Conception d'une place de marché pour la vente et la distribution d'énergie dans les *smart grids*

J. Cerquides[1] **Gauthier Picard**[2,3] J. A. Rodríguez-Aguilar[1]

[1]IIIA-CSIC
[2]Connected Intelligence, LaHC UMR CNRS 5516
[3]Institut Henri Fayol, MINES Saint-Étienne

Les *smart grids*

Contexte technologique et économique

Smart grids : promesses & bénéfices attendus

Smart Energy Aware Systems

- Nouvelle logique de production : **production décentralisée**
  - Démocratisation de la production décentralisée : équilibrage de charge local et réduction des pertes énergétiques
- Nouvelles informations de contexte : **conscience énergétique** (ou *energy awareness*)
  - Données captées fréquemment (consommation, production, tarification, etc.) impactent la mise-à-jour des échanges
- Nouvelle logique d'échanges commerciaux : **prosommation**

Smart grids : promesses & bénéfices attendus

Smart Energy Aware Systems

- Nouvelle logique de production : **production décentralisée**
  - Démocratisation de la production décentralisée : équilibrage de charge local et réduction des pertes énergétiques
- Nouvelles informations de contexte : **conscience énergétique** (ou *energy awareness*)
  - Données captées fréquemment (consommation, production, tarification, etc.) impactent la mise-à-jour des échanges
- Nouvelle logique d'échanges commerciaux : **prosommation**

Comment concevoir un marché décentralisé pour l'échange et la distribution de l'énergie ?

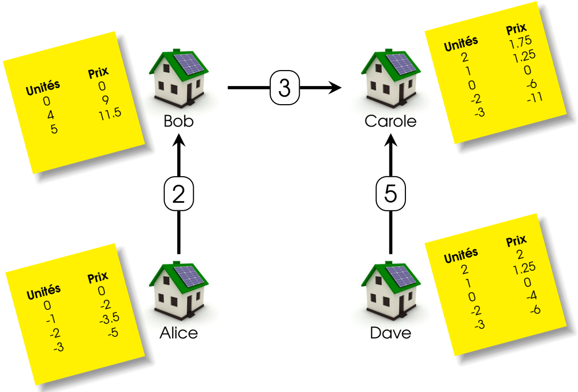
Exemple : scénario d'échange d'énergie

- *Prosommateurs* ($j \in P$)

Exemple : scénario d'échange d'énergie

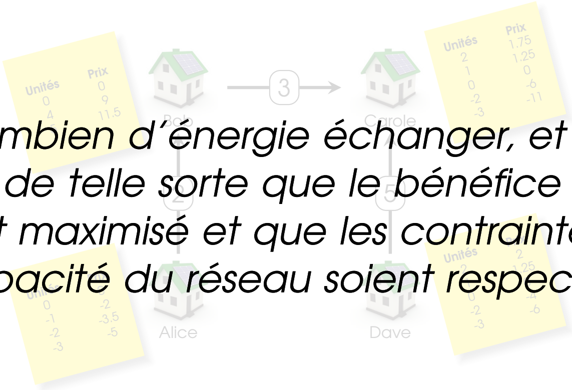
- *Prosommateurs* ($j \in P$)
- *Offres* ($o_j : \mathbb{Z} \rightarrow \mathbb{R} \cup \{-\infty\}$)

Exemple : scénario d'échange d'énergie



- *Prosommateurs* ($j \in P$)
- *Offres* ($o_j : \mathbb{Z} \to \mathbb{R} \cup \{-\infty\}$)
- *Liens* ($\{i, j\}$) de capacité max (c_{ij})

Exemple : scénario d’échange d’énergie



Combien d’énergie échanger, et avec qui, de telle sorte que le bénéfice global soit maximisé et que les contraintes de capacité du réseau soient respectées ?

Définition : problème d'allocation d'énergie

Le problème d'allocation d'énergie (*EAP*) revient à trouver une allocation $\mathbf{Y}$ qui maximise le bénéfice global $Value(\mathbf{Y})$, avec

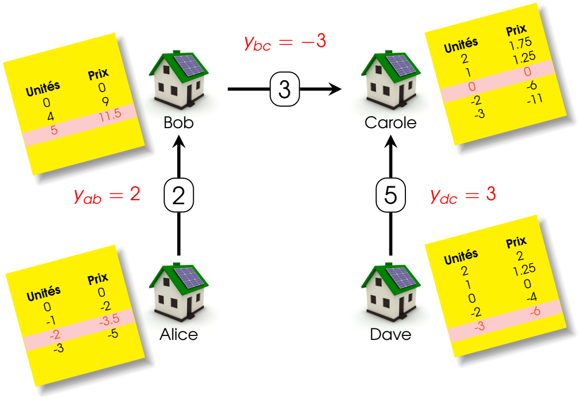
$$Value(\mathbf{Y}) = \sum_{i \in P} v_j(\mathbf{Y}_j)$$

$$v_j(\mathbf{Y}_j) = o_j(net(\mathbf{Y}_j))$$

$$net(\mathbf{Y}_j) = \sum_{i \in in(j)} \mathbf{y}_{ij} - \sum_{k \in out(j)} \mathbf{y}_{jk}$$

où $\mathbf{y}_{ij}$ représente le nombre d'unités que le prosommateur i vend à j (borné par c_{ij})

Exemple : scénario d'échange d'énergie (sol.)



$$Value(\mathbf{Y}) = o^{-2}_{Alice} + o^{5}_{Bob} + o^{0}_{Carole} + o^{-3}_{Dave} = -3.5 + 11.5 + 0 - 6 = 2$$

Techniques d'allocation distribuées

- Techniques basées sur des marchés
  - **Double ventes aux enchères** (*call market* ou CDA) où l'énergie est négociée sur une base journalière
  - La correspondance entre offre et demande est calculée par une autorité **centrale**
  - Les mécanismes actuels négligent les contraintes physiques du réseau
    $\rightarrow$ **Échange et distribution sont deux activités découplées**
- Techniques par envoi de messages
  - Programmation dynamique distribuée (MILLER, 2014 ; KUMAR et al., 2009)
  - Propagation de croyances (MILLER, 2014)

Notre contribution

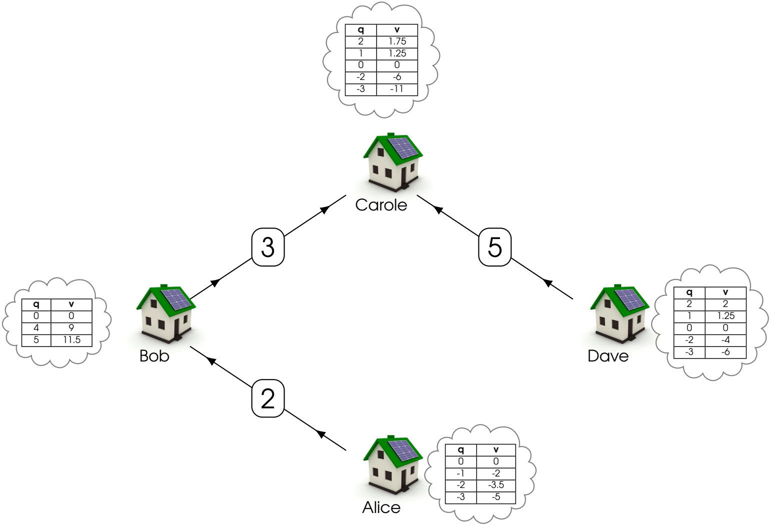
- Exploiter la structure **arborescente** des réseaux d'énergie (Gonen, 2014)
- Résoudre l'EAP comme un problème d'optimisation sous contraintes distribué (**DCOP**)
- Concevoir un algorithme par **envoi de messages exact** basé sur la programmation dynamique
  - Acyclic-Solving (Dechter, 2003)
- Calculer efficacement les messages en exploitant la structure algébrique des offres et des messages : **valuations**

Technique par envoi de messages

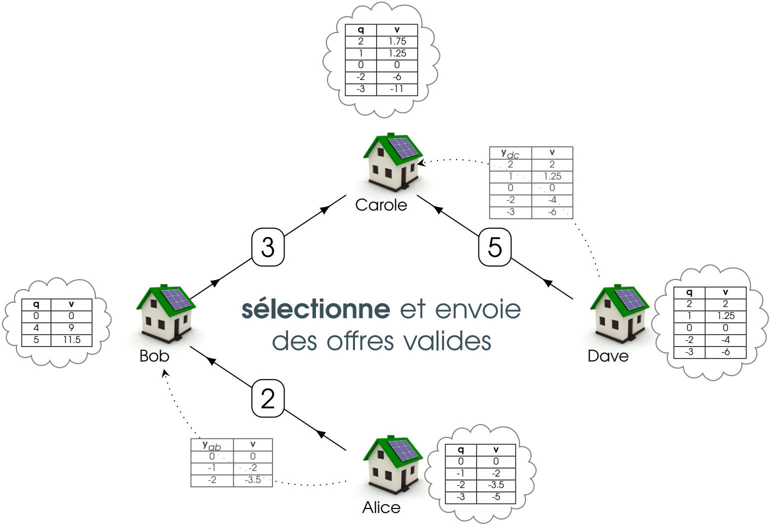
q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

Carole

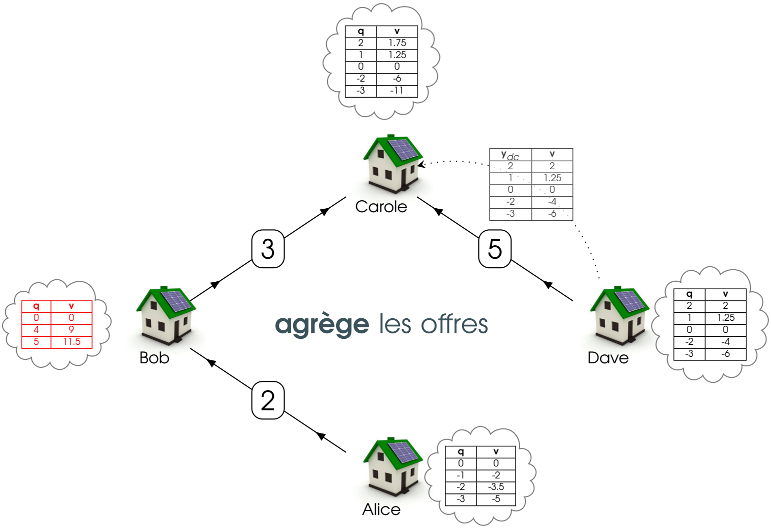
Technique par envoi de messages



Technique par envoi de messages



Technique par envoi de messages



q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

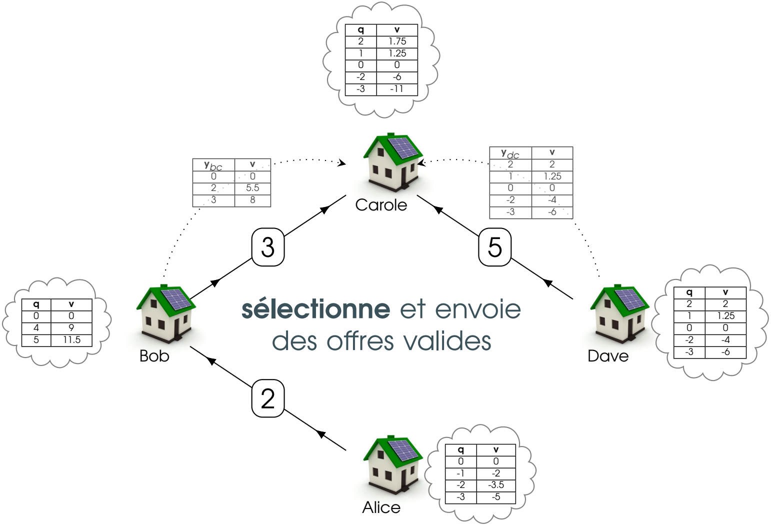
y_{dc}	v
2	2
1	1.25
0	0
-2	-4
-3	-6

q	v
0	0
4	9
5	11.5

q	v
2	2
1	1.25
0	0
-2	-4
-3	-6

q	v
0	0
-1	-2
-2	-3.5
-3	-5

Technique par envoi de messages



Carole:

q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

y_{bc}	v
0	0
2	5.5
3	8

y_{dc}	v
2	2
1	1.25
0	0
-2	-4
-3	-6

Bob:

q	v
0	0
4	9
5	11.5

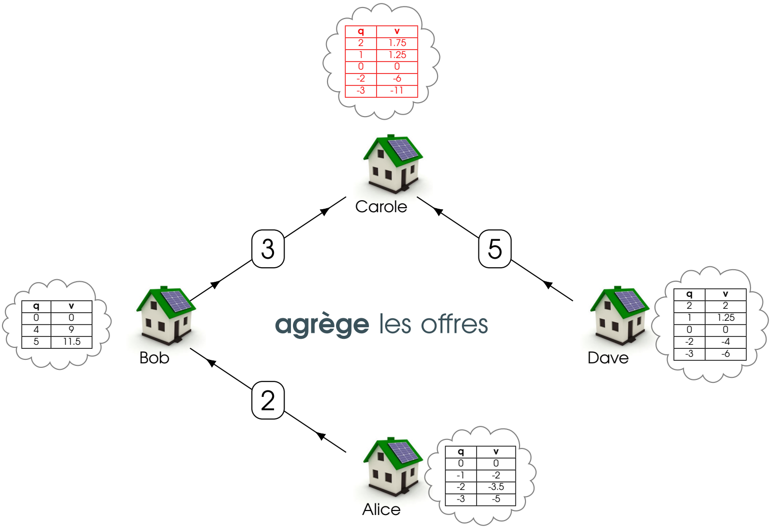
Dave:

q	v
2	2
1	1.25
0	0
-2	-4
-3	-6

Alice:

q	v
0	0
-1	-2
-2	-3.5
-3	-5

Technique par envoi de messages



Carole:

q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

Bob:

q	v
0	0
4	9
5	11.5

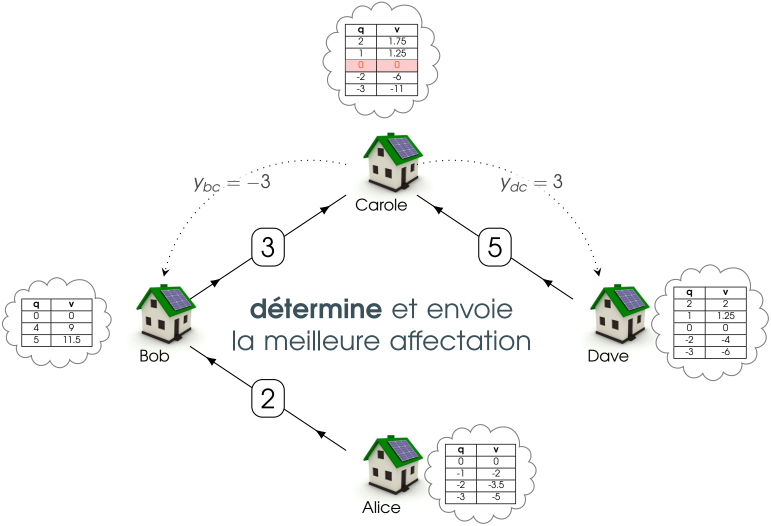
Dave:

q	v
2	2
1	1.25
0	0
-2	-4
-3	-6

Alice:

q	v
0	0
-1	-2
-2	-3.5
-3	-5

Technique par envoi de messages



Carole:

q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

Bob:

q	v
0	0
4	9
5	11.5

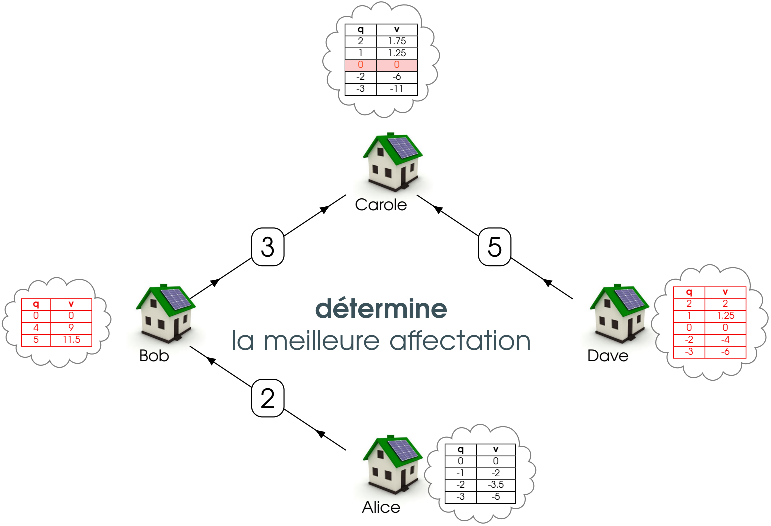
Dave:

q	v
2	2
1	1.25
0	0
-2	-4
-3	-6

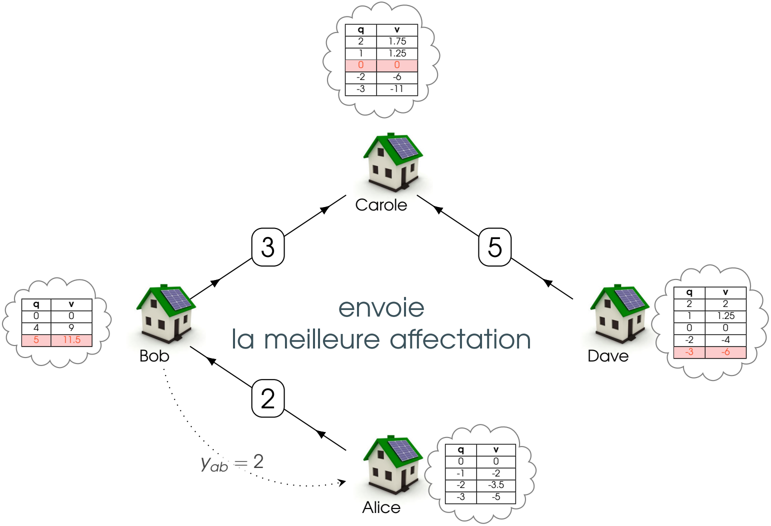
Alice:

q	v
0	0
-1	-2
-2	-3.5
-3	-5

Technique par envoi de messages



Technique par envoi de messages



Carole :

q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

Bob :

q	v
0	0
4	9
5	11.5

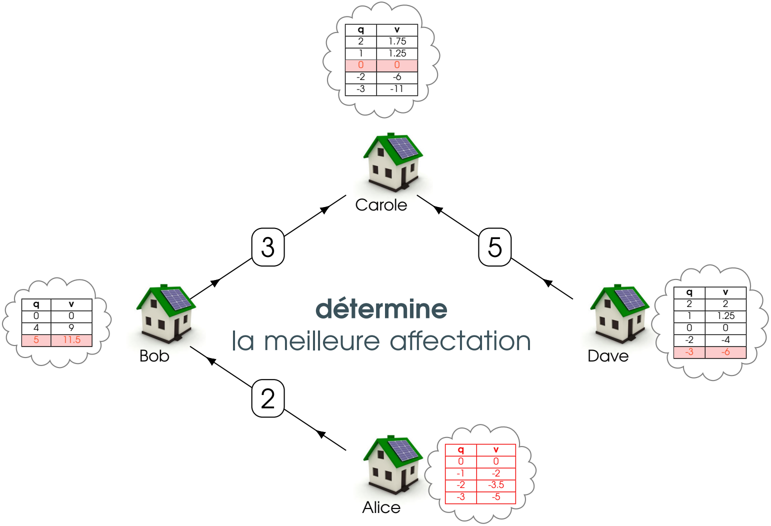
Dave :

q	v
2	2
1	1.25
0	0
-2	-4
-3	-6

Alice :

q	v
0	0
-1	-2
-2	-3.5
-3	-5

Technique par envoi de messages



Carole:

q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

Bob:

q	v
0	0
4	9
5	11.5

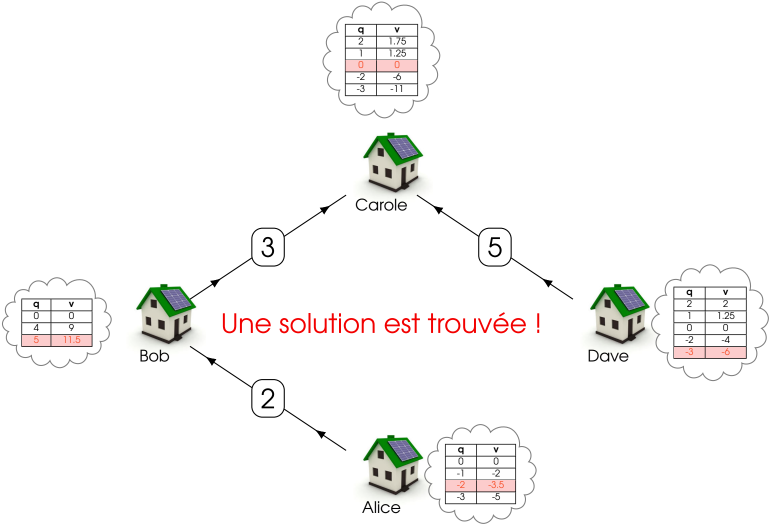
Dave:

q	v
2	2
1	1.25
0	0
-2	-4
-3	-6

Alice:

q	v
0	0
-1	-2
-2	-3.5
-3	-5

Technique par envoi de messages



Carole:

q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

Bob:

q	v
0	0
4	9
5	11.5

Dave:

q	v
2	2
1	1.25
0	0
-2	-4
-3	-6

Alice:

q	v
0	0
-1	-2
-2	-3.5
-3	-5

Calcul des messages

$$\mu_{j \to p_j}(\mathbf{y}_{jp_j}) = \max_{\mathbf{Y}_{j-p_j}} \left(v_j(\mathbf{y}_{jp_j}, \mathbf{Y}_{j-p_j}) + \sum_{k \in out(j) \setminus \{p_j\}} \mu_{j \to k}(\mathbf{y}_{jk}) + \sum_{i \in in(j)} \mu_{i \to j}(\mathbf{y}_{ij}) \right)$$

- C'est le point dur calculatoire
- Calculé en $\mathcal{O}((2C_j + 1)^{N_j})$
  - C_j est la capacité du lien le plus puissant
  - N_j est le nombre de voisins de j

$\Rightarrow$ Non applicable à des réseaux denses

Calcul des messages

$$\mu_{j\to p_j}(\mathbf{y}_{jp_j}) = \max_{\mathbf{Y}_{j-p_j}} \left(v_j(\mathbf{y}_{jp_j}, \mathbf{Y}_{j-p_j}) + \sum_{k\in out(j)\setminus\{p_j\}} \mu_{j\to k}(\mathbf{y}_{jk}) + \sum_{i\in in(j)} \mu_{i\to j}(\mathbf{y}_{ij}) \right)$$

- C'est le point dur calculatoire
- Calculé en $\mathcal{O}((2C_j + 1)^{N_j})$
  - C_j est la capacité du lien le plus puissant
  - N_j est le nombre de voisins de j

$\Rightarrow$ Non applicable à des réseaux denses

$\Rightarrow$ Calculer les messages plus efficacement !

Algèbre des *valuations*

- Exploiter une particularité des messages : **capacité bornée**
- Reformuler le calcul des messages grâce à trois opérations de base :
  - Restriction (linéaire) : $\alpha[D](k) = \begin{cases} \alpha(k) & k \in D \\ -\infty & \text{sinon} \end{cases}$
  - Complément (linéaire) : $\overline{\alpha}(k) = \alpha(-k)$
  - Agrégation (polynomiale) : $(\alpha \cdot \beta)(k) = \max_{\substack{i,j \\ k=i+j}} \alpha(i) + \beta(j)$

$$\mu_{j\to p_j} = \left(\overline{O_j} \cdot \prod_{k \in out(j)\setminus\{p_j\}} \overline{\mu_{j\to k}} \cdot \prod_{i \in in(j)} \mu_{i\to j} \right) [-D_{jp_j}]$$

L'algorithme RADPRO

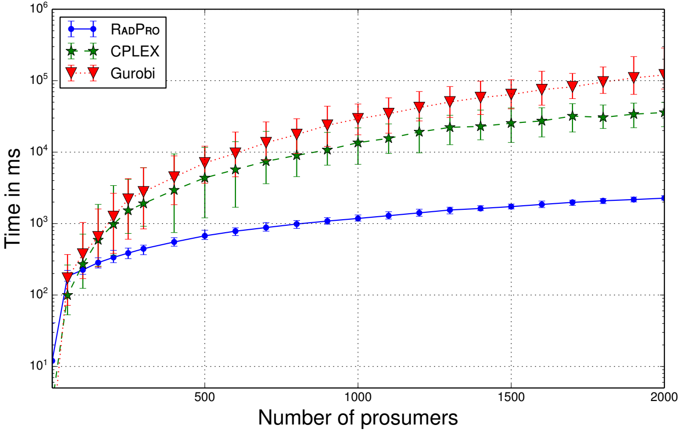
= ACYCLIC-SOLVING + calcul efficace des messages

- Complexité globale du calcul des messages : **polynomiale** en $\mathcal{O}(nN_{max}^2 C_{max}^2)$
  - Nombre de messages calculés en $\mathcal{O}(n(2C_{max}+1)^{N_{max}})$
  - Calcul d'un message en $\mathcal{O}(N_j C_j n_{o_j} + N_j^2 C_j^2)$
- Complexité communicationnelle : **linéaire** en $\mathcal{O}(nC_{max})$
  - $2n$ messages de taille max $2C_{max}+1$
- Facilement distribuable

	ACYCLIC-SOLVING	RADPRO
Messages	$\mathcal{O}(nC_{max})$	$\mathcal{O}(nC_{max})$
Temps	$\mathcal{O}(n(2C_{max}+1)^{N_{max}})$	$\mathcal{O}(nN_{max}^2 C_{max}^2)$

Comparaison avec les solveurs MIP

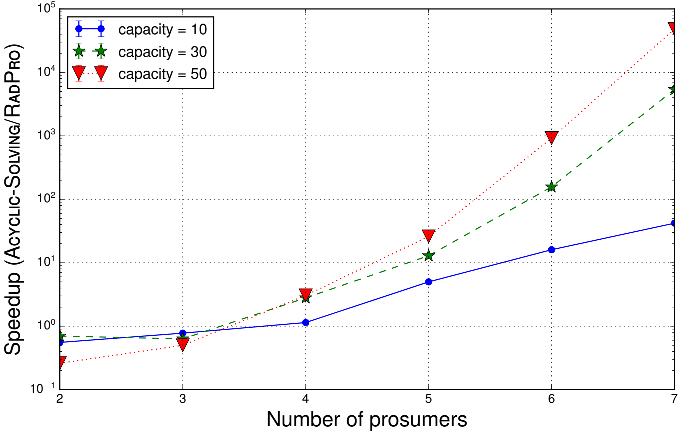
RADPRO surclasse CPLEX & Gurobi
(plus rapide de plus d'un ordre de grandeur que CPLEX !)



Réseaux aléatoires (distribution géométrique) avec $C_j = \mathcal{N}(100, 50)$

Efficacité du calcul des messages

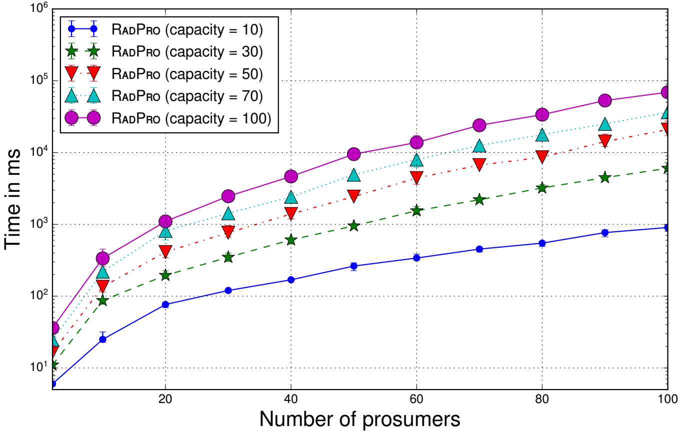
RADPRO surclasse ACYCLIC-SOLVING
(plus rapide de plus de trois ordres de grandeur qu'ACYCLIC-SOLVING !)



Réseaux en étoile avec $C_j \in \{10, 30, 50\}$ and $N_{max} \in [1..6]$

Passage à l'échelle

RADPRO résout de très grands EAP avec de grands facteurs de branchement
(résolution de problèmes avec capacité 100 et 100 voisins en moins d'une minute !)



Réseaux en étoile avec $C_j \in \{10, 30, 50, 70, 100\}$ et $N_{max} \in [1..99]$

Résumé des contributions

- **Problème d'allocation d'énergie (EAP)**
  - EAP est formulé dans le cadre des **DCOP**
  - EAP **généralise** le problème de flux d'énergie optimal (OPF)
- **Algèbre des *valuations***
  - Utilisée pour mettre en œuvre les **messages** et les **offres**
  - Opérations **efficaces** (agrégation, négation, sélection)
- **Algorithm RADPRO**
  - Basé sur la programmation dynamique pour résoudre des EAP **acycliques**
  - **Surclasse** les solveurs classiques et basés sur la programmation dynamique
  - Repose sur l'algèbre des *valuations* pour un **calcul des messages polynomial**

Perspectives

- **Prochaines étapes de RadPro**
  - Gérer la *fréquence des mises à jours* des offres
  - Gérer les réseaux **cycliques**
  - Étendre RadPro aux EAP *continus* : C-RadPro
- **Futurs usage d'approches à la RadPro**
  - Les marchés décentralisés vus comme **alternatives aux enchères journalières**
  - **Impact socio-économique** de la décentralisation (European Technology Platform, 2012)

CONNECTED
INTELLIGENCE

https://connected-intelligence.univ-st-etienne.fr

Références

- GONEN, Turan (2014). *Electric power distribution engineering*. CRC press.
- MILLER, Samuel John Odell (2014). "Decentralised Coordination of Smart Distribution Networks using Message Passing". Thèse de doct. University of Southhampton.
- EUROPEAN TECHNOLOGY PLATFORM (2012). *SmartGrids SRA 2035. Strategic Research Agenda. Update of the SmartGrids SRA 2007 for the needs by the year 2035*.
- KUMAR, Akshat, Boi FALTINGS et Adrian PETCU (2009). "Distributed Constraint Optimization with Structured Resource Constraints". In : *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*. AAMAS '09. Budapest, Hungary : International Foundation for Autonomous Agents et Multiagent Systems, p. 923–930. ISBN : 978-0-9817381-7-8. URL : http://dl.acm.org/citation.cfm?id=1558109.1558140.
- DECHTER, R (2003). *Constraint processing*. Morgan Kauffman. URL : http://books.google.com/books?hl=en\&lr=\&id=w4LG4EUOBCwC\&oi=fnd\&pg=PP2\&dq=Constraint+processing\&ots=ur_5y38Tbs\&sig=la9V-uFZ0kGza4iD4HM11F5-1Bo.

Résoudre EAP par la programmation en nombres entiers

$$
\begin{array}{rrl}
\textbf{maximiser} & \sum_{j=1}^{|P|} \sum_{l=1}^{n_{o_j}} x_j^l \cdot o_j^l & \\
\textbf{avec} & \sum_{l=1}^{n_{o_j}} x_j^l = 1 & \forall j \in P, 1 \leq l \leq n_{o_j} \\
& \sum_{i<j} y_{ij} - \sum_{k>j} y_{jk} = \sum_{l=1}^{n_{o_j}} x_j^l \cdot q_j^l & \forall j \in P \\
& y_{ij} \in D_{ij} & \forall (i,j) \in E \\
& x_j^l \in \{0, 1\} & \forall j \in P, 1 \leq l \leq n_{o_j}
\end{array}
$$

x_j^l : j vend l unités

n_{o_j} : nombre max d'unités dans l'offre

y_{ij} : nombre d'unités échangées entre i et j

$D_{ij} = [-c_{ij} .. c_{ij}]$: domaine de c_{ij}

C-RADPRO : extension aux offres continues

C-RADPRO surclasse CPLEX & Gurobi
(plus rapide de plus deux ordres de grandeur que CPLEX !)