ML-guided and Robust Constraint Acquisition

Dimos Tsouros, Hendrik ‘Henk’ Bierlee, Senne Berden, Tias Guns

2025-08-11 - Progress Towards The Holy Grail

Funded by the European Union

European Research Council
Established by the European Commission

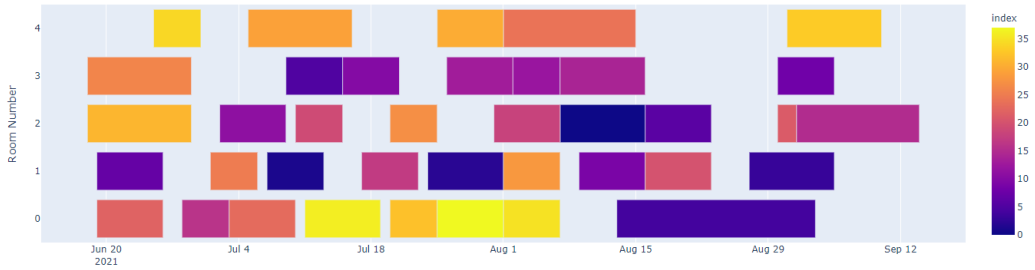
DTAI
Declarative Languages & Artificial Intelligence

KU LEUVEN

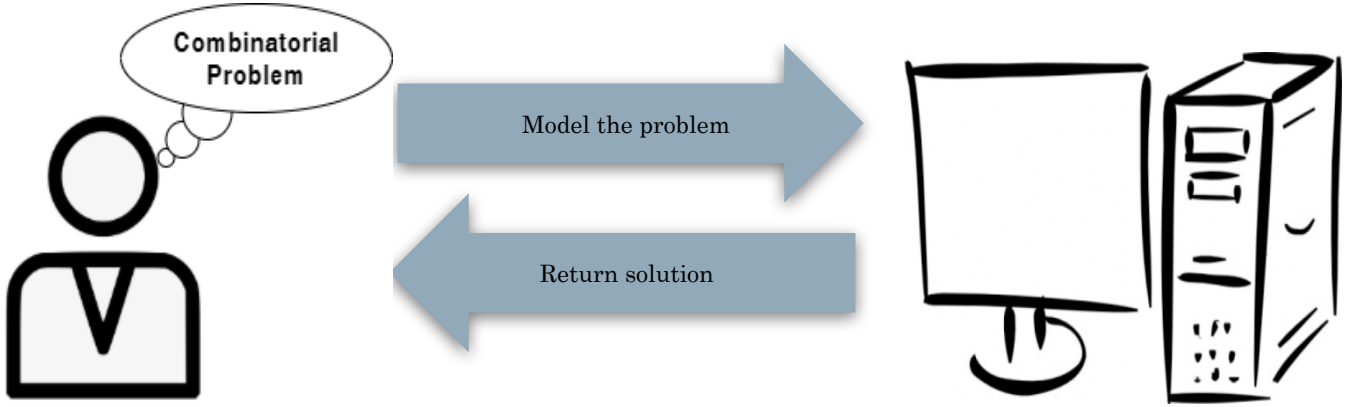
Introduction

- Constraint programming (CP)
  - Solving combinatorial problems in AI

	Week 1							Week 2							Total shifts
	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
name															
Megan	-	D	D	D	D	-	-	D	D	-	-	D	D	D	9
Katherine	D	D	D	D	D	-	-	-	D	D	-	-	D	D	9
Robert	D	D	D	-	-	D	D	D	-	-	D	D	-	-	8
Jonathan	D	D	-	-	-	D	D	D	D	D	-	-	-	-	7
William	-	D	D	D	D	-	-	D	D	-	-	D	D	D	9
Richard	D	D	D	-	-	-	D	D	D	D	-	-	-	-	7
Kristen	-	-	D	D	D	-	-	D	D	-	-	D	D	D	8
Kevin	D	D	-	-	D	D	-	-	D	D	D	D	-	-	8
Cover D	5/5	7/7	6/6	4/4	5/5	3/5	3/5	6/6	7/7	4/4	2/2	5/5	4/6	4/4	14



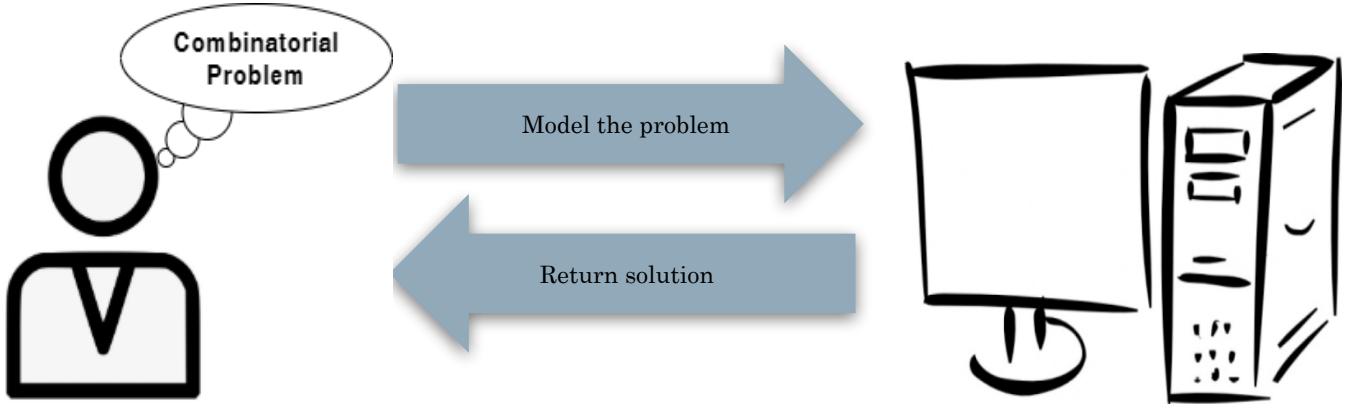
- Model + Solve paradigm



KU LEUVEN

Introduction

- Model + Solve paradigm



- Variables *X* with their domains *D*

```
# Variables
grid = cp.intvar(1, grid_size, shape=(grid_size, grid_size), name="grid")
```

- Constraints of the problem:

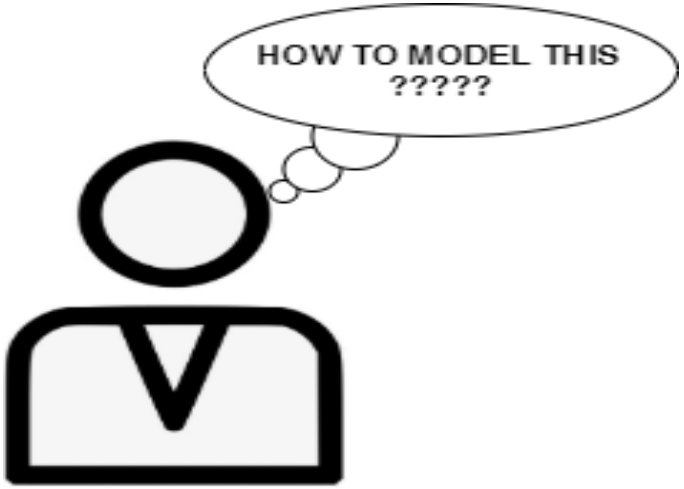
```
# Constraints on rows and columns
for row in grid:
    model += cp.AllDifferent(row)

for col in grid.T:  # numpy's Transpose
    model += cp.AllDifferent(col)

# Constraints on blocks
for i in range(0, grid_size, block_size_row):
    for j in range(0, grid_size, block_size_col):
        model += cp.AllDifferent(grid[i:i + block_size_row, j:j + block_size_col])
```

KU LEUVEN

Introduction

Modelling is not always trivial

- Requires expertise
- Bottleneck for the wider use of CP

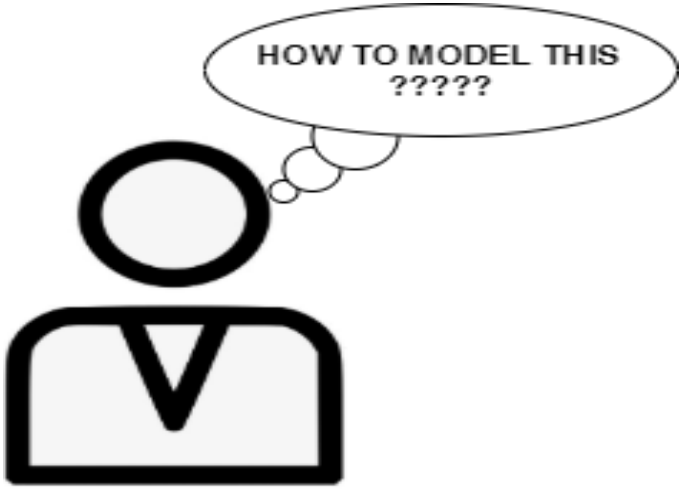


KU LEUVEN

Introduction

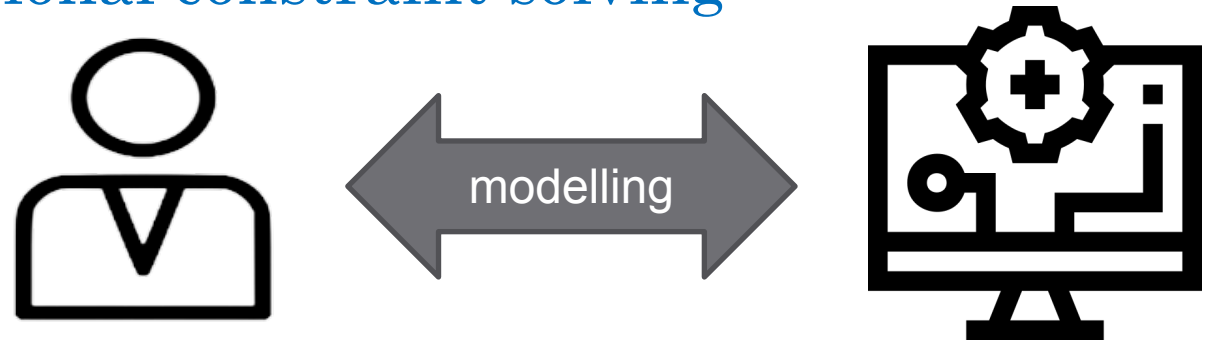
Modelling is not always trivial

- Requires expertise
- Bottleneck for the wider use of CP



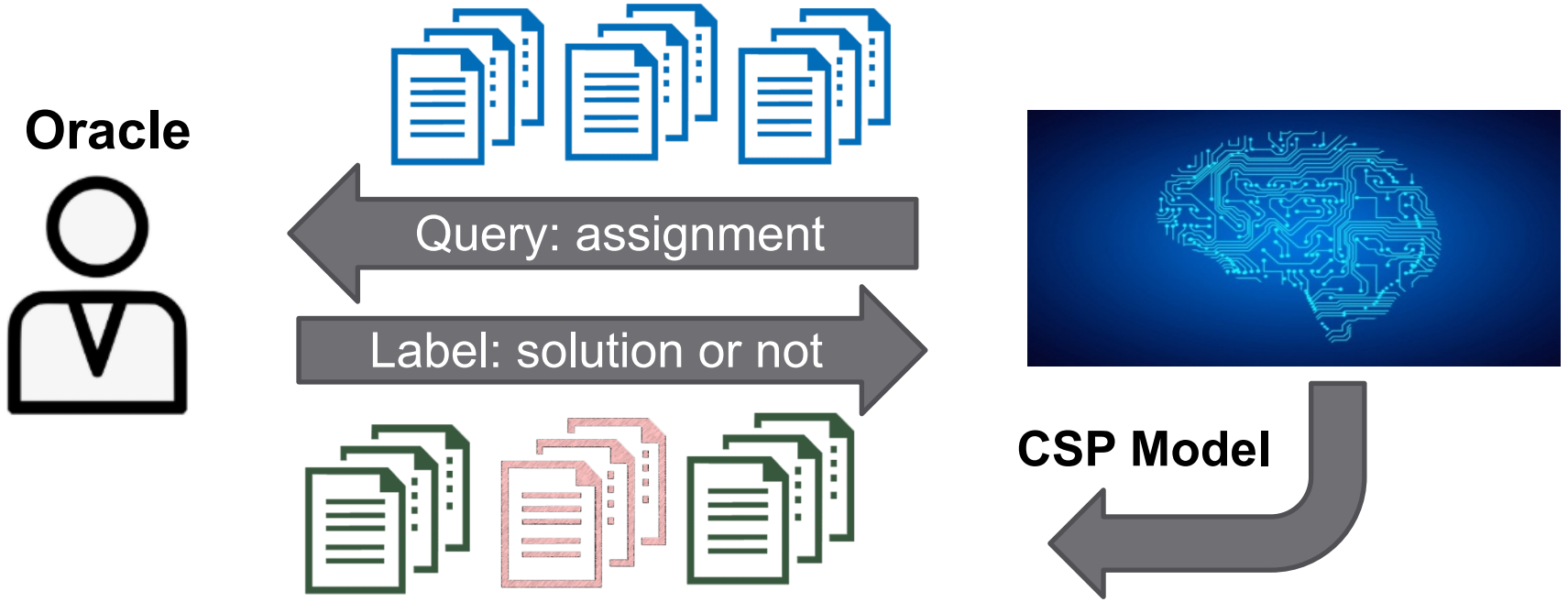
Constraint Acquisition:

- System and user modeling together!
- Conversational constraint solving



KU LEUVEN

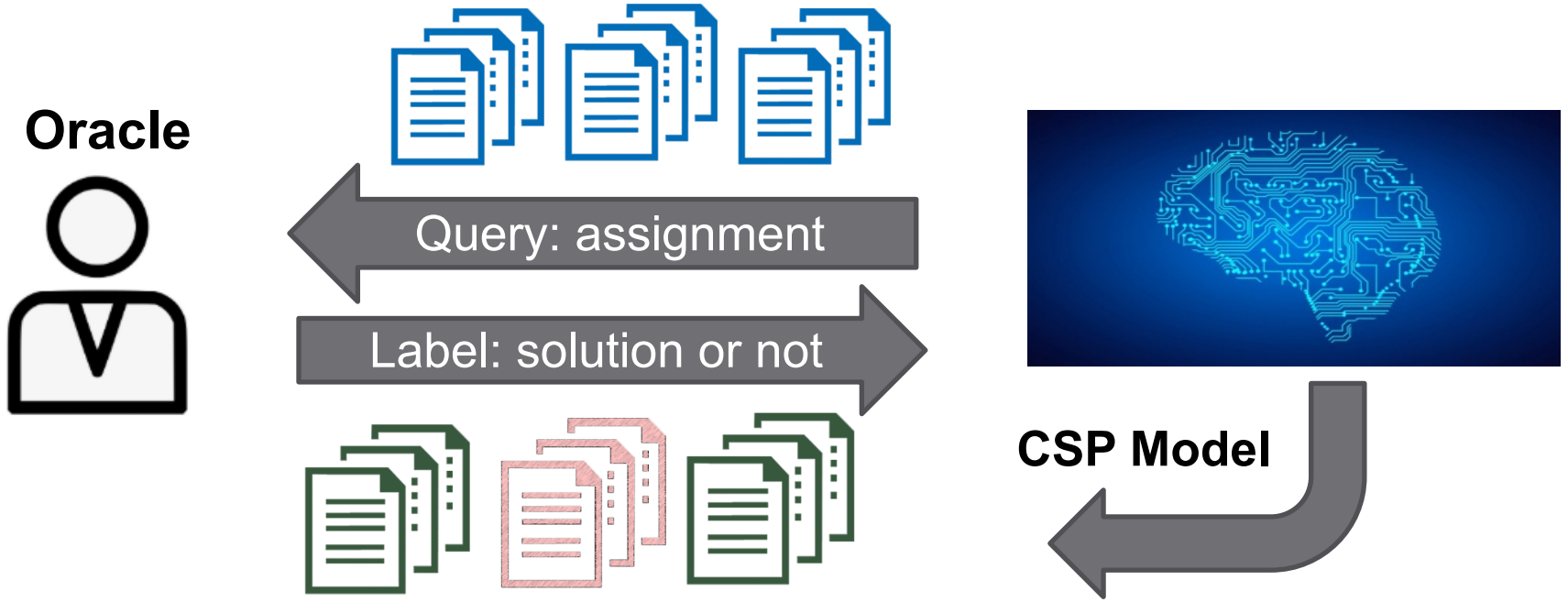
Interactive Constraint Acquisition



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

KU LEUVEN

Interactive Constraint Acquisition



Membership query

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

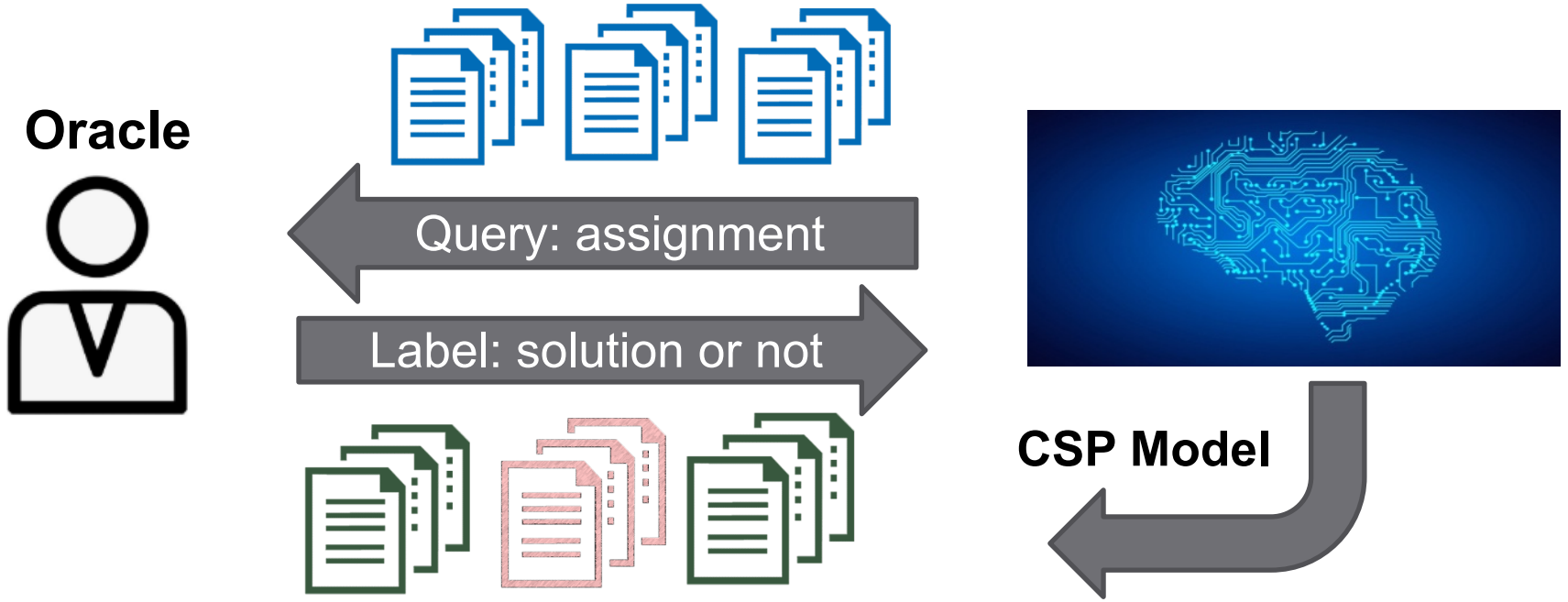
Partial query

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

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

KU LEUVEN

Interactive Constraint Acquisition



Membership query

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

Answer: "Non-solution" in both of them (a constraint is violated)

Partial query

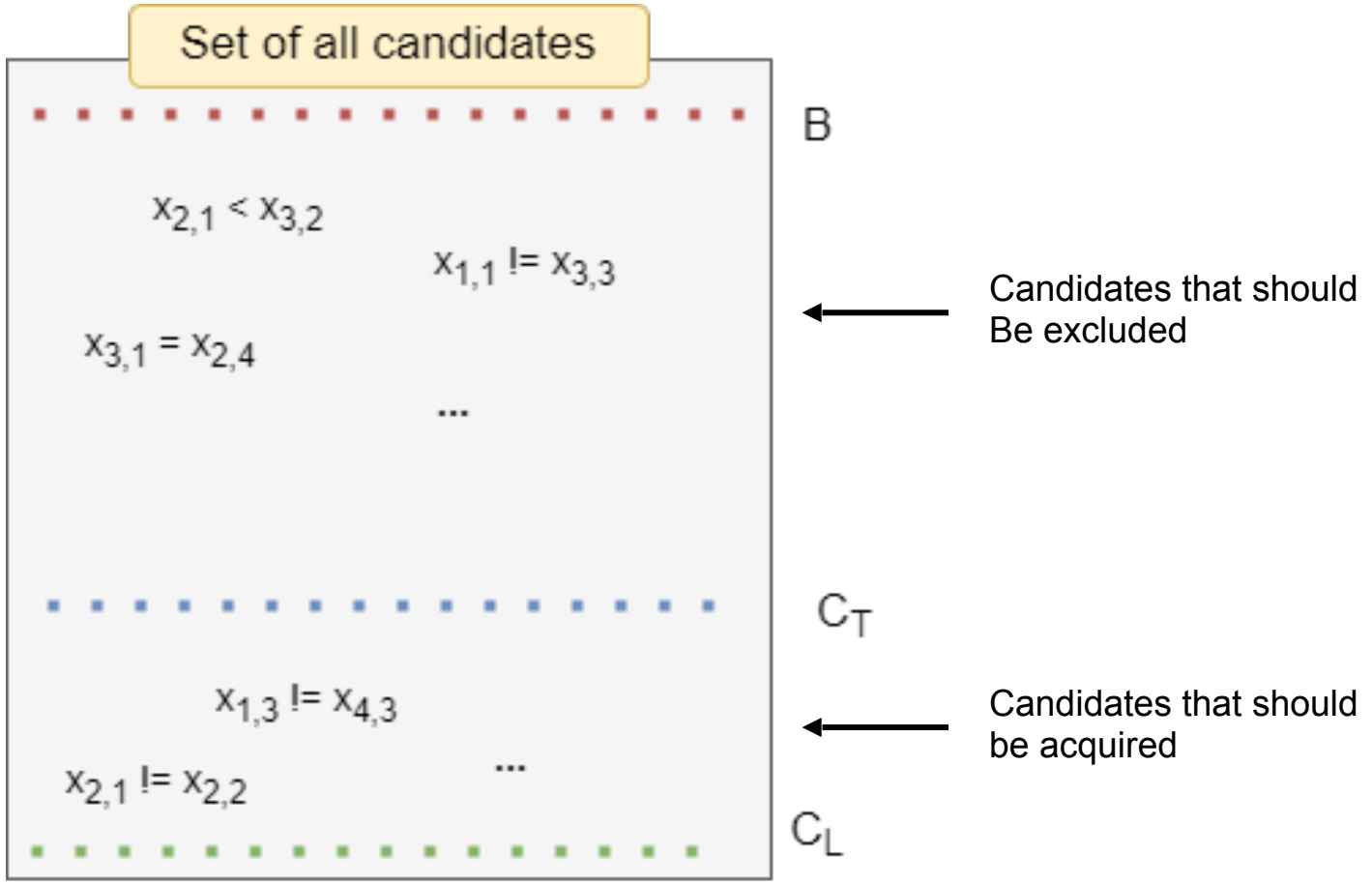
1	1	-	4
3	-	1	-
-	-	-	-
2	-	-	-

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

KU LEUVEN

Candidate Elimination

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

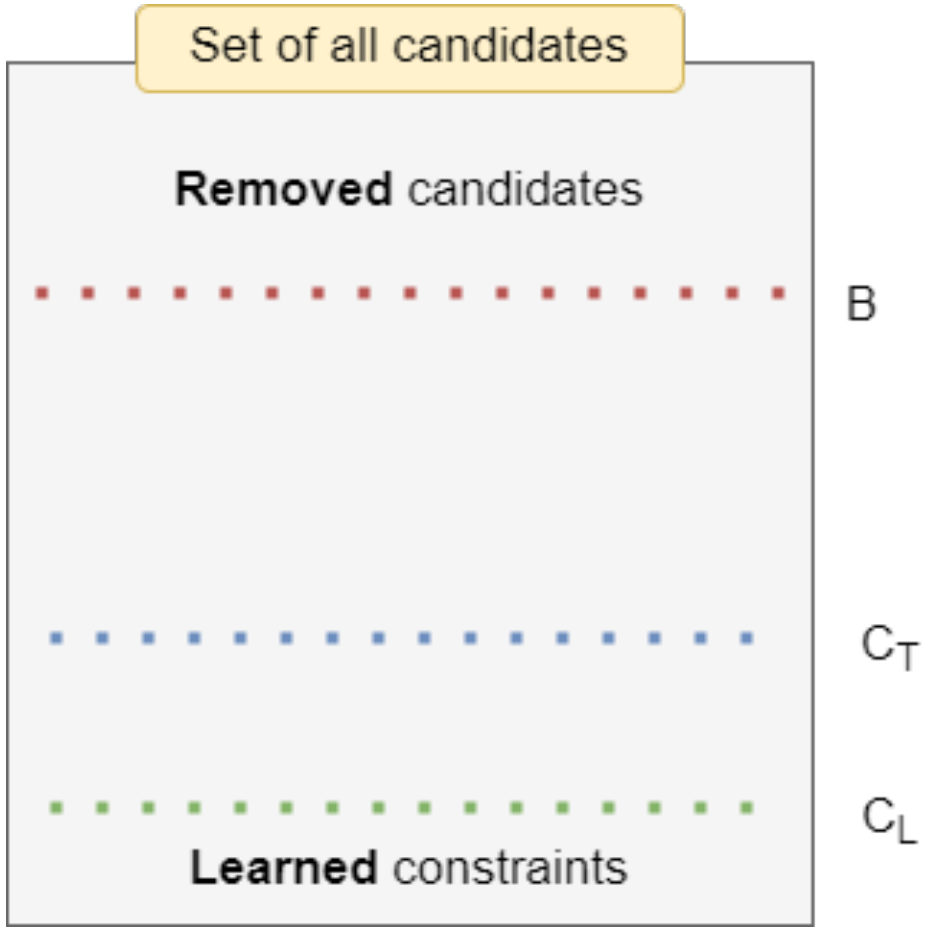


KU LEUVEN

Candidate Elimination

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

KU LEUVEN

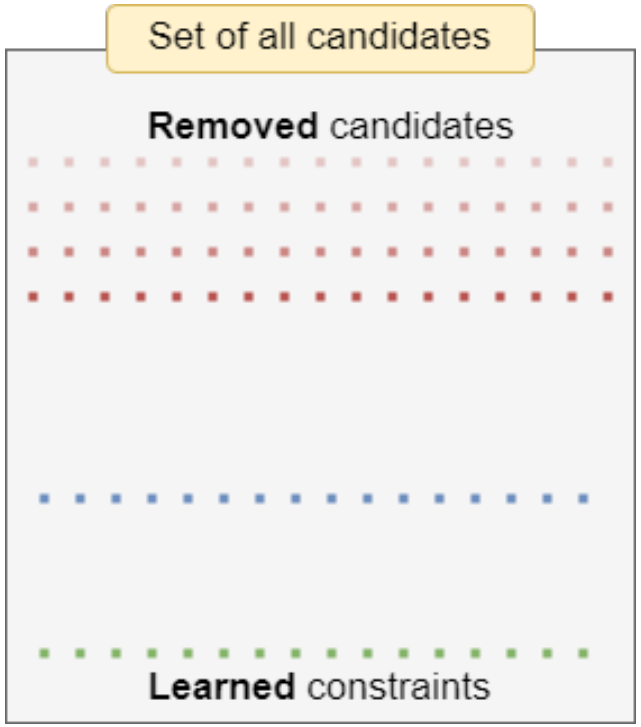
Learning from examples

- Examples: Assignments to the variables of the problem

- Learning from ***positive*** examples ("Solution"):

- Violated constraints cannot be part of the model
- Otherwise, it could not be a solution

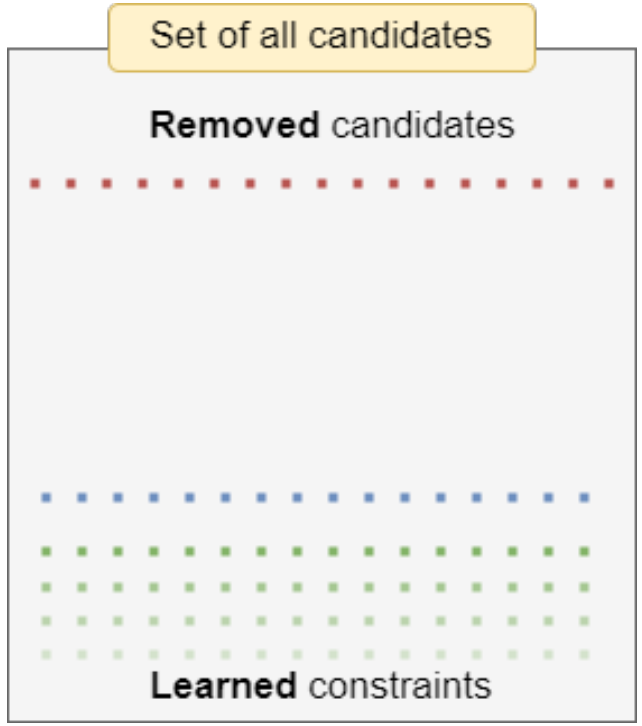
↳ Eliminating candidates



- Learning from ***negative*** examples ("Non-solution"):

- One (or more) violated constraint is a constraint of the problem
- Otherwise, it would be a solution

↳ Learning Constraints



KU LEUVEN

Interactive Constraint Acquisition

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

KU LEUVEN

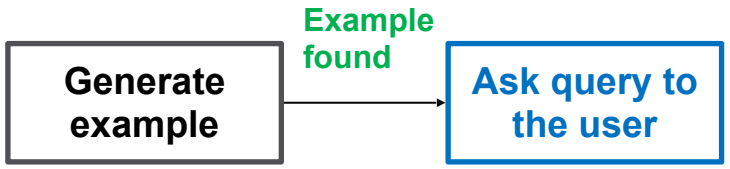
Interactive Constraint Acquisition

Generate example

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

KU LEUVEN

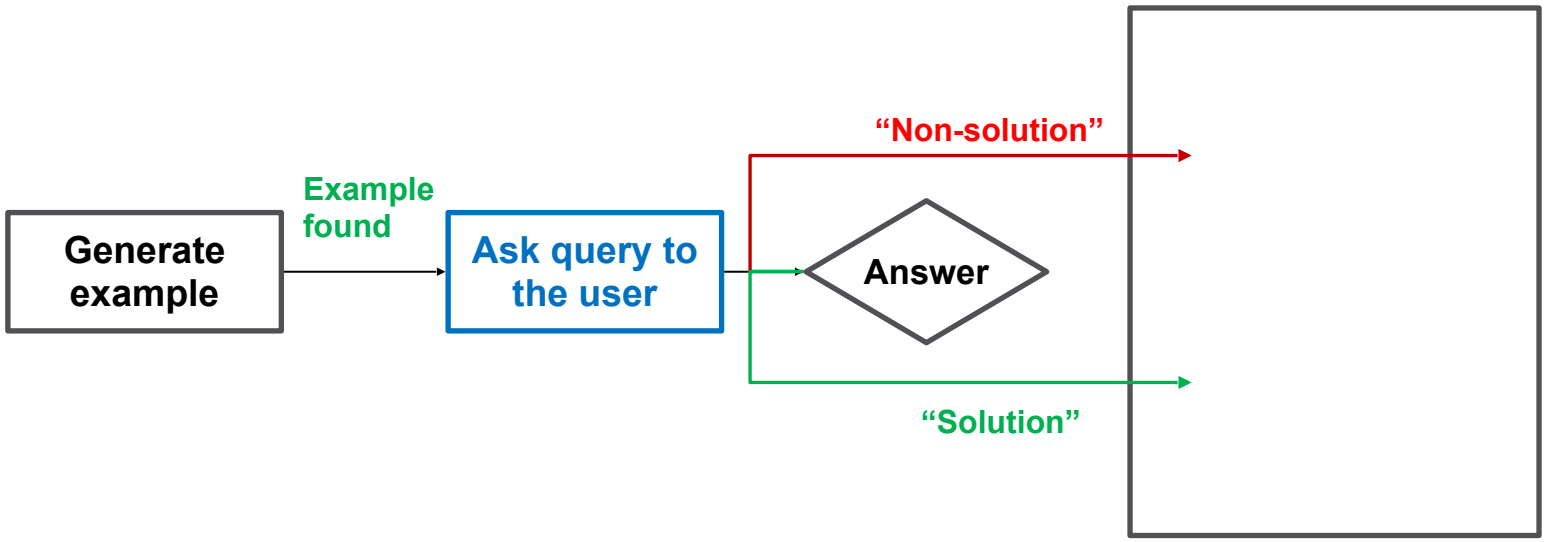
Interactive Constraint Acquisition



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

KU LEUVEN

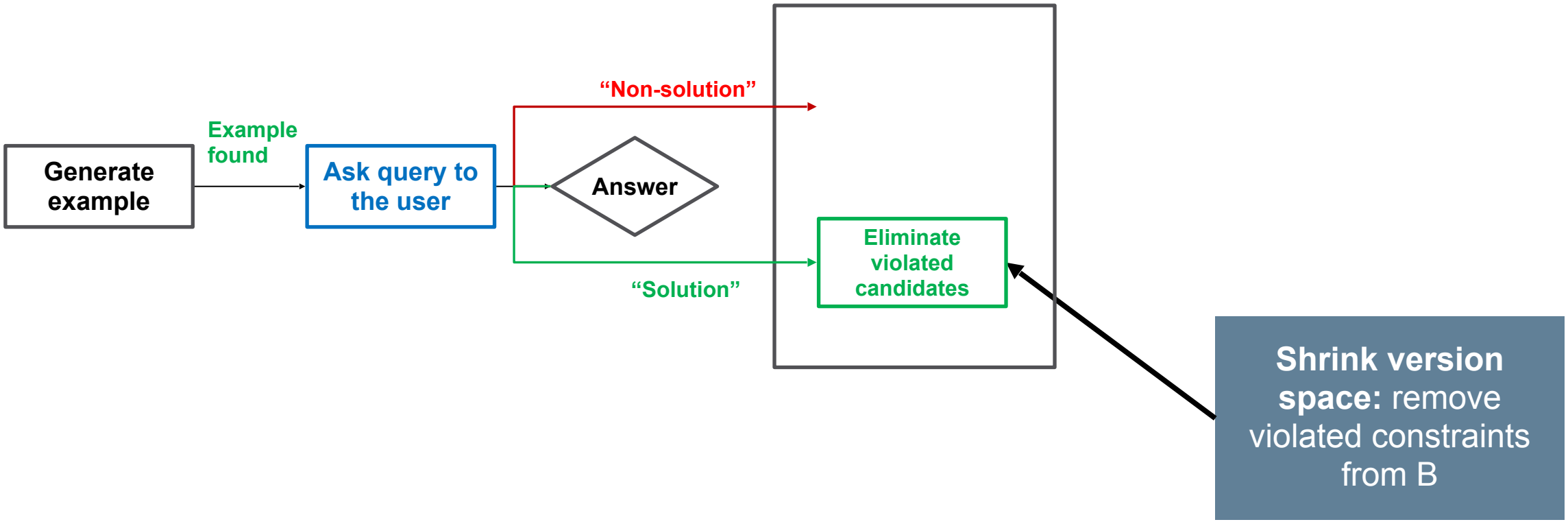
Interactive Constraint Acquisition



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

KU LEUVEN

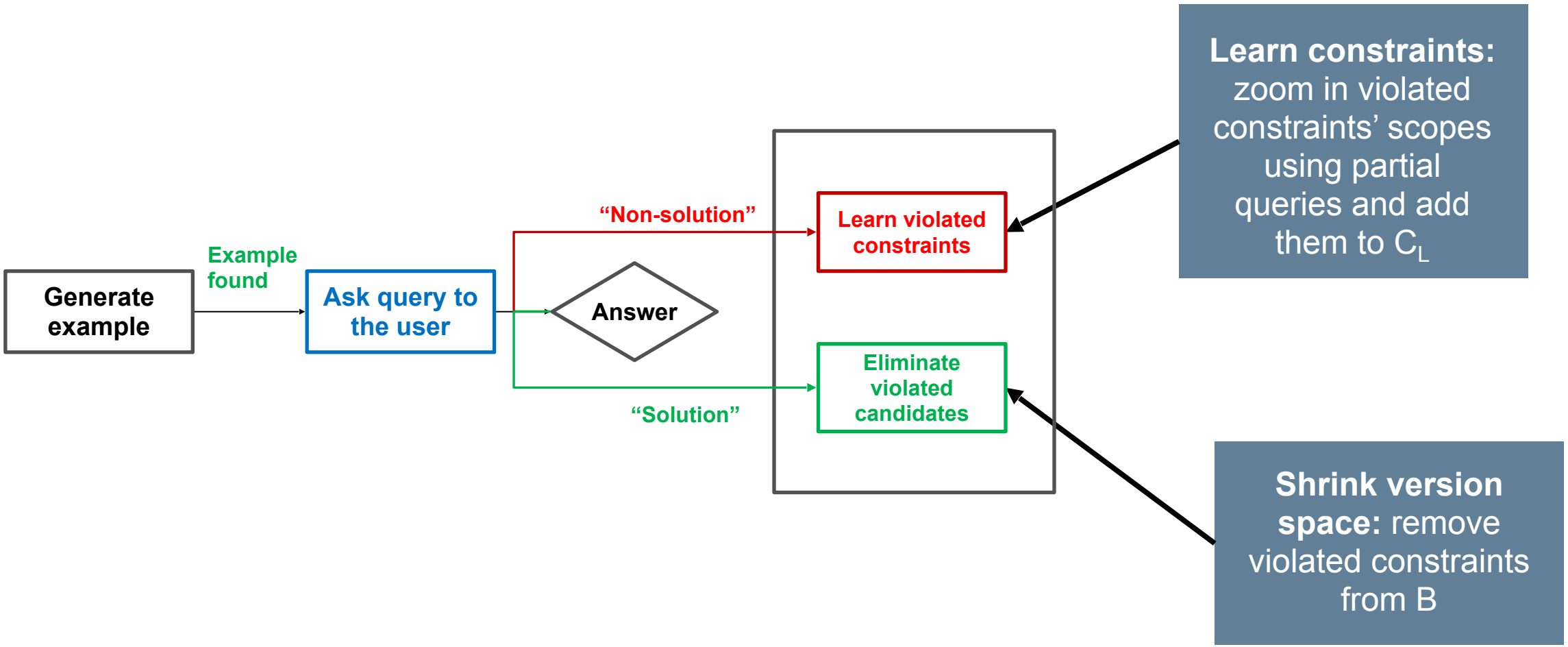
Interactive Constraint Acquisition



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

KU LEUVEN

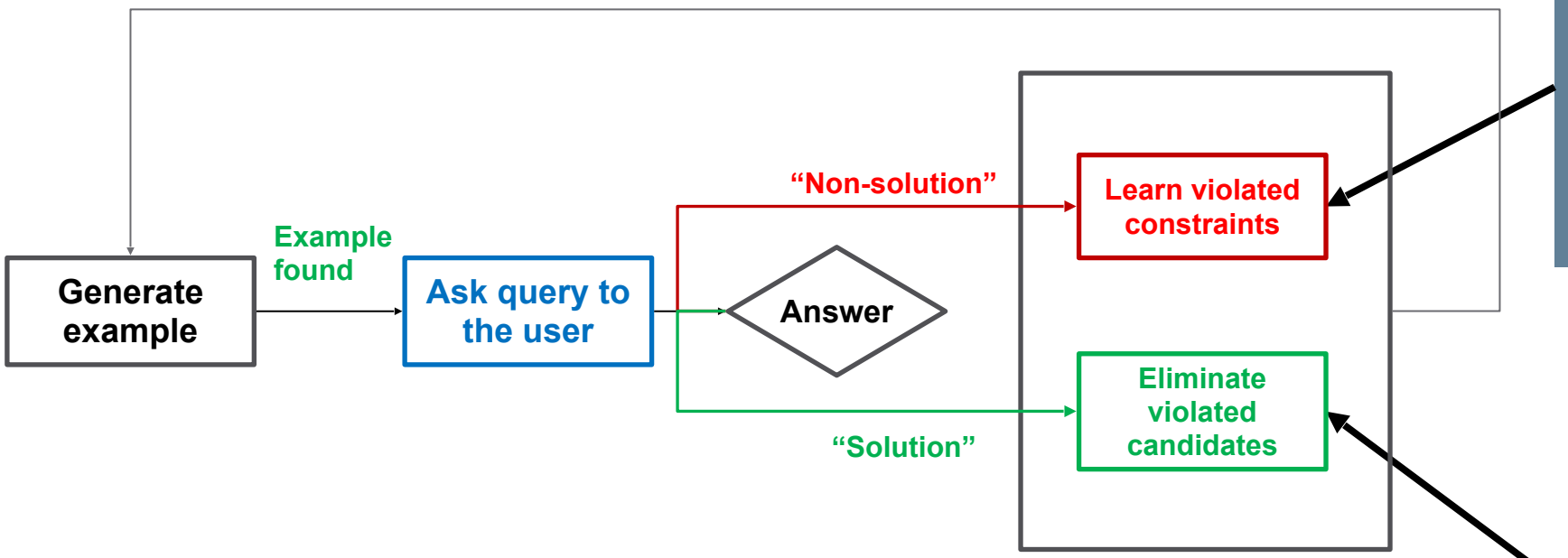
Interactive Constraint Acquisition



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

KU LEUVEN

Interactive Constraint Acquisition



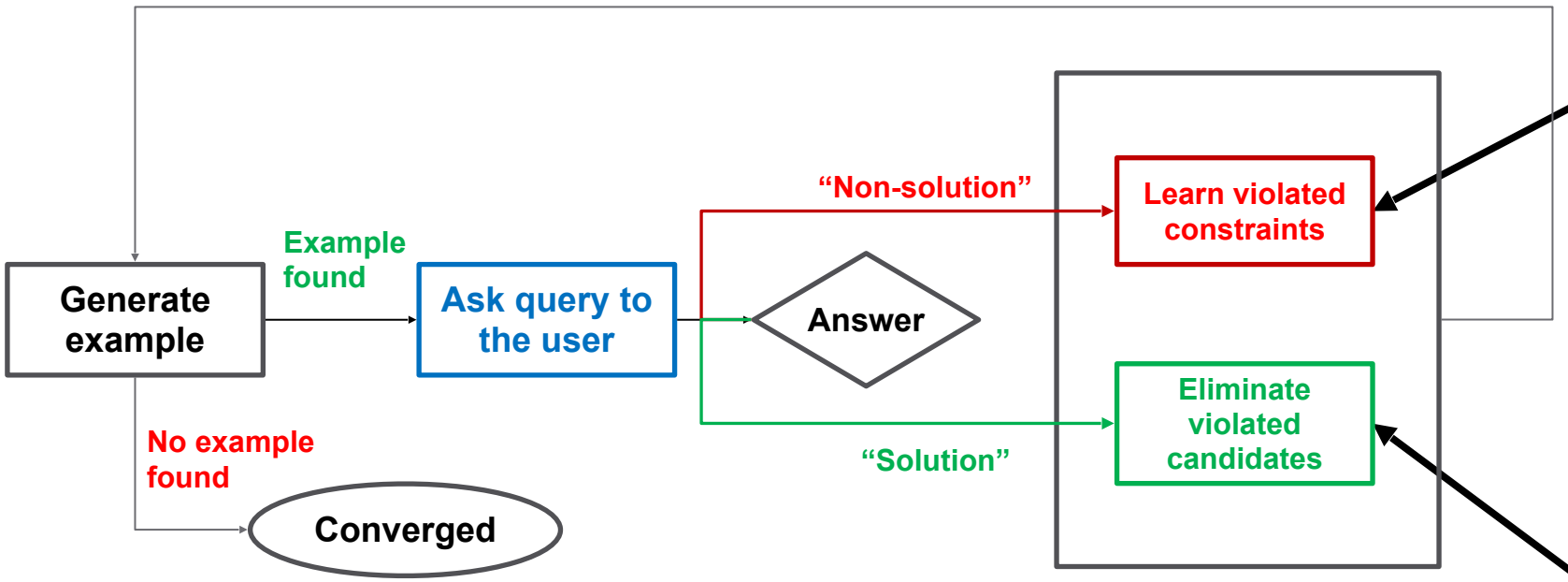
Learn constraints: zoom in violated constraints' scopes using partial queries and add them to C_L

Shrink version space: remove violated constraints from B

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

KU LEUVEN

Interactive Constraint Acquisition



Learn constraints: zoom in violated constraints' scopes using partial queries and add them to C_L

Shrink version space: remove violated constraints from B

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

KU LEUVEN

PyConA

https://github.com/CPMpy/PyConA

pycona 0.3

```
pip install pycona
```

Using the CPMpy modeling library

https://github.com/CPMpy/CPMpy

Hands-on on interactive CA Demos

Multiple Acquisition

```
[ ]  # Multiple Acquisition: includes MquAcq and MQuAcq2
     mqa = MQuAcq()
     learned_instance = mqa.learn(instance, oracle=oracle)

     mqa2 = MQuAcq2()
     learned_instance = mqa2.learn(instance, oracle=oracle)
```

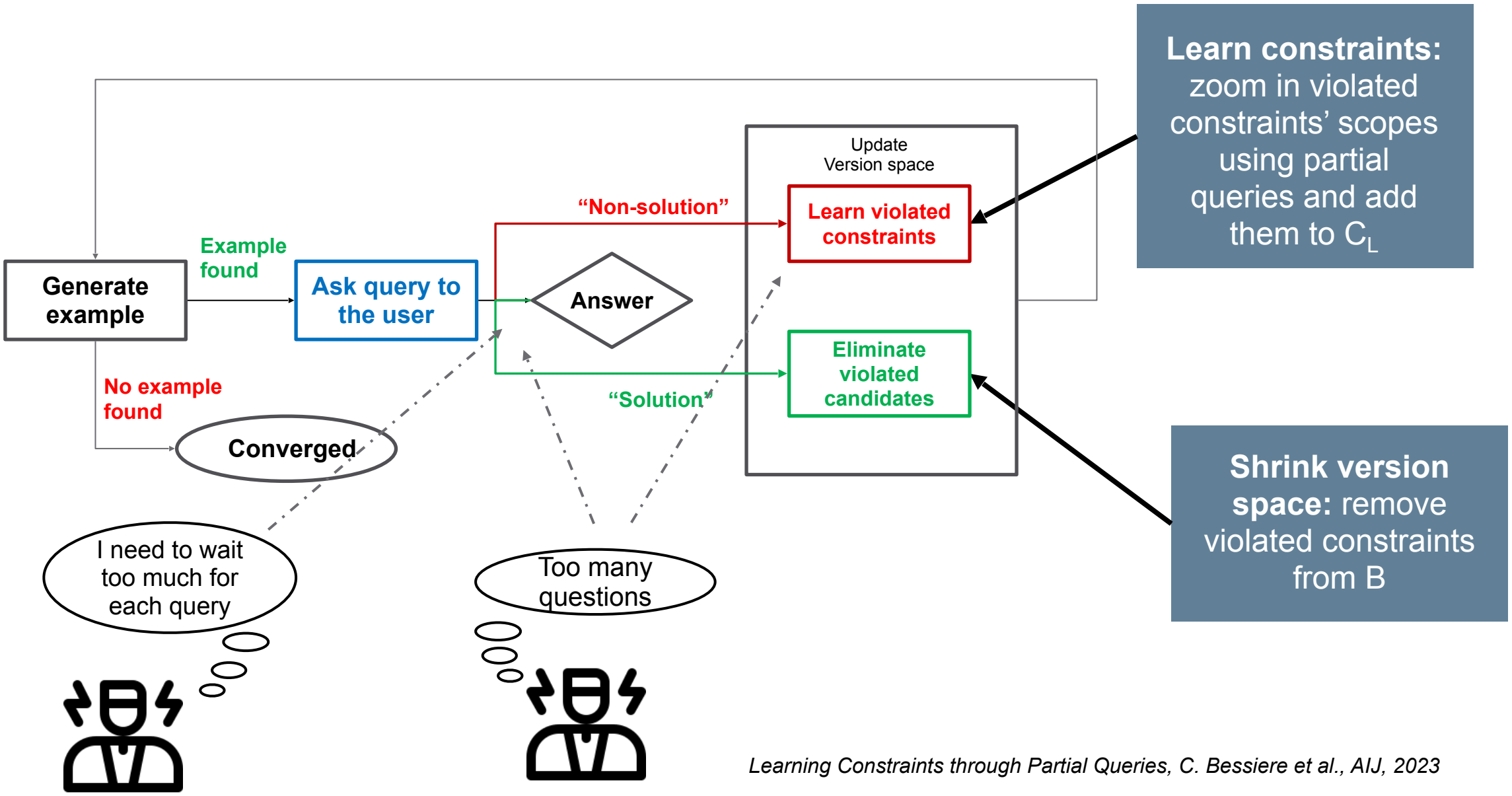
GrowAcq

```
# Creating an environment with the default settings
env = ActiveCAEnv()

# GrowAcq
ga = GrowAcq(env)
learned_instance = ga.learn(instance, oracle=oracle)

# we can compare the statistics:
out = pd.concat([a.env.metrics.short_statistics for a in [qa,mqa,mqa2,ga]])
out.index = ["quacq","mquacq","mquacq2","growacq"]
out
```

	CL	tot_q	top_lvl_q	tfs_q	tfc_q	avg_q_size	avg_gen_time	avg_t	max_t	tot_t	conv
quacq	34	196	38	142	16	6.6939	0.8317	0.4407	1.4866	86.3767	1
mquacq	34	161	63	30	68	2.7950	0.8812	0.0470	1.4936	7.5736	1
mquacq2	34	169	35	102	24	4.7988	0.5819	0.0454	1.2612	7.6738	1
growacq	34	179	64	84	31	4.5978	0.0544	0.0236	0.4465	4.2266	1

KU LEUVEN

Interactive Constraint Acquisition



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

KU LEUVEN

Query generation

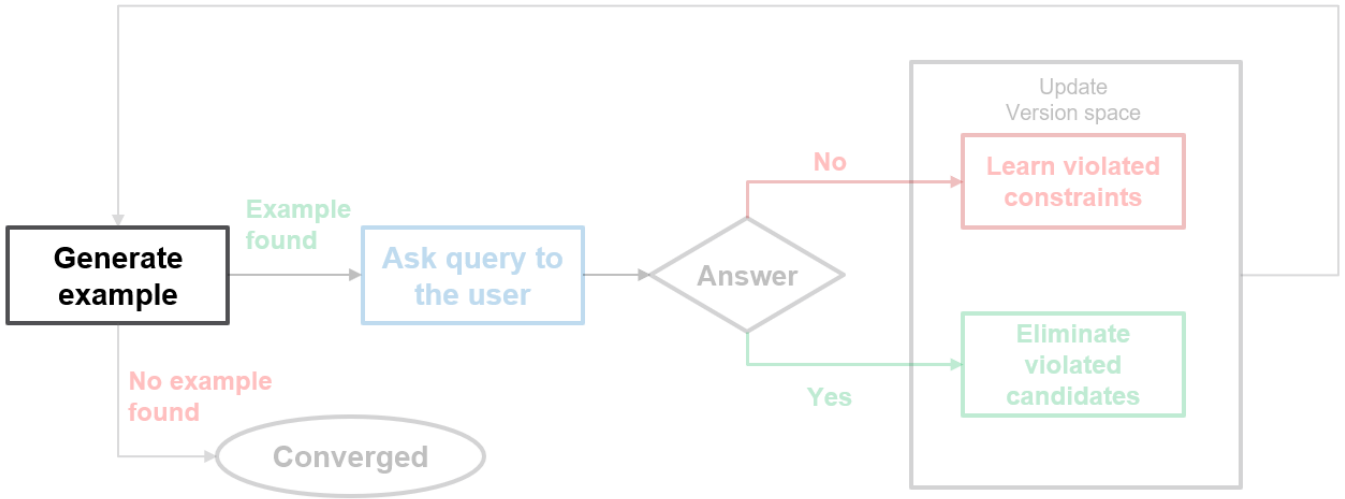
Informative query

- Generate “informative” examples
  - Solve a CSP

Quality of query

- Get the maximum amount of information
  - Solve a COP

Convergence



KU LEUVEN

Guiding Query Generation

Better generated examples lead to faster convergence

KU LEUVEN

Guiding Query Generation

Better generated examples lead to faster convergence

"Solution": Eliminate candidates fast

↓

The **more** candidates we have violated, the faster B will shrink

↓

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

KU LEUVEN

Guiding Query Generation

Better generated examples lead to faster convergence

"Solution": Eliminate candidates fast

The **more** candidates we have violated, the faster B will shrink

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

-"Non-solution": Find constraint(s) fast

The **less** candidates we have violated, the fewer additional queries needed to learn a constraint

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

KU LEUVEN

Guiding Query Generation

Better generated examples lead to faster convergence

"Solution": Eliminate candidates fast

The **more** candidates we have violated, the faster B will shrink

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

-"Non-solution": Find constraint(s) fast

The **less** candidates we have violated, the fewer additional queries needed to learn a constraint

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

Opposite objectives, based on the (**future**) answer

KU LEUVEN

Guiding Query Generation

Better generated examples lead to faster convergence

"Solution": Eliminate candidates fast

The **more** candidates we have violated, the faster B will shrink

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

-"Non-solution": Find constraint(s) fast

The **less** candidates we have violated, the fewer additional queries needed to learn a constraint

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

Opposite objectives, based on the (**future**) answer

KU LEUVEN

Guiding Query Generation

Better generated examples lead to faster convergence

"Solution": Eliminate candidates fast

-"Non-solution": Find constraint(s) fast

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

The **more** candidates we have violated, the faster B will shrink

The **less** candidates we have violated, the fewer additional queries needed to learn a constraint

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

Opposite objectives, based on the (**future**) answer

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

KU LEUVEN

Guiding Query Generation

Better generated examples lead to faster convergence

"Solution": Eliminate candidates fast

-"Non-solution": Find constraint(s) fast

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

The **more** candidates we have vi

The **less** candidates we have violated, the

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

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

Opposite objectives, based on the (**future**) answer

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

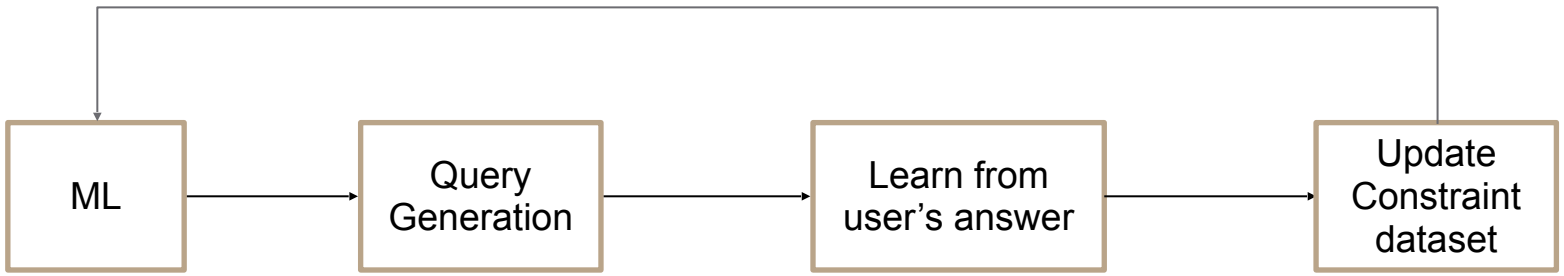
KU LEUVEN

How to guide query generation?

Use a classidier $O(c)$ to predict if a candidate is a constraint of the problem or not

It is a prediction problem

Use Machine Learning!!

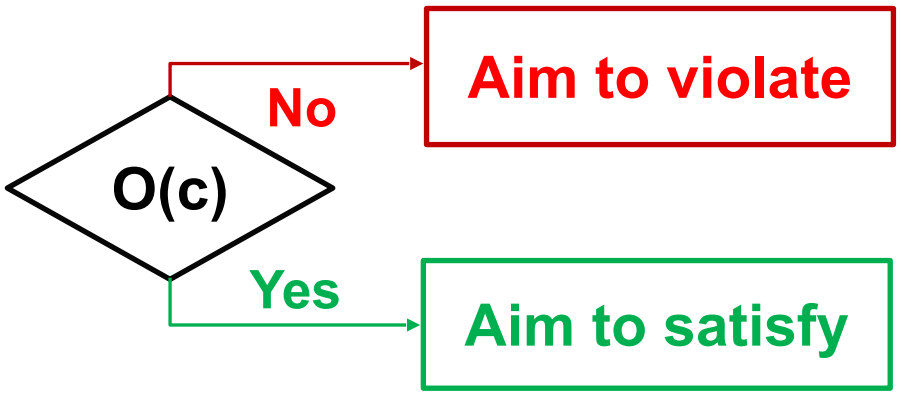


Learning to Learn in Interactive Constraint Acquisition, D. Tsouros et al., AAAI, 2024

KU LEUVEN

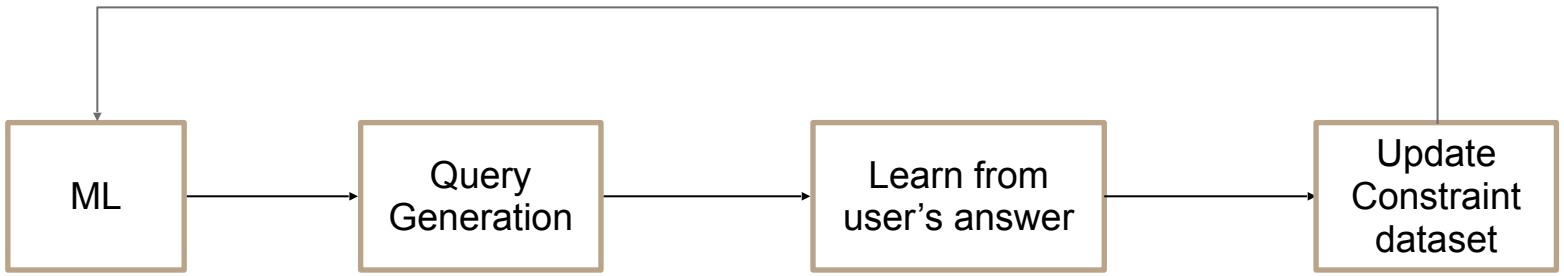
How to guide query generation?

Use a classidier $O(c)$ to predict if a candidate is a constraint of the problem or not



It is a prediction problem

Use Machine Learning!!

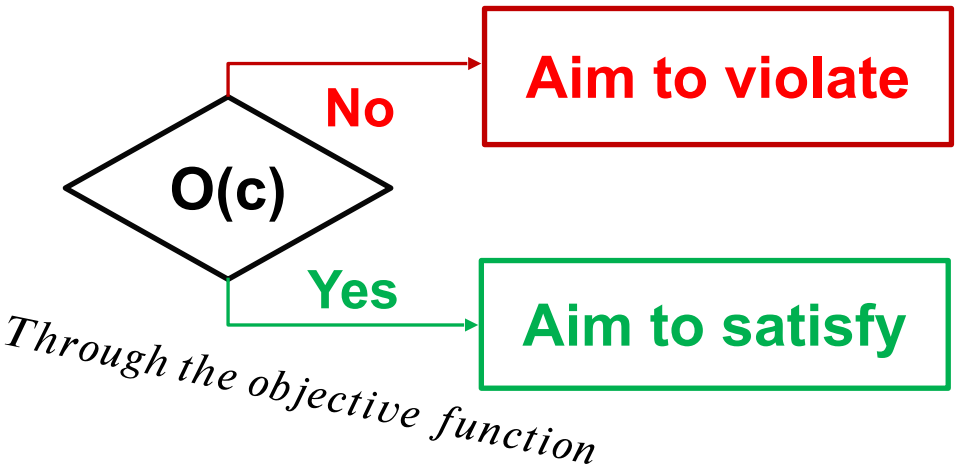


Learning to Learn in Interactive Constraint Acquisition, D. Tsouros et al., AAAI, 2024

KU LEUVEN

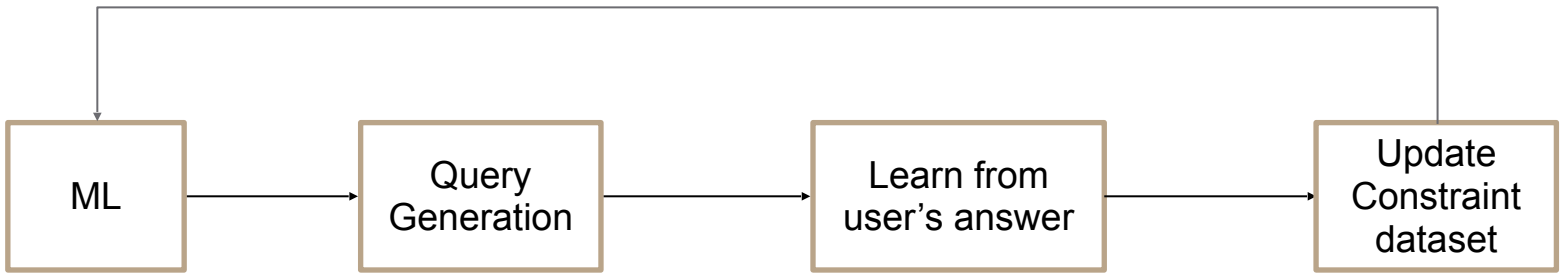
How to guide query generation?

Use a classidier $O(c)$ to predict if a candidate is a constraint of the problem or not



It is a prediction problem

Use Machine Learning!!

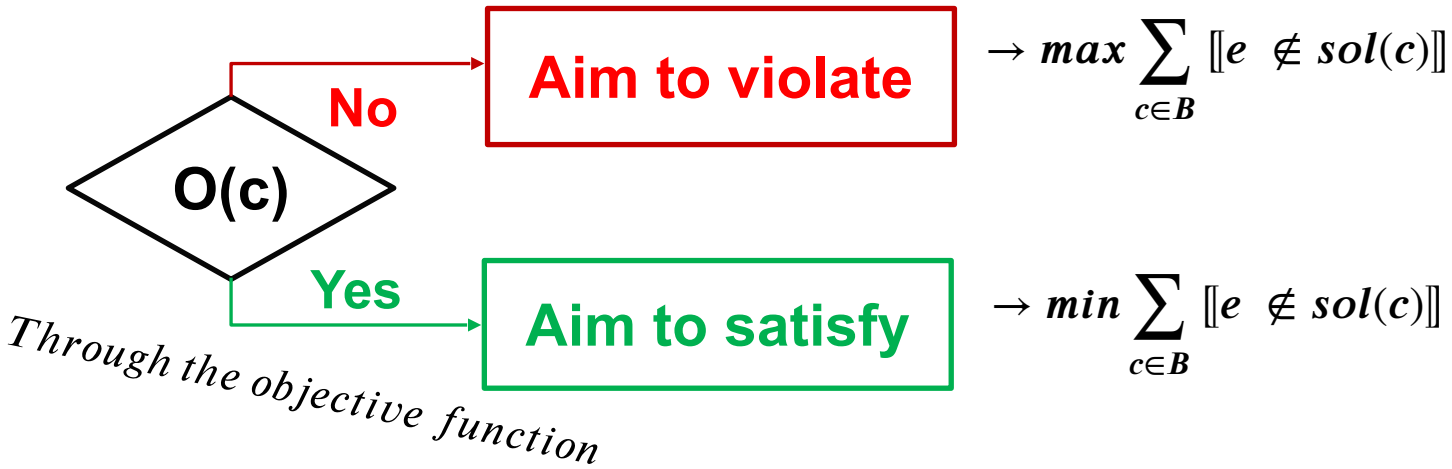


Learning to Learn in Interactive Constraint Acquisition, D. Tsouros et al., AAAI, 2024

KU LEUVEN

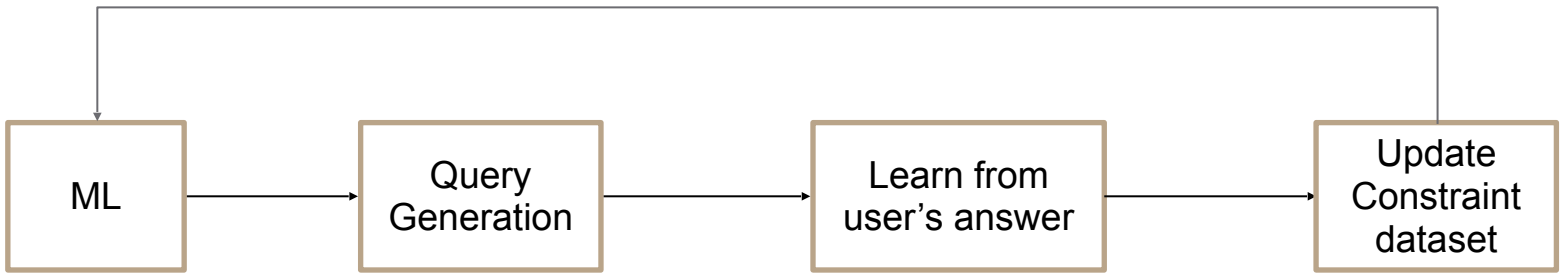
How to guide query generation?

Use a classidier $O(c)$ to predict if a candidate is a constraint of the problem or not



It is a prediction problem

Use Machine Learning!!

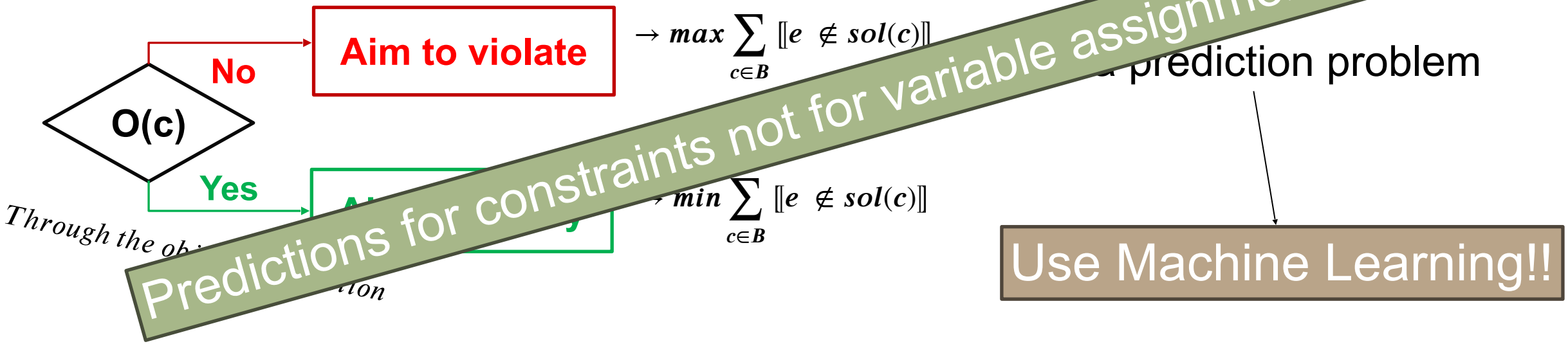


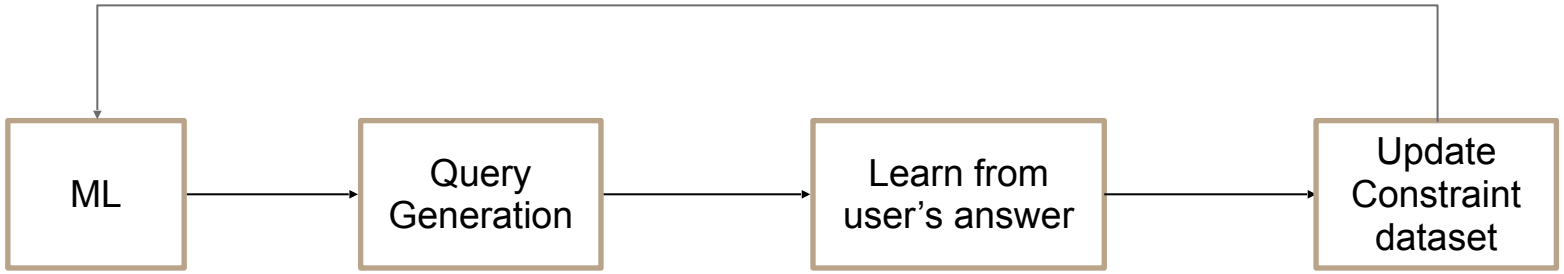
Learning to Learn in Interactive Constraint Acquisition, D. Tsouros et al., AAAI, 2024

KU LEUVEN

How to guide query generation?

Use a classidier $O(c)$ to predict if a candidate is a constraint of the problem or not





Learning to Learn in Interactive Constraint Acquisition, D. Tsouros et al., AAAI, 2024

KU LEUVEN

Using Machine Learning for the prediction

Dataset: Constraint features and class (True or False)
-Constructing during the acquisition process
-Constraints that we know are part of the problem or not
-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

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)	…

KU LEUVEN

Using Machine Learning for the prediction

Dataset: Constraint features and class (True or False)
-Constructing during the acquisition process
-Constraints that we know are part of the problem or not
-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

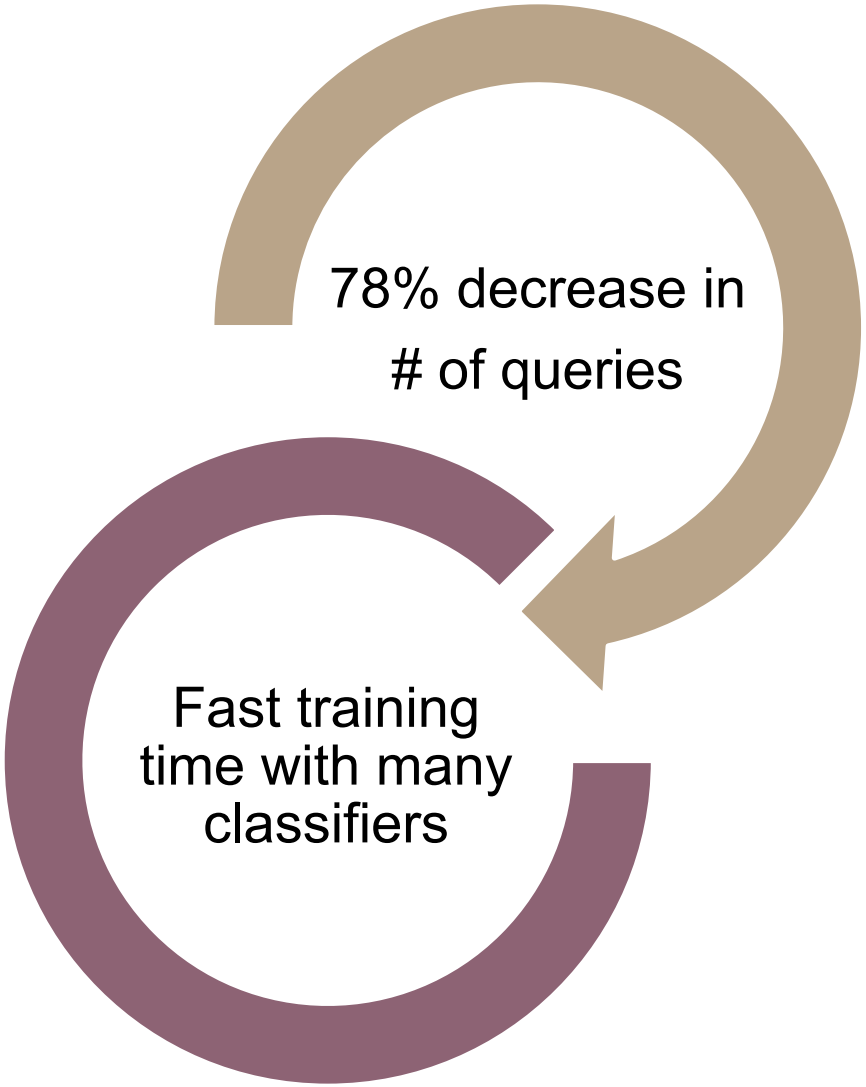
Example for constraint
$x_{1,1} \neq x_{1,2}$ *in Sudoku*

Relation-based features		Scope-based features	
Relation name (string)	≠	Dim[i] same_val (Bool)	True, False
Has constant (Bool)	False	Dim[i] avg (float)	1, 1.5
Constant value (int)	-1	Dim[i] distance (int)	0, 1
Arity (int)	2	…	

For the 2 dimensions

KU LEUVEN

Does using ML to guide queries help?



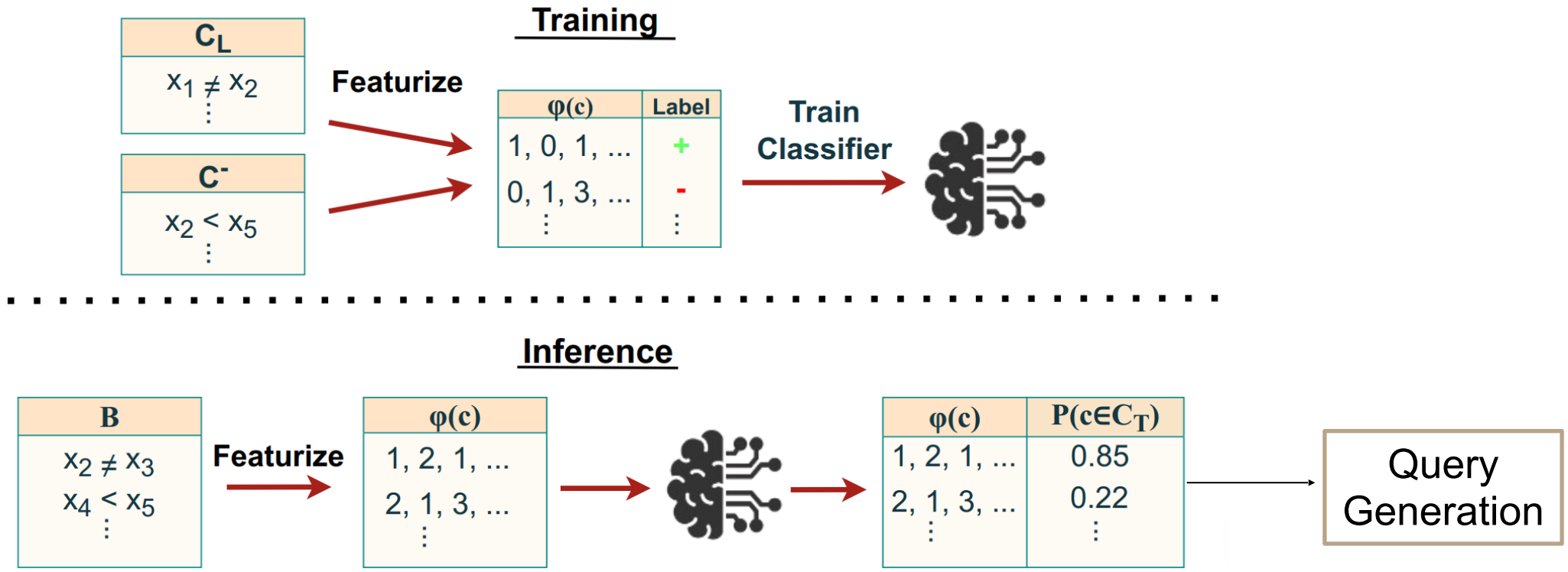
Work in progress

Extract what classifier has learned and ask generalization queries

Learning to Learn in Interactive Constraint Acquisition, D. Tsouros et al., AAAI, 2024

KU LEUVEN

Learning to Learn in Interactive CA



Statistical ML learns the structure implicitly, query-based learning makes it explicit

KU LEUVEN

Major limitation

Assumption of always correct answers!

How to make
interactive constraint acquisition
robust?

KU LEUVEN

Motivation

New active CA methods require fewer and fewer queries

However, a single false query answer can lead to an unrecoverable state

To enable adoption, robustness against noise is required

KU LEUVEN

Motivation

New active CA methods require fewer and fewer queries

However, a single false query answer can lead to an unrecoverable state

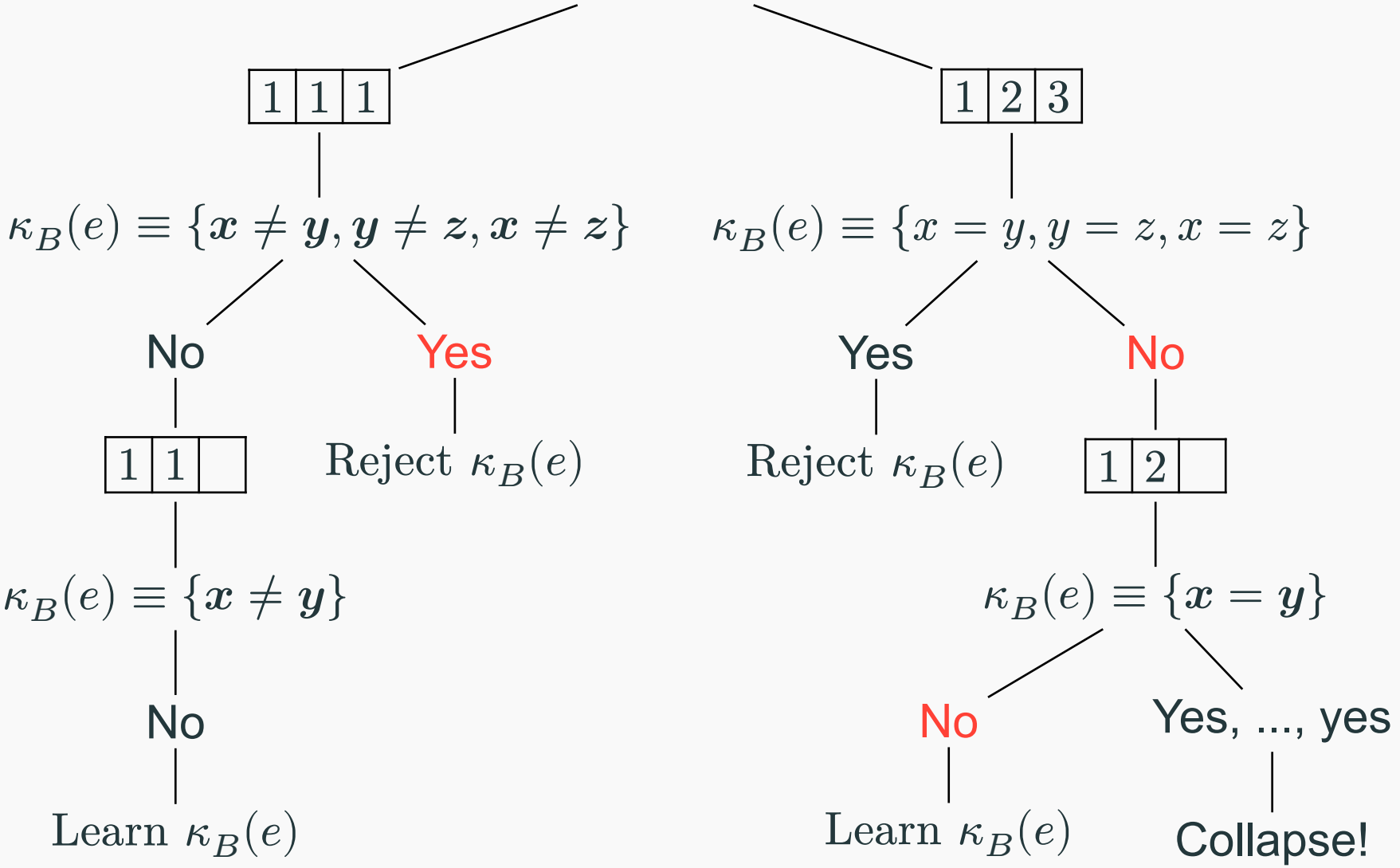
To enable adoption, robustness against noise is required

> “Uncertainty. How do we cope with imprecise, uncertain or noisy information?”
>
> – [E. C. Freuder]

KU LEUVEN

Example: AllDifferent(x, y, z) under noise

$C_T \equiv \{\boldsymbol{x \neq y}, \boldsymbol{y \neq z}, \boldsymbol{x \neq z}\}, B \equiv \{\boldsymbol{x \neq y}, \boldsymbol{y \neq z}, \boldsymbol{x \neq z}, x = y, y = z, x = z\}, C_L \equiv \{\}$



KU LEUVEN

What is robustness against noise?

Robust Active CA: acquire a C_L with a high probability of $C_L \leftrightarrow C_T$ under noise

Types of noise	**Example**
Uniform random	3% error rate
Based on query complexity	$\propto$ number of constraints the user needs to check (similar to [D. Angluin and D. K. Slonim])
Limited queries	Answering “I don’t know”

Robustness will come at a cost of extra queries and wait time

KU LEUVEN

Related work

In concept learning, there are robust methods using both membership **and** equivalence queries [L. Bisht, N. H. Bshouty, and L. Khoury]

In passive CA, the set of examples is fixed. Some passive CA methods are robust due to their statistical nature:

- `BayesAcq` [S. Prestwich]: Mimics a Naive Bayes classifier with a linear constraint
- `SeqAcq` [S. Prestwich]: All violated constraints of a negative example receive positive evidence (no sub-queries!)

KU LEUVEN

Robust active CA

Traditional active CA permanently removes constraints from B after one query

Instead, our idea is to store for each constraint c..

$$\mathrm{lrn}(c) \equiv \text{How many times have we learned } c?$$
$$\mathrm{rej}(c) \equiv \text{How many times have we rejected } c?$$

..then use $\mathrm{lrn}(c), \mathrm{rej}(c)$ to extract B and C_L from the initial bias B_0

KU LEUVEN

Extracting B and C_L

One measure of how "learned" a constraint c is:

$$\rho(c) \equiv \mathrm{lrn}(c) - \mathrm{rej}(c)$$

Extract B, C_L from initial bias B_0 by setting a bias interval $R \equiv [l..u]$:

$$c \in B_0 \text{ is } \begin{cases} \text{candidate } (c \in B) & \text{if } l \leq \rho(c) \leq u \\ \text{learned } (c \in C_L) & \text{if } \rho(c) > u \\ \text{rejected} & \text{if } \rho(c) < l \end{cases}$$

In other words:

$$B \equiv \{c : c \in B_0, l \leq \rho(c) \leq u\}, C_L \equiv \{c : c \in B_0, \rho(c) > u\}$$

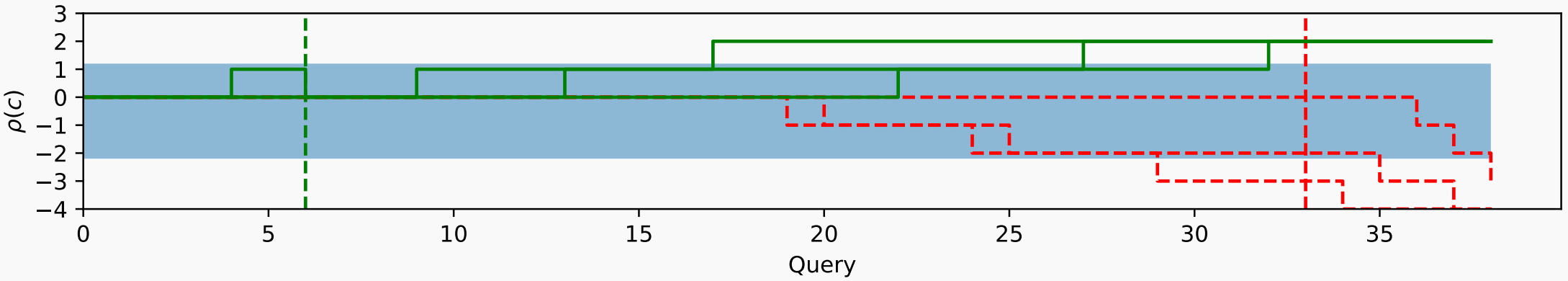
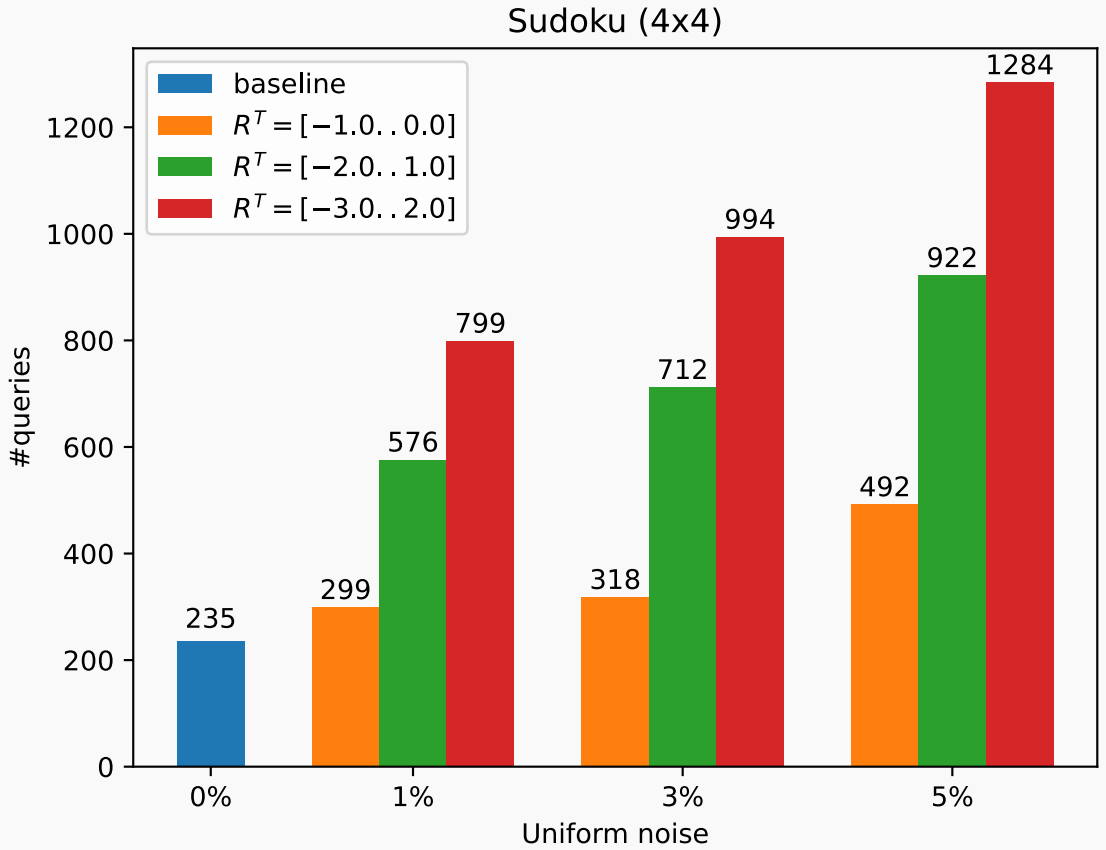


Figure 1: Learning $\mathrm{AllDifferent}(x, y, z)$ with one false positive and one false negative

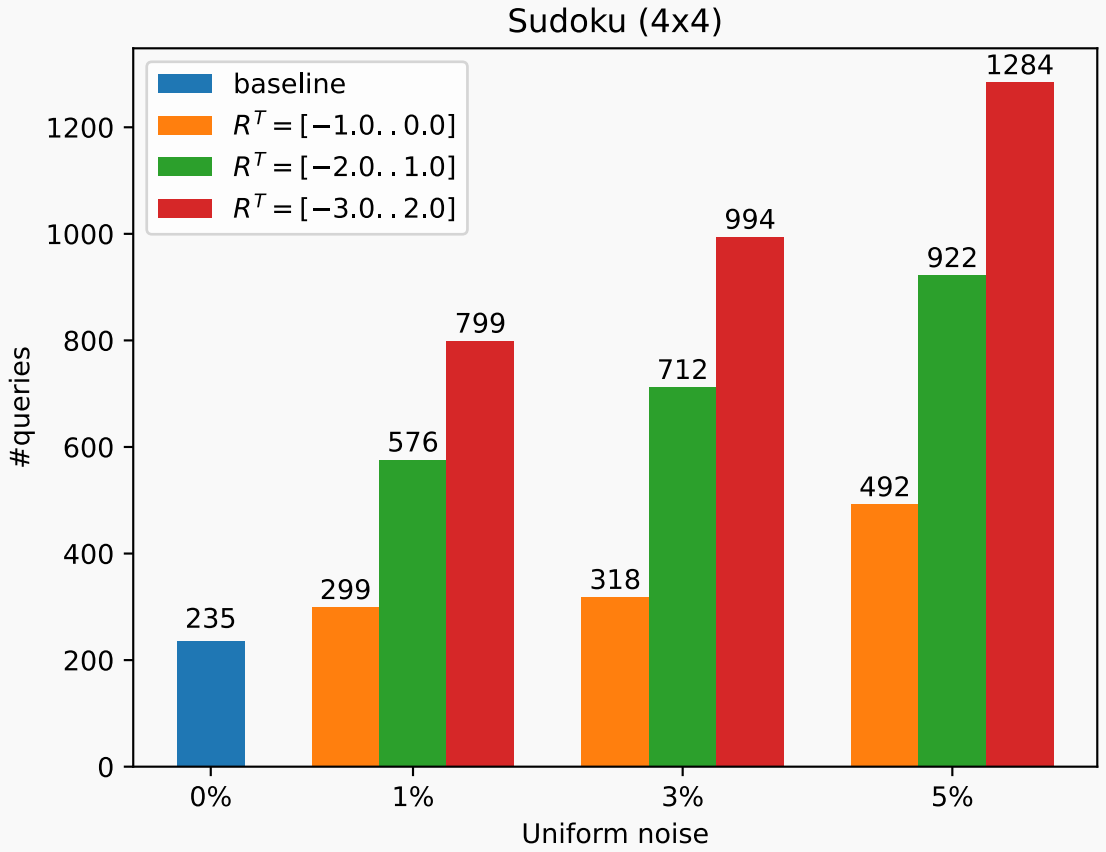
KU LEUVEN

Small benchmark results



KU LEUVEN

Small benchmark results



Unfortunately, the method fails to converge correctly on large benchmarks!

A more probabilistic direction

- Probabilistic interpretation of the evidence
- Accounting for structural evidence from ML
- Take inspiration from statistical learning like `SeqAcq`

Conclusion and discussion

- Active CA is paving a path towards the Holy Grail
  - Queries are (simpler) statements
  - But: noise-free assumption
- Robust active CA is possible but hard to scale
- An inexact setting perhaps requires inexact methods

KU LEUVEN

Bibliography

[1] E. C. Freuder, “In Pursuit of the Holy Grail,” *ACM Comput. Surv.*, vol. 28, no. 4es, p. 63, 1996, doi: 10.1145/242224.242304.

[2] D. Angluin and D. K. Slonim, “Randomly fallible teachers: Learning monotone DNF with an incomplete membership oracle,” *Machine Learning*, vol. 14, no. 1, pp. 7–26, Jan. 1994, doi: 10.1007/BF00993160.

[3] L. Bisht, N. H. Bshouty, and L. Khoury, “Learning with errors in answers to membership queries,” *Journal of Computer and System Sciences*, vol. 74, no. 1, pp. 2–15, 2008, doi: https://doi.org/10.1016/j.jcss.2007.04.010.

[4] S. Prestwich, “Bayesian Constraint Acquisition.”

[5] S. Prestwich, “Robust Constraint Acquisition by Sequential Analysis,” *ECAI 2020*. IOS Press, pp. 355–362, 2020. doi: 10.3233/FAIA200113.