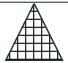
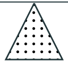
f_i
min $e[0]$
f_{j1}
f_{k1}
min $e[1] + e[2]$


f_i
min $e[0]$
f_{j1}
f_{k1}
min $e[1]$ + min $e[2]$


f_i
f_j
f_j
f_i
f_j
f_i

SEARCH – CACHING



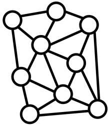
	yes	no	hash
	f_i f_j		f_i, f_j-
	f_i	f_j	$f_i - f_j$

	$N_{\min}=1$			$N_{\min}=5$			
	DL8	BinOCT	CP	DL8	CP	CP-c	CP-m
Proven optimality	49(64%)	13(17%)	**57**(75%)	54(71%)	56(74%)	56(74%)	**58**(76%)
Best solution found	49(64%)	21(28%)	**76**(100%)	54(71%)	**74**(97%)	**74**(97%)	70(92%)
Fastest	23(30%)	11(14%)	**49**(64%)	28(37%)	**40**(53%)	33(43%)	22(29%)
Time out	27(36%)	63(83%)	**19**(25%)	22(29%)	21(28%)	21(28%)	**19**(25%)

23 instances, depths from 2 to 5, 10 min TO

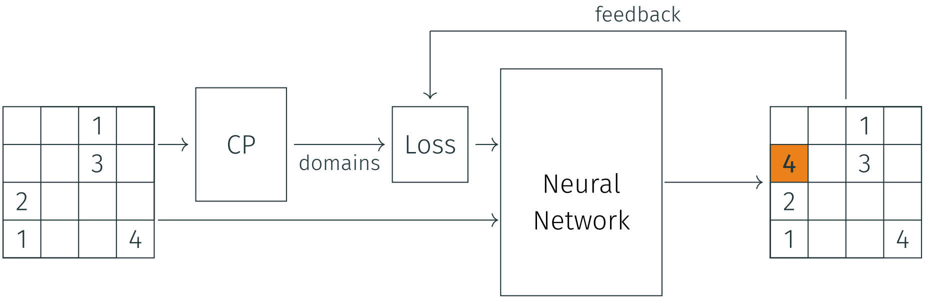
DL8: Dynamic programming approach using frequent itemsets mining

BinOCT: MIP-based approach running on CPLEX



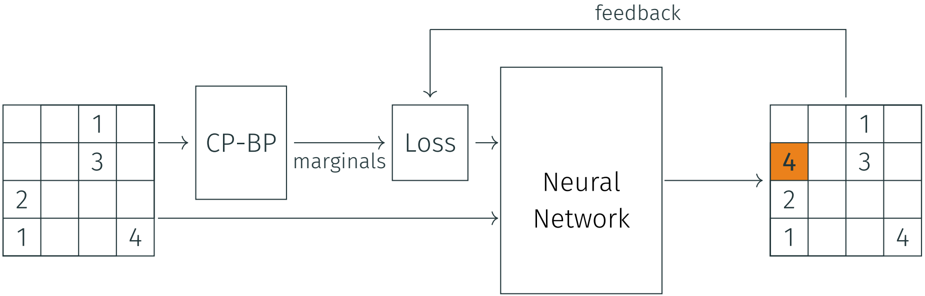
- Type of graph used: Decision trees
- Key proprieties:
  - Sub-tree independence
  - Path equivalence
- How this helps:
  - Reduction of symmetries
  - Caching possible

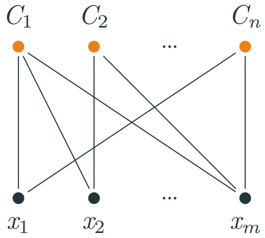
WHEN CP HELPS ML: CP-BP FOR LEARNING



M. Silvestri, M. Lombardi, and M. Milano. ”Injecting domain knowledge in neural networks: a controlled experiment on a constrained problem”. In CPAIOR 2021.

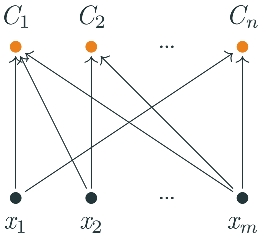
THE PROBLEM: MAKE NN OUTPUT SATISFY CONSTRAINTS





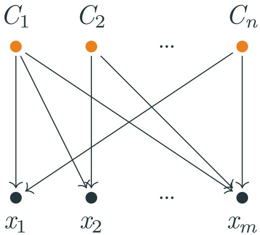
G. Pesant. "From support propagation to belief propagation in constraint programming". In JAIR 2019.

Message passing between variables and constraints



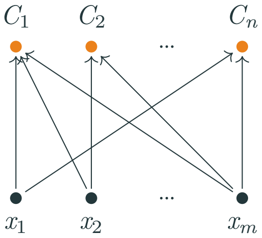
G. Pesant. "From support propagation to belief propagation in constraint programming". In JAIR 2019.

Message passing between variables and constraints



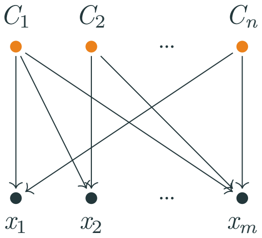
G. Pesant. "From support propagation to belief propagation in constraint programming". In JAIR 2019.

Message passing between variables and constraints



G. Pesant. "From support propagation to belief propagation in constraint programming". In JAIR 2019.

Message passing between variables and constraints



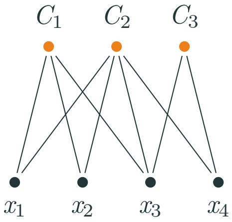
G. Pesant. "From support propagation to belief propagation in constraint programming". In JAIR 2019.

Variables: x_a, x_b, x_c, x_d

- $D_{x_a} = D_{x_a} = D_{x_a} = D_{x_a} = \{1, 2, 3, 4\}$

Constraints:

- $C_1 := \mathit{AllDifferent}(x_a, x_b, x_c)$
- $C_2 := x_a + x_b + x_c + x_d = 7$
- $C_3 := x_c \leq x_d$



True marginals (target)

	1	2	3	4
θ_{x_a}	0	.5	.5	0
θ_{x_b}	0	.5	.5	0
θ_{x_c}	1	0	0	0
θ_{x_d}	1	0	0	0

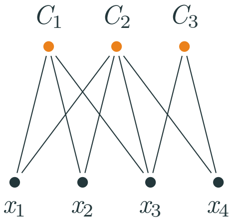
Two solutions: $(2, 3, 1, 1)$ and $(3, 2, 1, 1)$

Variables: x_a, x_b, x_c, x_d

- $D_{x_a} = D_{x_a} = D_{x_a} = D_{x_a} = \{1, 2, 3, 4\}$

Constraints:

- $C_1 := \mathit{AllDifferent}(x_a, x_b, x_c)$
- $C_2 := x_a + x_b + x_c + x_d = 7$
- $C_3 := x_c \leq x_d$



Two solutions: $(2, 3, 1, 1)$ and $(3, 2, 1, 1)$

True marginals (target)

	1	2	3	4
θ_{x_a}	0	.5	.5	0
θ_{x_b}	0	.5	.5	0
θ_{x_c}	1	0	0	0
θ_{x_d}	1	0	0	0

Marginals at iteration 0

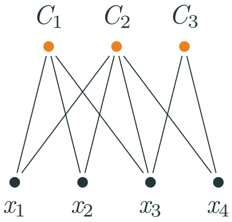
	1	2	3	4
$\hat{\theta}_{x_a}$	.25	.25	.25	.25
$\hat{\theta}_{x_b}$	.25	.25	.25	.25
$\hat{\theta}_{x_c}$	.25	.25	.25	.25
$\hat{\theta}_{x_d}$	.25	.25	.25	.25

Variables: x_a, x_b, x_c, x_d

- $D_{x_a} = D_{x_a} = D_{x_a} = D_{x_a} = \{1, 2, 3, 4\}$

Constraints:

- $C_1 := AllDifferent(x_a, x_b, x_c)$
- $C_2 := x_a + x_b + x_c + x_d = 7$
- $C_3 := x_c \leq x_d$



Two solutions: $(2, 3, 1, 1)$ and $(3, 2, 1, 1)$

True marginals (target)

	1	2	3	4
θ_{x_a}	0	.5	.5	0
θ_{x_b}	0	.5	.5	0
θ_{x_c}	1	0	0	0
θ_{x_d}	1	0	0	0

Marginals at iteration 1

	1	2	3	4
$\hat{\theta}_{x_a}$	.50	.30	.15	.05
$\hat{\theta}_{x_b}$	.50	.30	.15	.05
$\hat{\theta}_{x_c}$	.62	.28	.09	.01
$\hat{\theta}_{x_d}$	.29	.34	.26	.11

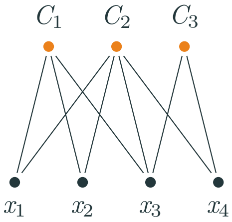
BELIEF PROPAGATION: EXAMPLE

Variables: x_a, x_b, x_c, x_d

- $D_{x_a} = D_{x_a} = D_{x_a} = D_{x_a} = \{1, 2, 3, 4\}$

Constraints:

- $C_1 := \mathit{AllDifferent}(x_a, x_b, x_c)$
- $C_2 := x_a + x_b + x_c + x_d = 7$
- $C_3 := x_c \leq x_d$



Two solutions: $(2, 3, 1, 1)$ and $(3, 2, 1, 1)$

True marginals (target)

	1	2	3	4
θ_{x_a}	0	.5	.5	0
θ_{x_b}	0	.5	.5	0
θ_{x_c}	1	0	0	0
θ_{x_d}	1	0	0	0

Marginals at iteration 10

	1	2	3	4
$\hat{\theta}_{x_a}$	.01	.52	.46	.01
$\hat{\theta}_{x_b}$	.01	.52	.46	.01
$\hat{\theta}_{x_c}$	.98	.02	.00	.00
$\hat{\theta}_{x_d}$	.90	.10	.00	.00

KU LEUVEN

Domains

	1	2	3	4
D_{x_a}	0	1	1	0
D_{x_b}	0	1	1	0
D_{x_c}	1	0	0	0
D_{x_d}	1	0	0	0

Marginals

	1	2	3	4
$\hat{\theta}_{x_a}$	.01	.52	.46	.01
$\hat{\theta}_{x_b}$	.01	.52	.46	.01
$\hat{\theta}_{x_c}$	.98	.02	.00	.00
$\hat{\theta}_{x_d}$	.90	.10	.00	.00

KU LEUVEN

$$Loss(x,y) = \underbrace{-\langle y, \log(\frac{1}{Z}\overbrace{f(x)}^{\hat{y}})\rangle}_{\texttt{cross entropy}} + \overbrace{\lambda}^{\texttt{weight}} \cdot \underbrace{t(x)}_{\texttt{CP feedback}}$$

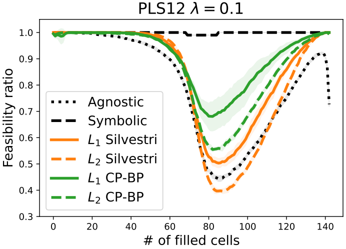
Domains

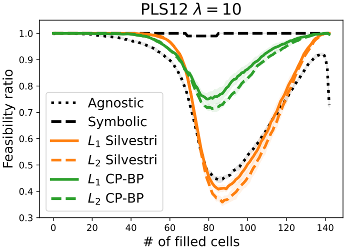
$$\begin{aligned} t(x) = L_1(x, C) &= \sum_k |C_k(x) - f_k(x)| \\ t(x) = L_2(x, C) &= \sum_k (C_k(x) - f_k(x))^2 \\ C_k(x) &\in \{0,1\} \end{aligned}$$

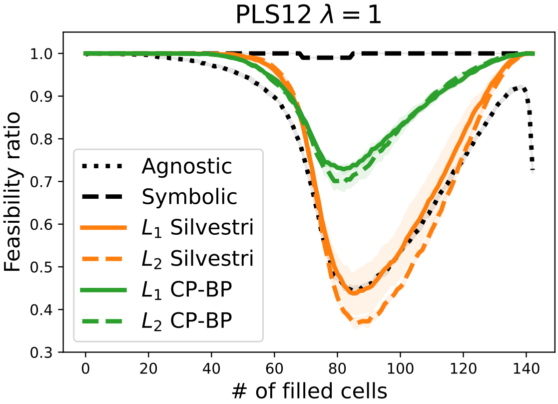
Marginals

$$\begin{aligned} t(x) = L_1(x, \hat{\theta}) &= \sum_k |\hat{\theta}_k(x) - f_k(x)| \\ t(x) = L_2(x, \hat{\theta}) &= \sum_k (\hat{\theta}_k(x) - f_k(x))^2 \\ \hat{\theta}_k(x) &\in [0,1] \end{aligned}$$

RESULTS – SUPERVISED







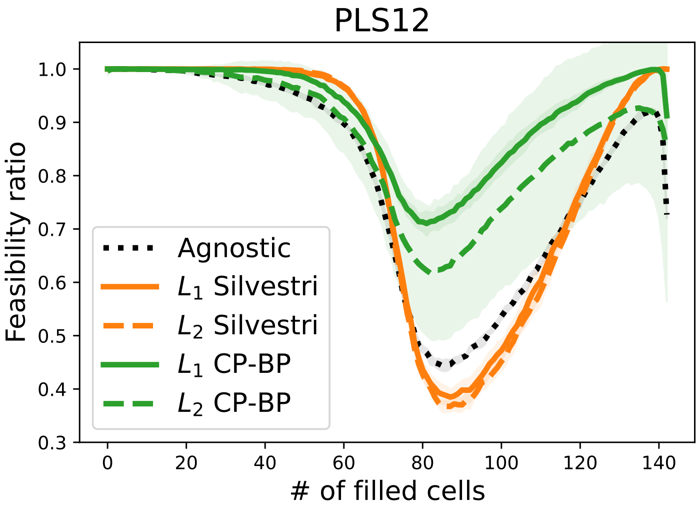
$$Loss(x,y) = \cancel{\underbrace{-\langle y, \log(\frac{1}{Z}\overbrace{f(x)}^{\hat{y}})\rangle}_{\texttt{cross entropy}}} + \cancel{\overbrace{\lambda}^{\texttt{weight}}} \cdot \underbrace{t(x)}_{\texttt{CP feedback}}$$

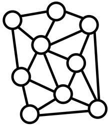
Domains

$$\begin{aligned} t(x) = L_1(x, C) &= \sum_k |C_k(x) - f_k(x)| \\ t(x) = L_2(x, C) &= \sum_k (C_k(x) - f_k(x))^2 \\ C_k(x) &\in \{0, 1\} \end{aligned}$$

Marginals

$$\begin{aligned} t(x) = L_1(x, \hat{\theta}) &= \sum_k |\hat{\theta}_k(x) - f_k(x)| \\ t(x) = L_2(x, \hat{\theta}) &= \sum_k (\hat{\theta}_k(x) - f_k(x))^2 \\ \hat{\theta}_k(x) &\in [0, 1] \end{aligned}$$


PLS12
Feasibility ratio
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100
120
140
# of filled cells
Agnostic
L_1 Silvestri
L_2 Silvestri
L_1 CP-BP
L_2 CP-BP

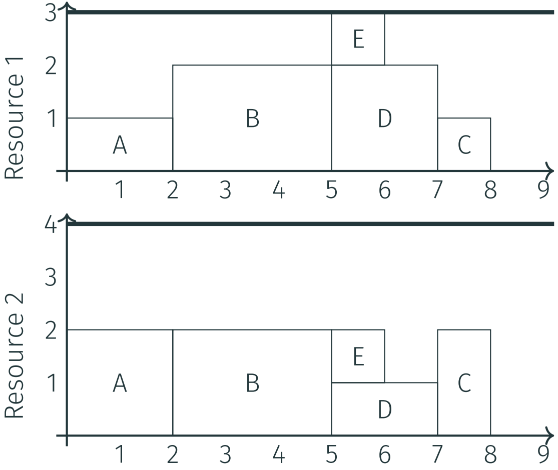


- Type of graph used: Bipartite heterogeneous graph
- Key technology:
  - Message passing
- How this helps:
  - Computation of marginals

WHEN ML HELPS CP: RCPSP USING GNNS

Task	p_i	c_{ir_1}	c_{ir_2}	succ
A	2	1	2	B C D
B	3	2	2	E
C	1	1	2	
D	2	2	1	C
E	1	1	1	C

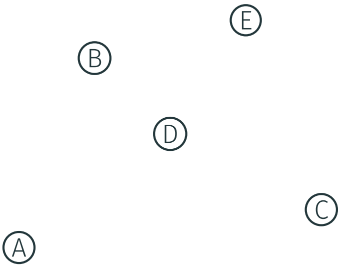
$C_{r_1} = 3$ and $C_{r_2} = 4$



PRECEDENCE GRAPH

Task	p_i	c_{ir_1}	c_{ir_2}	succ
A	2	1	2	B C D
B	3	2	2	E
C	1	1	2	
D	2	2	1	C
E	1	1	1	C

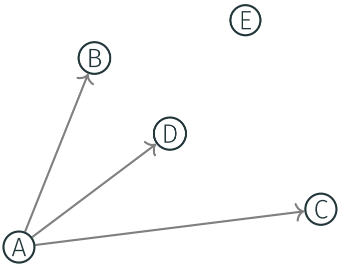
$C_{r_1} = 3$ and $C_{r_2} = 4$



PRECEDENCE GRAPH

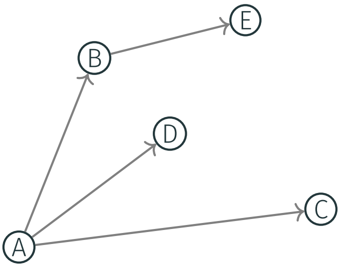
Task	p_i	c_{ir_1}	c_{ir_2}	succ
A	2	1	2	B C D
B	3	2	2	E
C	1	1	2	
D	2	2	1	C
E	1	1	1	C

$C_{r_1} = 3$ and $C_{r_2} = 4$



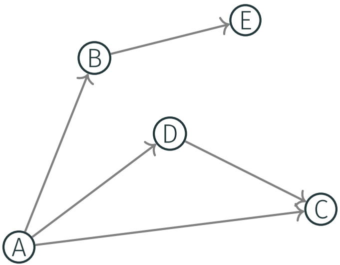
Task	p_i	c_{ir_1}	c_{ir_2}	succ
A	2	1	2	B C D
B	3	2	2	E
C	1	1	2	
D	2	2	1	C
E	1	1	1	C

$C_{r_1} = 3$ and $C_{r_2} = 4$



Task	p_i	c_{ir_1}	c_{ir_2}	succ
A	2	1	2	B C D
B	3	2	2	E
C	1	1	2	
D	2	2	1	C
E	1	1	1	C

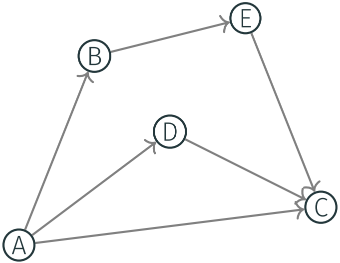
$C_{r_1} = 3$ and $C_{r_2} = 4$



KU LEUVEN

Task	p_i	c_{ir_1}	c_{ir_2}	succ
A	2	1	2	B C D
B	3	2	2	E
C	1	1	2	
D	2	2	1	C
E	1	1	1	C

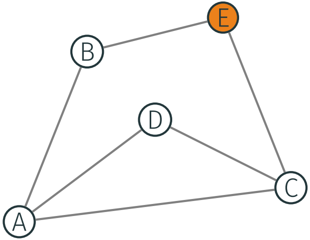
$C_{r_1} = 3$ and $C_{r_2} = 4$



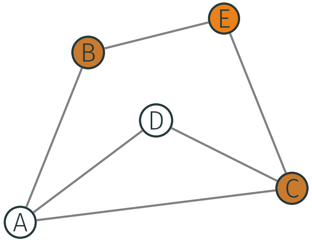
Main principle: for each node, creating an embedding of its neighborhood

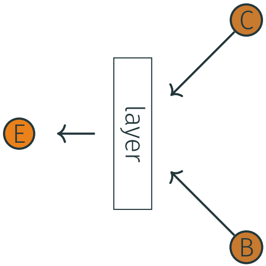
Tasks:

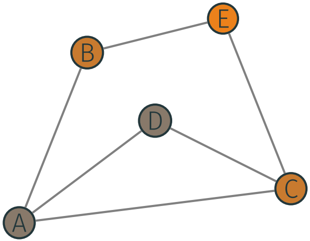
- Graph classification
- Node prediction
- Link prediction


E
B
D
C
A

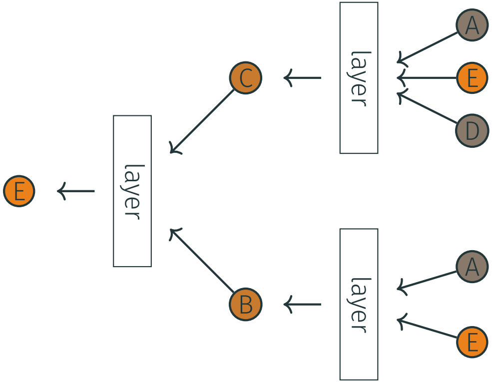

E


B
E
D
C
A

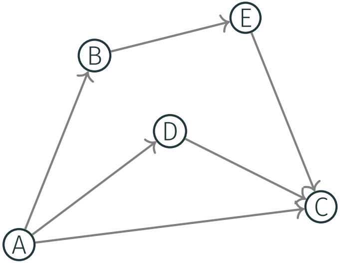

C
E
layer
B

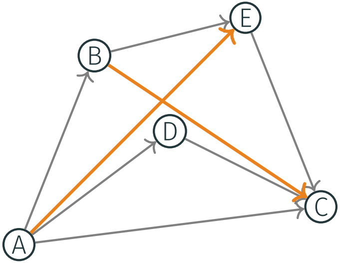


B
E
D
A
C



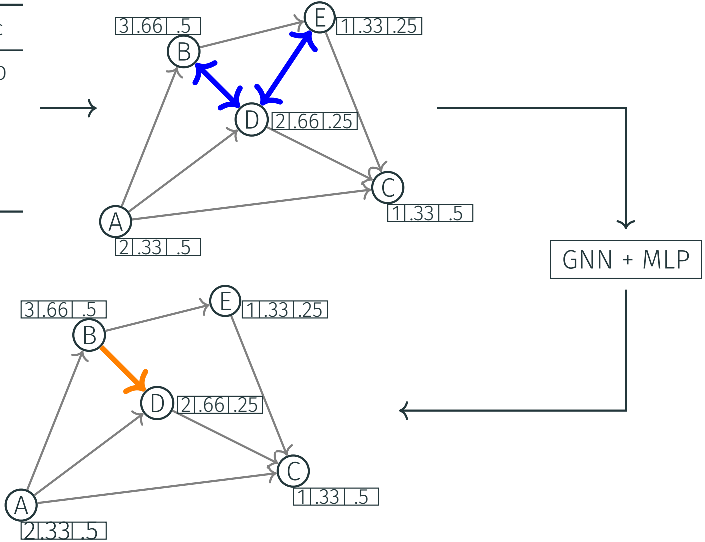
E
layer
C
B
layer
layer
A
E
D
A
E

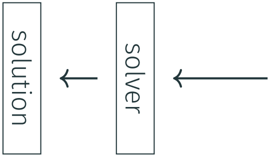

B
E
D
C
A


B
E
D
C
A

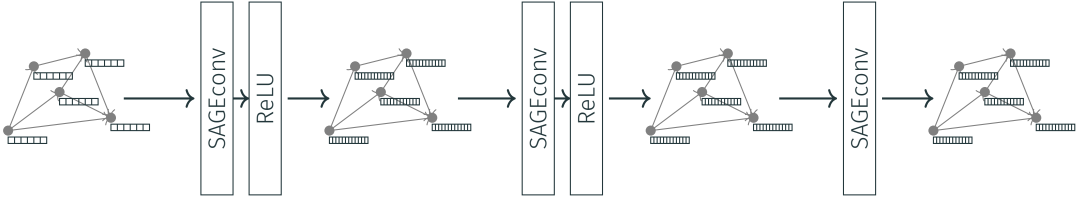
KU LEUVEN

Task	p_i	c_{ir_1}	c_{ir_2}	succ
A	2	1	2	B C D
B	3	2	2	E
C	1	1	2	
D	2	2	1	C
E	1	1	1	C

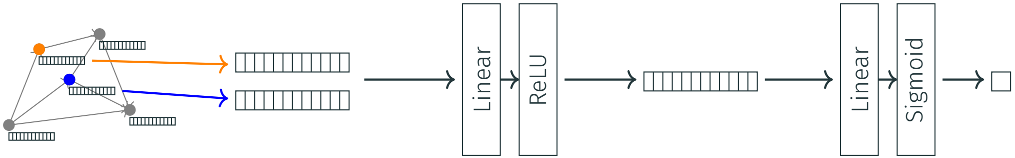




Goal: creates, for each node, embedding of the neighborhood

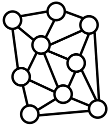


Goal: evaluate, given a candidate edge, its likeliness to exist



Two usages of the learned precedences:

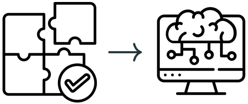
- additional constraints:
  - reduces search space
  - restriction of the problem
  - improve solution for a few instances
- task ordering:
  - preserve solutions
  - best first solution



- Type of graph used: Homogeneous directed graphs
- Key technology:
  - GNNs
- Key operation:
  - Transitive closure
- How this helps:
  - Reduction of the diameter
  - Better generalization
  - Computation of embeddings

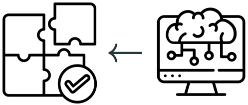
CONCLUSION

CONCLUSION – CP/ML SYNERGIES



CP helps ML

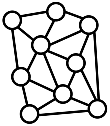
- Help with satisfying (hard) constraint



ML helps CP

- Deal with (big) data

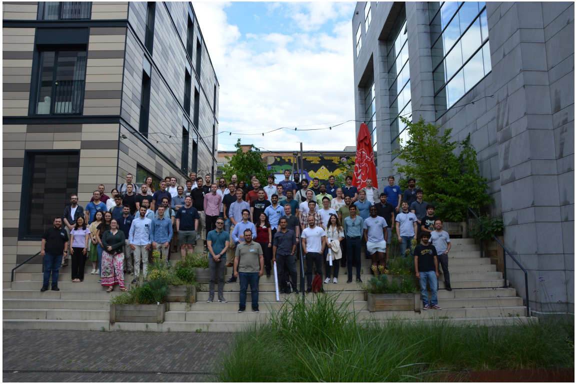
Always look at the graph side of problems



- Designs benefits from properties of underlying graphs
- Lots of tools/library/algorithms for graphs ready to use
- Known operations can create the graph you need

ACP SUMMER SCHOOL 2023 - RECORDINGS

KU LEUVEN



https://youtube.com/playlist?list=PLcByDTr7vRTYJ2s6DL-3bzjGwtQif33y3

Thank you for listening!

Any questions?

`https://hverhaeghe.bitbucket.io/`

REFS

ML for CP

- Sunny-CP: R. Amadini, M. Gabbrielli, and J. Mauro. ”A Multicore Tool for Constraint Solving”. In IJCAI 2015.
- SeaPearl: F. Chalumeau, I. Coulon, Q. Cappart, and L.-M. Rousseau. ”SeaPearl: A Constraint Programming Solver Guided by Reinforcement Learning”. In CPAIOR 2021. https://corail-research.github.io/seapearl/
- Constraint acquisition: S. Prestwich, E. Freuder, B. O’Sullivan, and D. Browne. ”Classifier-based constraint acquisition”. In AMAI 2021.

CP for ML

- Clustering: T. Guns, T.-B.-H. Dao, C. Vrain, and K.-C. Duong. ”Repetitive Branch-and-Bound Using Constraint Programming for Constrained Minimum Sum-of-Squares Clustering”. In ECAI 2016.
- Visual Sudoku: M. Mulamba, J. Mandi, R. Canoy, and T. Guns. ”Hybrid classification and reasoning for image-based constraint solving”
- PLS experiment: M. Silvestri, M. Lombardi, and M. Milano. ”Injecting domain knowledge in neural networks: a controlled experiment on a constrained problem”. In CPAIOR 2021.