

# Approche par optimisation distribuée pour la configuration autonome et spontanée d'environnements intelligents

Pierre Rust<sup>1,2</sup>   Gauthier Picard<sup>1</sup>   Fano Ramparany<sup>2</sup>

<sup>1</sup>MINES Saint-Étienne, CNRS  
Lab Hubert Curien UMR 5516

<sup>2</sup>Orange Labs



# Contexte : l'Internet des Objets / IoT

- Sujet très à la mode aujourd'hui (marketing ?)
- 25 milliards d'objets connectés en 2020 ? (Gartner)
- Forte baisse du coût du hardware et des solutions de communications
- Des objets contraints :
  - ▶ mémoire et puissance de calcul limités
  - ▶ capacité de communication contraintes
- Coordination centralisée, implémentée au niveau "Cloud"

# Coordination entre objets connectés

Quelle approche pour l'IoT ?

- Coordination décentralisée :
  - ▶ pas de point de défaillance unique,
  - ▶ pas de goulot d'étranglement sur les communications,
  - ▶ meilleure montée en charge,
  - ▶ exploitation de la localité des interactions.
  
- Problème distribué d'optimisation sous contraintes :
  - ▶ distribuer les calculs directement sur les objets connectés,
  - ▶ algorithmes par envoi de messages.
  
- Application à l'intelligence ambiante et à la maison intelligente

# Optimisation distribuée

Un problème d'optimisation sous contraintes, où les variables sont portées par des *agents* :

- Chaque agent ne connaît que les contraintes impliquant ses propres variables
- Chaque agent ne connaît que les agents voisins (via les contraintes)
- Les agents communiquent uniquement avec leurs voisins
- Les choix d'affectation des valeurs sont pris, de manière autonome, par ces *agents*

# DCOP

## Problème d'Optimisation sous Contraintes Distribué

Un DCOP est un tuple  $\langle \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mu \rangle$ , où :

- $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$  un ensemble d'agents ;
- $\mathcal{X} = \{x_1, \dots, x_n\}$  un ensemble de variables ;
- $\mathcal{D} = \{\mathcal{D}_{x_1}, \dots, \mathcal{D}_{x_n}\}$  les domaines des variables  $x_i$  ;
- $\mathcal{C} = \{c_1, \dots, c_m\}$  des contraintes souples, où chaque  $c_i$  définit un coût  $\in \mathbb{R} \cup \{\infty\}$  pour chaque combinaison d'affectation aux variables impliquées dans cette contrainte ;
- $\mu : \mathcal{X} \rightarrow \mathcal{A}$  une application associant les variables à leur agent.

Une *solution* au DCOP est une affectation de valeurs à toutes les variables qui minimise la somme totale des coûts  $\sum_i c_i$ .

# Optimisation distribuée

## Pourquoi la distribution ?

- Confidentialité : sur les contraintes, sur les affectations choisies
- Résilience et robustesse : pas de point central
- Communications : locales, pas besoin de faire remonter les informations, puis de faire redescendre les affectations
- Rapidité / parallélisation
- Peut être utiliser pour modéliser un problème distribué de coordination

# Algorithmes DCOP

Algorithmes complets : ADOPT, **DPOP**, Action GDL, etc

- DPOP : programmation dynamique, opère sur un pseudo-arbre en faisant remonter des coûts vers la racine,
- complet, avec un nombre de messages linéaire, exponentiel dans les cas fortement cycliques

Algorithmes approchés : MGM, DSA, **Max-Sum**, etc.

- Max-Sum : inférence par propagation de croyance,
- complet dans le cas d'un graph acyclique, sinon sous optimal, convergence non-garantie, mais très bon résultats expérimentaux.

# Le modèle SECP

SECP : *Smart Environment Configuration Problem*

Problème de Configuration d'Environnements Intelligents

- Coordonner les objets de la maison
- Modéliser
  - ▶ les objets,
  - ▶ les liens entre les objets et l'environnement,
  - ▶ les objectifs de l'utilisateur.
- Formuler le problème comme un problème d'optimisation



# Le modèle SECP



## Effecteurs :

Ampoules connectés, TV, Volets roulants, ...

## Capteurs :

Capteur de présence, luminosité, etc.

## Modèle de dépendance physique :

E.g. Modèle d'éclairage du salon

## Préférences utilisateur :

Exprimées comme des règles :

IF	presence_living_room	=	1
AND	light_sensor_living_room	<	60
THEN	light_level_living_room	←	60
AND	shutter_living_room	←	0

# Le modèle SECP



## Effecteurs :

- Variable de *Decision*  $x_i$ , Domaine  $\mathbf{x}_i \in \mathcal{D}_{x_i}$
- Fonction de coût  $c_i : \mathcal{D}_{x_i} \rightarrow \mathbb{R}$

## Capteurs :

- Variable en *lecture seule*  $s_i$ , Domaine  $\mathbf{s}_i \in \mathcal{D}_{s_i}$

## Modèle de dépendance physique : $\langle y_j, \phi_j \rangle$

- Donne l'état de l'environnement en fonction de la valeurs des variables des effecteurs influençant ce modèle
- Variable  $y_j$  représente l'état *théorique* de l'environnement
- Fonction  $\phi_j : \prod_{s \in \sigma(\phi_j)} \mathcal{D}_s \rightarrow \mathcal{D}_{y_j}$

## Préférences utilisateur :

- Fonction d'utilité  $u_k$
- Distance entre l'état théorique courant de l'environnement et son état cible.

# Formulation du SECP en DCOP

- Problème d'optimisation multi-objectifs

$$\underset{x_i \in \nu(\mathfrak{A})}{\text{minimiser}} \sum_{i \in \mathfrak{A}} c_i \quad \text{et} \quad \underset{\substack{x_i \in \nu(\mathfrak{A}) \\ y_j \in \nu(\Phi)}}{\text{maximiser}} \sum_{k \in \mathfrak{R}} u_k$$

$$\text{avec } \phi_j(x_j^1, \dots, x_j^{\bar{\phi}_j}) = y_j \quad \forall y_j \in \nu(\Phi)$$

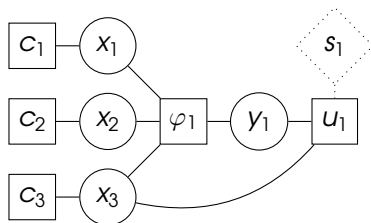
- DCOP mono-objectif :

$$\underset{\substack{x_i \in \nu(\mathfrak{A}) \\ y_j \in \nu(\Phi)}}{\text{maximiser}} \quad \omega_U \sum_{k \in \mathfrak{R}} u_k - \omega_C \sum_{i \in \mathfrak{A}} c_i + \sum_{\varphi_j \in \tilde{\nu}} \varphi_j$$

En reformulant les contraintes dures  $\phi_j$

# Formulation du SECP en DCOP

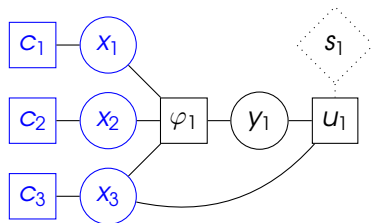
Représentation sous forme d'un graphe de facteurs, pour la représentation des contraintes n-aires.



# Distribution des calculs

déploiement sur les agents

- Distribution optimale : partitionnement de graphe, NP-complet
- Contraintes spécifiques au modèle SECP
  - ▶ Capacité mémoire
  - ▶ Optimiser pour la communication
  - ▶ Des calculs fixes, d'autres mobiles

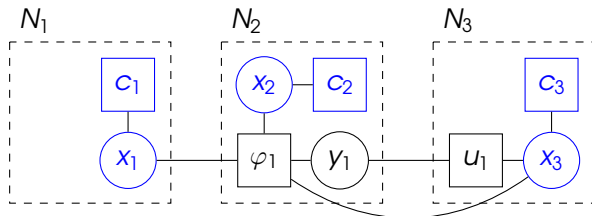


# Distribution des calculs

## déploiement sur les agents

### ■ Heuristique simple :

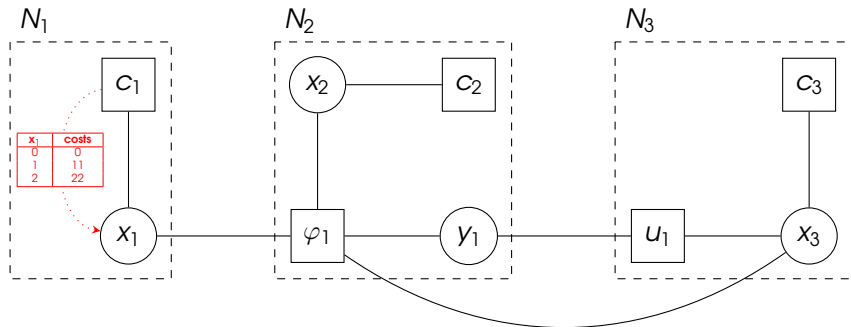
- ▶ Pas de calculs sur les objets 'dormants' (capteurs)
- ▶ Rapprocher les calculs des variables impactées
- ▶ Distribuer la charge de calcul sur les agents



# Résolution du SECP

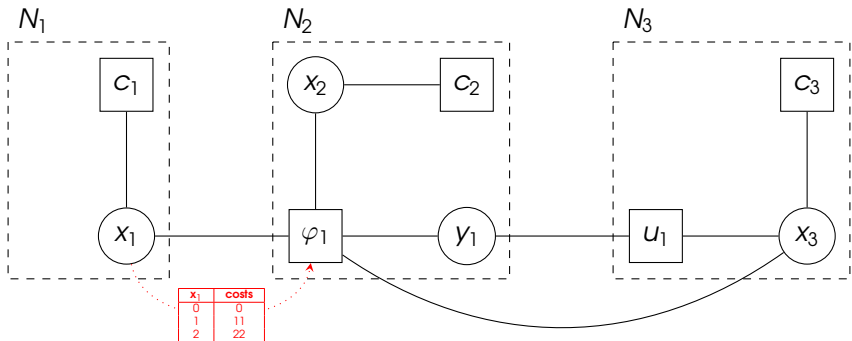
Optimisation via un algorithme par passage de messages

Exemple avec Max-Sum



# Résolution du SECP

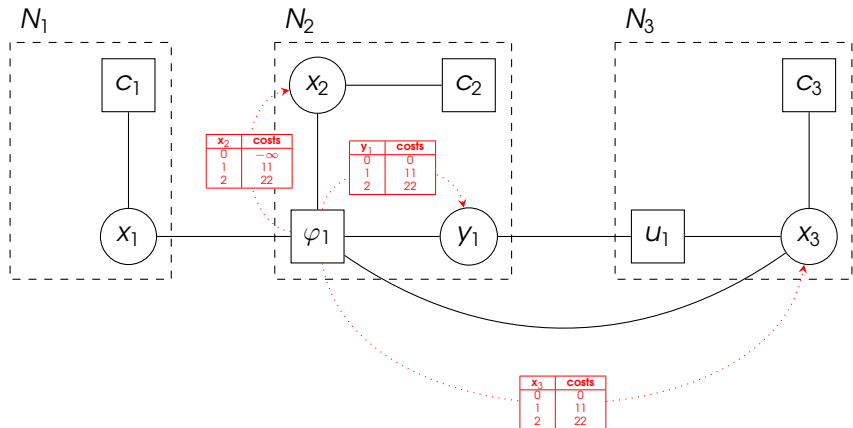
Optimisation via un algorithme par passage de messages





# Résolution du SECP

Optimisation via un algorithme par passage de messages

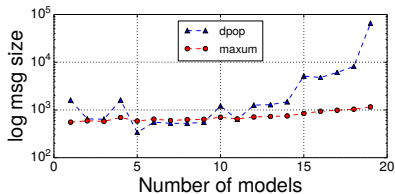
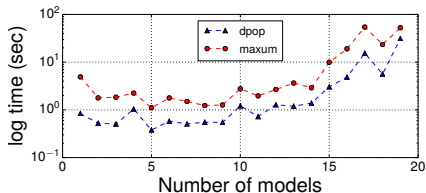
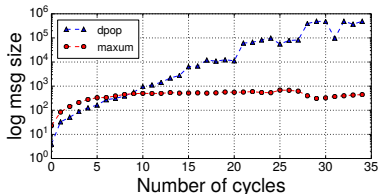


# Expérimentation

- Instances générées aléatoirement, maison réaliste
- Uniquement des graphes connexes
- 3 expérimentations :
  - ▶ Nombre croissant de modèle (10 effecteurs & 5 règles)
  - ▶ Nombre croissant de règles (10 effecteurs & 5 modèles)
  - ▶ Nombre croissant de cycles
- Résolution avec deux algorithmes DCOP : Max-Sum and DPOP
  - ▶ DPOP est complet and sert d'étalon
  - ▶ Max-Sum est approché, mais léger

# Results : DPOP vs. Max-Sum

- Dpop est un peu plus rapide
- Max-Sum génère un volume de message inférieur
- Max-Sum est presque toujours optimal
- La complexité dépend du nombre de cycle









# Conclusions

- Le modèle SECP permet de représenter les objets connectés, leur lien avec l'environnement et les objectifs spécifiés par l'utilisateur
- Ce modèle est une approche viable pour la coordination **décentralisée** et **autonome** des objets dans un environnement intelligent
- L'algorithme Max-Sum est le plus approprié pour les objets (et les communications) contraints

# Perspectives

- Dynamicité de l'environnement, résilience du système
  - ▶ réparation, distribuée et en ligne de la distribution
- Prendre en compte les aspects temporels (planification, etc.)
- Apprentissage des modèles de dépendances physique
- Adaptation de Max-Sum pour exploiter les caractéristiques spécifiques de notre problème

## References

-  AJI, S.M. et R.J. McELIECE (2000). "The generalized distributive law". In : *Information Theory, IEEE Transactions on* 46.2, p. 325–343. ISSN : 0018-9448. DOI : 10.1109/18.825794.
-  ARCHOS (2016). *Smart Home*.  
<http://www.archos.com/us/products/objects/chome/ash/index.html>.  
(accessed January 26).
-  BICHOT, C.-E. et P. SIARRY, éd. (2011). *Graph Partitioning*. Wiley.
-  CERQUIDES, J., A. FARINELLI, P. MESEGUER et S. D. RAMCHURN (2014). "A Tutorial on Optimization for Multi-Agent Systems". In : *The Computer Journal* 57.6, p. 799–824. DOI : 10.1093/comjnl/bxt146. URL : <http://dx.doi.org/10.1093/comjnl/bxt146>.
-  DECHTER, R. (2003). *Constraint processing*. Morgan Kaufman. URL : [http://books.google.com/books?hl=en&lr=&id=w4LG4EU0BCwC&oi=fnd&pg=PP2&dq=Constraint+processing&ots=ur\\\_5y38Tbs&sig=la9V-uFZ0kGza4iD4HM11F5-1Bo](http://books.google.com/books?hl=en&lr=&id=w4LG4EU0BCwC&oi=fnd&pg=PP2&dq=Constraint+processing&ots=ur\_5y38Tbs&sig=la9V-uFZ0kGza4iD4HM11F5-1Bo).
-  DEGELER, V. et A. LAZOVIK (2013). "Dynamic Constraint Reasoning in Smart Environments". In : *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, p. 167–174.

## References (cont.)



FARINELLI, A., A. ROGERS, A. PETCU et N. R. JENNINGS (2008). "Decentralised Coordination of Low-power Embedded Devices Using the Max-sum Algorithm". In : *International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*, p. 639–646. ISBN : 978-0-9817381-1-6. URL : <http://dl.acm.org/citation.cfm?id=1402298.1402313>.



JAIN, M., M. TAYLOR, M. TAMBE et M. YOKOO (2009). "DCOPs Meet the Real World : Exploring Unknown Reward Matrices with Applications to Mobile Sensor Networks". In : *International Joint Conference on Artificial Intelligence (IJCAI'09)*, p. 181–186.



LYNCH, N.A. (1997). *Distributed Algorithms*. Morgan Kaufmann.









MAHESWARAN, R.T., J.P PEARCE et M. TAMBE (2004). "Distributed Algorithms for DCOP : A Graphical-Game-Based Approach". In : *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems (PDCS), San Francisco, CA*, p. 432–439.



MATSUI, T., M. SILAGHI, K. HIRAYAMA, M. YOKOO et H. MATSUO (2012). "Distributed Search Method with Bounded Cost Vectors on Multiple Objective DCOPs". In : *15th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*. Springer, p. 137–152.

## References (cont.)

-  MODI, P. J., W. SHEN, M. TAMBE et M. YOKOO (2005a). "ADOPT : Asynchronous Distributed Constraint Optimization with Quality Guarantees". In : *Artificial Intelligence* 161.2, p. 149–180.
-  MODI, P. J., W. SHEN, M. TAMBE et M. YOKOO (2005b). "ADOPT : Asynchronous distributed constraint optimization with quality guarantees." In : *Artificial Intelligence Journal*.
-  ORANGE (2016). *Homelive*. <http://homelive.orange.fr>. (accessed January 26).
-  PECORA, F. et A. CESTA (2007). "DCOP for Smart Homes : a case study". en. In : *Computational Intelligence* 23.4, p. 395–419. ISSN : 08247935. DOI : 10.1111/j.1467-8640.2007.00313.x. URL : <http://doi.wiley.com/10.1111/j.1467-8640.2007.00313.x> (visité le 30/10/2015).
-  PETCU, A. et B. FALTINGS (2005). "A scalable method for multiagent constraint optimization". In : *International Joint Conference on Artificial Intelligence (IJCAI'05)*, p. 266–271.
-  RASMUSSEN, C.E. et C. WILLIAMS (2006). *Gaussian Processes for Machine Learning*. MIT Press.



## References (cont.)



ROGERS, A., A. FARINELLI, R. STRANDERS et N.R. JENNINGS (2011). "Bounded approximate decentralised coordination via the max-sum algorithm". In : *Artificial Intelligence* 175.2, p. 730–759. ISSN : 0004-3702. DOI : <http://dx.doi.org/10.1016/j.artint.2010.11.001>. URL : <http://www.sciencedirect.com/science/article/pii/S0004370210001803>.



SAMSUNG (2016). *SmartThings*. <http://www.samsung.com/us/smart-home/>. (accessed January 26).



STIMSON, A. (1974). *Photometry and Radiometry for Engineers*. Wiley et Son.



VINYALS, Meritxell, Juan A. RODRIGUEZ-AGUILAR et Jesús CERQUIDES (2010). "Constructing a unifying theory of dynamic programming DCOP algorithms via the generalized distributive law". In : *Autonomous Agents and Multi-Agent Systems* 22.3, p. 439–464. ISSN : 1573-7454. DOI : [10.1007/s10458-010-9132-7](https://doi.org/10.1007/s10458-010-9132-7). URL : <http://dx.doi.org/10.1007/s10458-010-9132-7>.