



**HAL**  
open science

## Coordination et résilience optimales d'objets intelligents

Gauthier Picard, Pierre Rust, Fano Ramparany

► **To cite this version:**

Gauthier Picard, Pierre Rust, Fano Ramparany. Coordination et résilience optimales d'objets intelligents. Un état des lieux sur les activités de recherche sur l'intelligence artificielle dans les écoles de l'IMT, Apr 2019, Paris, France. emse-02102291

**HAL Id: emse-02102291**

**<https://hal-emse.ccsd.cnrs.fr/emse-02102291v1>**

Submitted on 17 Apr 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

### Parties prenantes



### Auteurs

Gauthier Picard

Henri Fayol Institute  
Laboratoire Hubert Curien  
CNRS UMR 5516

picard@emse.fr

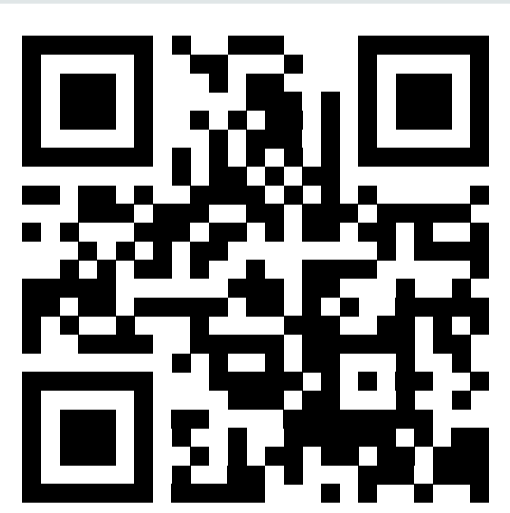
Pierre Rust

Orange Labs  
Laboratoire Hubert Curien  
CNRS UMR 5516

pierre.rust@orange.com

Fano Ramparany

Orange Labs  
fano.ramparany@orange.com



### Partenaires



## Coordination distribuée et autonome pour l'IoT

### Motivations

- ▶ 25 milliards d'objets en 2020
- ▶ Appareils bon marché mais contraints
- ▶ Aujourd'hui : coordination centralisée au niveau du cloud

### Coordination décentralisée

- ▶ Montée à l'échelle, interactions locales
- ▶ Pas de point d'échec central (SPOF)
- ▶ Pas de goulet d'étranglement pour les communications

### DCOP : Problème d'optimisation distribuée sous contraintes

Un tuple  $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu \rangle$

- ▶  $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$ , est un ensemble d'agents
- ▶  $\mathcal{X} = \{x_1, \dots, x_n\}$ , est un ensemble de variables
- ▶  $\mathcal{D} = \{\mathcal{D}_{x_1}, \dots, \mathcal{D}_{x_n}\}$ , les domaines de variables,  $\mathcal{D}_{x_i} = \{v_1, \dots, v_k\}$
- ▶  $\mathcal{C} = \{c_1, \dots, c_m\}$ , est un ensemble de contraintes souples,  $c_i$  est une fonction de coût  $\in \mathbb{R} \cup \{\infty\}$  pour toutes affectations aux variables dans sa portée
- ▶  $\mu : \mathcal{X} \rightarrow \mathcal{A}$ , une fonction assignant chaque variable à un calcul

**Solution** : une affectation à toutes les variables qui minimise la somme des coûts  $\sum_i c_i$

### Algorithmes par envoi de messages

- ▶ **Complets** ADOPT, DPOP, etc.
- ▶ **Approchés** MGM, DSA, Max-Sum, etc.

## pyDCOP une librairie d'étude DCOP

### Fonctionnalités

- ▶ Implémentation des algorithmes majeurs : DSA, DPOP, MGM, Max-Sum, etc.
- ▶ Utilitaires pour le développement de nouveaux algorithmes
- ▶ Fournit toute l'infrastructure logicielle pour l'étude des algorithmes DCOP: instanciation des agents, déploiement de l'algorithme, contexte d'exécution, envoi des messages, etc.
- ▶ Générateur de problèmes de benchmark
- ▶ Monitoring et production de métriques: temps de résolution, nombre de cycles, charge réseau, coût et qualité de la solution, etc.

### pyDCOP pour l'IoT

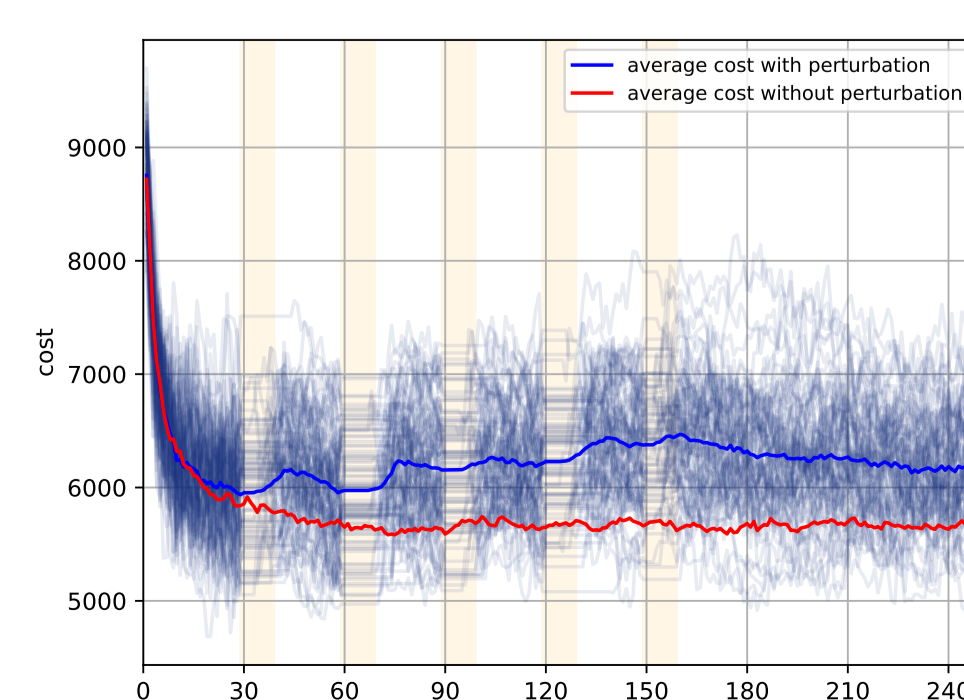
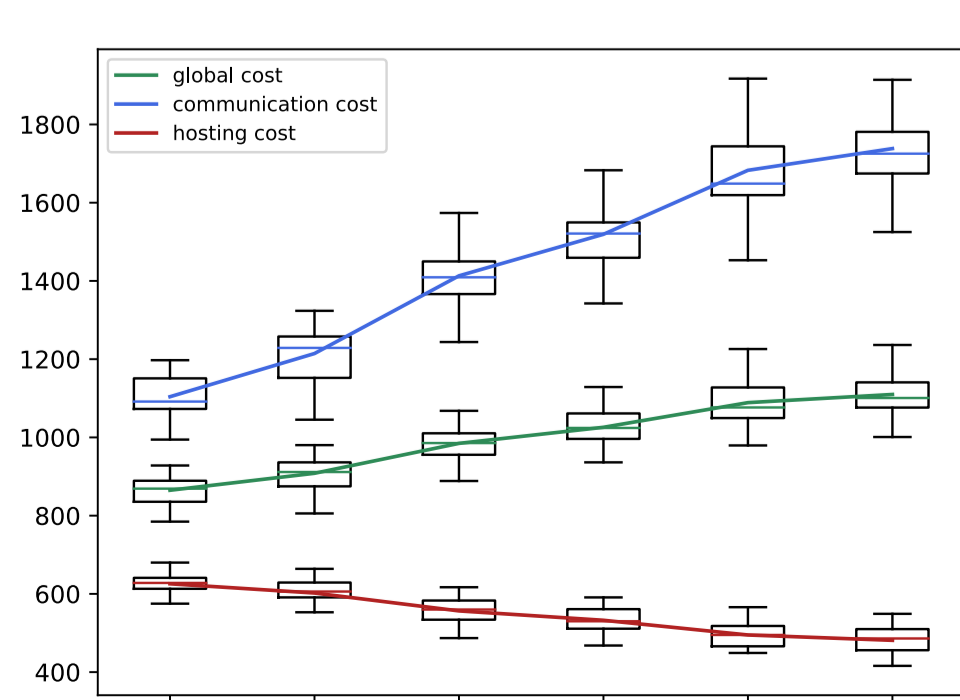
- ▶ Prise en compte de la problématique de la distribution des décisions sur l'infrastructure : optimisation du placement en fonction des coûts de communication, d'hébergement, etc.
- ▶ Résilience du système

### Prototypage avec pyDCOP

- ▶ Multi-plateforme
- ▶ Permet un déploiement *réel*: PC, Machine virtuelle, Raspberry Pi
- ▶ Interface graphique de monitoring

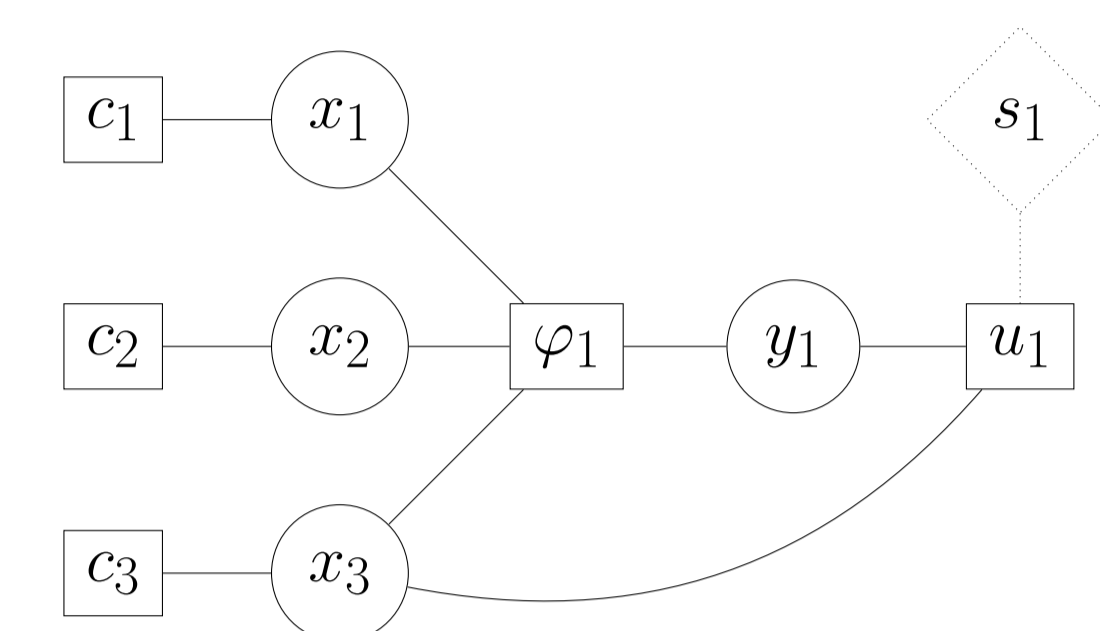
### Utilisation

- ▶ Source: <https://github.com/Orange-OpenSource/pyDcop>
- ▶ Documentation: <https://pydcop.readthedocs.io>
- ▶ en python



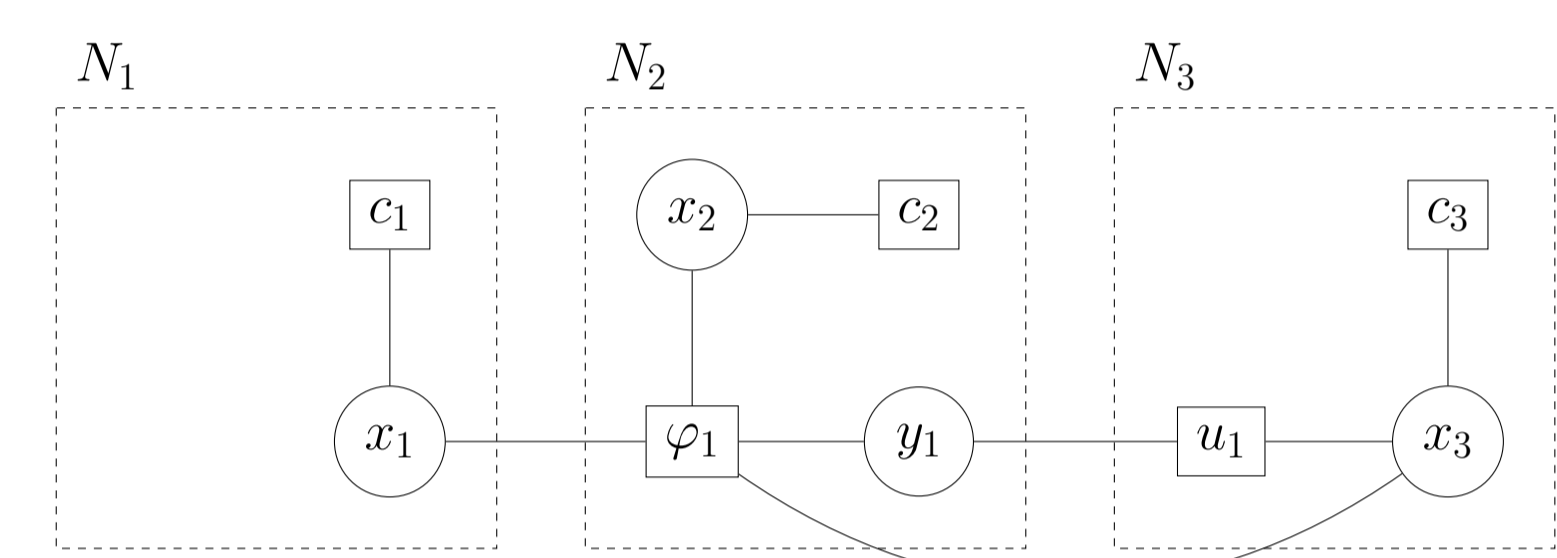
## Distribution et résolution d'un DCOP

### Représentation en graphe de calculs



- ▶ Graphe de facteurs
- ▶ Graphe de contraintes

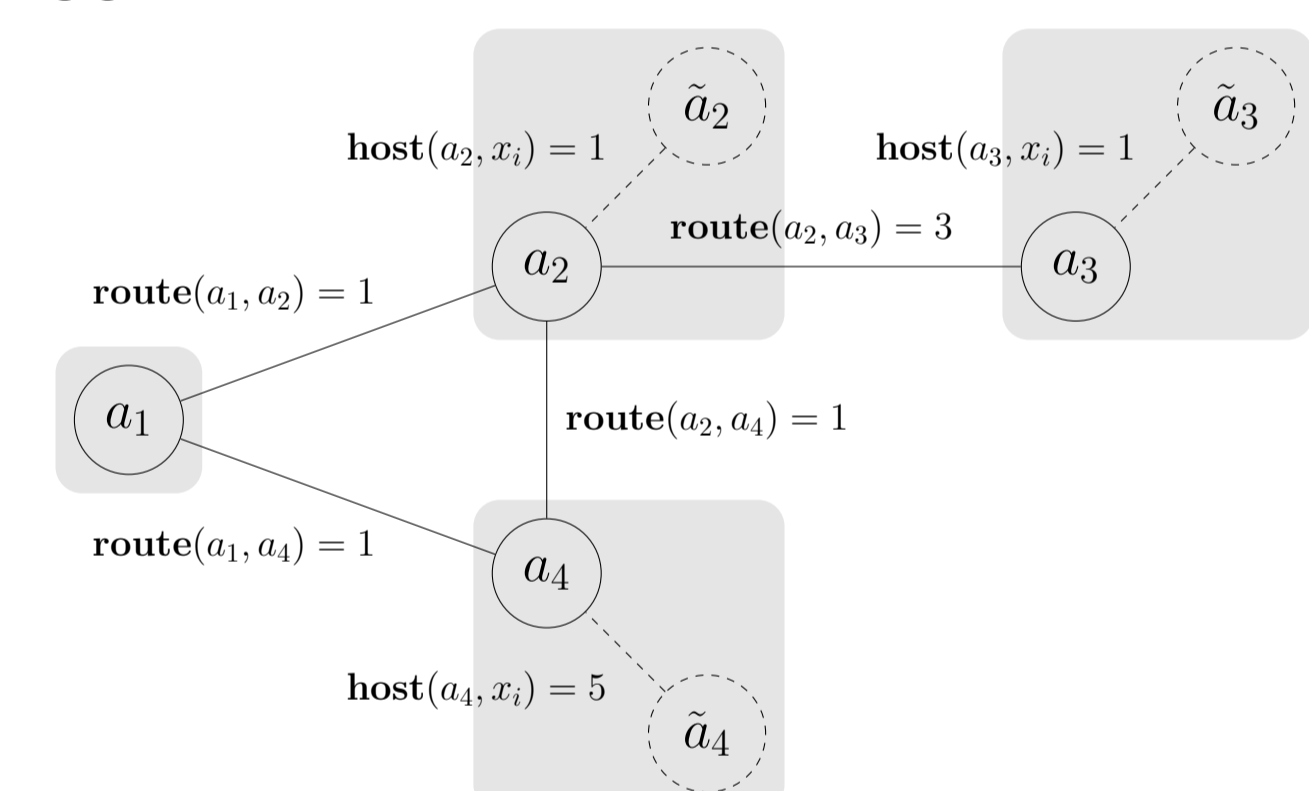
### Distribution du graphe de calculs



Un problème de partitionnement de graphe

- ▶ Heuristiques: garder les calculs proches des variables impactées
- ▶ Programmation Linéaire : minimiser les coûts de communication, d'hébergement, en respectant les capacités matérielles des agents
- ▶ etc.

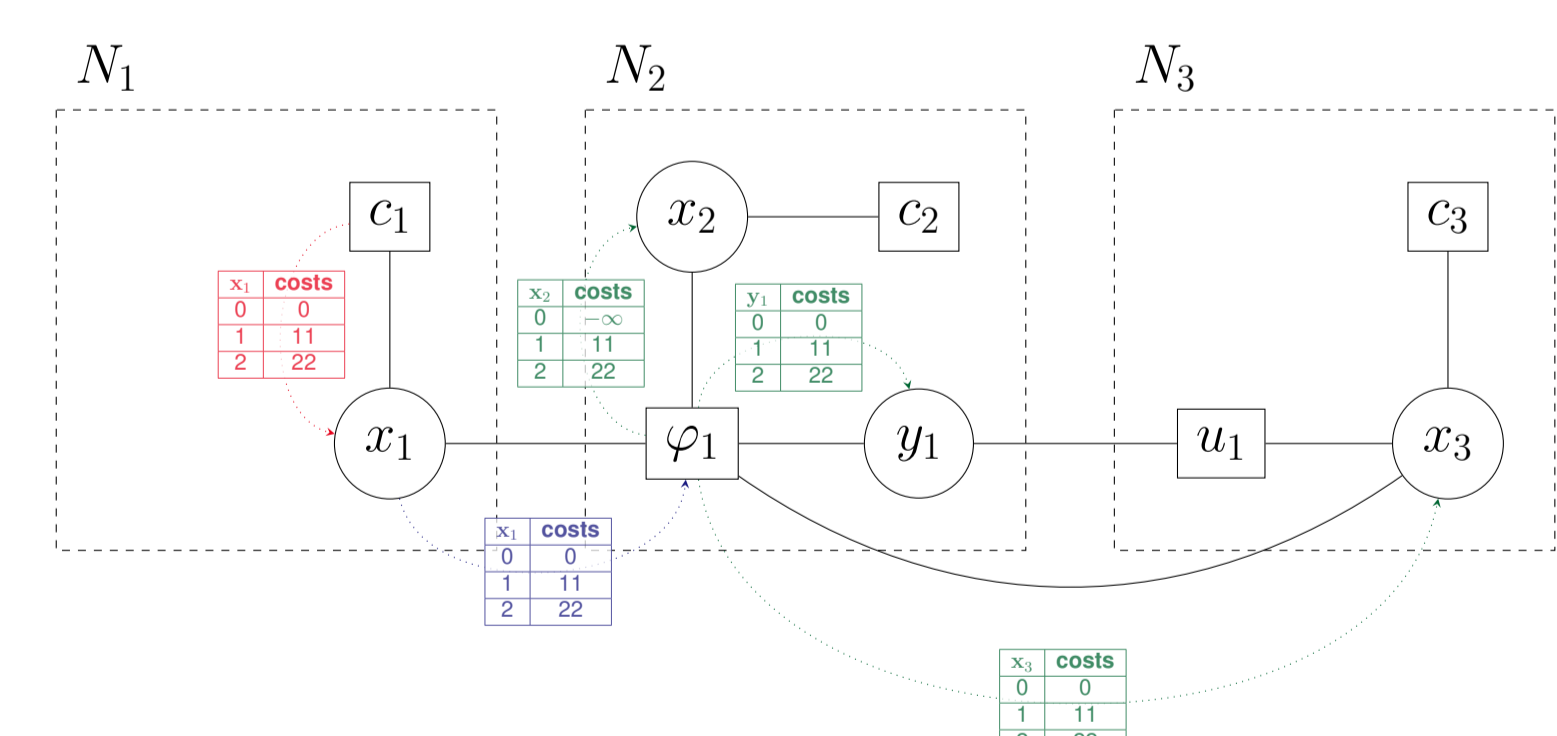
### K-Résilience



Survivre à la disparition d'éléments de l'infrastructure

- ▶ Réplication des calculs
- ▶ Réparation en ligne du déploiement

### Résolution



Avec des algorithmes DCOP asynchrones supportant la perte de messages.

- ▶ MaxSum, A-DSA pour le problème initial
- ▶ MGM2 pour la réparation du déploiement

## Références

- [1] P. Rust, G. Picard, and F. Ramparany. "Installing Resilience in Distributed Constraint Optimization Operated by Physical Multi-Agent Systems". In: *Autonomous Agents and Multiagent Systems (AAMAS)*. 2019.
- [2] P. Rust, G. Picard, and F. Ramparany. "On the Deployment of Factor Graph Elements to Operate Max-Sum in Dynamic Ambient Environments". In: *Autonomous Agents and Multiagent Systems – AAMAS 2017 Workshops, Best Papers, Sao Paulo, Brazil, May 8-12, 2017, Revised Selected Papers*. Vol. 10642. LNAI. Springer, 2017, pp. 116–137.
- [3] P. Rust, G. Picard, and F. Ramparany. "Using Message-passing DCOP Algorithms to Solve Energy-efficient Smart Environment Configuration Problems". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*. AAAI Press, 2016, pp. 468–474.