Motivations

Problem \[\rightarrow\] CSP \[\rightarrow\] CP Solver

\[\rightarrow\] solution
Motivations

• **Question:** How does the user write down the constraints of a problem?
• **Limitations:** modelling constraint networks require a fair expertise
  [Freuder99, Frisch et al.05, Smith06]

• **Need:** Simple way to build constraint model ➔ Modeller-assistant
Motivations

- **Question:** How does the user write down the constraints of a problem?
- **Limitations:** modelling constraint networks require a fair expertise
  
  [Freuder99, Frisch et al.05, Smith06]

- **Need:** Simple way to build constraint model ➔ Modeller-assistant
- **How:** In a Machine Learning way (passive/active, offline/online, by reinforcement...)

Learning process

- Solutions
- Non-solutions

[Diagram showing a flow from Problem to Solution through CSP and learning process]
Motivations

- **Question:** How does the user write down the constraints of a problem?
- **Limitations:** modelling constraint networks require a fair expertise
  
  [Freuder99, Frisch et al.05, Smith06]

- **Need:** Simple way to build constraint model ➔ Modeller-assistant
- **How:** In a Machine Learning way (passive/active, offline/online, by reinforcement...)
Acquisition using standard ML

- **Empirical Model Learning** [Lombardi and Milano, AIJ17]
  - Extracting an Empirical Model using Neural Networks and Decision Trees
  - Empirical Model in terms of variables/constraints

- **Model Agnostic Solution of CSPs via Deep Learning: A preliminary**
  [Galassi et al., CPAIOR18]

- **Study Boundary estimation for constraint optimization problem** [Spieker and Gotlieb, ISMP18]
  - Learning boundaries for objective variables
  - Based on supervised learning (data curation, regression models)
Version Space Learning (Overview) [Mitchell82]

- Let \( X = x_1, \ldots, x_n \) a set of attributes of domains \( D = D_1, \ldots, D_n \)

- A concept is a Boolean function \( f : X \rightarrow \{0, 1\} \)
  - \( f(x_i) = 0 \) => \( x_i \) is a negative instance
  - \( f(x_j) = 1 \) => \( x_j \) is a positive instance

Given a set of hypothesis \( H \), any subset of \( H \) represents a version space

- A concept to learn is the set of positive instances that can be represented by a version space
Version Space Learning (Overview) [Mitchell82]

Most specific concept to learn:

\[ f : (\forall x_i \in E^+ : f(x_i) = 1) \land (\forall x_i \in E^- : f(x_i) = 0) \]

\[ f \equiv h_2 \land h_6 \land h_9 \]
Constraint Acquisition as Version Space Learning

Constraint Programming:

Constraint network
Constraint Acquisition Problem

Inputs:
- (X,D): Vocabulary
- \( \Gamma \): Constraint language
- \( B \): Bias (constraints/hypothesis)
- \( C_T \): Target Network (concept to learn)
- \( (E^+,E^-) \): training set

Output:
- \( C_L \): Learned network such that:
Example

- \( \Gamma = \{<,=\} \)
- \( B = \{x_i < x_j, x_i = x_j, \forall i, j\} \)
- \( C_T = \{x_1 = x_3, x_1 < x_2\} \)
- \( C_L = \{x_1 = x_3, x_3 < x_2\} \)
Constraint Acquisition Problem

- Convergence Problem:
  - $C_L$ agrees with $E$
  - For any other network’ $C' \subseteq B$
  agreeing with $E$, we have:

$$sol(C') = sol(C_L)$$

coNP-complete [Constraint Acquisition, AIJ17]
Convergence / Collapse states
Convergence / Collapse states
Acquisition using membership queries

- **CONACQ** [Bessiere et al. AIJ17]
  - SAT-Based constraint acquisition
  - Bidirectional search using Membership queries
  - Conacq1.0 (passive learning)
  - Conacq2.0 (active learning)

\[ K = \left( \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \right) \wedge \left( x_4 \vee x_5 \vee x_6 \vee x_7 \right) \ldots \]

Non-learnability using Membership queries [Constraint Acquisition, AIJ17]
Acquisition using complex queries

- **Matchmaker agents** [Freuder and Wallace wAAAI97]
- **Argument-Based CONACQ** [Friedrich et al.09]
- **ILP-Based Acquisition** [Lallouet et al. 10]
Structured problem acquisition

ModelSeeker [Beldiceanu and Simonis, CP11’12]

- A passive learning
- Based on global constraint catalogue (≈1000)
- Bottom-up search
- ModelSeeker learns constraints underlying the scheduling of the Bundesliga (the German Football Liga) from a single example schedule.
QUACQ: Quick Acquisition

- **QUACQ** [Bessiere et al. IJCAI13]
  - Active learning approach
  - Bidirectional search
    - But it can be top-down search if no positive example
  - Based on partial queries to elucidate the scope of the constraint to learn
  - Learnability using partial queries
ask(2, 8, 4, 2, 6, 5, 1, 6)
Partial Queries

ask(2, 8, 4, 2, 6, 5, 1, 6) = No
ask(2, 8, 4, 2, -, -, -, -) = No
Partial Queries

ask(2, 8, -, -, -, -, -, -) = Yes
Partial Queries

ask(2, 8, 4, -, -, -, -, -, -) = No
QUACQ: Quick Acquisition

yes
reduce(B)
ask(e)
Gen-query
QUACQ: Quick Acquisition

- yes
  - reduce(B)
- ask(e)
  - Gen-query
- No
  - partial-ask(e)
  - FindScope
QUACQ: Quick Acquisition
QUACQ: Quick Acquisition

- **ask(e)**
  - yes → **reduce(B)**
  - no → **partial-ask(e)**

- **Gen-query**

- **FindScope**

- **FindC**

- **Update(C_L)**
QUACQ: Quick Acquisition

Yes:
- reduce(B)
- Gen-query
- B=ø
- Update(C_L)

No:
- partial-ask(e)
- FindScope
- FindC
Algorithm 1: QUACQ: Acquiring a constraint network $C_T$ with partial queries

1. $C_L \leftarrow \emptyset$;
2. while true do
3.   if $\text{sol}(C_L) = \emptyset$ then return "collapse";
4.   choose $e$ in $D^X$ accepted by $C_L$ and rejected by $B$;
5.   if $e = \text{nil}$ then return "convergence on $C_L$";
6.   if $\text{ASK}(e) = \text{yes}$ then $B \leftarrow B \setminus \kappa_B(e)$;
7.   else
8.     $c \leftarrow \text{FindC}(e, \text{FindScope}(e, \emptyset, X, \text{false}))$;
9.     if $c = \text{nil}$ then return "collapse";
10.    else $C_L \leftarrow C_L \cup \{c\}$;
The number of queries required to find the target concept is in:

$$O(|C_T| \cdot (\log |X| + |\Gamma|))$$

The number of queries required to converge is in:

$$O(|B|)$$
In practice?

Limitations:

• QUACQ needs more than 8000 queries to learn the Sudoku model
• Generating a query can be time-consuming

Need:

• Reduce the dialogue with the user and the waiting time

How:

• Eliciting more information on why a complete instantiation is classified as negative by the user [MultiAcq, IJCAI16]
• Eliciting more information by asking complex queries to the user [ECAI14, ICTAI15, IJCAI16]
• Time-bounded query generator [T-QUACQ, CPAIOR’18]
Partial Queries

ask(2, 8, 4, 2, 6, 5, 1, 6) = No
In practice?

Limitations:
- QUACQ needs more than 8000 queries to learn the Sudoku model
- Generating a query can be time-consuming

Need:
- Reduce the dialogue with the user and the waiting time

How:
- Eliciting more information on why a complete instantiation is classified as negative by the user [MultiAcq, IJCAI16]
- Eliciting more information by asking complex queries to the user [ECAI14, ICTAI15, IJCAI16]
- Time-bounded query generator [T-QUACQ, CPAIOR’18]
Variables and Types

A type is a subset of variables defined by the user as having a common property.

Example (School Timetabling Problem)
A type is a subset of variables defined by the user as having a common property.

Example (School Timetabling Problem)

Can C1 be generalized to all Teachers, Rooms and Courses?
In practice?

Limitations:
- QUACQ needs more than 8000 queries to learn the Sudoku model
- Generating a query can be time-consuming

Need:
- Reduce the dialogue with the user and the waiting time

How:
- Eliciting more information on why a complete instantiation is classified as negative by the user [MultiAcq, IJCAI16]
- Eliciting more information by asking complex queries to the user [ECAI14, ICTAI15, IJCAI16]
- Time-bounded query generator [T-QUACQ, CPAIOR’18]
Conclusions

- Formal definition of Constraint Acquisition Problem
- Architectures for acquiring constraint networks

Future works:
- Taxonomy of queries
- Constraint Acquisition toolbox
Constraint Acquisition

Nadjib Lazaar

Thank you!!

U. Montpellier, France
LIRMM - CNRS - COCONUT team
21-05-19
NordConsNet19 - SIMULA - Oslo