Machine Learning-Based Generalization Queries for Constraint Acquisition

Dimos Tsouros · Senne Berden · Tias Guns

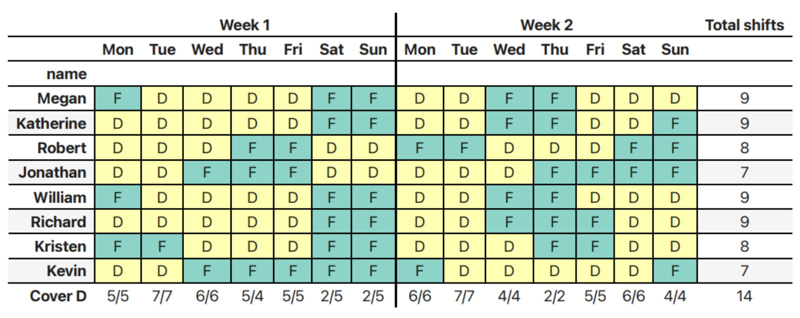
dtsouros@uowm.gr· senne.berden@kuleuven.be · tias.guns@kuleuven.be

UNIVERSITY OF WESTERN MACEDONIA

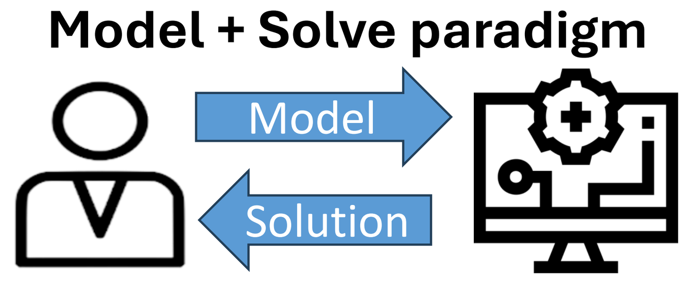
KU LEUVEN

Constraint Solving

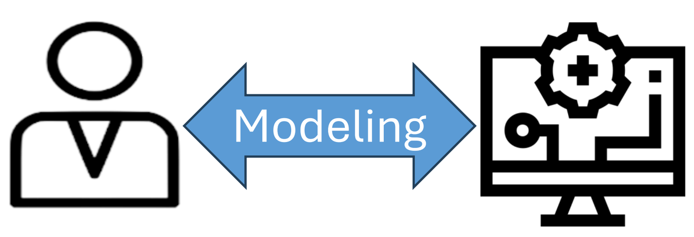
Solving Combinatorial problems with constraints



Model + Solve paradigm



Constraint Acquisition



Interactive Constraint Acquisition

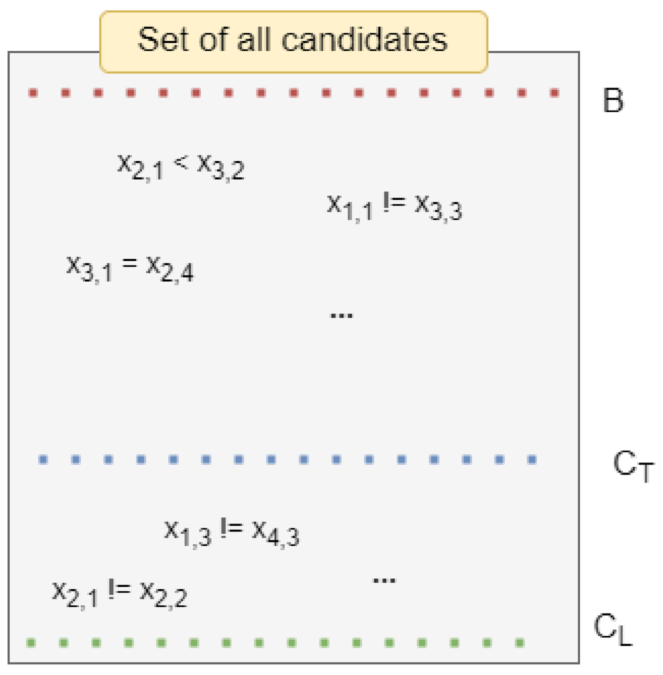
Membership query

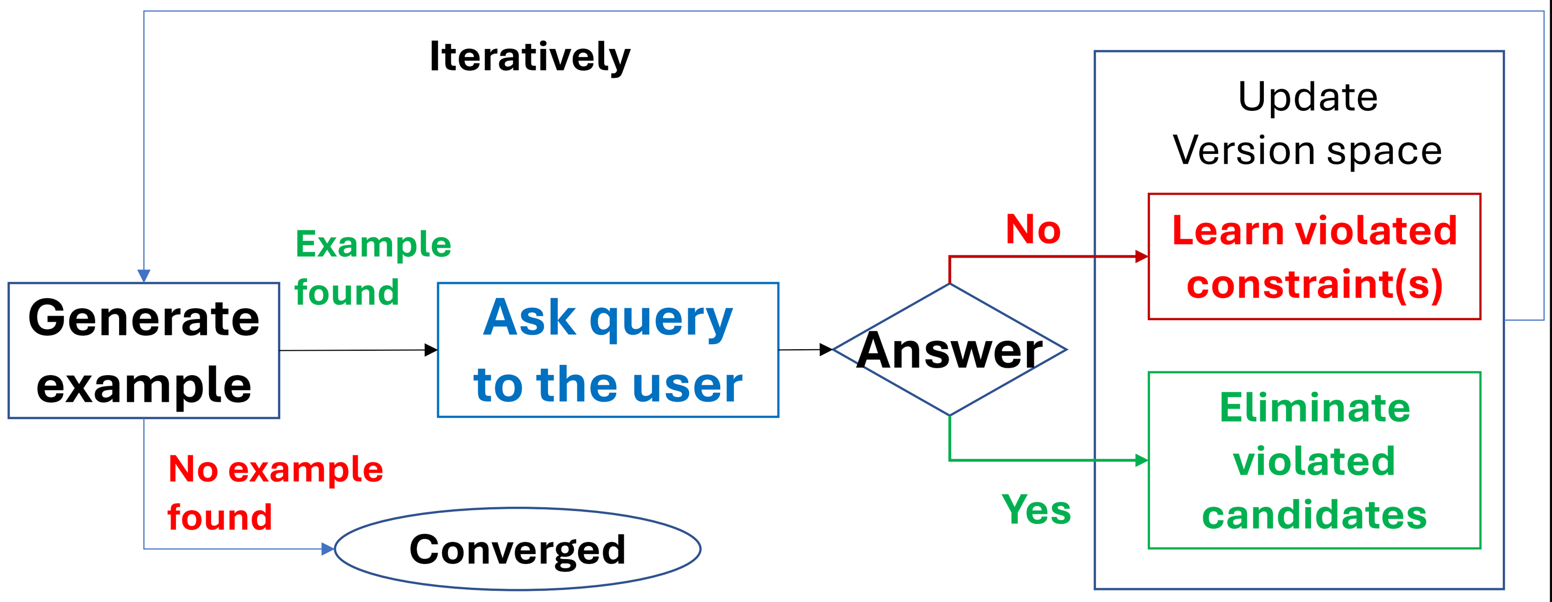
1	1	3	4
3	2	1	1
2	2	3	1
2	3	4	3

Partial query

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

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





Challenge: slow learning

Standard interactive CA learns one ground constraint at a time. Even when many constraints share the same underlying **pattern**.

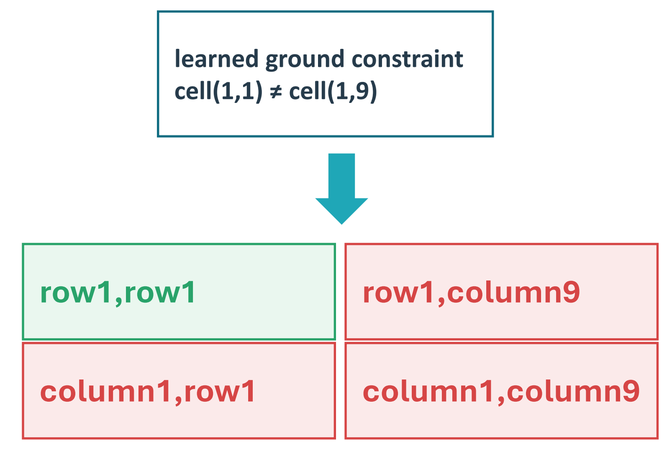
Example from Exam Timetabling

$course[i,j] \neq course[k,l]$ → Broader rule: all courses should be in different slots

Key point: learning each constraint separately repeats work; learning the pattern can **reduce** the number of queries.

Existing type-based generalization

- **Generalization based on user-given variable types (e.g. row, column)**
- **Ask whether a learned ground constraint should hold for every pair of variables with those types.**



Constraint Specification

Capturing
problem-level requirements

- Relation r: What constraint?
- Partitioning function G: How to group the variables?
- Sequence conditions S: Which combinations in each group?

```
Foreach Y ∈ 𝒫(V,T):
  Foreach scope ∈ combinations(Y, arity(r), S):
    c ← (rel(c) = r, var(c) = scope)
```

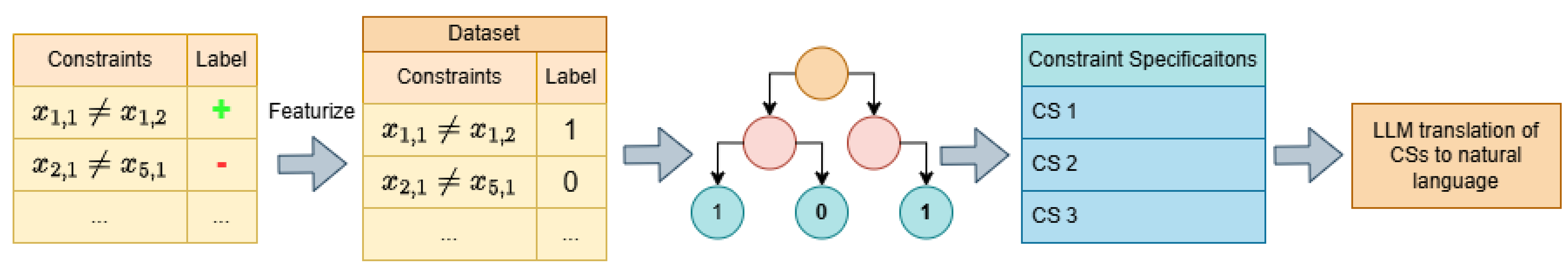
Extract CSs using ML pattern recognition

Feature Representation of Constraints

Feature group	Example features
Relation	relation name, constants, arity
Partitioning	same row / column / latent dims
Conditioning	index avg, min/max, distances
Label	constraint is in target C_T or not

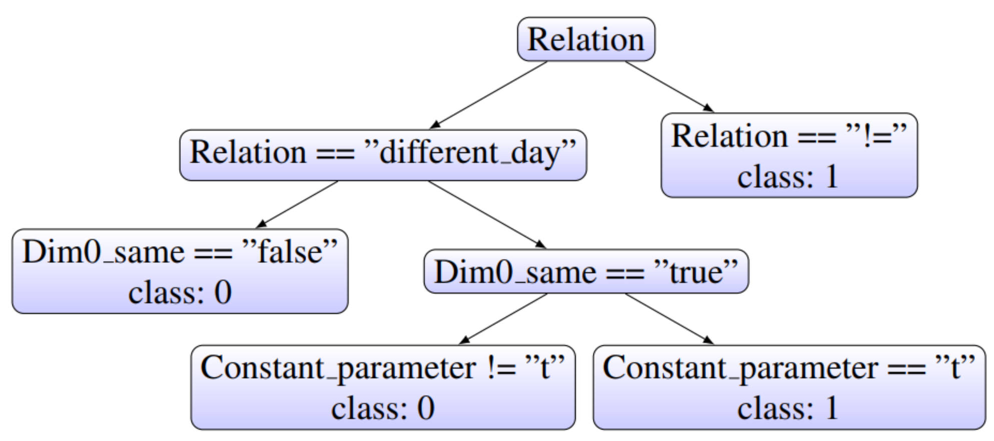
Feature groups: building blocks for extracting CSs from *positive* decision rules.

1. Build dataset of constraints during acquisition
2. Extract Constraint Specifications to query the user



ML-based generalization queries to CSs – Example on Exam Timetabling

1. Decision Tree learned



2. Decision Rules

```
r1: Relation == "different_day"
& Dim0_same == "false"
then 0
r2: Relation == "different_day"
& Dim0_same == "true"
& Constant_parameter != "t"
then 0
r3: Relation == "different_day"
& Dim0_same == "true"
& Constant_parameter == "t"
then 1
r4: Relation == "!=" then 1
```

3. Extracted Constraint Specifications

```
Foreach row ∈ all_rows:
  Foreach scope ∈ all_pairs(row):
    c ← (rel(c) = "different_day(t)",
                   var(c) = scope)
```

```
Foreach scope ∈ all_pairs(V):
  c ← (rel(c) = "!=", var(c) = scope)
```

4. Natural language queries

Within each semester, should every pair of exams be scheduled on different days?

In the entire timetable, should every pair of exams be assigned to different slots?

Evaluation

Benchmarks:

Fully structured

- Latin
- Sudoku
- Exam
- Nurse
- Grid

Partially structured

- Purdey
- Murder
- Zebra
- Greater-than Sudoku.

Decision trees were used for interpretable rule extraction.

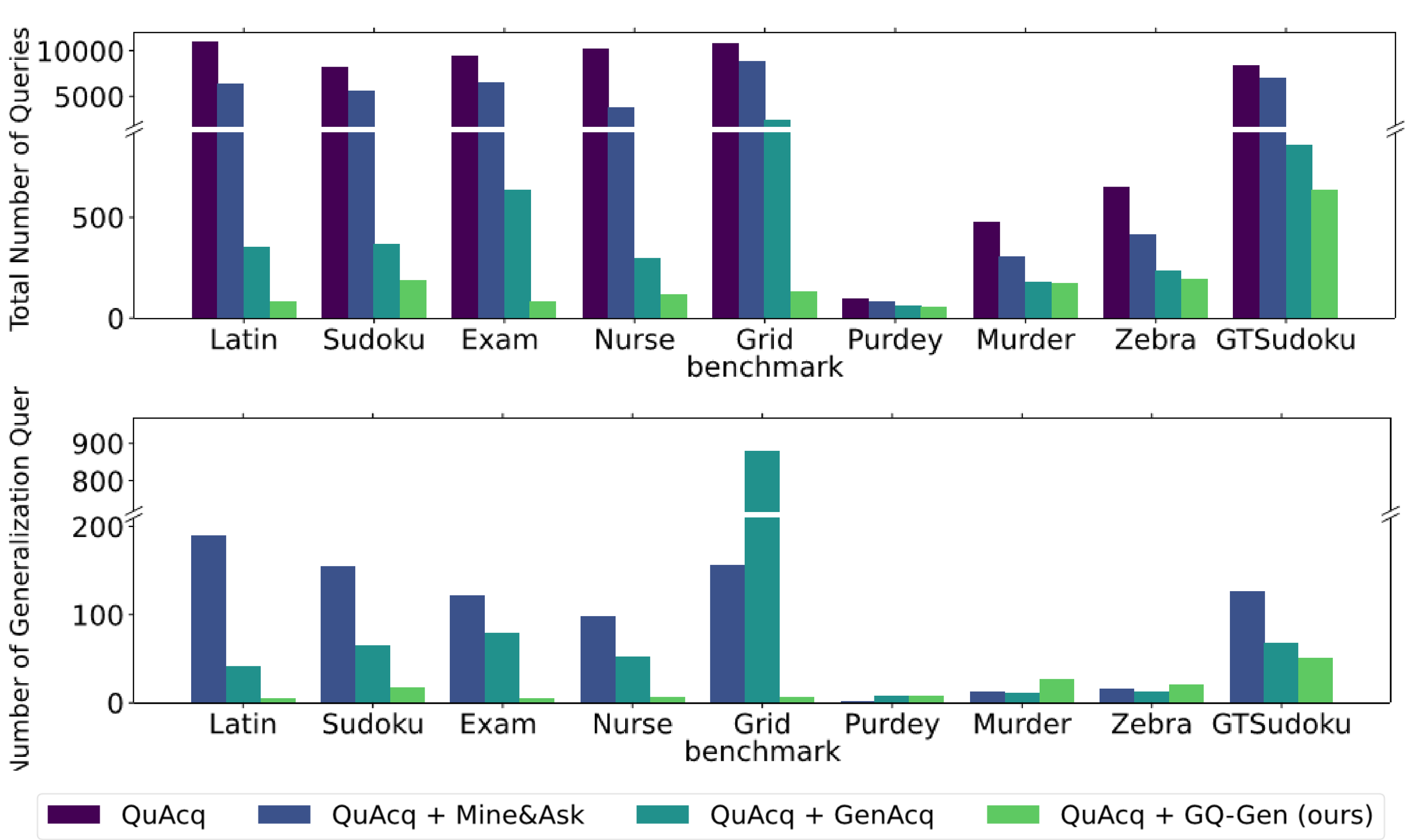
Compared against

1. QuAcq (No generalization)
2. GenAcq (type-based generalization, given types)
3. Mine&Ask (detecting types for generalization)

Results summary

Up to 95% fewer queries vs GenAcq

Up to 99% fewer than QuAcq at scale



User Study

Natural-language formulations of CS-based queries were evaluated by 19 participants. Results indicate that users can understand and answer the generated queries quickly and accurately.

79%	89%	69%	70%
clarity score 4–5	correct answers	high confidence	<30s response

Conclusions

Pattern recognition with ML moves from guiding learning to being the **main learning component.**

CS-based generalization can make interactive CA substantially more query-efficient and more human-friendly.

This research received funding from:

- the framework of H.F.R.I call "4th Call for H.F.R.I.'s Research Projects to Support Postdoctoral Researchers" (H.F.R.I. Project Number: 28553).
- the European Research Council (ERC) under the EU Horizon 2020 research and innovation programme (Grant No.101002802, CHAT-Opt);
- the Research Foundation-Flanders (FWO) (Grant No. 11PQ024N).