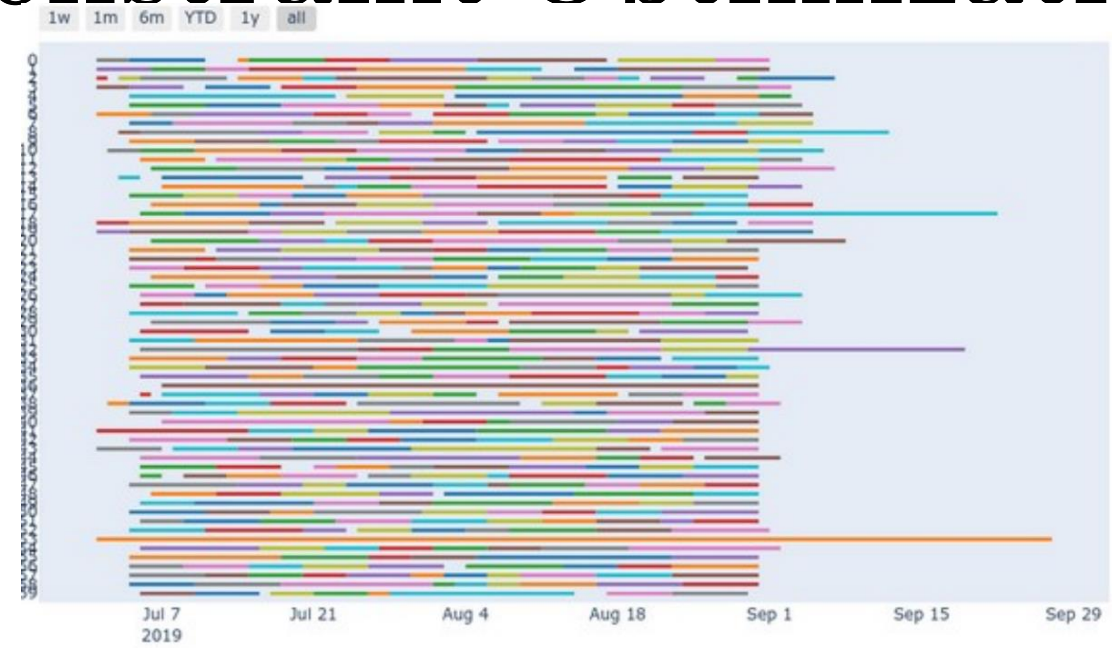
ML-guided Interactive Constraint Acquisition

published at JAIR:
https://www.jair.org/index.php/jair/article/view/19524

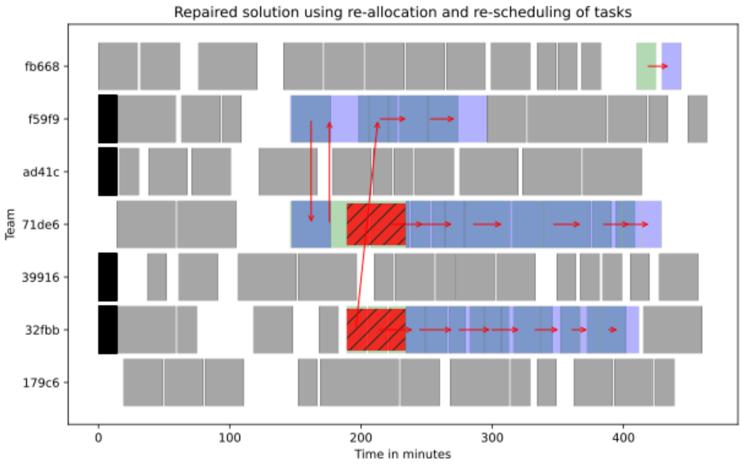
Dimos Tsouros, Senne Berden, Tias Guns

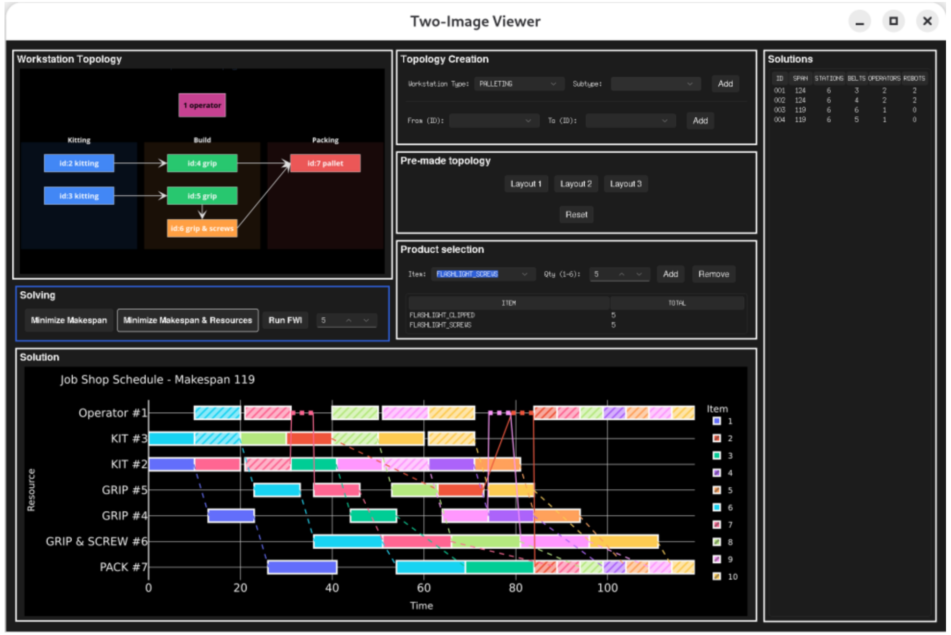
29/05/2026

Constraint Optimization Problems



	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
Kevin	D	D	F	F	F	F	F	F	D	D	D	D	D	F	7
Cover D	5/5	7/7	6/6	5/4	5/5	2/5	2/5	6/6	7/7	4/4	2/2	5/5	6/6	4/4	14

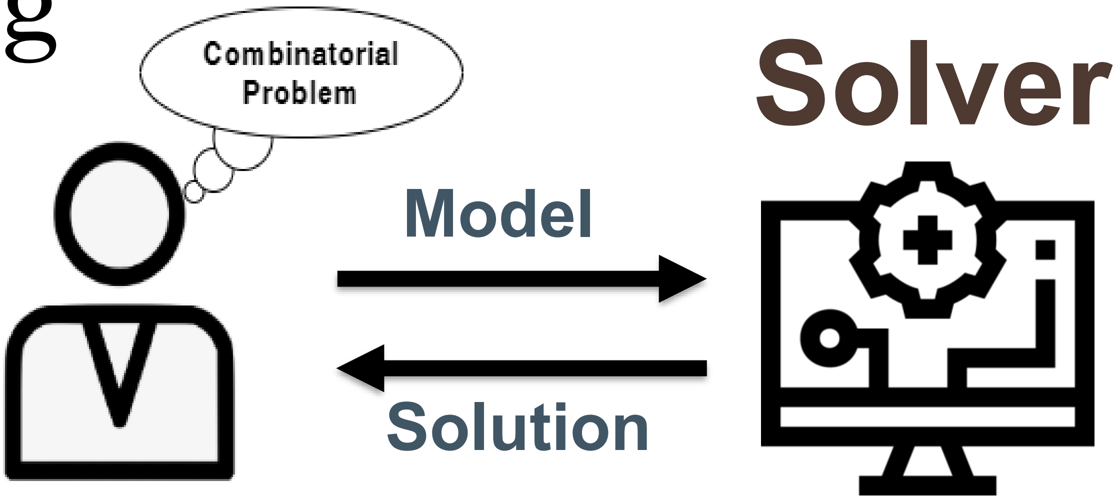




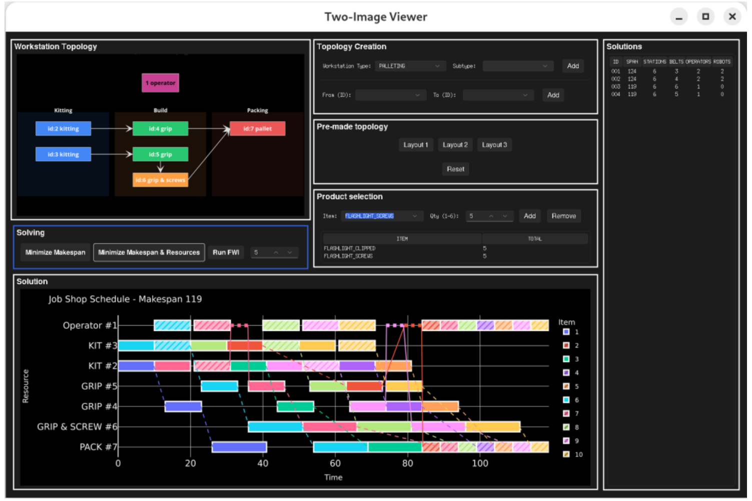
Solved and visualized using the CPMpy library in python

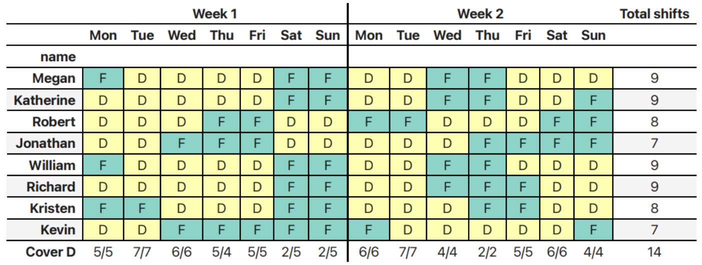
Constraint Modeling

Model + Solve paradigm



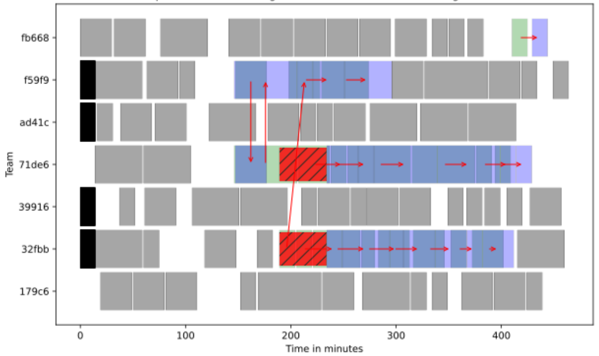
- Decision Variables
- Domains
- Constraints
- Objective (optional)



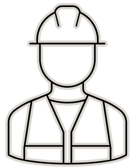


	Week 1							Week 2							Total shifts
	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
name															
Megan	F	D	D	D	D	F	F	D	D	F	F	D	D	D	9
Katherine	D	D	D	D	D	F	F	D	D	F	F	D	D	F	9
Robert	D	D	D	F	F	D	D	F	F	D	D	D	F	F	8
Jonathan	D	D	F	F	F	D	D	D	D	D	F	F	F	F	7
William	F	D	D	D	D	F	F	D	D	F	F	D	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
Kevin	D	D	F	F	F	F	F	F	D	D	D	D	D	F	7
Cover D	5/5	7/7	6/6	5/4	5/5	2/5	2/5	6/6	7/7	4/4	2/2	5/5	6/6	4/4	14

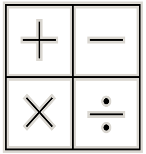
Repaired solution using re-allocation and re-scheduling of tasks



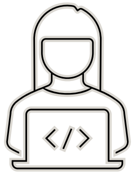
Constraint Solving technologies are powerful, but modeling requires expertise in:



Domain Knowledge



Mathematical Skills



Solver Formalisms

Making Constraint Solving More Accessible

Modelling Tools

Constraint Acquisition

Users provide or classify (non-)solutions

From Natural Language

Textual descriptions of combinatorial problems

Making Constraint Solving More Accessible

Modelling Tools

Modeling is not a one-shot procedure!

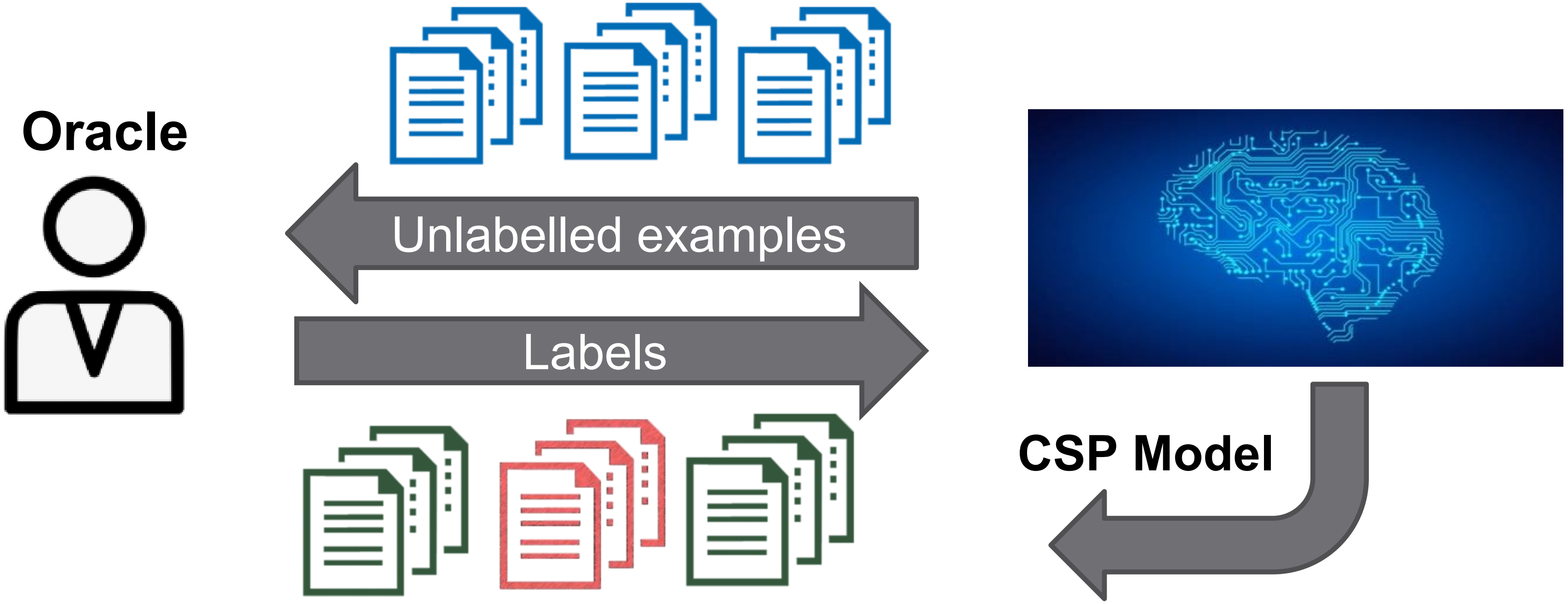
Constraint Acquisition

Users provide or classify (non-)solutions

From Natural Language

Textual descriptions of combinatorial problems

Interactive Constraint Acquisition (CA)



Membership query

1	1	3	4
3	2	1	1
2	2	3	1
2	3	4	3

Answer: Negative in both of them (a constraint is violated)

Partial query

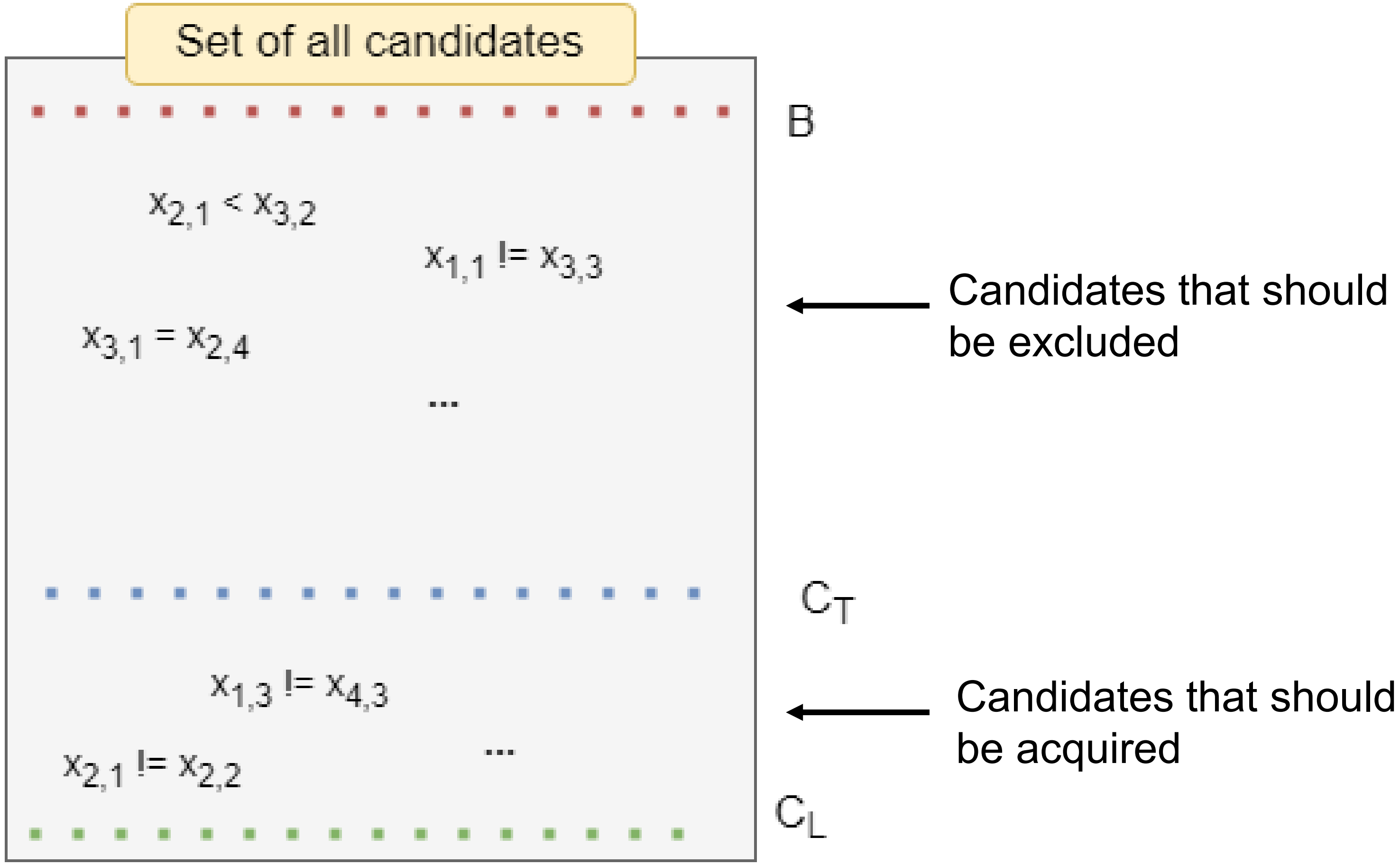
1	1	-	4
3	-	1	-
-	-	-	-
2	-	-	-

How to choose which examples to query to the user?

Learning Constraints through Partial Queries, C. Bessiere et al., AIJ, 2023

Interactive CA

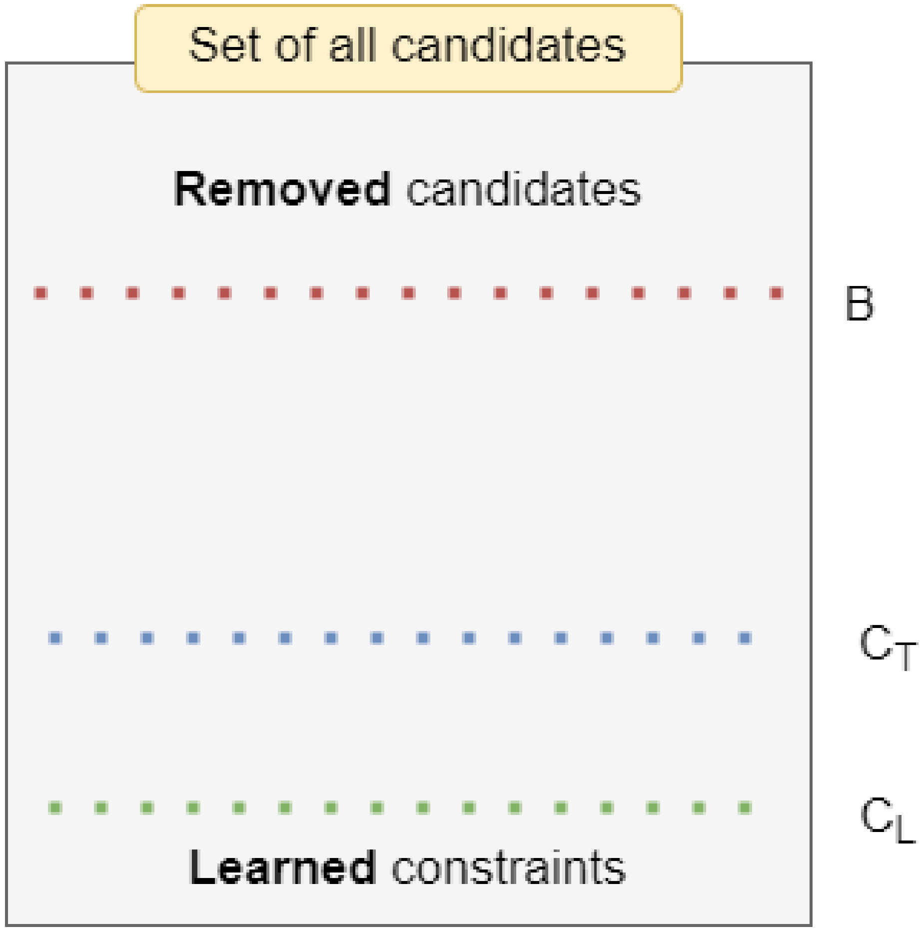
- **B**: set of (remaining) candidate constraints
- **C_T**: target set of constraints
- **C_L**: learned set of constraints



Interactive CA

During the learning process:

- Constraints are removed from *B*
- Constraints are added to C_L

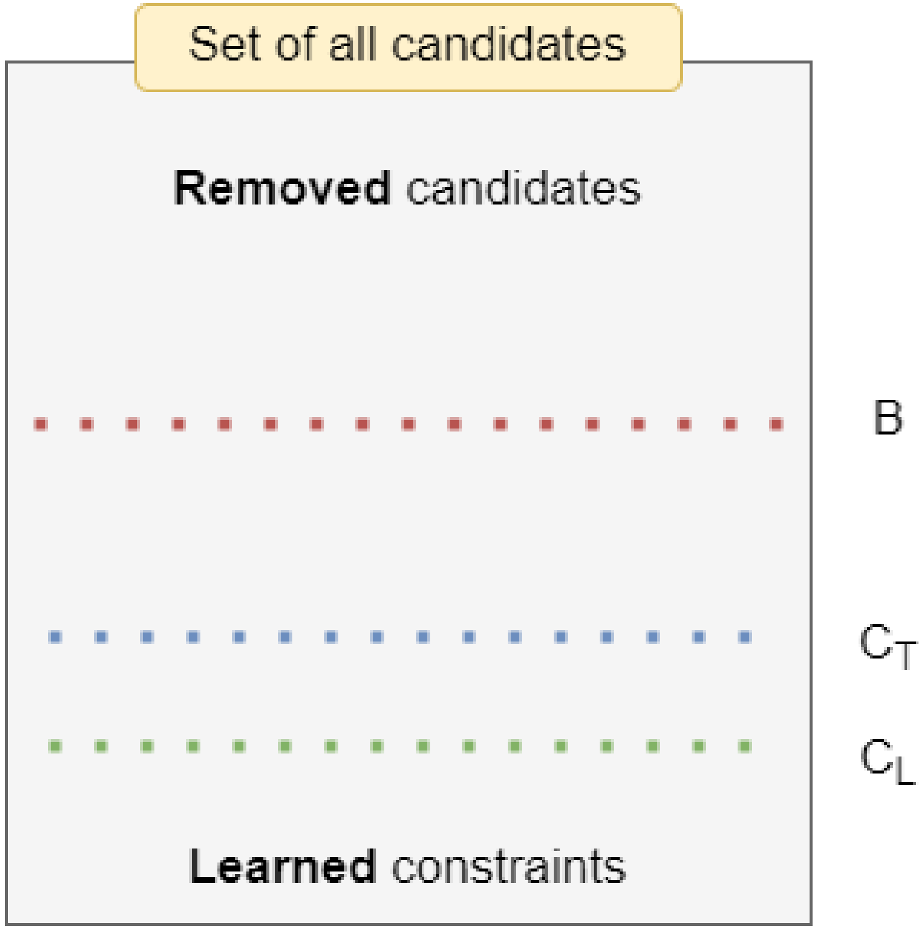


- **B**: set of (remaining) candidate constraints
- **C_T**: target set of constraints
- **C_L**: learned set of constraints

Interactive CA

During the learning process:

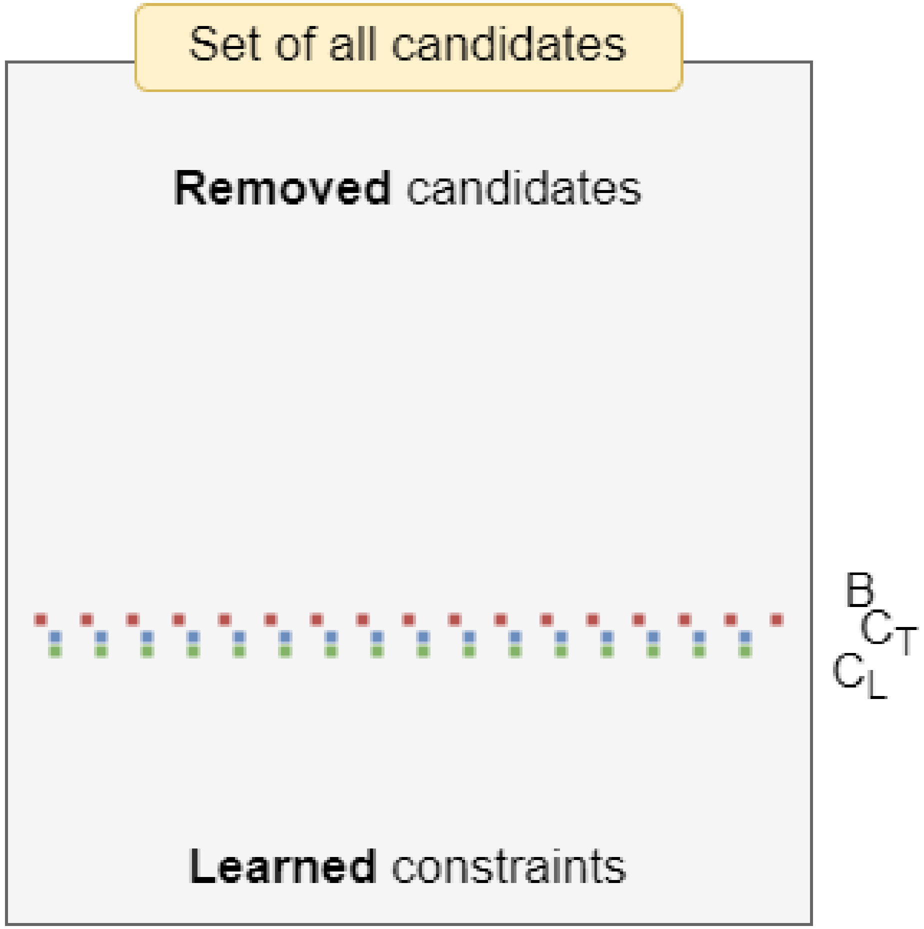
- Constraints are removed from B
- Constraints are added to C_L



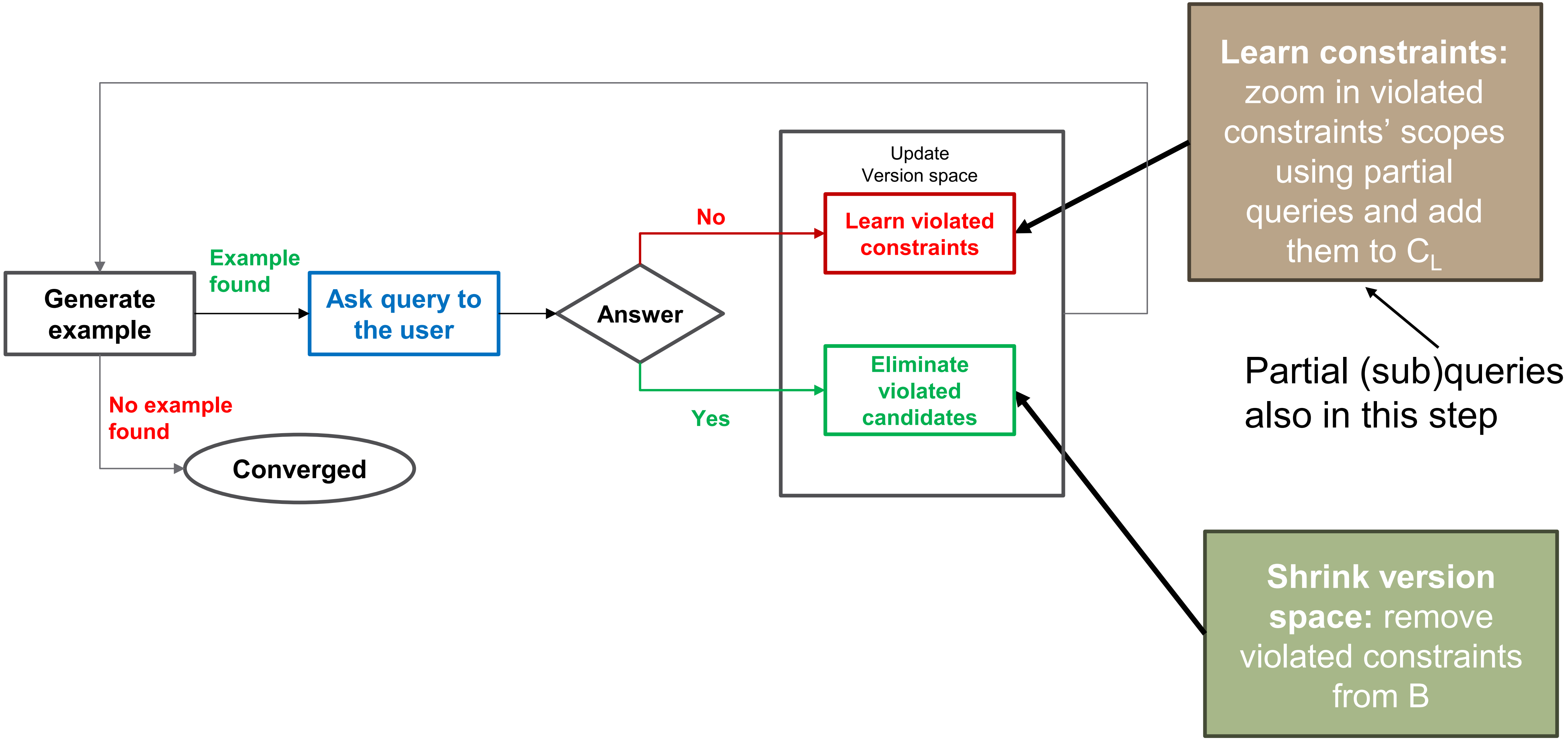
- **B**: set of (remaining) candidate constraints
- $\mathbf{C_T}$: target set of constraints
- $\mathbf{C_L}$: learned set of constraints

Interactive CA

If C_T is representable by B the CA system will eventually ***converge*** to a C_L equivalent to C_T



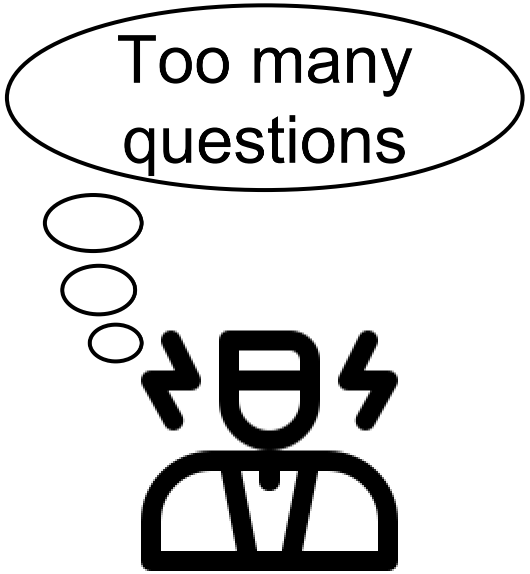
Interactive Constraint Acquisition



Learning Constraints through Partial Queries, C. Bessiere et al., AIJ, 2023

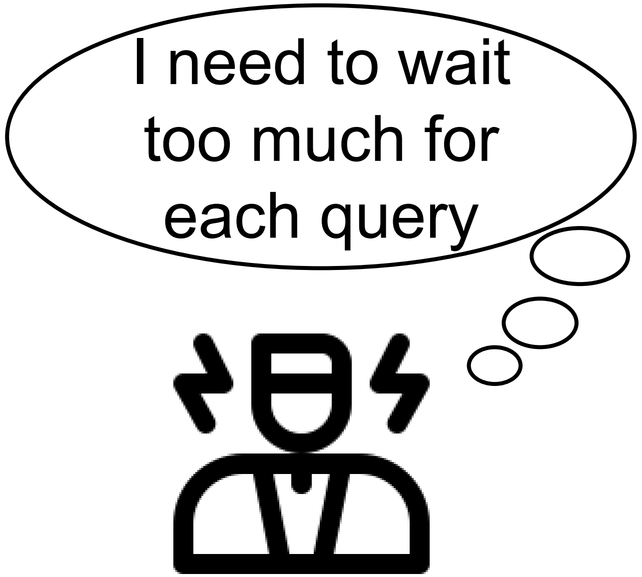
Challenges for interactive CA

Minimum number of queries



Even if oracle is a software system:
Simulations can be expensive.

Minimum waiting time for the user



Even if oracle is a software system:
Acquisition time can be important.

Learning constraint models from data, D Tsouros et al., 2023

Query generation

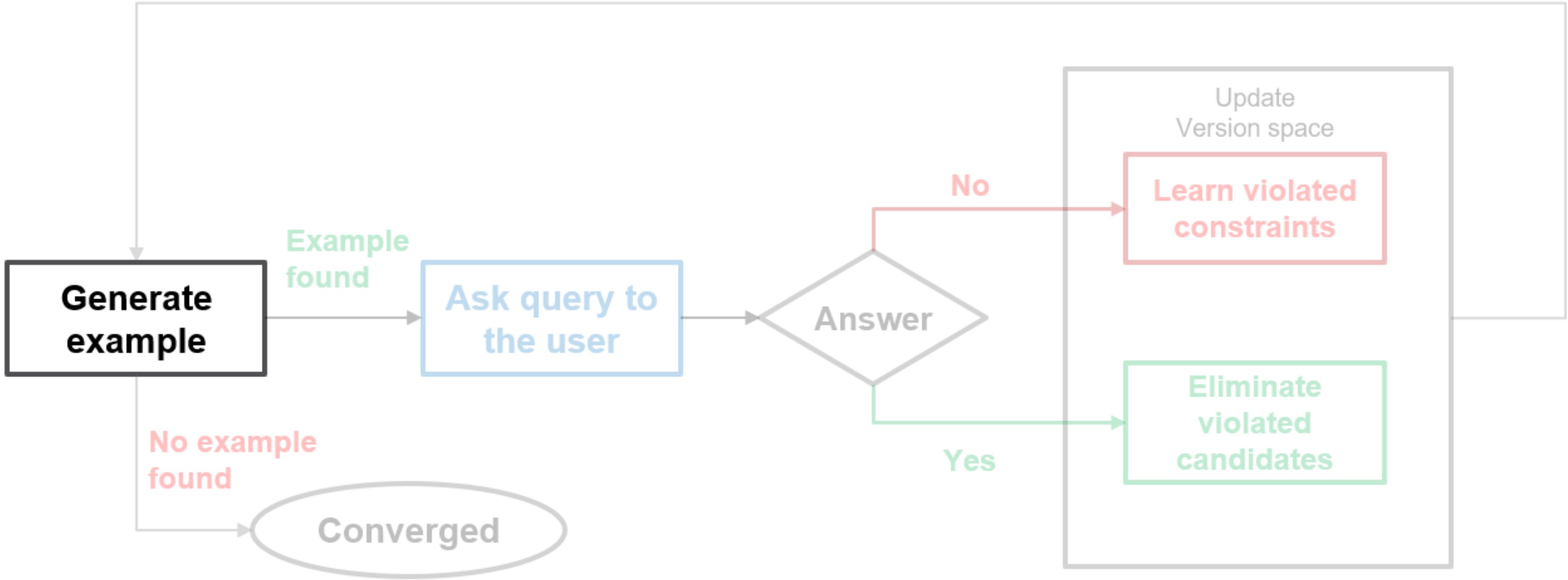
Why??

Find an Informative ("irredundant") query

- Not violating any learned constraint in C_L
- Violating at least one constraint from *B*

Queries give information only about the **violated** candidates

It is a ***CSP***: Find $e \in sol(C_L \wedge \bigvee_{c \in B} \sim c)$



Learning Constraints through Partial Queries, C. Bessiere et al., AIJ, 2023

Query generation

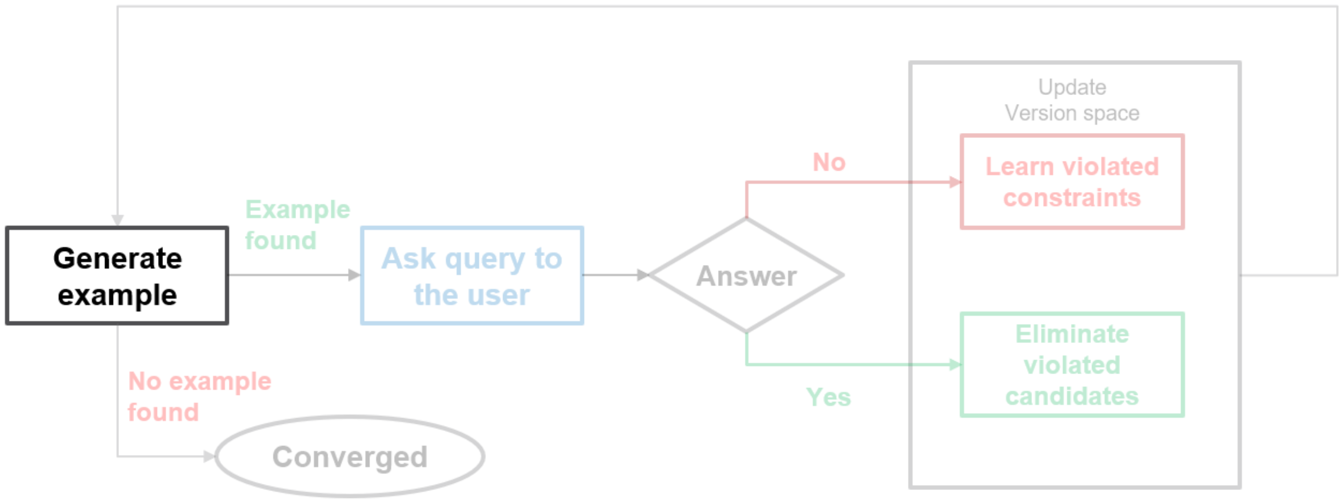
Quality of query

- *Better* generated examples lead to faster convergence
  - More information per query -> less queries needed

Literature: maximize candidate violations

$$\max \sum_{c \in B} \sim c$$

Not fully aligning with the goal!!



Learning Constraints through Partial Queries, C. Bessiere et al., AIJ, 2023

ML-guided Query Generation

Better generated examples lead to faster convergence

Positive answers: shrink *B* fast

- Negative answers: Find the conflict fast

The more we have violated the faster B will shrink

The less candidates we have violated, the less queries we need to find the constraint(s)

$$\boldsymbol{max} \sum_{c \in B} [\![e \notin sol(c)]\!]$$

Opposite objectives based on the (future) answer

$$\boldsymbol{min} \sum_{c \in B} [\![e \notin sol(c)]\!]$$

ML-guided Query Generation

Better generated examples lead to faster convergence

Positive answers: shrink *B* fast

- Negative answers: Find the conflict fast

We cannot know the answer of the user before we ask the query → max violations

The
fast

But what if we can predict if a candidate is a constraint of the problem or not?

(and relation) of the constraint(s)

$$\boldsymbol{max} \sum_{c \in B} [\![e \notin sol(c)]\!]$$

Opposite objectives based on the (future) answer

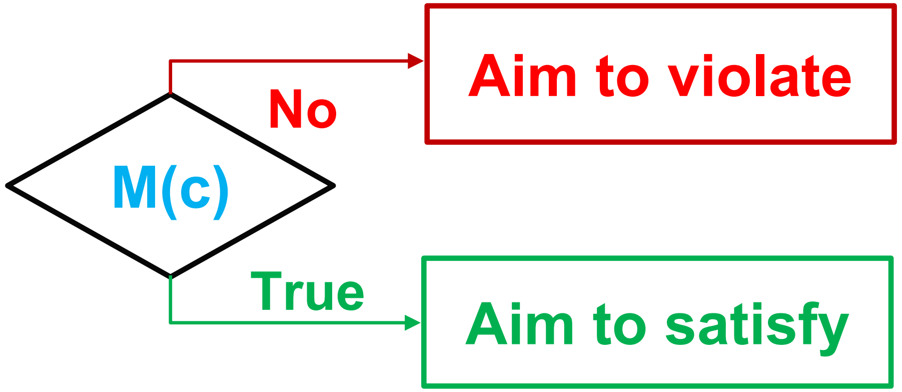
$$\boldsymbol{min} \sum_{c \in B} [\![e \notin sol(c)]\!]$$

ML-guided Query Generation

- Positive answers: shrink B fast → $\boldsymbol{max} \sum_{c \in B} [\![\boldsymbol{e} \notin \boldsymbol{sol}(\boldsymbol{c})]\!]$
- Negative answers: Find the conflict fast → $\boldsymbol{min} \sum_{c \in B} [\![\boldsymbol{e} \notin \boldsymbol{sol}(\boldsymbol{c})]\!]$

Use of *model* $\mathrm{M(c)} = (\mathrm{c} \in C_T)$, to guide query generation based on the *prediction of the constraint*

$$e = \underset{e \in Sol(C_L \wedge B)}{\operatorname{argmax}} \sum_{c \in B} \overbrace{[\![e \notin sol(c)]\!]}^{\text{If the constraint } c \text{ is violated}} \cdot (\overbrace{1}^{\text{Increase objective value by 1}} - \overbrace{k \cdot [\![M(c)]\!]}^{\text{If it is a constraint predicted to be true: reduce objective value significantly}})$$



Why?

1. Aim for positive answers first: $\boldsymbol{max}(\sum_{c \in B} [\![\boldsymbol{e} \notin \boldsymbol{sol}(\boldsymbol{c})]\!])$

2. When a (probably true) constraint has to be violated, leading to a *negative answer* $\boldsymbol{min}(\sum_{c \in B} [\![\boldsymbol{e} \notin \boldsymbol{sol}(\boldsymbol{c})]\!])$

ML-guided Query Generation

- Positive answers: shrink *B* fast → $\boldsymbol{max} \sum_{c \in B} [\![\boldsymbol{e} \notin \boldsymbol{sol}(\boldsymbol{c})]\!]$
- Negative answers: Find the conflict fast → $\boldsymbol{min} \sum_{c \in B} [\![\boldsymbol{e} \notin \boldsymbol{sol}(\boldsymbol{c})]\!]$

Predictions for <u>constraints</u> not for variable assignments!!

Use of *model* $\mathrm{M(c)} = (\mathrm{c} \in C_T)$, to guide query generati… on the *prediction of the constraint*

…straint *c* is violated

Increase objective value by 1

If it is a constraint predicted to be true: reduce objective value significantly

$$e = \ldots (C_L \wedge B) \sum_{c \in B} [\![e \notin sol(c)]\!] \cdot (1 - k \cdot [\![M(c)]\!])$$

M(c)

No → **Aim to violate**

True → **Aim to satisfy**

Why?

1. Aim for positive answers first: $\boldsymbol{max}(\sum_{c \in B} [\![\boldsymbol{e} \notin \boldsymbol{sol}(\boldsymbol{c})]\!])$

2. When a (probably true) constraint has to be violated, leading to a *negative answer* $\boldsymbol{min}(\sum_{c \in B} [\![\boldsymbol{e} \notin \boldsymbol{sol}(\boldsymbol{c})]\!])$

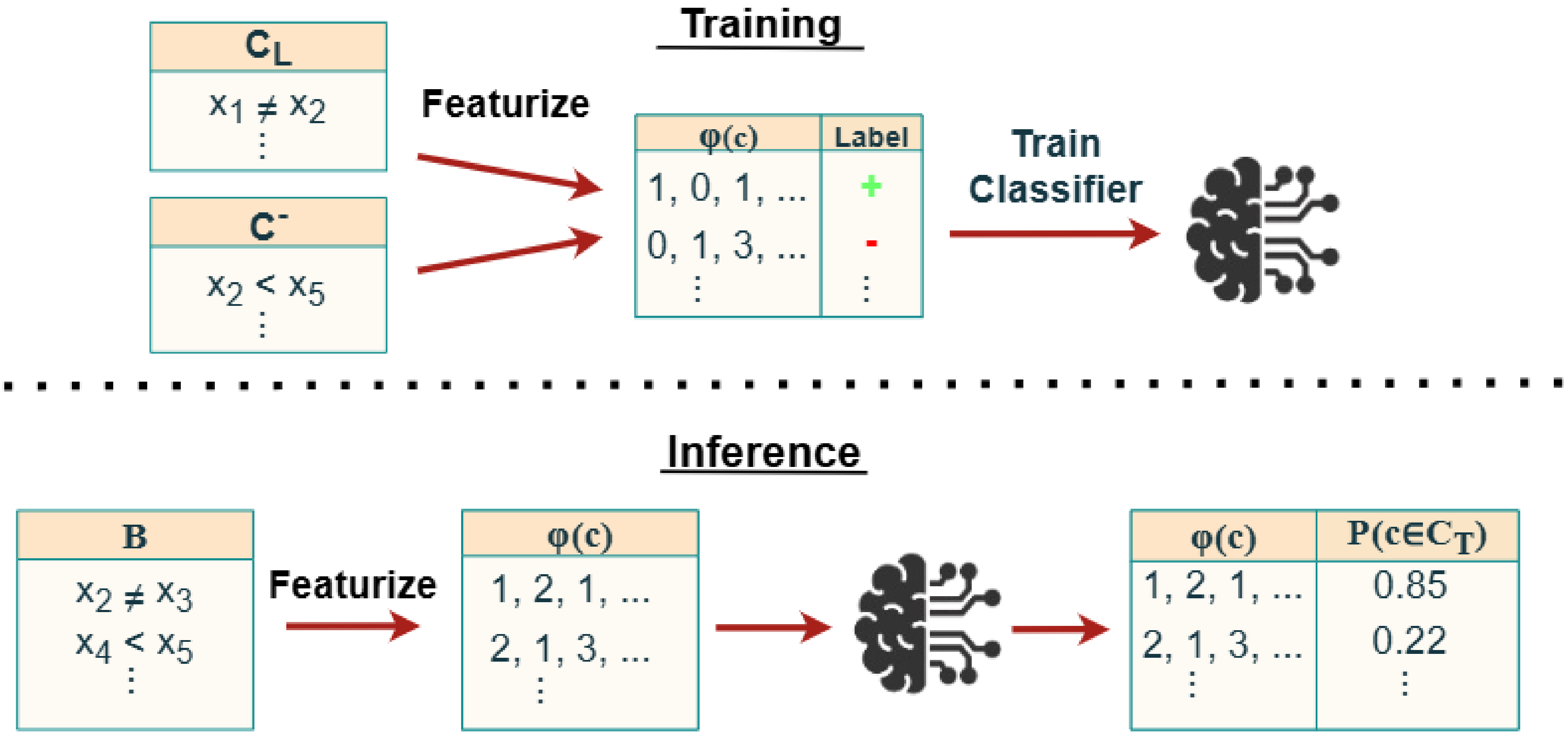
Using Machine Learning for the prediction

Dataset: Constraint features and class (True or False)
- Constructing during the acquisition process
- When a constraint is learned add a positive instance
- When a constraint is removed from *B add a negative instance*
- Use both relation and scope features

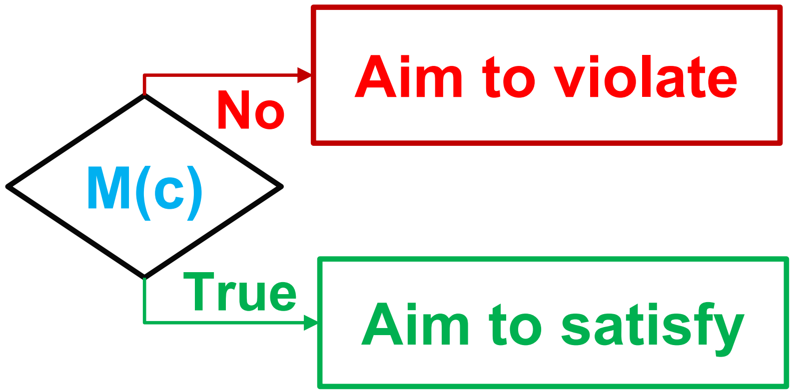
φ(c)	Label
1, 0, 1, ...	+
0, 1, 3, ...	-
⋮	⋮

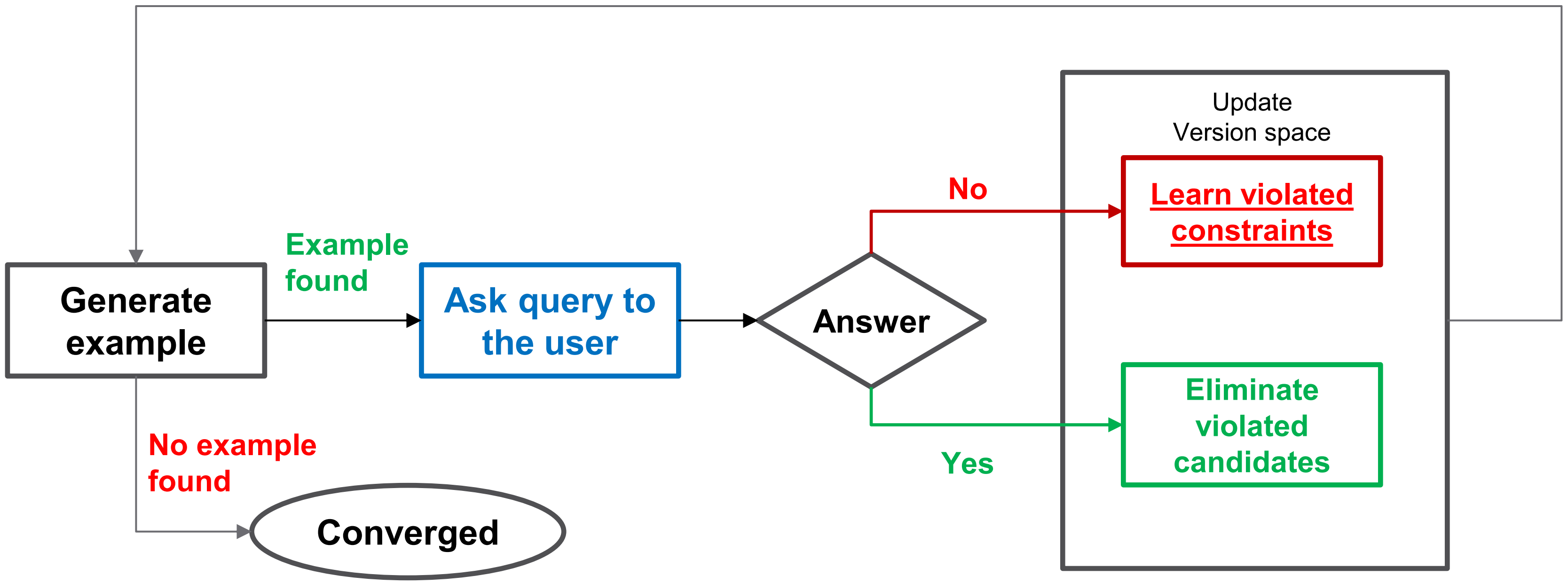
Relation-based features	Scope-based features
Relation name (string)	Dim[i] same_val (Bool)
Has constant (Bool)	Dim[i] avg (float)
Constant value (int)	Dim[i] distance (int)
Arity (int)	…

Learning to Learn in Interactive CA



In the paper: How to guide all parts of Interactive CA

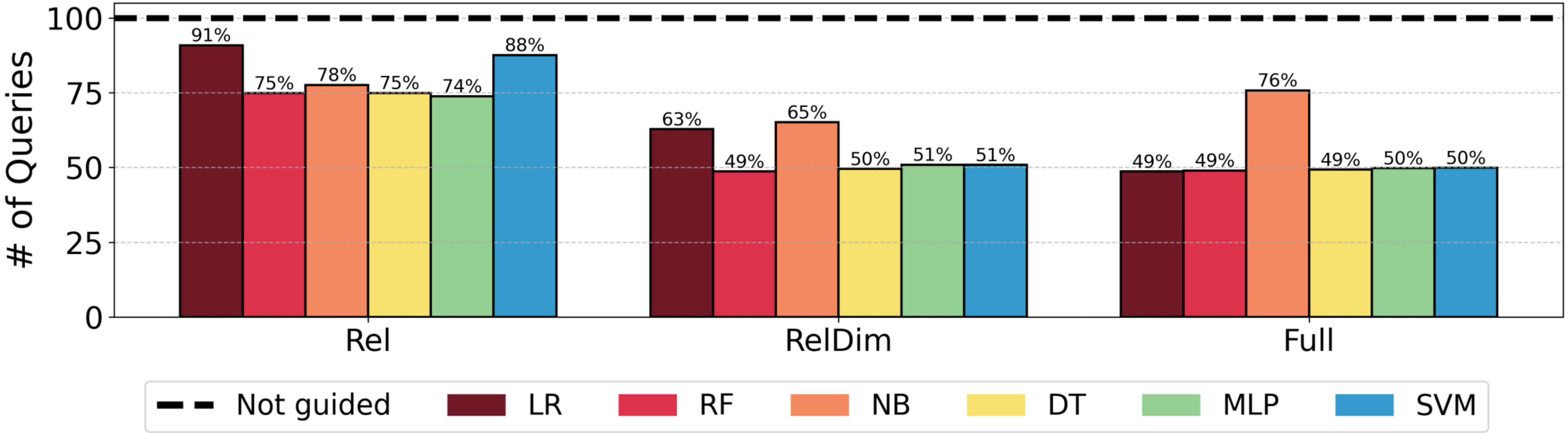




Guiding *also partial queries when, after a negative query, the system* ***finds the scope,*** *and the* ***exact constraint****!*

+ new FindScope function with integrated query generation

Results



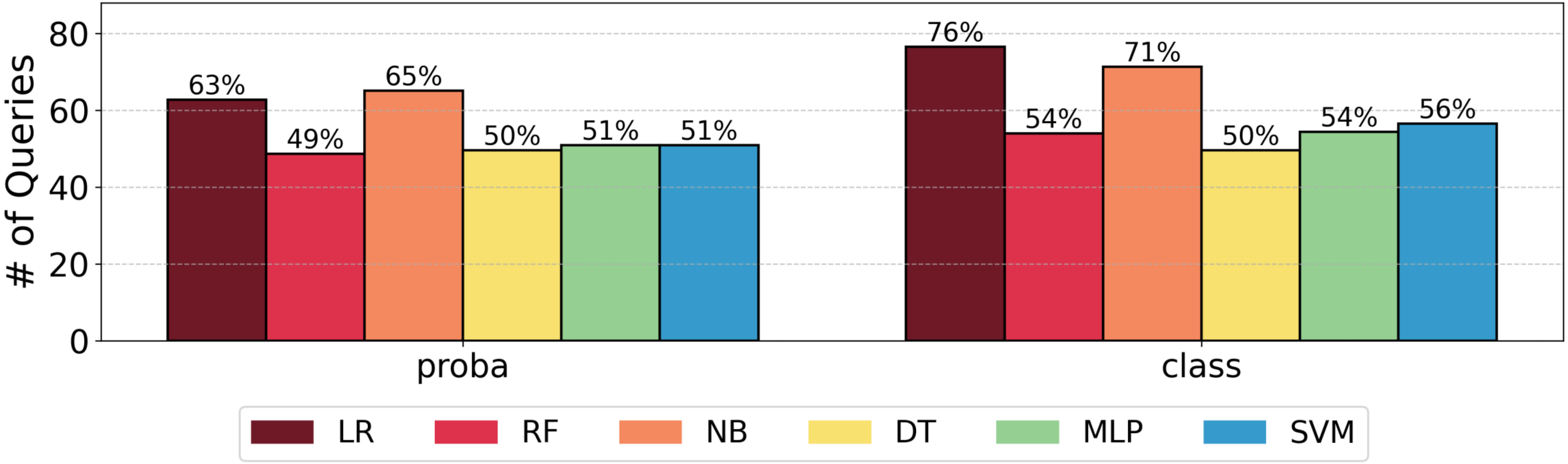
Results when guiding query generation:

- ➢ Different **classifiers**
- ➢ Different **expressivity on feature representation**:
  - *Rel* (low expressivity),
  - *RelDim* (Medium expressivity),
  - *Full* (all features)

Average over 7 benchmarks:

1. Sudoku
2. JSudoku
3. GTSudoku
4. Random
5. Job-shop
6. Exam timetabling
7. Nurse rostering

Results



Average results comparing the use of

- **probabilities**
- **predicted class**

Future work

Number of Queries

- More natural interaction
- Generalize while learning to patterns

Need of tractable bias

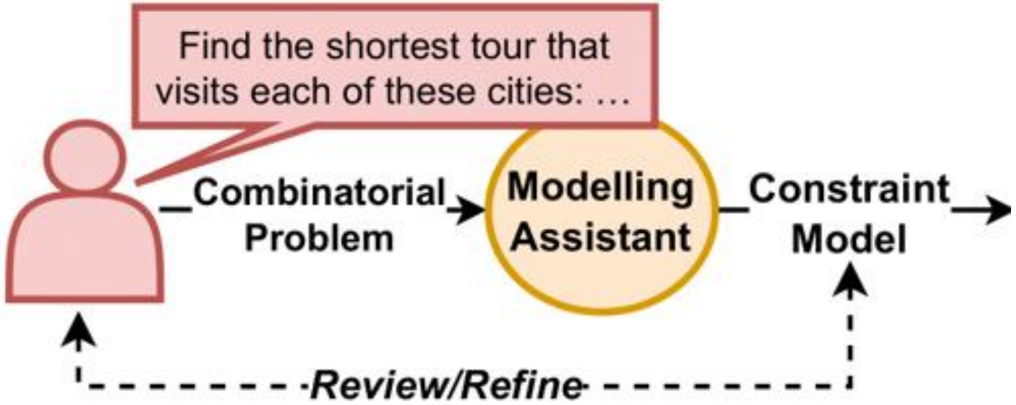
- LLM-based interaction can avoid the need of Bias

Noisy data

- Robust CA methods for handling noise.

With LLM-based interaction

+integrate in LLM assistant modeling framework



ML-guided Interactive Constraint Acquisition

Paper

https://doi.org/10.1613/jair.1.19524

Code

https://github.com/Dimosts/ML-guided-CA

PyConA

https://github.com/CPMpy/PyConA

CP + ML Survey - JAIR

https://jair.org/index.php/jair/article/view/19533