Designing a Marketplace for the Trading and Distribution of Energy in the Smart Grid

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

[1]IIIA-CSIC

[2]LaHC UMR CNRS 5516, MINES Saint-Étienne

Smart grids: promises & expected outcomes

- New distribution rationale: **decentralized production**
  - Democratization of decentralized production: local balancing and reducing energy loss
- New context information: **energy awareness**
  - Frequently sensed data (consumption, production, pricing) impacts trading updates
- New trading rationale: **prosumption**

Smart grids: promises & expected outcomes

- New distribution rationale: **decentralized production**
  - Democratization of decentralized production: local balancing and reducing energy loss
- New context information: **energy awareness**
  - Frequently sensed data (consumption, production, pricing) impacts trading updates
- New trading rationale: **prosumption**

How to design a decentralized market for the trading and distribution of energy?

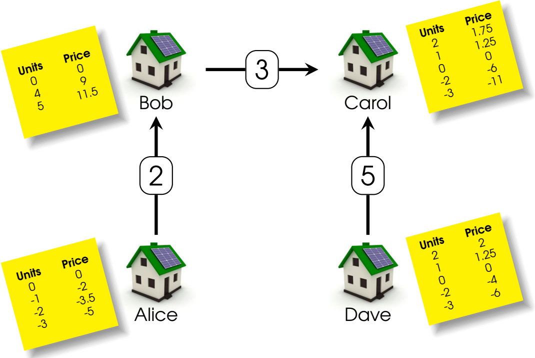
Example: energy trading scenario

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

Example: energy trading scenario

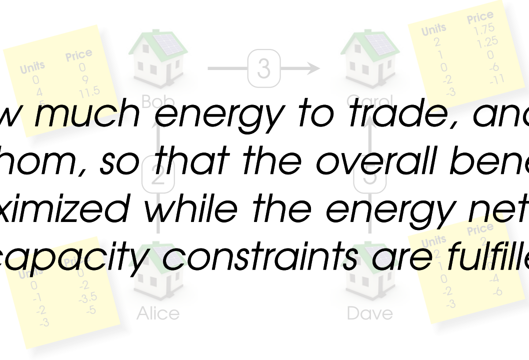
- *Prosumers* ($j \in P$)
- *Offers* ($o_j : \mathbb{Z} \rightarrow \mathbb{R} \cup \{-\infty\}$)

Example: energy trading scenario



- *Prosumers* ($j \in P$)
- *Offers* ($o_j : \mathbb{Z} \to \mathbb{R} \cup \{-\infty\}$)
- *Links* ($\{i, j\}$) w/ some max capacity (c_{ij})

Example: energy trading scenario



How much energy to trade, and with whom, so that the overall benefit is maximized while the energy network's capacity constraints are fulfilled?

Definition: energy allocation problem

The energy allocation problem (*EAP*) amounts to finding an allocation $\mathbf{Y}$ that maximizes the overall benefit $Value(\mathbf{Y})$, with

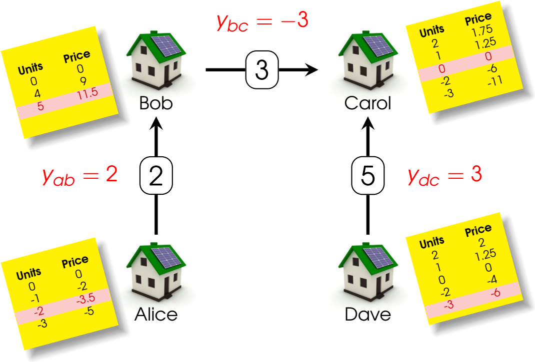
$$Value(\mathbf{Y}) = \sum_{i \in P} v_j(\mathbf{Y}_j)$$

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

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

where $\mathbf{y}_{ij}$ stands for the number of units that prosumer i sells to prosumer j (bounded by c_{ij})

Example: energy trading scenario (solution)



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

Distributed allocation techniques

- Market-based
  - **Double auction** (call market or CDA) where energy is traded on a day-ahead basis
  - Matching between supply and demand computed by **central** authority
  - Current market mechanisms disregard grid constraints $\rightarrow$ **Trading and distribution as decoupled activities**
- Message passing
  - Dynamic programming (Miller, 2014; Kumar et al., 2009)
  - Belief-propagation (Miller, 2014)

Our contribution

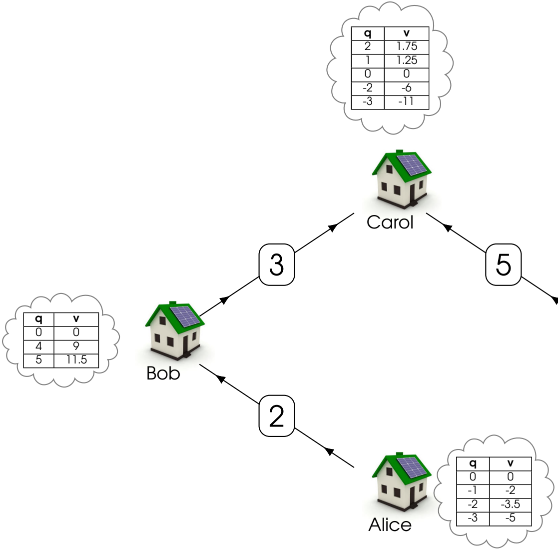
- Exploit the **tree** structure of energy networks (GONEN, 2014)
- Solve EAP as a distributed constraint optimization problem (**DCOP**)
- Design an **exact message passing** algorithm based on dynamic programming
  - ACYCLIC-SOLVING (DECHTER, 2003)
- Assess efficiently messages by exploiting the algebraic structure of offers and messages : **valuations**

Message passing solution

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

Carol

Message passing solution



Carol:

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

Bob:

q	v
0	0
4	9
5	11.5

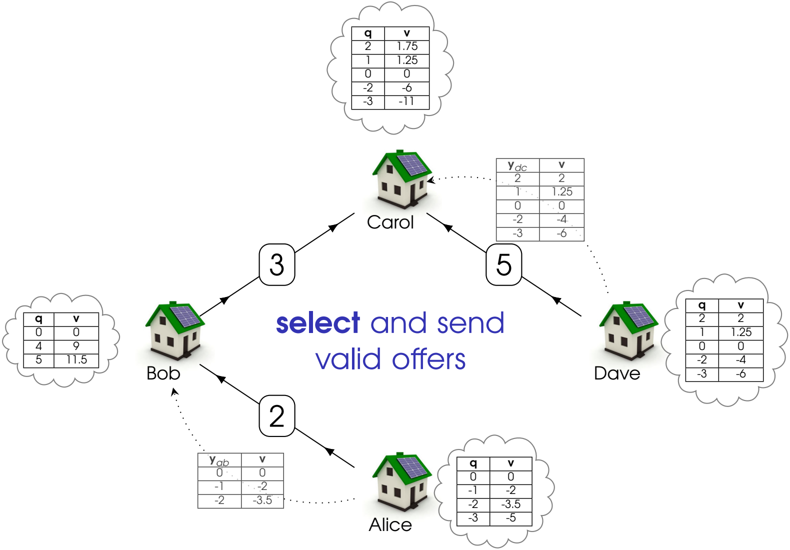
Dave:

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

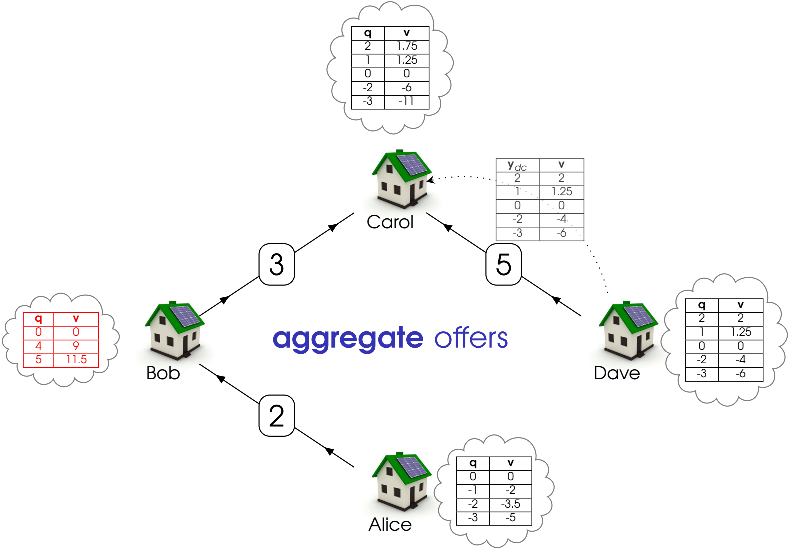
Alice:

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

Message passing solution



Message passing solution



Carol:

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

Message from Dave to Carol:

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

Bob:

q	v
0	0
4	9
5	11.5

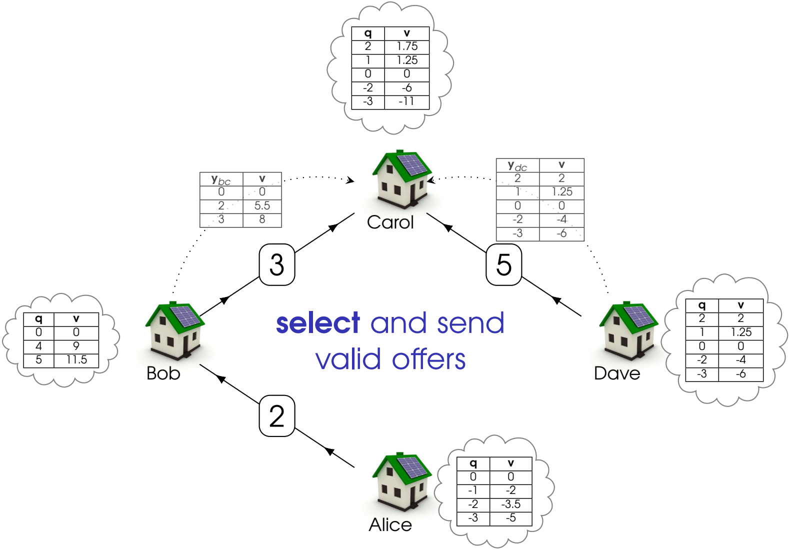
Dave:

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

Message passing solution



Carol:

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

Bob → Carol message:

y_{bc}	v
0	0
2	5.5
3	8

Dave → Carol message:

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

Bob:

q	v
0	0
4	9
5	11.5

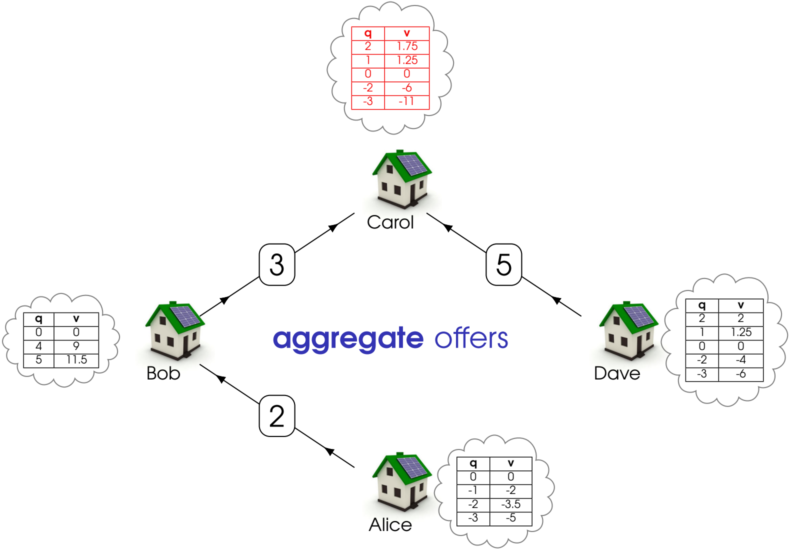
Dave:

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

Message passing solution



Carol:

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

Bob:

q	v
0	0
4	9
5	11.5

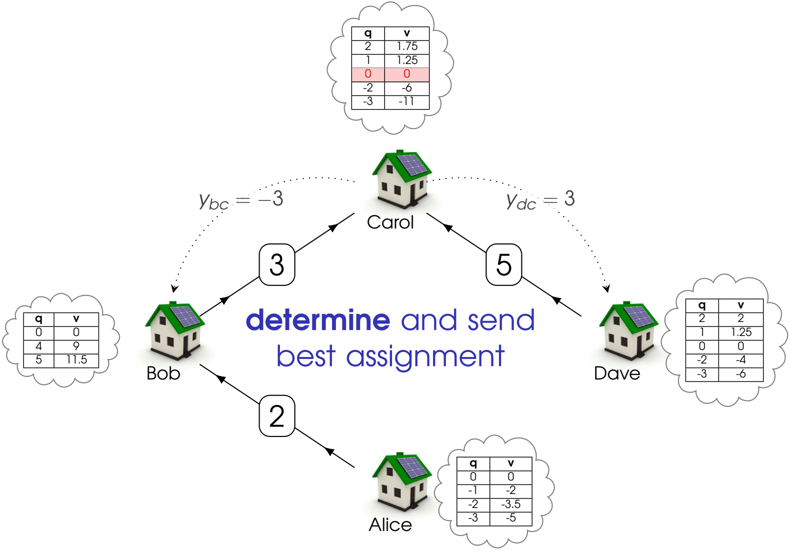
Dave:

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

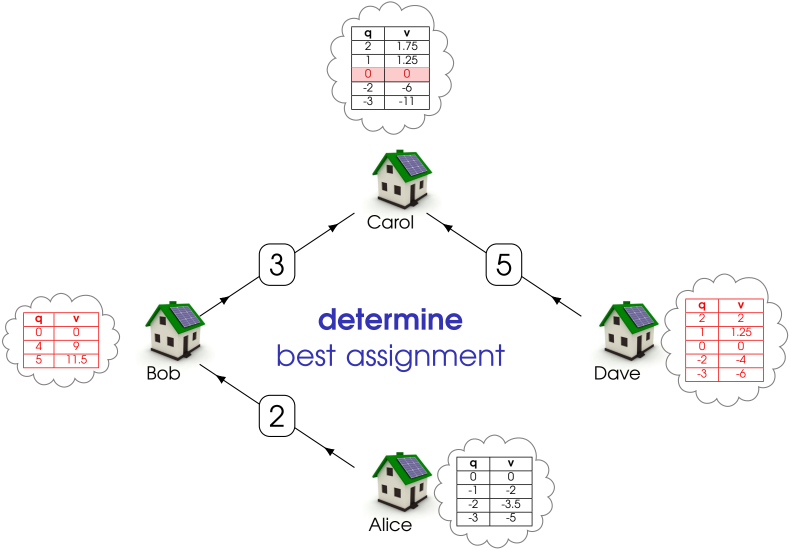
Alice:

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

Message passing solution



Message passing solution



Carol:

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

Bob:

q	v
0	0
4	9
5	11.5

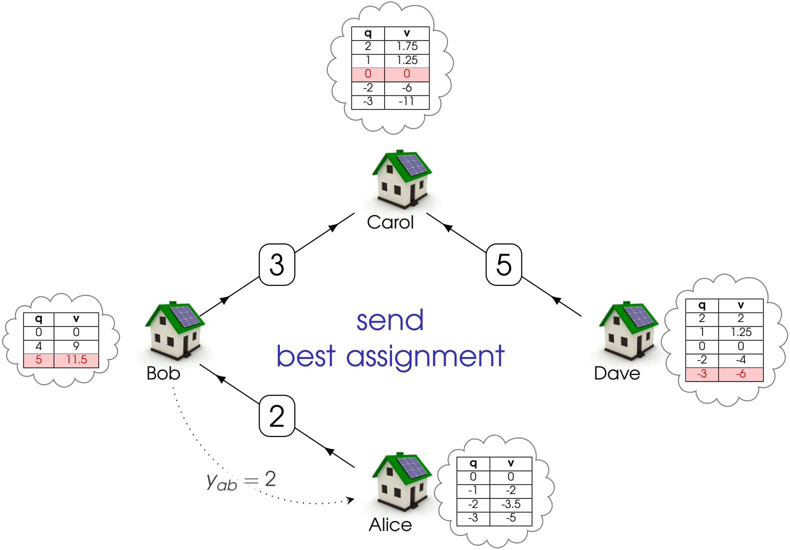
Dave:

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

Message passing solution



Carol:

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

Bob:

q	v
0	0
4	9
5	11.5

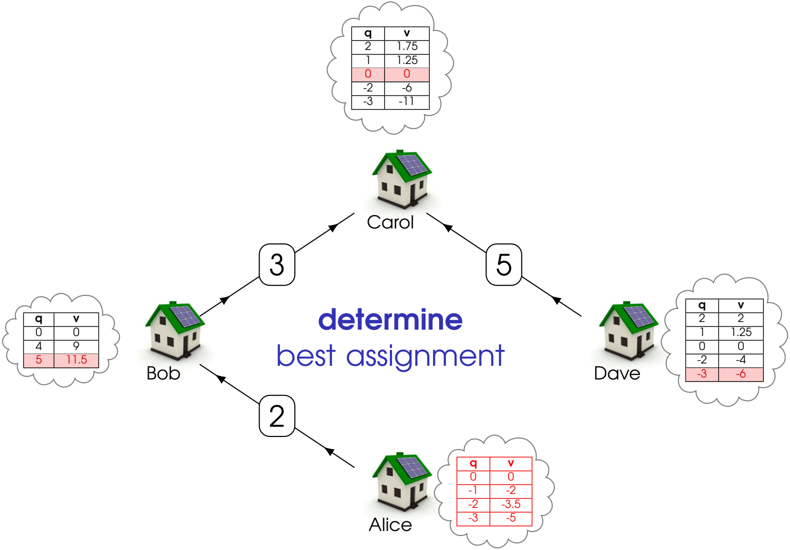
Dave:

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

Message passing solution



Carol:

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

Bob:

q	v
0	0
4	9
5	11.5

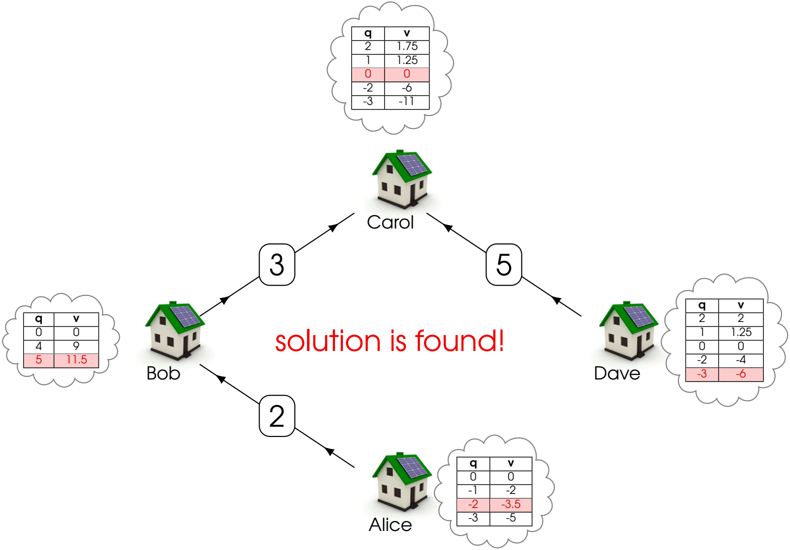
Dave:

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

Message passing solution



Carol:

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

Message assessment

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

- This is the computational hard point
- Computed in $\mathcal{O}((2C_j + 1)^{N_j})$
  - C_j is the capacity of the most powerful link
  - N_j is the number of neighbors of j

$\Rightarrow$ Not applicable to dense networks

Message assessment

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

- This is the computational hard point
- Computed in $\mathcal{O}((2C_j + 1)^{N_j})$
  - C_j is the capacity of the most powerful link
  - N_j is the number of neighbors of j

$\Rightarrow$ Not applicable to dense networks

Assess messages more efficiently!

Algebra of valuations

- Take advantage of a particularity of the messages: **restricted capacity**
- Reformulate message assessment with 3 operations:
  - Restriction (linear): $\alpha[D](k) = \begin{cases} \alpha(k) & k \in D \\ -\infty & \text{otherwise} \end{cases}$
  - Complement (linear): $\overline{\alpha}(k) = \alpha(-k)$
  - Aggregation (polynomial): $(\alpha \cdot \beta)(k) = \max_{\substack{i,j \\ k=i+j}} \alpha(i) + \beta(j)$

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

RADPRO algorithm

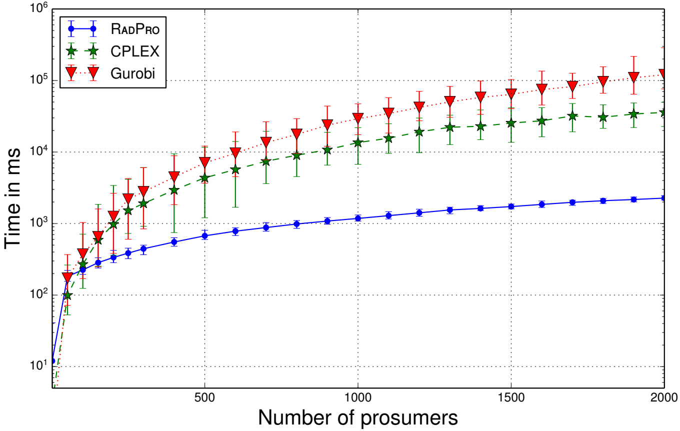
= ACYCLIC-SOLVING + efficient message assessment

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

- Global complexity of message assessment: **polynomial** in $\mathcal{O}(nN_{max}^2 C_{max}^2)$
  - Number of message assessments in $\mathcal{O}(n)$
  - Single message assessment in $\mathcal{O}(N_j C_j n_{o_j} + N_j^2 C_j^2)$
- Communication complexity: **linear** in $\mathcal{O}(nC_{max})$
  - $2n$ messages of max size $2C_{max}+1$
- Easily distributable

Comparison with MIP solvers

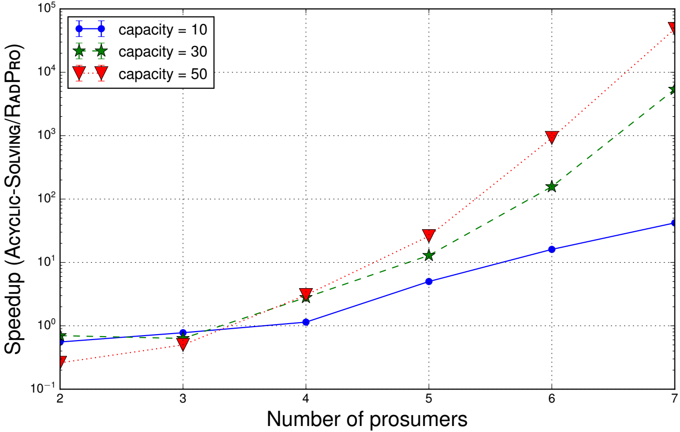
RadPro outperforms CPLEX & Gurobi (more than one order of magnitude faster than CPLEX!)



Random networks (geometric distribution) with $C_j = \mathcal{N}(100, 50)$

Efficiency of message assessment

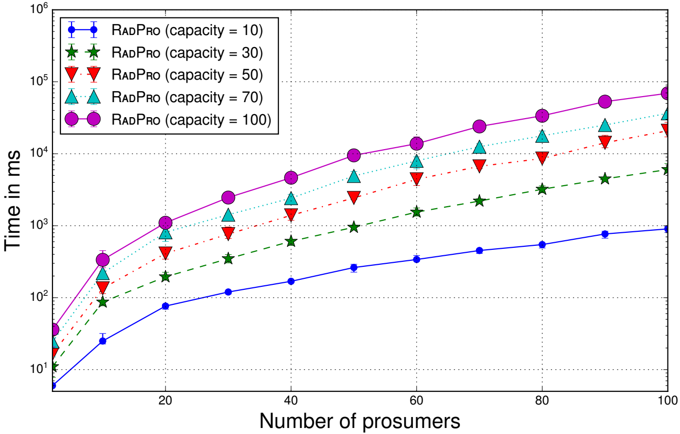
RADPRO outperforms ACYCLIC-SOLVING
(more than three orders of magnitude faster than ACYCLIC-SOLVING!)



Star-shaped networks (hubs) with $C_j \in \{10, 30, 50\}$ and $N_{max} \in [1..6]$

Scalability

RADPRO solves large-scale EAP with high branching factor (solving problems with capacity 100 and 100 neighbors in less than 1 min!)



Star-shaped networks (hubs) with $C_j \in \{10, 30, 50, 70, 100\}$ and $N_{max} \in [1..99]$

Summary of the contribution

- **Energy Allocation Problem**
  - EAP is formulated as a **DCOP**
  - EAP **generalizes** Optimal Power Flow Problem (OPF)
- **Valuation algebra**
  - Used to implement **messages** and **offers**
  - **Efficient** operations (aggregation, negation, selection)
- **RADPRO algorithm**
  - Based on dynamic programming to solve **acyclic** EAP
  - **Outperforms** MIP and DP-based solvers
  - Based on the valuation algebra for **polynomial message computation**

Perspectives

- **Next steps of RADPRO**
  - Handle **high frequency** of incoming offers
  - Cope with **cyclic** networks
  - Extend RADPRO to **continuous** EAP
- **Potential future uses of RADPRO**
  - Decentralized markets as **alternatives to daily auctions**
  - **Socio-economical impact** of decentralized markets (EUROPEAN TECHNOLOGY PLATFORM, 2012)

References (contd)

GONEN, Turan (2014). *Electric power distribution engineering.* CRC press.

MILLER, Samuel John Odell (2014). "Decentralised Coordination of Smart Distribution Networks using Message Passing". PhD thesis. 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, and 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 and Multiagent Systems, pp. 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.

Solving EAP through MIP

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

x_j^l : j sells l units

n_{o_j} : maximum number of units in the offer

y_{ij} : number of units exchanged between i and j

$D_{ij} = [-c_{ij} \,..\, c_{ij}]$: domain of c_{ij}