pyDCOP
a DCOP library in python for IoT and dynamic systems

Pierre Rust$^{1,2}$ Gauthier Picard$^1$ Fano Ramparany$^2$

$^1$MINES Saint-Étienne, CNRS Lab Hubert Curien UMR 5516

$^2$Orange Labs
pyDCOP is an open source software library for the study of DCOP algorithms and their use in real world application

https://github.com/Orange-OpenSource/pyDcop
pyDCOP

- DCOPs
- Features
- Usage
- Architecture
- Contributing / Extending pyDCOP
Distributed Constraints Optimization Problem

A set of agents *coordinate* to select a joint assignment that *optimizes* a global objective
Distributed Constraints Optimization Problem

Formally, A DCOP is a tuple \( \langle A, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu \rangle \)

- \( A = \{a_1, \ldots, a_{|A|}\} \) are agents
- \( \mathcal{X} = \{x_1, \ldots, x_n\} \) are variables
- \( \mathcal{D} = \{\mathcal{D}_{x_1}, \ldots, \mathcal{D}_{x_n}\} \) are finite domains for the variables
- \( \mathcal{C} = \{c_1, \ldots, c_m\} \) are of soft constraints
- \( \mu : \mathcal{X} \rightarrow A \) associates variables to agents

A solution to the DCOP is an assignment to all variables that minimizes

\[
\sum_i c_i
\]
DCOP’s are being studied for many applications
DCOP’s are being studied for many applications

- Sensor networks
DCOP’s are being studied for many applications

- Sensor networks
- Resource allocation
DCOP’s are being studied for many applications

- Sensor networks
- Resource allocation
- Smart transport, Traffic lights
DCOP’s are being studied for many applications

- Sensor networks
- Resource allocation
- Smart transport, Traffic lights
- Internet-of-Things
DCOP’s are being studied for many applications

- Sensor networks
- Resource allocation
- Smart transport, Traffic lights
- Internet-of-Things
- Smart home
DCOP’s are being studied for many applications

- Sensor networks
- Resource allocation
- Smart transport, Traffic lights
- Internet-of-Things
- Smart home
- ...

More generally

Distributed Coordination and Problem Solving in Multi-Agent Systems
pyDCOP allows studying existing algorithms

- Implementations of standard algorithms
- Extensive metrics collection (cycles, messages, etc.)
- Evaluating meta-parameter effects
- Simulating large-scale agent’s networks
- Prototyping real-life scenario
pyDCOP makes it easy to develop new DCOP algorithms

- All infrastructure is provided
- Easy to code new ideas
- Base classes and utilities are provided
- Only need to code your algorithm’s logic
**pyDCOP** is multi-platform

- Linux, Mac OS, and Windows
- PC, Virtual machines, Raspberry-Pi and other single-board computers
**pyDCOP** is multi-process and multi-machine

**pyDCOP** can be executed...

- in a single process, using threads and in-memory communication
- as several processes, each hosting several agents
- as independent processes, running on different machines with network communication
**pyDCOP** is multi-process and multi-machine

**pyDCOP** can be executed...
- in a single process, using threads and in-memory communication
- as several processes, each hosting several agents
- as independent processes, running on different machines with network communication

⇒ **Large-scale simulations and physical prototypes**
pyDCOP user interface

- Command-line interface for easy experiment scripting
- Graphical interface
pyDCOP implements novel approaches for dynamic IoT systems

Several computation **distribution** options

- Optimize placement of variables and/or constraints on agents
- Depending on network and devices capabilities constraints
pyDCOP implements novel approaches for dynamic IoT systems

Resilient coordination: Self-repair and Dynamic DCOPs

- Replication of problem definitions
- Distributed self-repair mechanism
**pyDCOP - Concepts & Principles**

1. **Problem**
2. **DCOP**
   - Agents, Variables, Constraints, etc.
3. **Computation Graph**
4. **Computations On Agents**
5. **Solve / Run-time**
6. **Algorithm**
   - Computations definitions
7. **Instantiate**
8. **Distribute**
9. **Model**
Simple YAML-based file format for DCOP description
Optionally: Agents’ capacity, communication costs, hosting costs
When you don’t need to compute a distribution.
When you want to force a specific distribution.
Solving a DCOP, the simplest way:

- Start threaded agents
- Computes a distribution
- Deploys the computations
- Monitor agents
- Collects metrics and results
Other commands:

- `distribute`: only computes distribution
- `run`: for dynamic DCOPs
- `generate`: generates problems
- `replica_dist`: replica distribution
- `agent`: standalone agent
- `orchestrator`: standalone orchestrator

All commands are documented on:
pyDCOP

- DCOPs
- Features
- Usage
- Architecture
- Contributing / Extending pyDCOP
pyDCOP Architecture - Communication

Agent 1

C1

C2

Communication layer

Agent 2

C3

Communication layer

In-Memory
HTTP + JSON

Directory
pyDCOP Architecture - Agent

Agent 1

- UI
- Rep.
- Mgt
- C1
- C2
- C3

Communication layer

- DCOP's algorithm computations
- Expose agent's Internals through Websocket server
- Replicate computations on neighbors (resilience)
- Add / remove computations
- Repair
- Metrics
- Discovery

Add / remove computations
Repair
Metrics
Discovery
■ DCOPs
■ Features
■ Usage
■ Architecture
■ Contributing / Extending pyDCOP
Extension points

- DCOP algorithms
- Distribution
- Communication Layer
- Directory / Discovery
- Repair mechanism: replication & migration
pyDCOP - Implementing an algorithm

Simply extend one of the base Computation classes:
TODO include a very simple example? DSA?
Call for action

All contributions are welcome!
- DCOP algorithms implementations
- Distribution
- Repair mechanisms: replication & repair
- Documentation
- Any feedback on ease-of-use, implementation, etc.
**pyDCOP** is already used in academia

- Several publications [IJCAI’16, OPTMAS’17, OPTMAS’18, AAMAS’19]
- Teaching (master classes, conference tutorials, and summer schools)

Additionally, **pyDCOP** has extensive user documentation, including introductory tutorials:

[https://pydcop.readthedocs.io](https://pydcop.readthedocs.io)
pyDCOP

**Source code:**
https://github.com/Orange-OpenSource/pyDcop

**Documentation:**
https://pydcop.readthedocs.io