
IJCAI 2023

# Learning Constraint Networks over Unknown Constraint Languages

Christian Bessiere | Clément Carbonnel | Areski Himeur

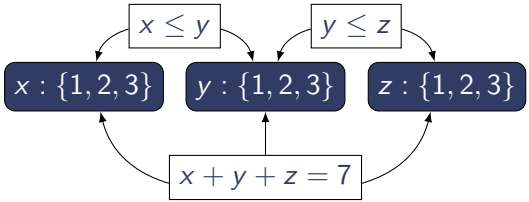
University of Montpellier, CNRS, LIRMM, Montpellier, France
{bessiere, clement.carbonnel, areski.himeur}@lirmm.fr


Work supported by EU Horizon 2020 TAILOR (GA N° 952215),
ANITI (GA N° ANR-19-PI3A-0004) and ANR AXIAUM (GA N° ANR-20-THIA-0005-01)

# Background

A **constraint network**:

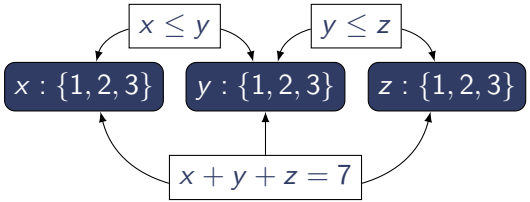
- a set of **variables** over a finite **domain**
- a set of **constraints**, i.e. relations between variables that must be satisfied in any **solution**



# Background

A **constraint network**:

- a set of **variables** over a finite **domain**
- a set of **constraints**, i.e. relations between variables that must be satisfied in any **solution**




A **constraint language** is a set of relations over a domain.

- Language of a constraint network : set of all relations that appear in its constraints

# Constraint Programming & Constraint Acquisition

Constraint programming is an **expressive**, **flexible**, **efficient** paradigm for solving problems.

# Constraint Programming & Constraint Acquisition

Constraint programming is an **expressive**, **flexible**, **efficient** paradigm for solving problems.

**Challenge** | Designing a constraint network representing a given problem can be difficult.

- To overcome this, **constraint acquisition** learns a constraint network automatically.

## **Definition** | Passive Constraint Acquisition

**Instance:** Set of examples, labelled as solutions and non-solutions.

**Goal:** Find a constraint network that correctly classifies the examples.

## **Definition** | Passive Constraint Acquisition

**Instance:** Set of examples, labelled as solutions and non-solutions.

**Goal:** Find a constraint network that correctly classifies the examples.


ConAcq.1 (Bessiere et al., 2006, 2017), ModelSeeker (Beldiceanu and Simonis, 2012), BayesAcq (Prestwich et al., 2021), Count-CP (Kumar et al., 2022)

**Problem** | All current approaches require some knowledge of the constraint language of the output network.

# Our contribution

We develop a constraint acquisition method that
**constructs a constraint language as part of the learning process**.

In general, given a set of examples, a large number of constraint languages can be used.

In general, given a set of examples, a large number of constraint languages can be used.

**Problem** | Some languages are clearly unsatisfactory from a practical point of view

In general, given a set of examples, a large number of constraint languages can be used.

**Problem** | Some languages are clearly unsatisfactory from a practical point of view

- Examples:
  - $(x = 1, y = 2, z = 3)$ is a solution.
  - $(x = 3, y = 2, z = 1)$ and $(x = 1, y = 3, z = 2)$ are non-solutions.

In general, given a set of examples, a large number of constraint languages can be used.

**Problem** | Some languages are clearly unsatisfactory from a practical point of view

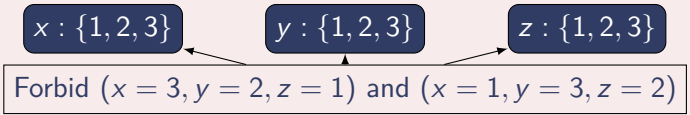
- Examples:
  - $(x = 1, y = 2, z = 3)$ is a solution.
  - $(x = 3, y = 2, z = 1)$ and $(x = 1, y = 3, z = 2)$ are non-solutions.
- A good network: 

In general, given a set of examples, a large number of constraint languages can be used.

**Problem** | Some languages are clearly unsatisfactory from a practical point of view

- Examples:
  - $(x = 1, y = 2, z = 3)$ is a solution.
  - $(x = 3, y = 2, z = 1)$ and $(x = 1, y = 3, z = 2)$ are non-solutions.
- A good network:


- A bad network:



**Intuition** | The best constraint language is the simplest.

First approximation: the **smallest** language in terms of arity and number of relations.

**Intuition** | The best constraint language is the simplest.

First approximation: the **smallest** language in terms of arity and number of relations.

**Problem** | Language-Free Acq

**Instance:** Set of examples, labelled as solutions and non-solutions, two integers $k$ and $r$.

**Question:** Is there a constraint network **over a language of size at most** $k$ **and arity at most** $r$ that correctly classifies the examples?

**Intuition** | The best constraint language is the simplest.

First approximation: the **smallest** language in terms of arity and number of relations.

**Problem** | Language-Free Acq

**Instance:** Set of examples, labelled as solutions and non-solutions, two integers $k$ and $r$.

**Question:** Is there a constraint network **over a language of size at most $k$ and arity at most $r$** that correctly classifies the examples?

Language-Free Acq is NP-complete even for $(k, r) = (1, 1)$.

**Method** | Compute a constraint network with minimum $(k, r)$ that correctly classifies the examples.

- **Strategy:** minimize $k + r^2$

**Method** | Compute a constraint network with minimum $(k, r)$ that correctly classifies the examples.

- **Strategy:** minimize $k + r^2$
- **Tie-breaking:** lower arity, more constraints, tighter constraints

**Method** | Compute a constraint network with minimum $(k, r)$ that correctly classifies the examples.

- **Strategy:** minimize $k + r^2$
- **Tie-breaking:** lower arity, more constraints, tighter constraints
- **Construct and solve a model for each** $(k, r)$**:**
  - Convert to an instance Weighted Partial Max-Sat
  - Compute the optimal network or prove that none exists
  - Output the first constraint network found

# Experimental Results

- Learn the **target language** and the **target network**:
  - Sudoku [200 examples]
  - Schur's Lemma [800 examples]

# Experimental Results

- Learn the **target language** and the **target network**:
  - Sudoku [200 examples]
  - Schur's Lemma [800 examples]
- Learn the **target language** and an **equivalent network**:
  - Jigsaw Sudoku [200-1400 examples depending on shape]

# Experimental Results

- Learn the **target language** and the **target network**:
  - Sudoku [200 examples]
  - Schur's Lemma [800 examples]
- Learn the **target language** and an **equivalent network**:
  - Jigsaw Sudoku [200-1400 examples depending on shape]
- Learn an **equivalent network** on another language:
  - Subgraph Isomorphism [800 examples]
  - Golomb Ruler (with 10 variables) [3200 examples]

# Experimental Results

- Learn the **target language** and the **target network**:
  - Sudoku [200 examples]
  - Schur's Lemma [800 examples]
- Learn the **target language** and an **equivalent network**:
  - Jigsaw Sudoku [200-1400 examples depending on shape]
- Learn an **equivalent network** on another language:
  - Subgraph Isomorphism [800 examples]
  - Golomb Ruler (with 10 variables) [3200 examples]
- Neither learn the target language nor an equivalent network:
  - 8-Queens (coordinates model)

# Recap and Future Work

We proposed a novel constraint acquisition method that
**does not require any knowledge on the constraint language of the target network**.


Experiments show promising results, but has limitations. Could be addressed by using **more sophisticated notions of simplicity** and **detecting topological information**.

Thank you for your time and attention.

Work supported by EU Horizon 2020 TAILOR (GA N° 952215),
ANITI (GA N° ANR-19-PI3A-0004) and ANR AXIAUM (GA N° ANR-20-THIA-0005-01)
Experiments performed with the MESO@LR-Platform at University of Montpellier

# References


Beldiceanu, N., & Simonis, H. (2012). A model seeker: Extracting global constraint models from positive examples. In M. Milano (Ed.), *Principles and practice of constraint programming - 18th international conference, CP 2012, québec city, qc, canada, october 8-12, 2012. proceedings* (pp. 141–157). Springer. https://doi.org/10.1007/978-3-642-33558-7_13. (Cit. on pp. 6, 7)

Bessiere, C., Coletta, R., Koriche, F., & O'Sullivan, B. (2006). Acquiring constraint networks using a sat-based version space algorithm. *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, 1565–1568. http://www.aaai.org/Library/AAAI/2006/aaai06-251.php

Bessiere, C., Koriche, F., Lazaar, N., & O'Sullivan, B. (2017). Constraint acquisition. *Artif. Intell.*, *244*, 315–342. https://doi.org/10.1016/j.artint.2015.08.001

Kumar, M., Kolb, S., & Guns, T. (2022). Learning constraint programming models from data using generate-and-aggregate. In C. Solnon (Ed.), *28th international conference on principles and practice of constraint programming, CP 2022, july 31 to august 8, 2022, haifa, israel* (29:1–29:16). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.CP.2022.29. (Cit. on pp. 6, 7)

Prestwich, S. D., Freuder, E. C., O'Sullivan, B., & Browne, D. (2021). Classifier-based constraint acquisition. *Ann. Math. Artif. Intell.*, *89*(7), 655–674. https://doi.org/10.1007/s10472-021-09736-4 (cit. on pp. 6, 7)