



MONASH
University

KU LEUVEN

Towards Step-wise Explanations of Large Search Trees

CP 2026 – Lisbon, Portugal

Ignace Bleukx

KU Leuven, Belgium

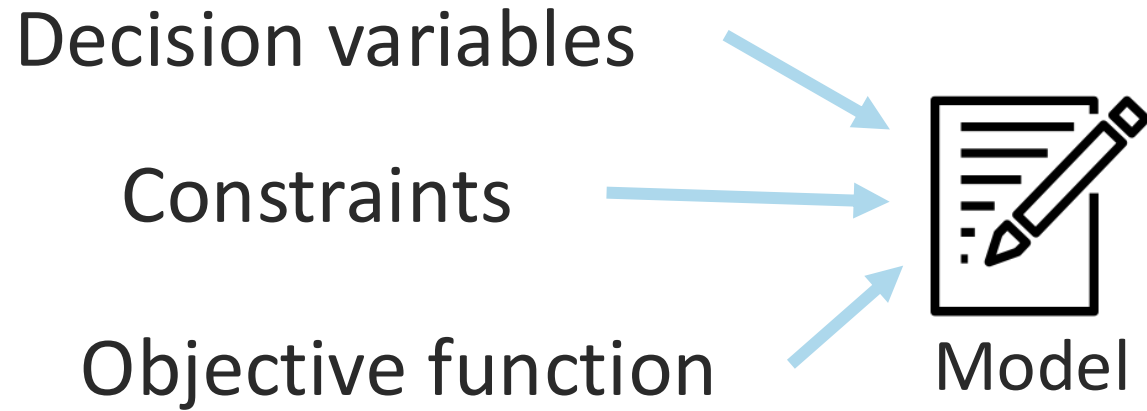
Peter J. Stuckey

Monash University, Australia

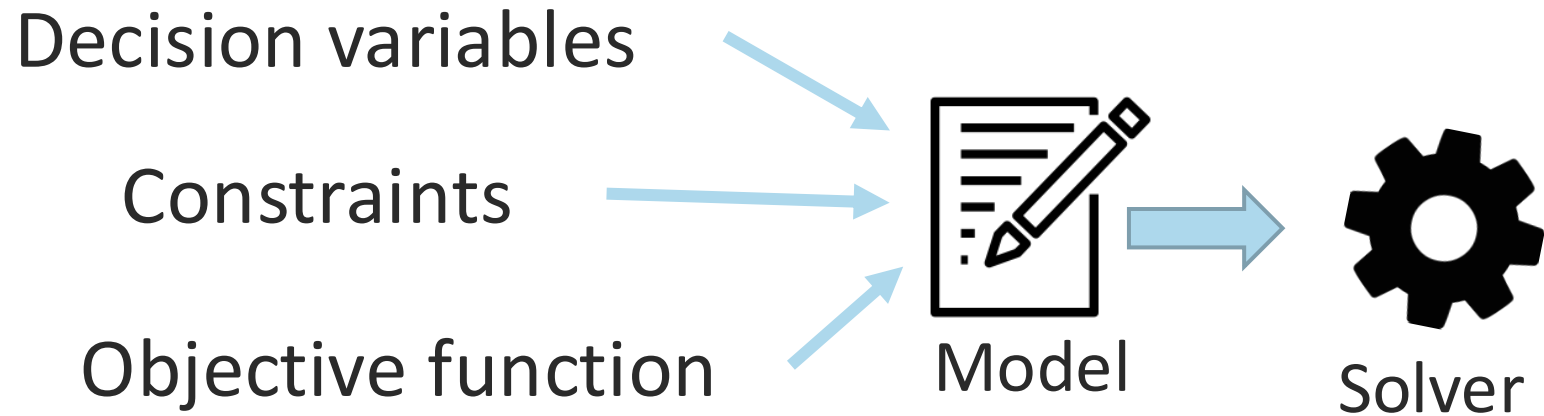
Tias Guns

KU Leuven, Belgium

Constraint Solving



Constraint Solving



Constraint Solving

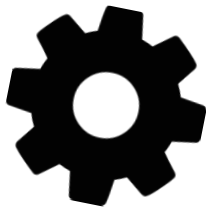
Decision variables

Constraints

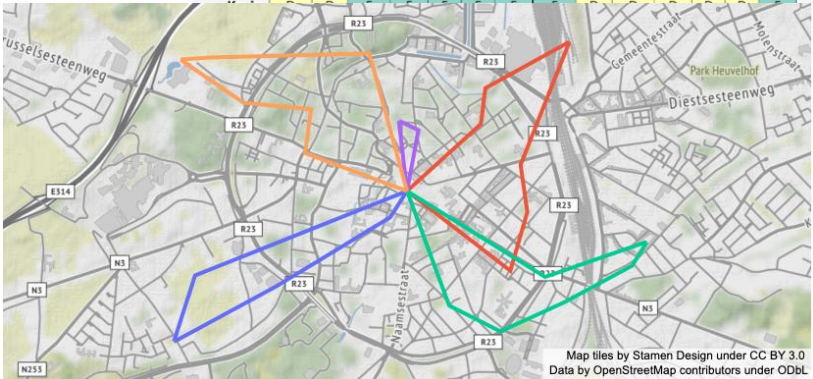
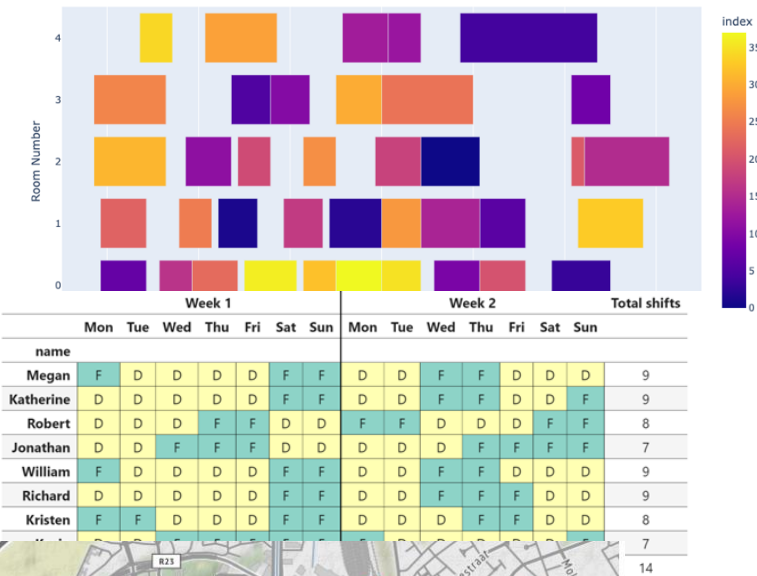
Objective function



Model



Solver



Explainable Constraint Solving

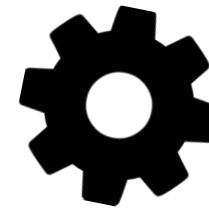
Decision variables

Constraints

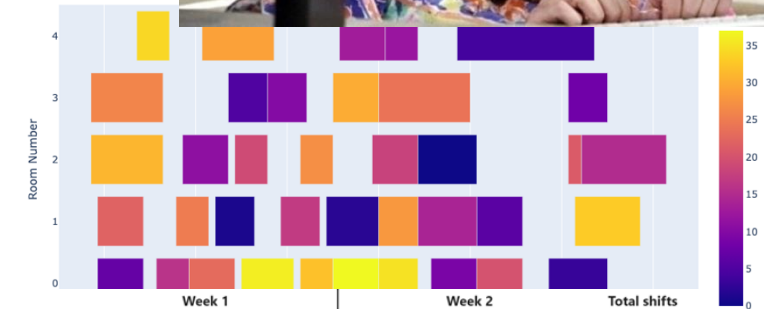
Objective function



Model



Solver



	Week 1							Week 2							Total shifts
	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
name															
Megan	F	D	D	D	D	F	F	D	D	F	F	D	D	D	9
Katherine	D	D	D	D	D	F	F	D	D	F	F	D	D	F	9
Robert	D	D	D	F	F	D	D	F	F	D	D	D	F	F	8
Jonathan	D	D	F	F	F	D	D	D	D	D	F	F	F	F	7
William	F	D	D	D	D	F	F	D	D	F	F	D	D	D	9
Richard	D	D	D	D	D	F	F	D	D	F	F	F	D	D	9
Kristen	F	F	D	D	D	F	F	D	D	D	F	F	D	D	8
															7
															14



Map tiles by Stamen Design under CC BY 3.0
Data by OpenStreetMap contributors under ODbL

Our focus



Focus on explaining UNSAT
“Why is there no solution?”

Our focus



Focus on explaining UNSAT

“Why is there no solution?”

Covers variety of other questions:

“Why is objective value x^* optimal?”

“Why is it impossible to assign $x = 42$?”

“Why is it *suboptimal* to assign $x = 42$?”

Step-wise Explanations [1]

Explain step-by-step, using easy to understand steps, how to arrive to the conclusion

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2		8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{2,2} = 7 \wedge x_{3,2} = 9 \wedge \dots \wedge$$

$$AllDiff(row_5) \wedge AllDiff(col_2)$$

$$\Rightarrow x_{5,2} = 2$$

Step-wise Explanations [1]

Explain step-by-step, using easy to understand steps, how to arrive to the conclusion

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2		8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{2,2} = 7 \wedge x_{3,2} = 9 \wedge \dots \wedge$$

$$AllDiff(row_5) \wedge AllDiff(col_2)$$

$$\Rightarrow x_{5,2} = 2$$

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2	6	8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{5,1} = 4 \wedge x_{5,2} = 2 \wedge \dots \wedge$$

$$AllDiff(row_5) \wedge AllDiff(block_6)$$

$$\Rightarrow x_{5,3} = 6$$

[1] Bart Bogaerts, Emilio Gamba, and Tias Guns. "A framework for step-wise explaining how to solve constraint satisfaction problems." *Artificial Intelligence* 300 (2021): 103550.

Step-wise Explanations [1]

Explain step-by-step, using easy to understand steps, how to arrive to the conclusion

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2		8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{2,2} = 7 \wedge x_{3,2} = 9 \wedge \dots \wedge$$

$$AllDiff(row_5) \wedge AllDiff(col_2)$$

$$\Rightarrow x_{5,2} = 2$$

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2	6	8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{5,1} = 4 \wedge x_{5,2} = 2 \wedge \dots \wedge$$

$$AllDiff(row_5) \wedge AllDiff(block_6)$$

$$\Rightarrow x_{5,3} = 6$$

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

$$x_{4,4} = 7 \wedge x_{4,5} = 6 \wedge \dots \wedge$$

$$AllDiff(block_5)$$

$$\Rightarrow x_{4,6} = 1$$

[1] Bart Bogaerts, Emilio Gamba, and Tias Guns. "A framework for step-wise explaining how to solve constraint satisfaction problems." *Artificial Intelligence* 300 (2021): 103550.

Step-wise Explanations [1]

Explain step-by-step, using easy to understand steps, how to arrive to the conclusion

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2		8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{2,2} = 7 \wedge x_{3,2} = 9 \wedge \dots \wedge$$

$$AllDiff(row_5) \wedge AllDiff(col_2)$$

$$\Rightarrow x_{5,2} = 2$$

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2	6	8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{5,1} = 4 \wedge x_{5,2} = 2 \wedge \dots \wedge$$

$$AllDiff(row_5) \wedge AllDiff(block_6)$$

$$\Rightarrow x_{5,3} = 6$$

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

$$x_{4,4} = 7 \wedge x_{4,5} = 6 \wedge \dots \wedge$$

$$AllDiff(block_5)$$

$$\Rightarrow x_{4,6} = 1$$

“Easy to understand” means using few facts and constraints

[1] Bart Bogaerts, Emilio Gamba, and Tias Guns. "A framework for step-wise explaining how to solve constraint satisfaction problems." *Artificial Intelligence* 300 (2021): 103550.

Step-wise Explanations for UNSAT CSPs [2]

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2		8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{2,2} = 7 \wedge x_{3,2} = 9 \wedge \cdots \wedge$$

$$AllDiff(row_5) \wedge AllDiff(col_2)$$

$$\Rightarrow x_{5,2} = 2$$

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2	6	8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{5,1} = 4 \wedge x_{5,2} = 2 \wedge \cdots \wedge$$

$$AllDiff(row_5) \wedge AllDiff(block_6)$$

$$\Rightarrow x_{5,3} = 6$$

			6	6				

$$x_{4,4} = 6 \wedge x_{4,5} = 6 \wedge \cdots \wedge$$

$$AllDiff(block_5)$$

$$\Rightarrow False$$

Derive “false” at the end of sequence

[2] Bleukx, Ignace, et al. "Simplifying Step-Wise Explanation Sequences." 29th International Conference on Principles and Practice of Constraint Programming (CP 2023).

Step-wise Explanations from proofs [3]

```
      ./.
      ,pkku.
      mkkiiu, .
      ,kkn
      'kkn.
      .:kkkip, .
      .',',',mpkkkkkkkp:pupu:,
      .,.,.,mp::::ukpmppppppppuuppuuumup; .
      ,:,:pp::::pp;:::::p;::::pupp::::
      .:uuuumpuuuup,,:ppuuuup:p:uuuupuuuu:p'
      ,:puuuumpuuuuu, :uuuuuuuuuuuuuuuuuuuuuuuu;
      .:uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      puuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      .uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      'puuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      .kuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      kmuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      ;kuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      kmuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      ,kmuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      pkmuupkuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      :kpuukpuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      .pkppkpuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      .pkkkkmuupkpuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'
      .',;ukkkkkkkkppppppkkkkkkkkku;'.
      .,.,.,.,.,.
```

Certifying solvers (e.g., Pumpkin) generate an “explanation sequence”: a **proof**

We can find explanation sequences by starting from a proof

Step-wise Explanations from proofs [3]

[illegible]

1 $C_1 \rightarrow (x \geq 1) \vee (y = 2) \vee (z \leq 5)$

2 $C_2 \rightarrow (x \leq 0) \vee (q \neq 2)$

$$\textcircled{3} \textcircled{1} \wedge \textcircled{2} \rightarrow (y = 2) \vee (z \leq 5) \vee (q \neq 2)$$

...

● ③ $\wedge \dots \rightarrow y = 2$

...

● *False*

Step-wise Explanations from proofs [3]

[illegible]

$$\textcircled{1} \ C_1 \rightarrow (x \geq 1) \vee (y = 2) \vee (z \leq 5)$$

② $C_2 \rightarrow (x \leq 0) \vee (q \neq 2)$

$$\textcircled{3} \textcircled{1} \wedge \textcircled{2} \rightarrow (y = 2) \vee (z \leq 5) \vee (q \neq 2)$$

• • •

- **③** $\wedge \dots \rightarrow y = 2$ $MUS(\{U_1, U_2, \dots, y \neq 2\})$

• • •

- *False*

User-level constraints

Step-wise Explanations from proofs [3]

Certifying solvers (e.g., Pumpkin) generate an “explanation sequence”: a **proof**

We can find explanation sequences by starting from a proof



Efficient and works for large CSPs

[illegible]

Step-wise Explanations from proofs [3]

Certifying solvers (e.g., Pumpkin) generate an “explanation sequence”: a **proof**

We can find explanation sequences by starting from a proof



Efficient and works for large CSPs

“What about problems with search?”

- Peter at my Phd defence

[illegible]

“Problems with search”

Cannot derive UNSAT at root-node of search tree (using **any** propagation strength)

Step-wise explanations with negative literals [2]

Do not require search [4]

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	12 45		8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{2,2} = 7 \wedge x_{2,2} = 7 \wedge \dots \wedge$$

AllDiff(col₂)

$$\Rightarrow x_{5,2} \neq 3, x_{5,2} \neq 6, \\ x_{5,2} \neq 7, x_{5,2} \neq 8, x_{5,2} \neq 9$$

5	3			7				
6	7		1	9	5			
1	9	8		4			6	
8				6				3
4	2		8	5	3			1
7				2				6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3				8		1	7	9

$$x_{5,1} = 4 \wedge x_{5,2} = 2 \wedge \dots \wedge$$

AllDiff(row₅)

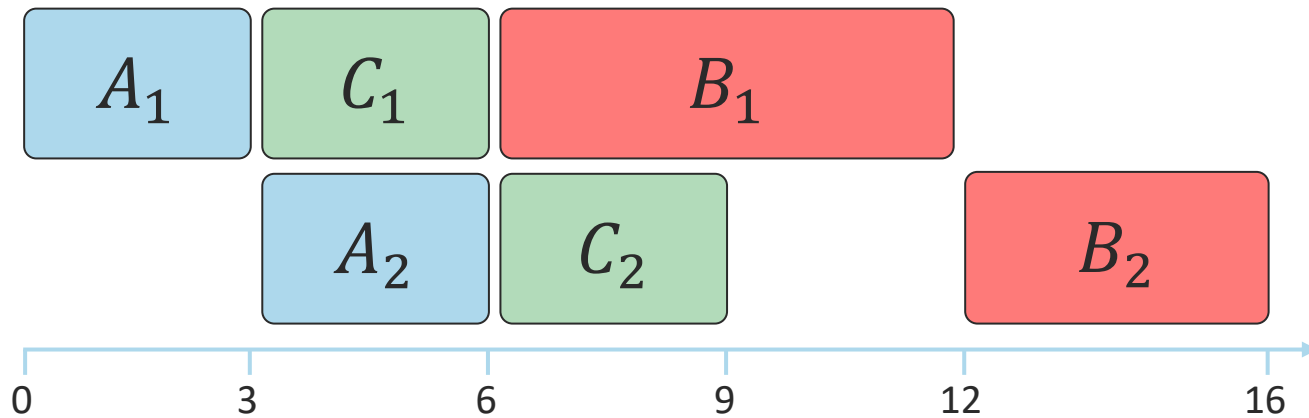
$$\Rightarrow x_{5,2} \neq 4, x_{5,2} \neq 5, x_{5,2} \neq 1$$

....

[2] Bleukx, Ignace, et al. "Simplifying Step-Wise Explanation Sequences." *29th International Conference on Principles and Practice of Constraint Programming (CP 2023)*.

[4] Lagerkvist Mikael Zayenz, "Scaling Sudoku as a Constraint Problem" at MODREF 2026

An example - jobshop



$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$NoOverlap((A_2, 3), (B_2, 4), (C_2, 3))$

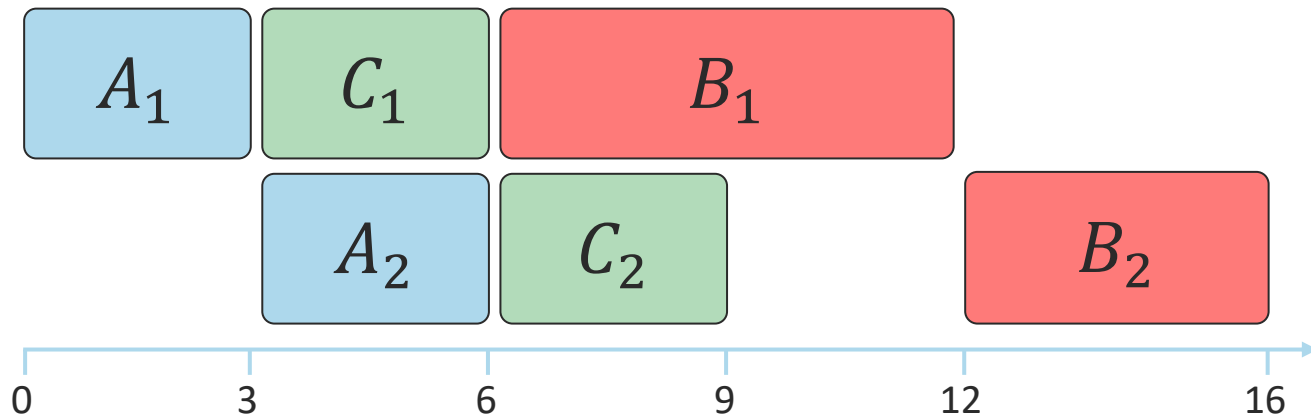
$$A_1 + 3 \leq A_2$$

$$B_1 + 6 \leq B_2$$

$$C_1 + 3 \leq C_2$$

$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

An example - jobshop



$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$NoOverlap((A_2, 3), (B_2, 4), (C_2, 3))$

$$A_1 + 3 \leq A_2$$

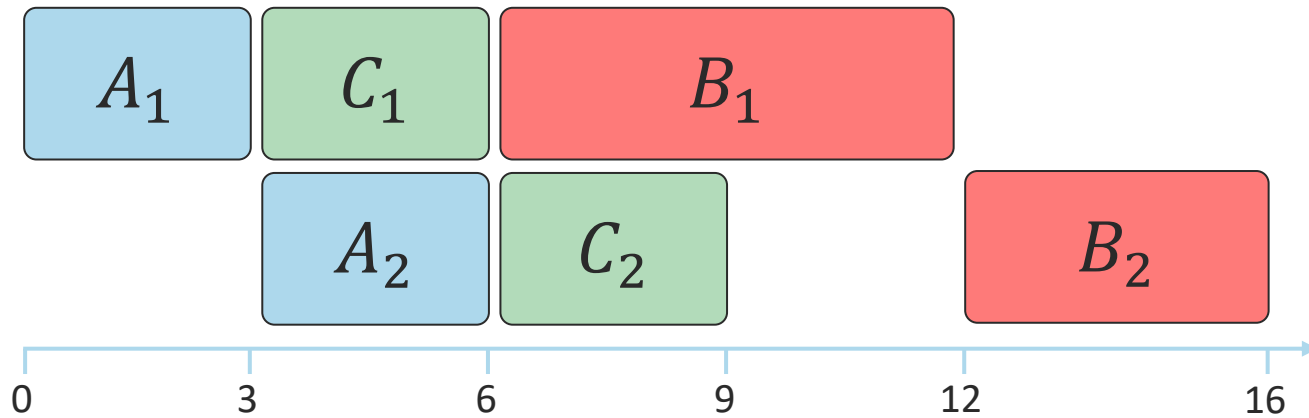
$$B_1 + 6 \leq B_2$$

$$C_1 + 3 \leq C_2$$

$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

“Why is there no solution with a shorter makespan?”

An example - jobshop



$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$NoOverlap((A_2, 3), (B_2, 4), (C_2, 3))$

$$A_1 + 3 \leq A_2$$

$$B_1 + 6 \leq B_2$$

$$C_1 + 3 \leq C_2$$

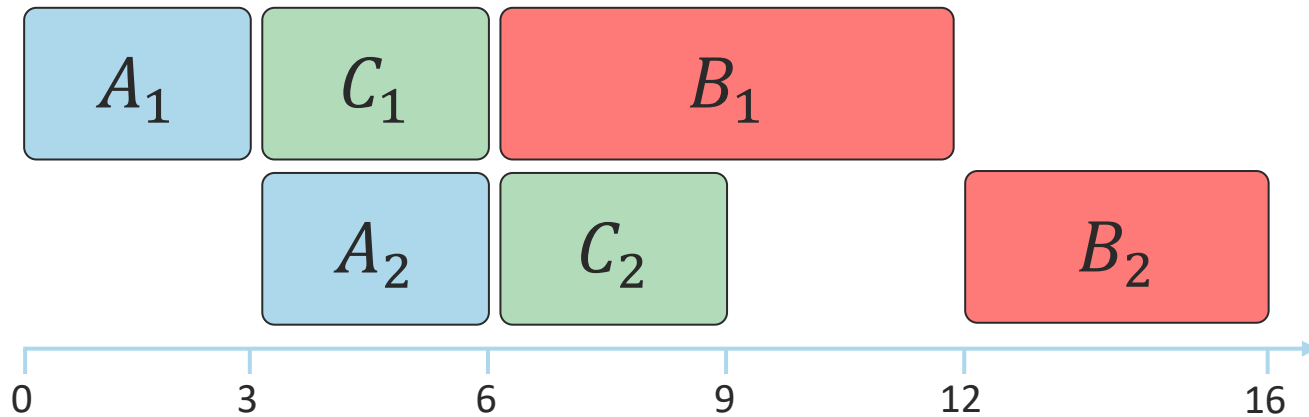
$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

$$makespan < 16$$

$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

$$A_2 \leq 12, B_2 \leq 11, C_2 \leq 12$$

An example - jobshop



$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$NoOverlap((A_2, 3), (B_2, 4), (C_2, 3))$

$$A_1 + 3 \leq A_2$$

$$B_1 + 6 \leq B_2$$

$$C_1 + 3 \leq C_2$$

$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

$$makespan < 16$$

$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

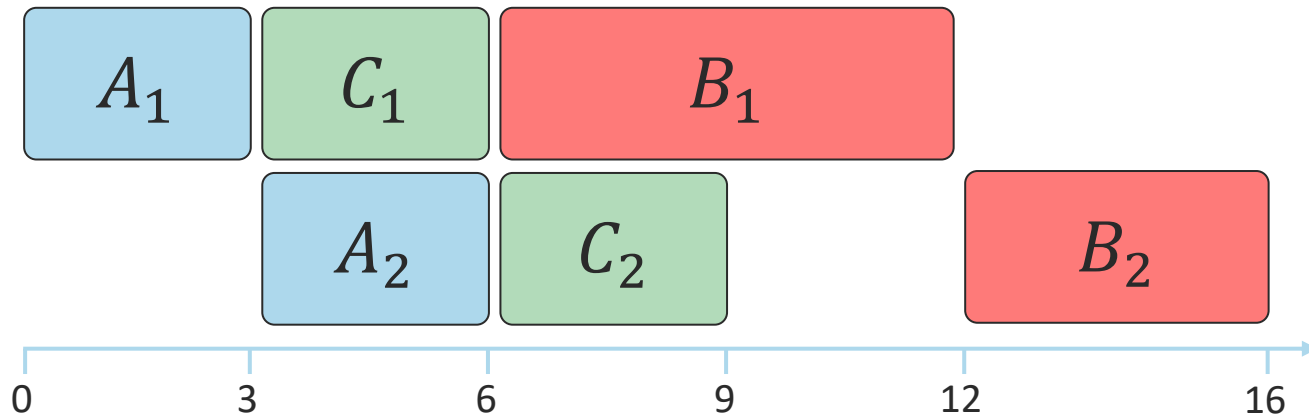
$$A_2 \leq 12, B_2 \leq 11, C_2 \leq 12$$

$$B_1 \geq 0, B_2 \leq 11$$

$$B_1 + 6 \leq B_2$$

$$B_1 \leq 5, B_2 \geq 6$$

An example - jobshop



$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$NoOverlap((A_2, 3), (B_2, 4), (C_2, 3))$

$$A_1 + 3 \leq A_2$$

$$B_1 + 6 \leq B_2$$

$$C_1 + 3 \leq C_2$$

$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

$$makespan < 16$$

$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

$$A_2 \leq 12, B_2 \leq 11, C_2 \leq 12$$

$$B_1 \geq 0, B_2 \leq 11$$

$$B_1 + 6 \leq B_2$$

$$B_1 \leq 5, B_2 \geq 6$$

$$A_2 \leq 12, 6 \leq B_2 \leq 11, C_2 \leq 12$$

$$A_1 + 3 \leq A_2$$

$$C_1 + 3 \leq C_2$$

$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$NoOverlap((A_2, 3), (B_2, 4), (C_2, 3))$

$$B_1 \geq 3$$

“What about problems with search?”

Step-wise explanations derive *domain reduction* using set of constraints

Sometimes need lots of constraints to derive domain reduction

“What about problems with search?”

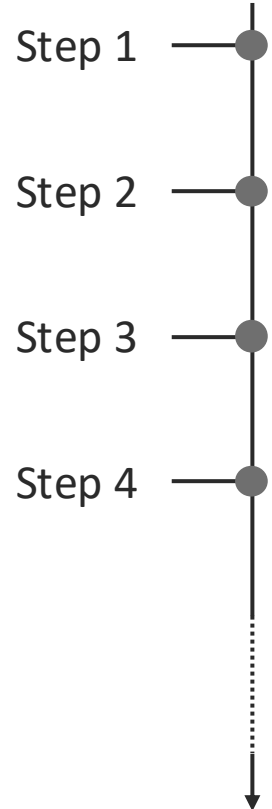
Step-wise explanations derive *domain reduction* using set of constraints

Sometimes need lots of constraints to derive domain reduction



Bad interpretability for large CSPs...

Solution: nested explanations [1]



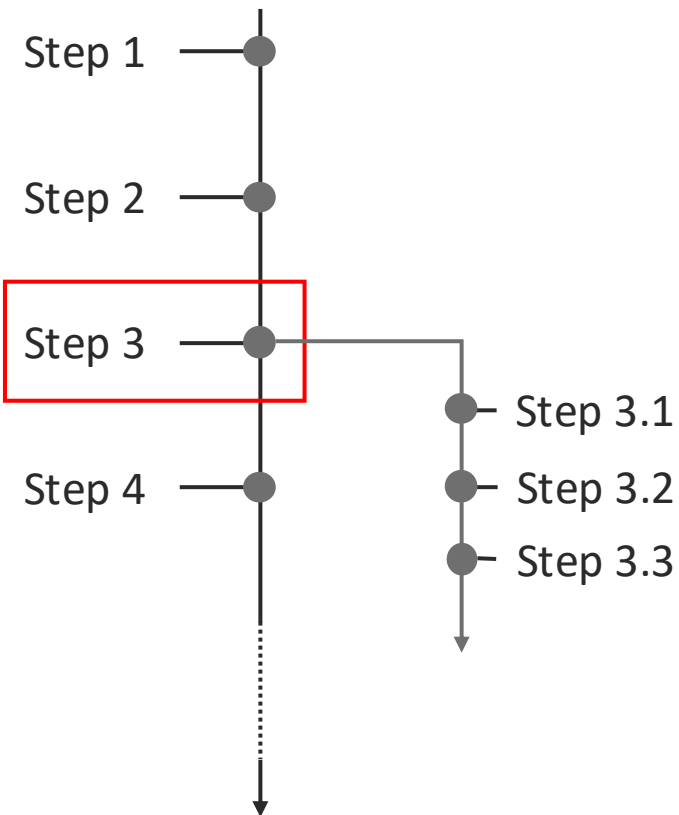
Given a “complex” step:

$$\text{Input} \wedge \text{Constraints} \Rightarrow \text{Output}$$

Recursively explain by contradiction

“Why is $\text{Input} \wedge \text{Constraints} \wedge \neg \text{Output}$ unsat?”

Solution: nested explanations [1]



Given a “complex” step:

$$\text{Input} \wedge \text{Constraints} \Rightarrow \text{Output}$$

Recursively explain by contradiction

“Why is $\text{Input} \wedge \text{Constraints} \wedge \neg \text{Output}$ unsat?”

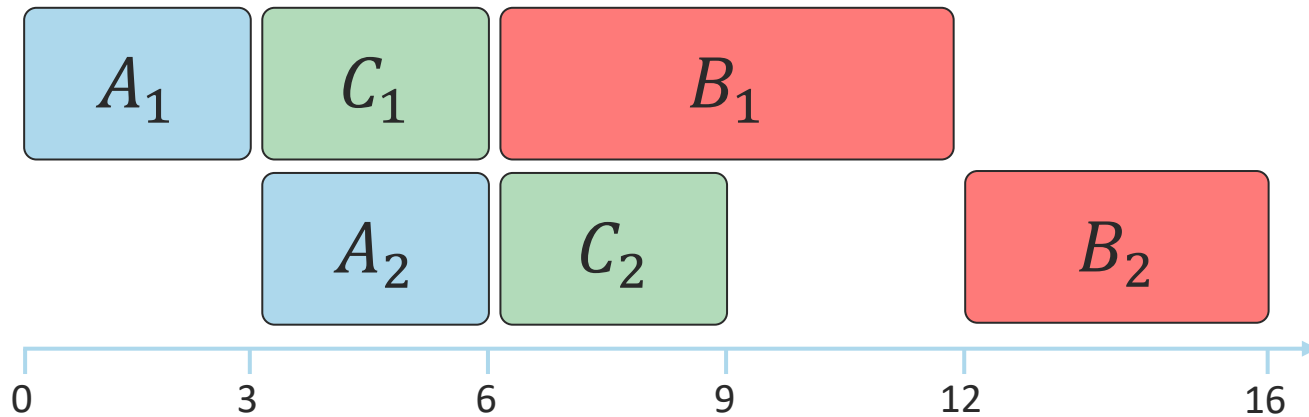
Contributions

Generate step-wise explanations to explain **why large scale CSPs are UNSAT**

Enabling us to compute nested explanations **until at most 1 (global) constraint per step**

Experimental evaluation and comparison to **proof-size**

An example - jobshop



$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$NoOverlap((A_2, 3), (B_2, 4), (C_2, 3))$

$$A_1 + 3 \leq A_2$$

$$B_1 + 6 \leq B_2$$

$$C_1 + 3 \leq C_2$$

$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

$$makespan < 16$$

$$makespan = \max(A_2 + 3, B_2 + 4, C_2 + 3)$$

$$A_2 \leq 12, B_2 \leq 11, C_2 \leq 12$$

$$B_1 \geq 0, B_2 \leq 11$$

$$B_1 + 6 \leq B_2$$

$$B_1 \leq 5, B_2 \geq 6$$

$$A_2 \leq 12, 6 \leq B_2 \leq 11, C_2 \leq 12$$

$$A_1 + 3 \leq A_2$$

$$C_1 + 3 \leq C_2$$

$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$NoOverlap((A_2, 3), (B_2, 4), (C_2, 3))$

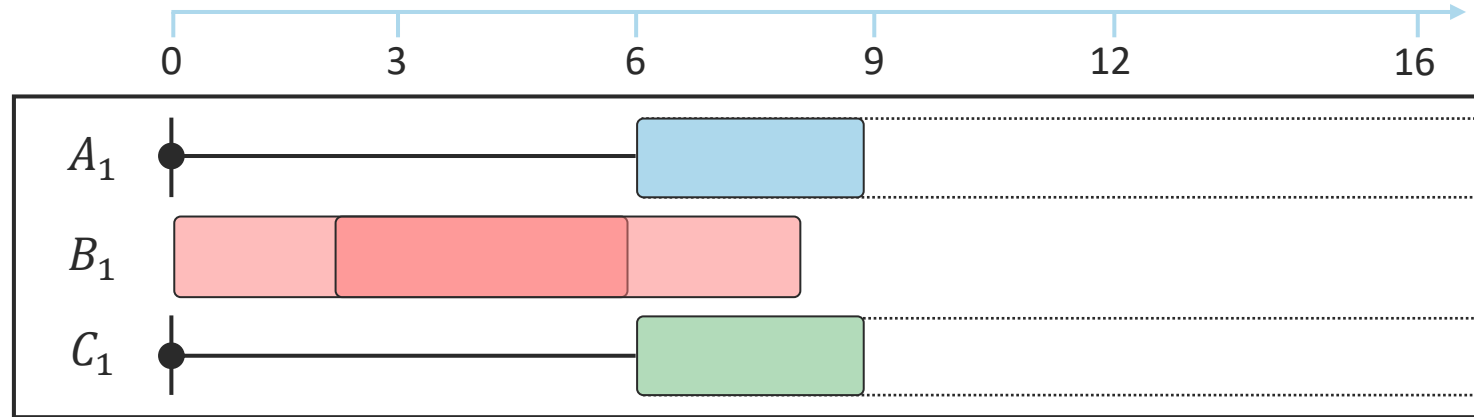
$$B_1 \geq 3$$

Example - nested

$$B_1 < 3$$

NoOverlap(($A_1, 3$), ($B_1, 6$), ($C_1, 3$))

$$A_1 \geq 6, C_1 \geq 6$$



Example - nested

$$B_1 < 3$$

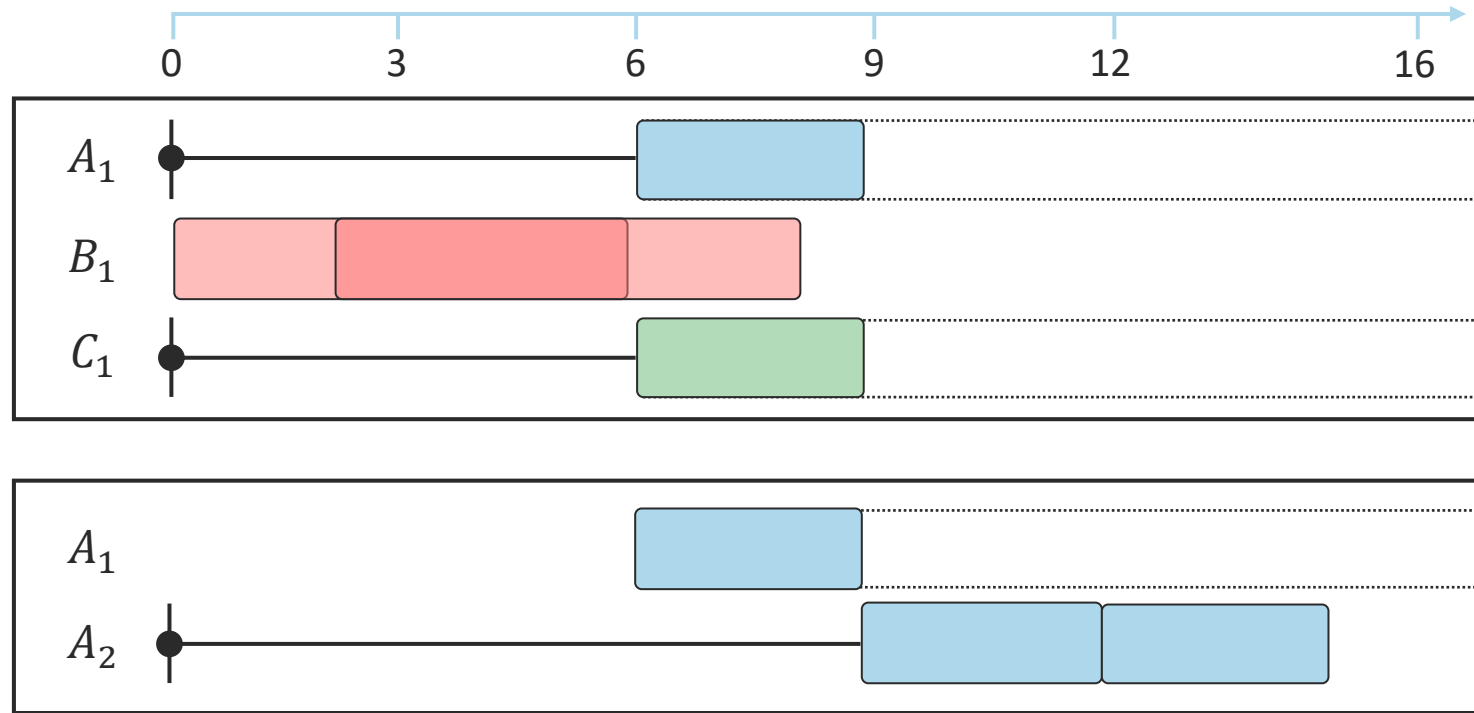
$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$$A_1 \geq 6, C_1 \geq 6$$

$$A_1 \geq 6$$

$$A_1 + 3 \leq A_2$$

$$A_2 \geq 9$$



Example - nested

$$B_1 < 3$$

$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$$A_1 \geq 6, C_1 \geq 6$$

$$A_1 \geq 6$$

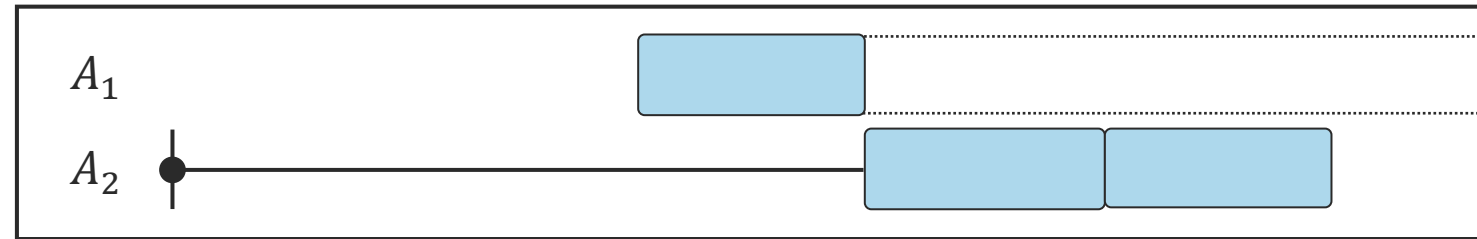
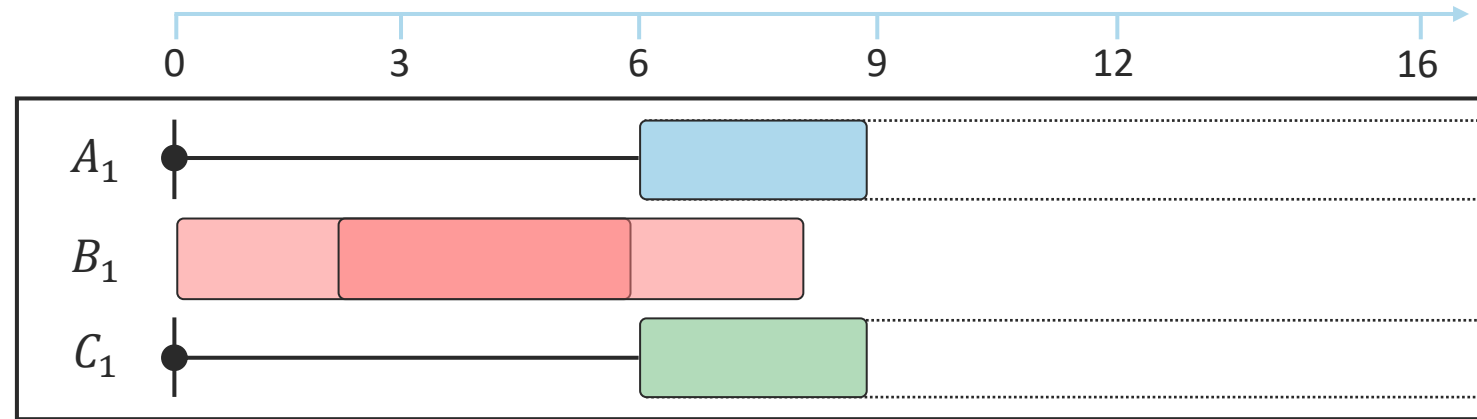
$$A_1 + 3 \leq A_2$$

$$A_2 \geq 9$$

$$C_2 \geq 6$$

$$C_1 + 3 \leq C_2$$

$$C_2 \geq 9$$



Example - nested

$$B_1 < 3$$

$NoOverlap((A_1, 3), (B_1, 6), (C_1, 3))$

$$A_1 \geq 6, C_1 \geq 6$$

$$A_1 \geq 6$$

$$A_1 + 3 \leq A_2$$

$$A_2 \geq 9$$

$$C_2 \geq 6$$

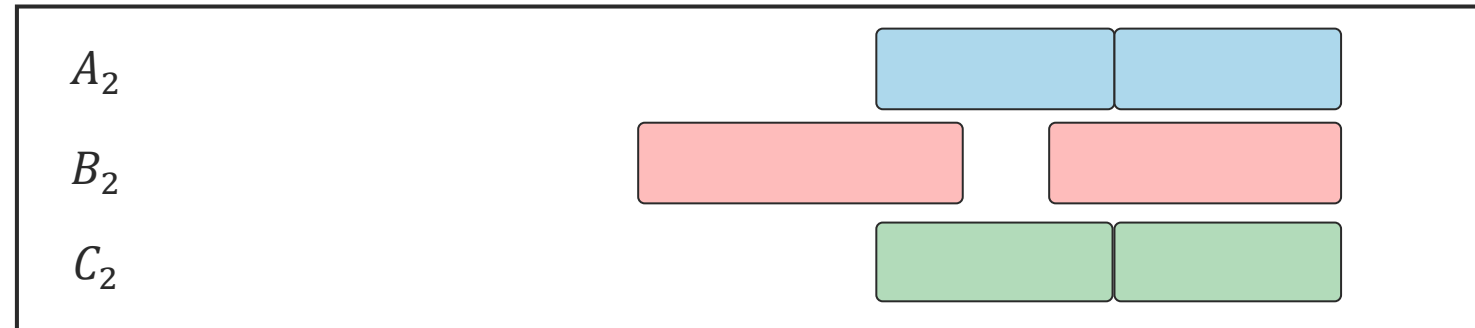
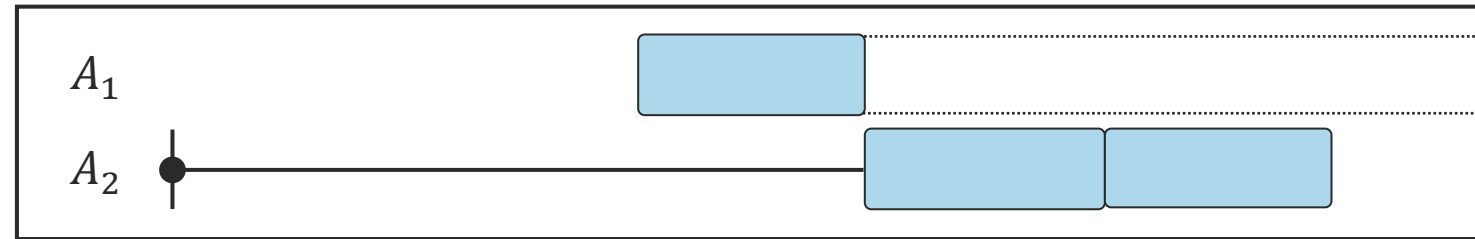
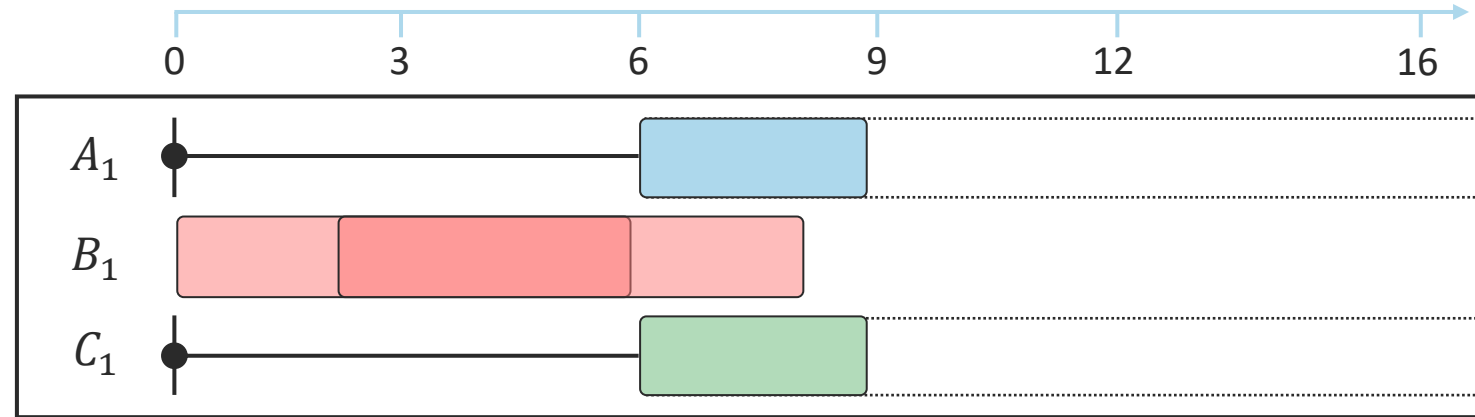
$$C_1 + 3 \leq C_2$$

$$C_2 \geq 9$$

$$9 \leq A_2 \leq 12, 6 \leq B_2 \leq 11, 9 \leq C_2 \leq 12$$

$NoOverlap((A_2, 3), (B_2, 4), (C_2, 3))$

False



Edge case: nested explanation of False

Nested step for

$$\boxed{\textit{Input}} \wedge \boxed{\textit{Constraints}} \Rightarrow \boxed{\textit{False}}$$

Edge case: nested explanation of False

Nested step for

$$\boxed{\textit{Input}} \wedge \boxed{\textit{Constraints}} \Rightarrow \boxed{\textit{False}}$$

$$\boxed{\textit{Input}} \wedge \boxed{\textit{Constraints}} \wedge \boxed{\textit{True}}$$

Edge case: nested explanation of False

Nested step for

$$\boxed{\textit{Input}} \wedge \boxed{\textit{Constraints}} \Rightarrow \boxed{\textit{False}}$$

$$\boxed{\textit{Input}} \wedge \boxed{\textit{Constraints}} \wedge \boxed{\textit{True}}$$

Need “dummy” literal to “start of propagation”

Solution: generate trimmed proof, take first **decision**

Experiments

We are interested in

1. How much nesting do we need?
2. If we're doing all this nesting, aren't we better off reading the proof?

Benchmarks



Benchmark 2: RCPSP on PSPlib
“Why is there no better solution?”

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Megan	-	D	D	D	D	-	-
Katherine	D	D	D	D	D	-	-
Robert	D	D	D	-	-	D	D
Jonathan	D	D	-	-	-	D	D
William	-	D	D	D	D	-	-
Richard	D	D	D	D	D	-	-
Kristen	-	-	D	D	D	-	-
Kevin	D	D	-	-	-	-	-
Cover/Req	5/5	7/7	6/6	5/4	5/5	2/5	2/5

Benchmark 2: Nurserostering (NR)
“Why is this MUS UNSAT?”

Experimental setup



CSPs modeled in CPMpy

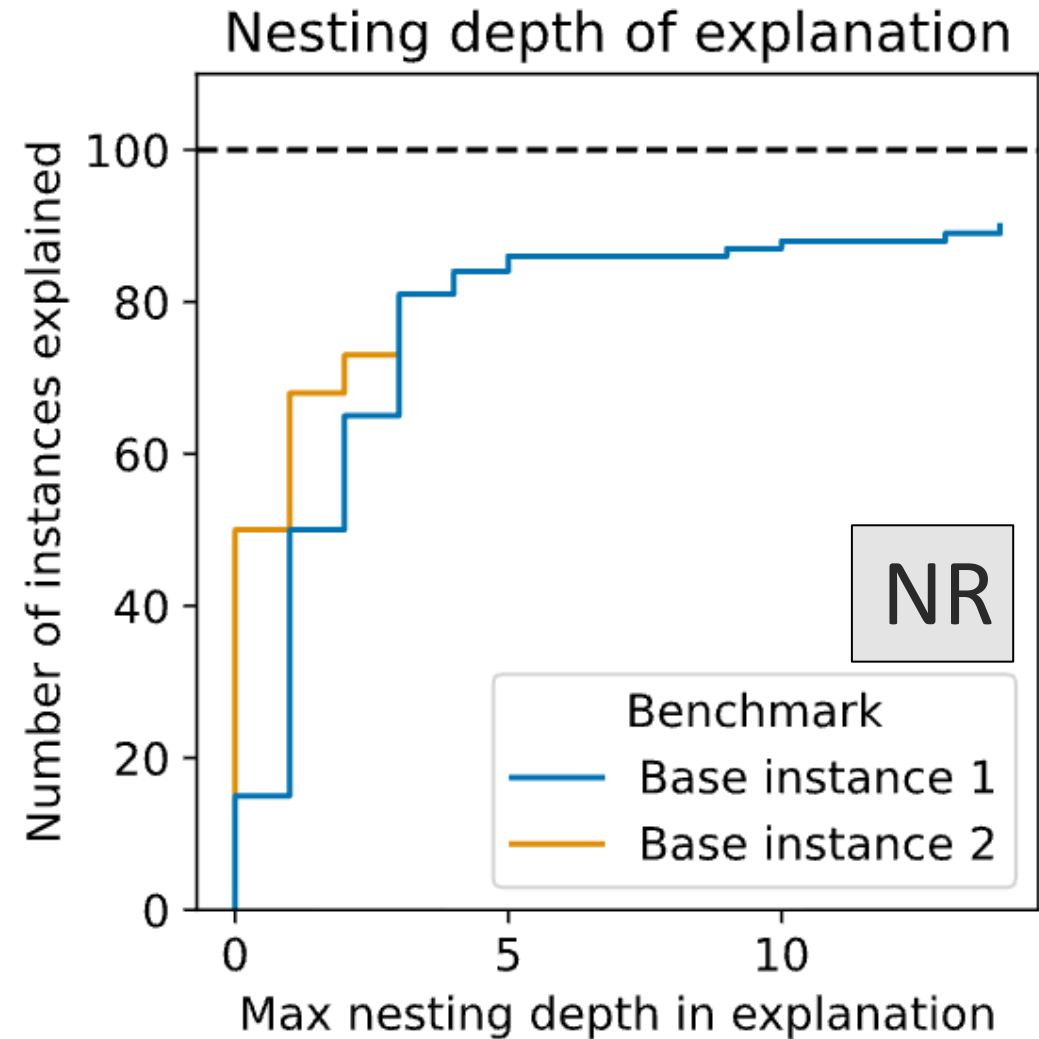
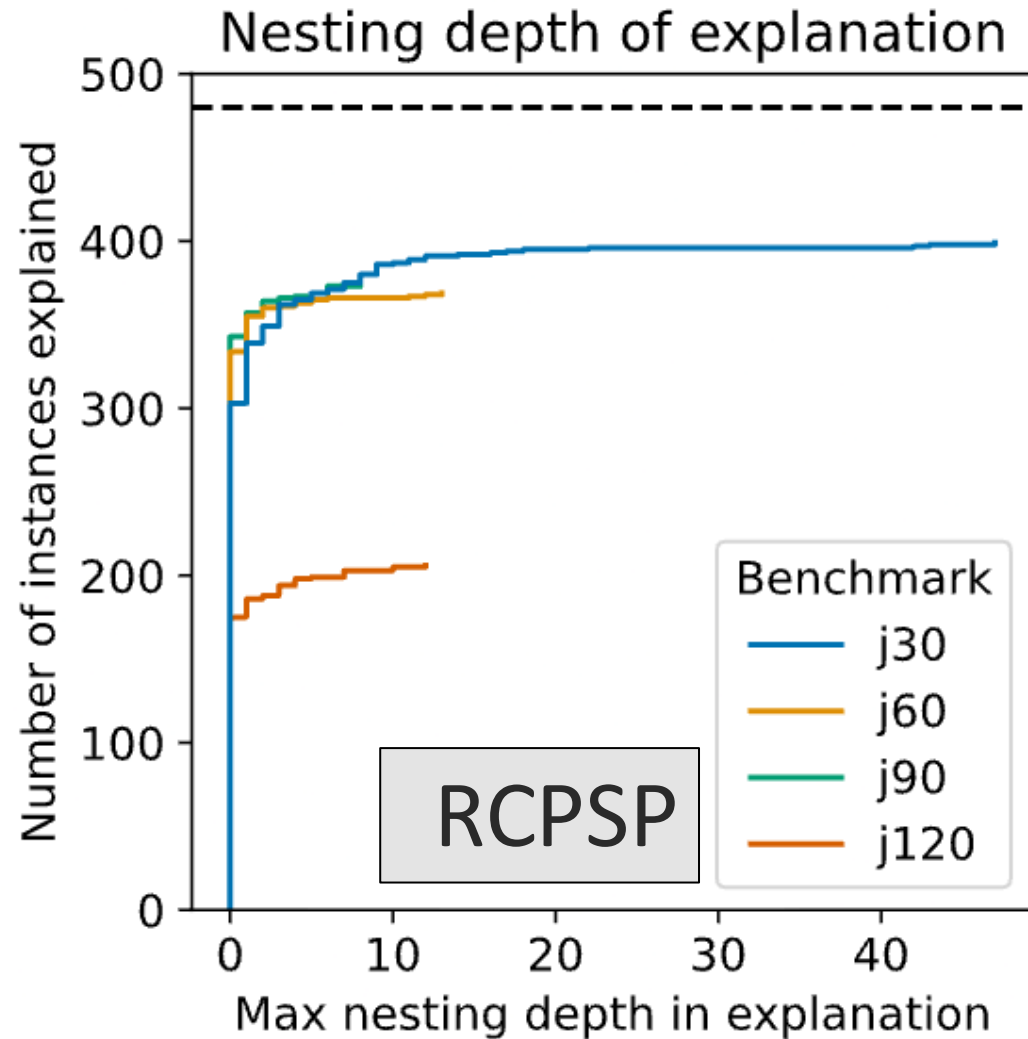
Find step-wise explanation sequence with
Pumpkin prooflog

To nest or not to nest?

Anytime we need >1 constraint

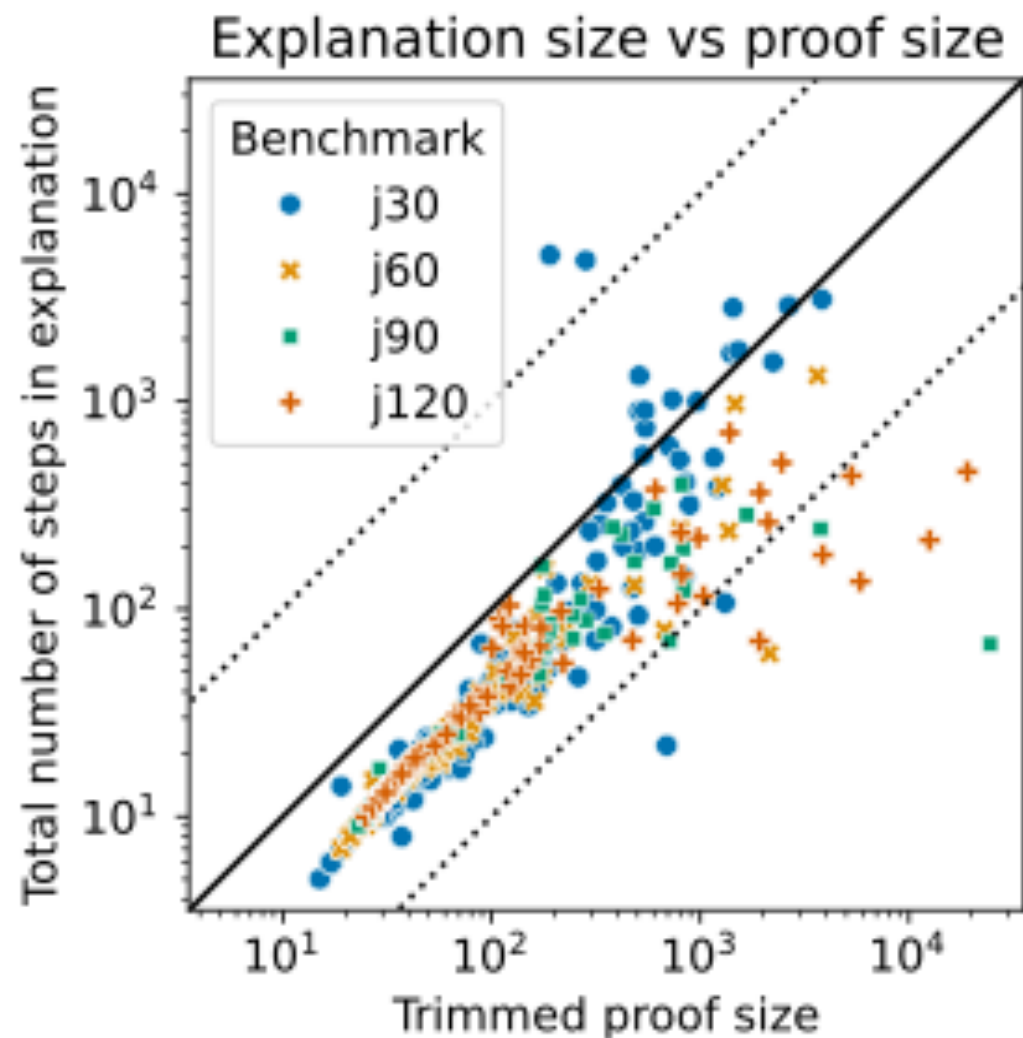
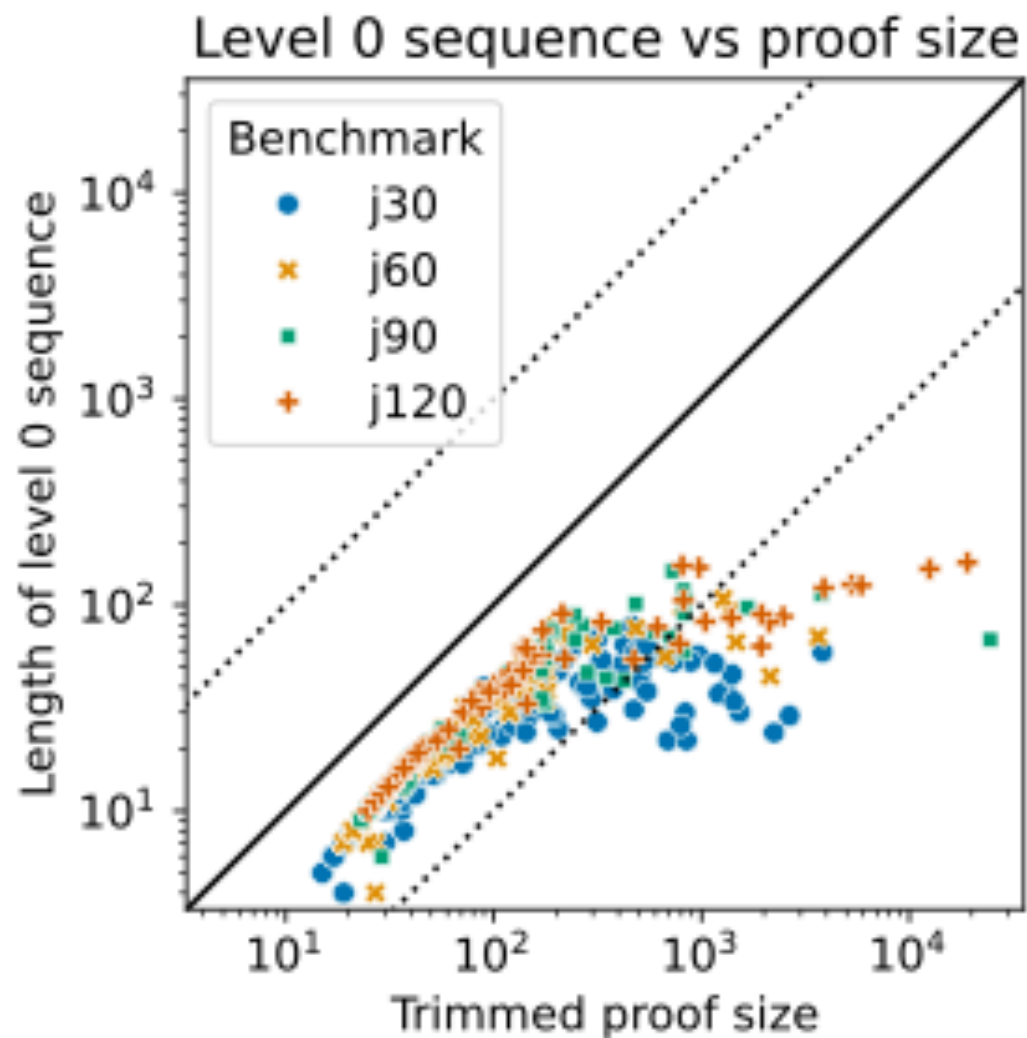
“Worst case scenario”

EQ1: How much nesting do we need?



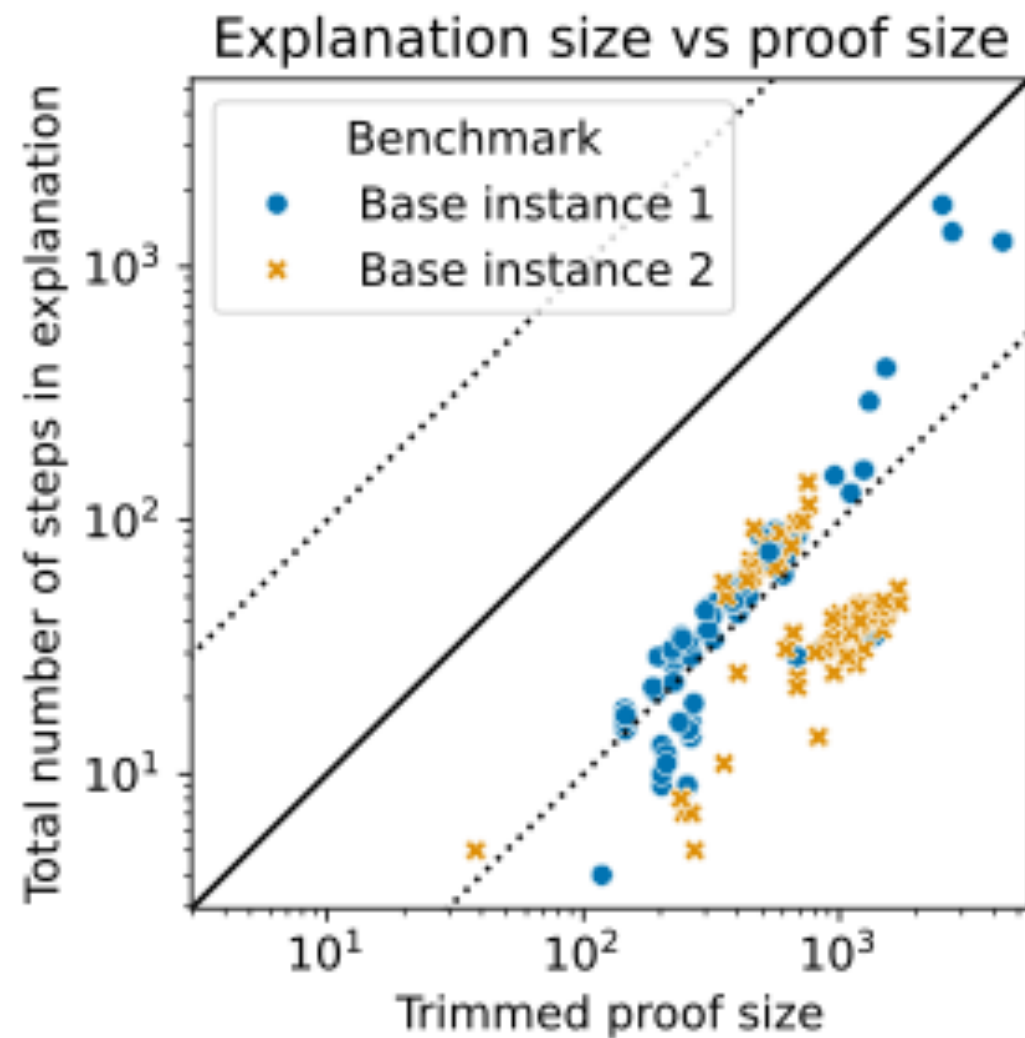
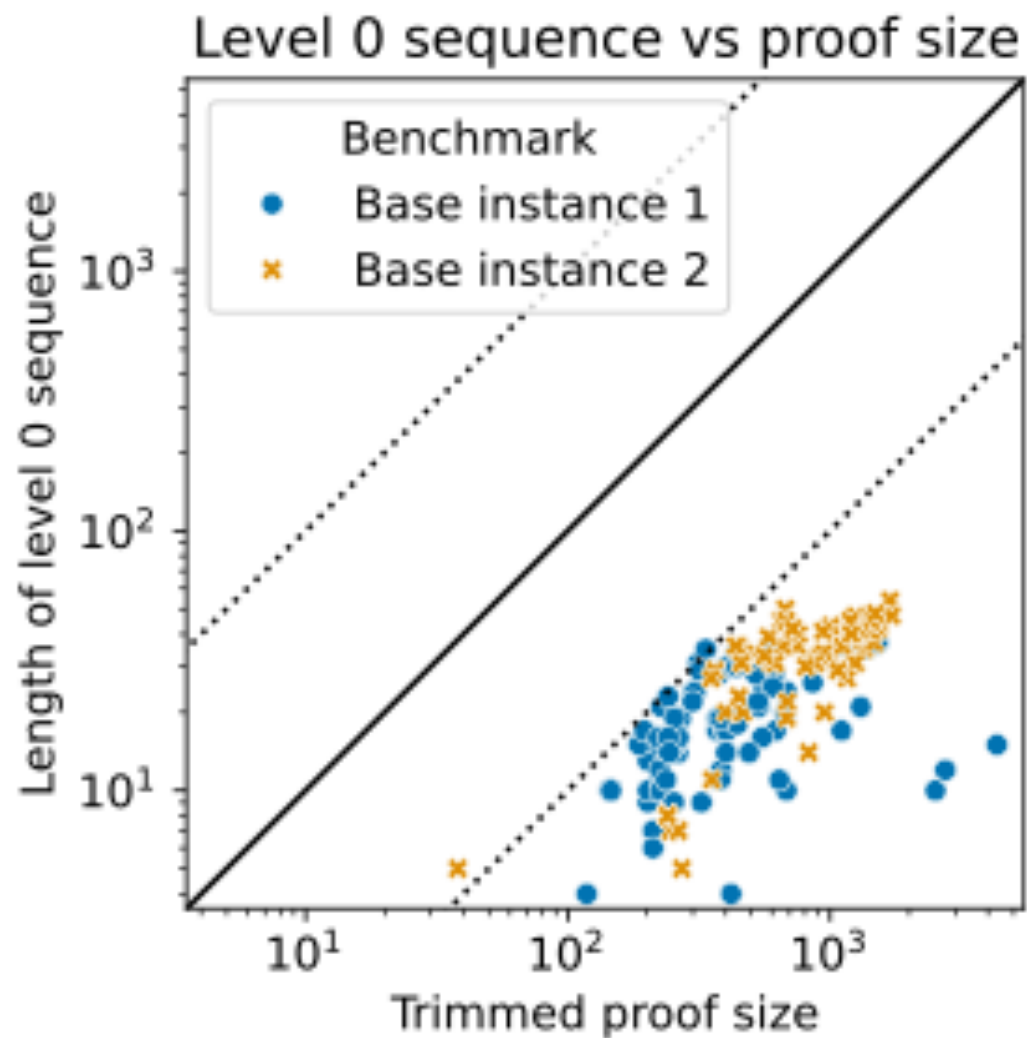
EQ2: Comparison to proof

RCPSP



EQ2: Comparison to proof

NR



Conclusions

Nested explanations can be used to explain CSPs with large search trees

Although surprisingly: no nesting needed for many instances!

Also several instances with exponential number of explanations steps

Outlook

Can we influence solvers to produce proofs corresponding to short explanation sequences?

- To improve explanation quality?

- To improve solving time?

What are alternative approaches to nesting?

- Use auxiliary variables in the explanation?

- Allow more complex derivations, beyond domain reductions



Questions?