Defining a Continuous 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?

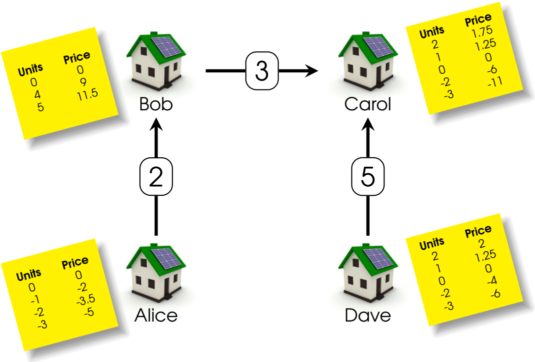
Example: energy trading scenario

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

Example: energy trading scenario

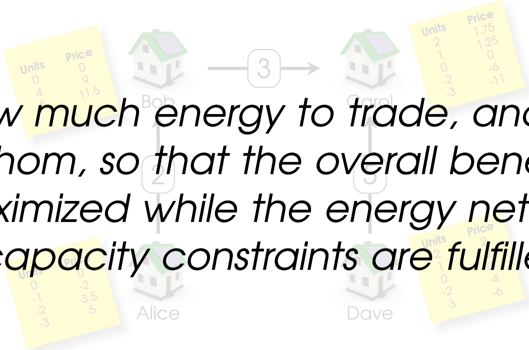
- *Prosumers* ($j \in P$)
- *Offers* ($o_j : \mathbb{Z} \rightarrow \mathbb{R} \cup \{-\infty\}$)

Example: energy trading scenario



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

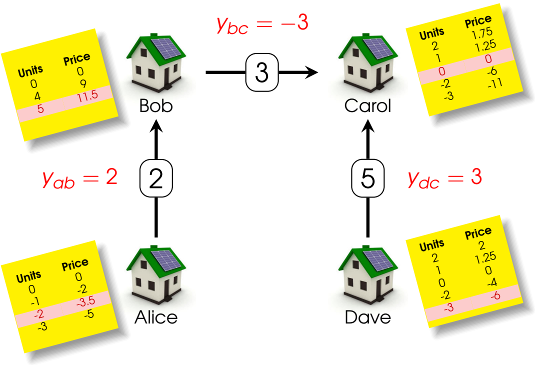
$$Value(\mathbf{Y}) = \sum_{j \in P} v_j(\mathbf{Y}_j)$$

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

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

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)



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

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 → **Trading and distribution as decoupled activities**
- Message passing
  - Dynamic programming (MILLER, 2014; KUMAR et al., 2009)
  - Belief-propagation (MILLER, 2014)

Our contribution

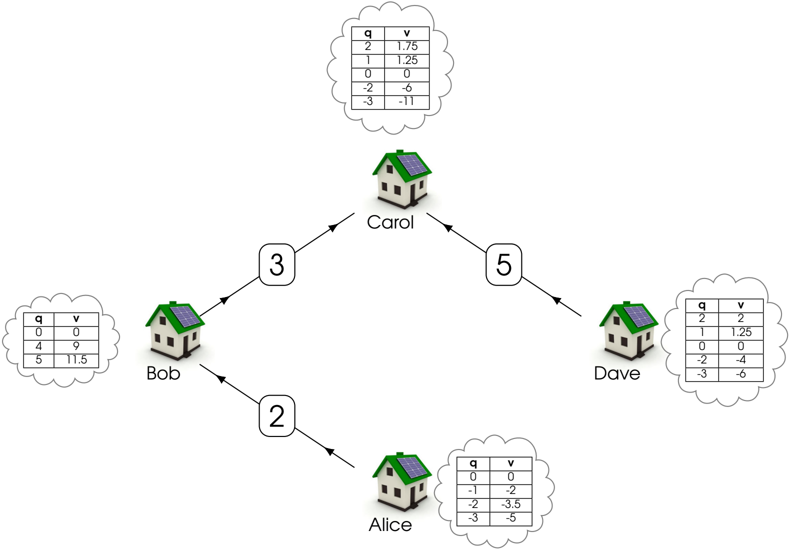
- Exploit the **tree** structure of energy networks (GONEN, 2014)
- Solve EAP as a distributed contraint optimization problem (**DCOP**)
- Design an **exact message passing** algorithm based on dynamic programming
  - ACYCLIC-SOLVING (DECHTER, 2003)
- Assess efficiently messages by exploiting the algebraic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

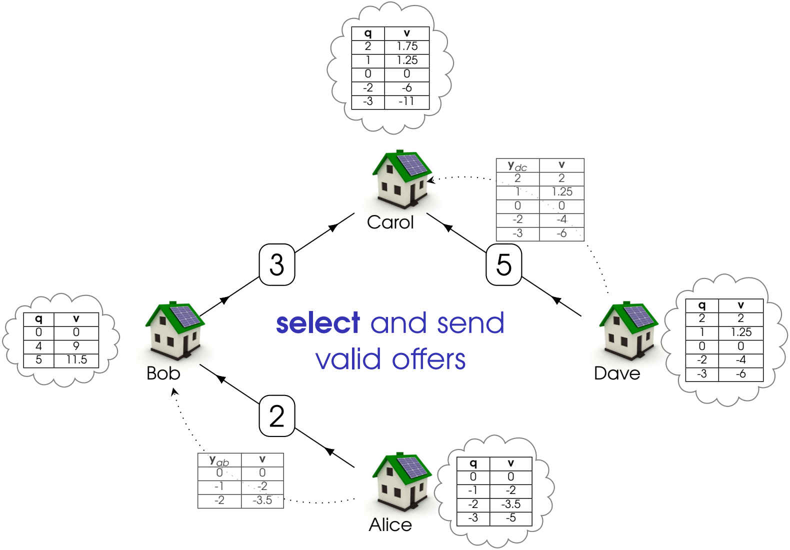
Dave:

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

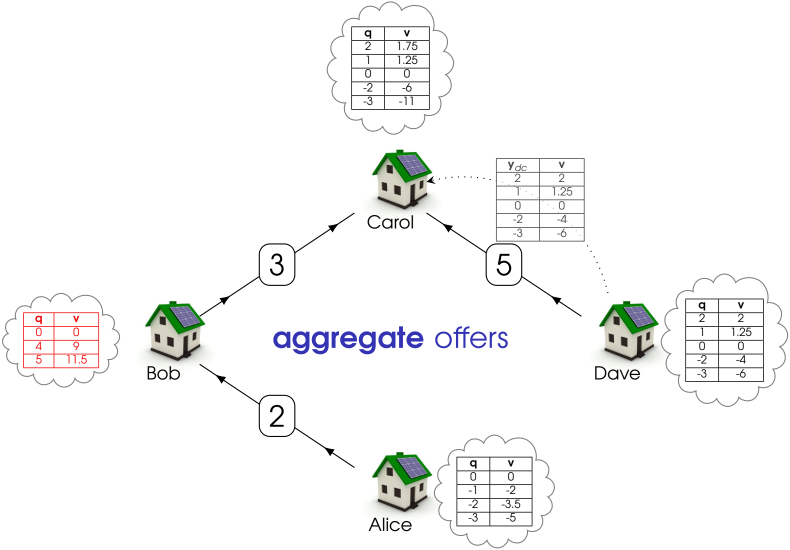
Alice:

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 5:

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

Bob:

q	v
0	0
4	9
5	11.5

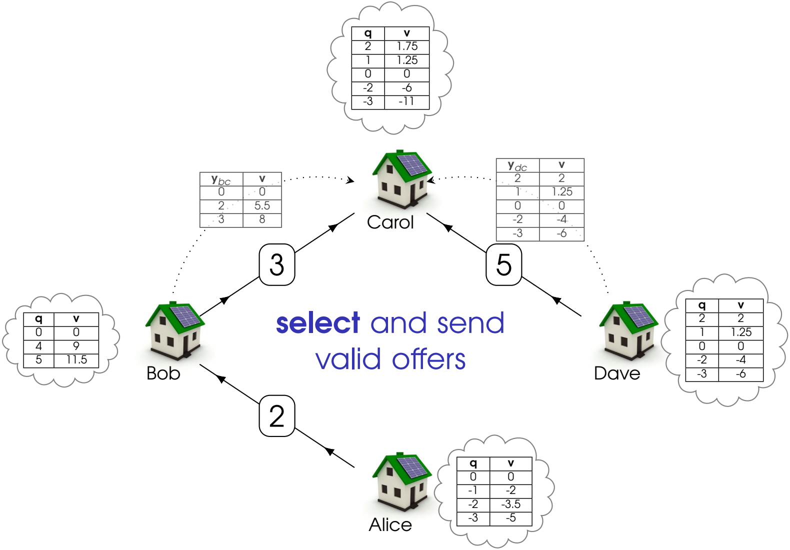
Dave:

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

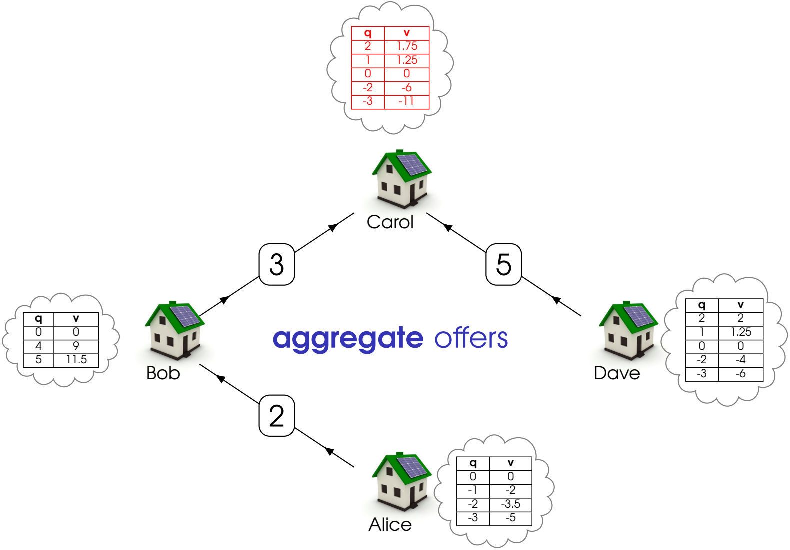
Alice:

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

Message passing solution



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

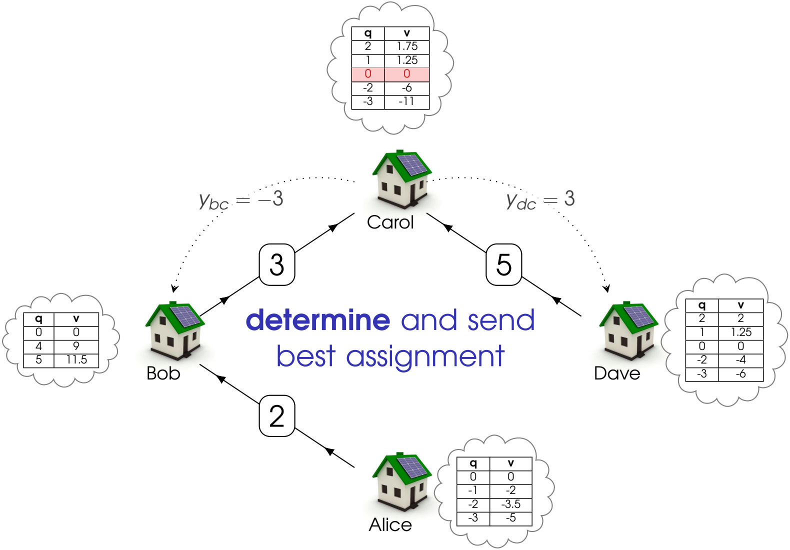
Dave:

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



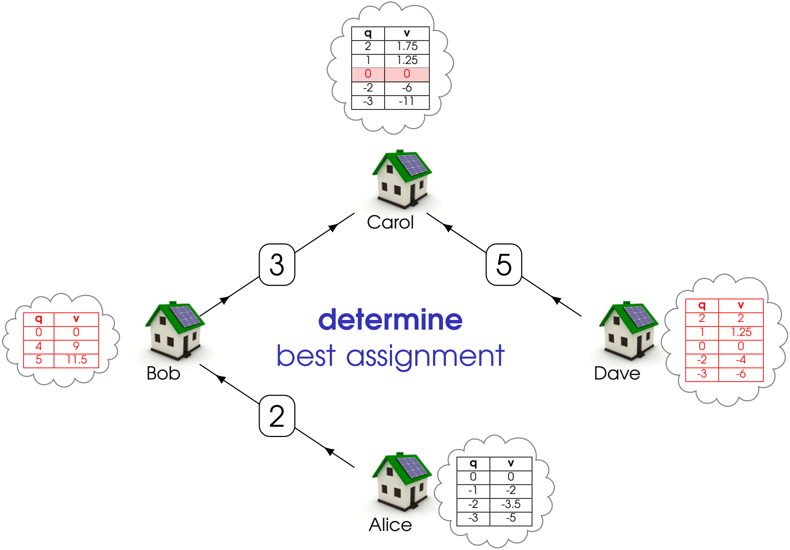
q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

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

Message passing solution



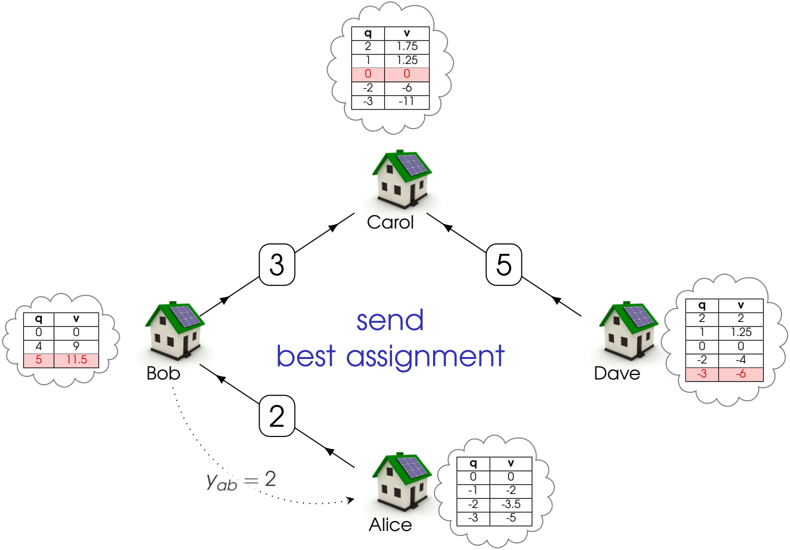
q	v
2	1.75
1	1.25
0	0
-2	-6
-3	-11

q	v
0	0
4	9
5	11.5

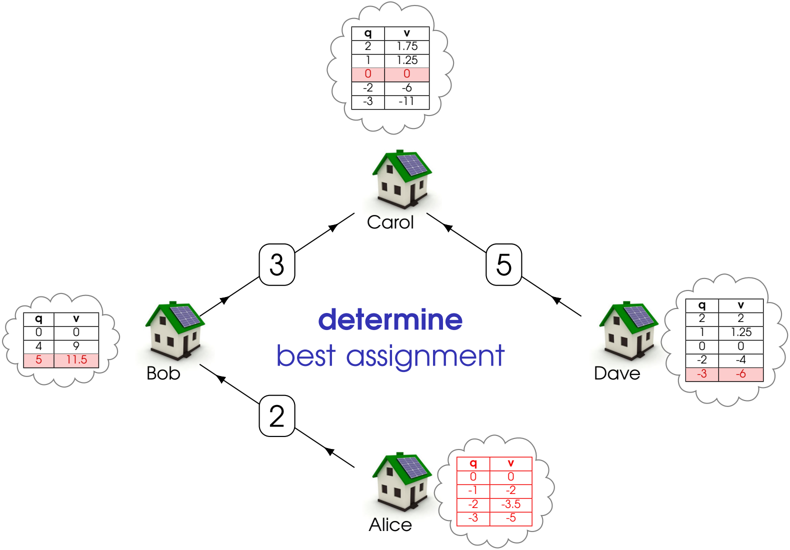
q	v
2	2
1	1.25
0	0
-2	-4
-3	-6

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

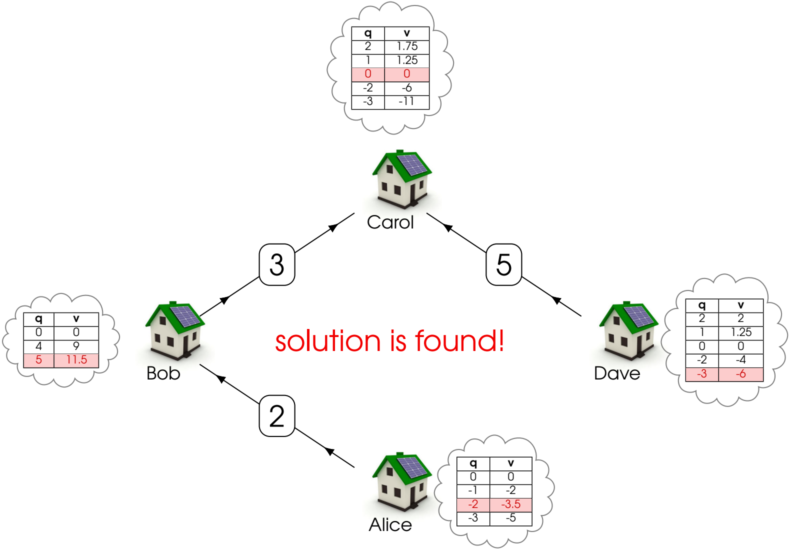
Message passing solution



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 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

$\Rightarrow$ Assess message 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

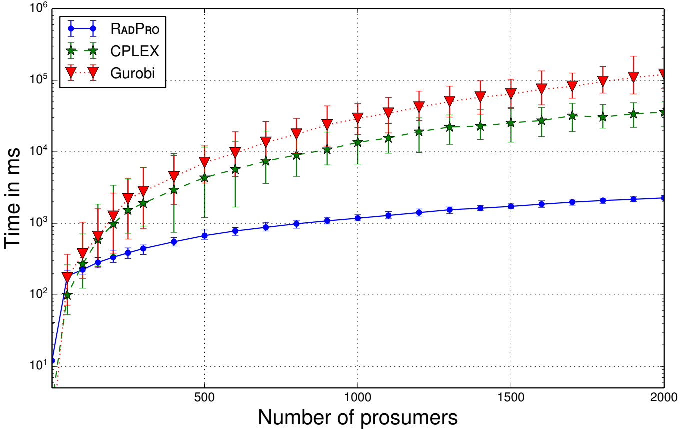
= Acyclic-Solving + efficient message assessment

- Global complexity of message assessment: **polynomial** in $\mathcal{O}(nN_{max}^2 C_{max}^2)$
  - Number of message assessments in $\mathcal{O}(n(2C_{max}+1)^{N_{max}})$
  - 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

	Acyclic-Solving	RadPro
Communication	$\mathcal{O}(nC_{max})$	$\mathcal{O}(nC_{max})$
Computation	$\mathcal{O}(n(2C_{max}+1)^{N_{max}})$	$\mathcal{O}(nN_{max}^2 C_{max}^2)$

Comparison with MIP solvers

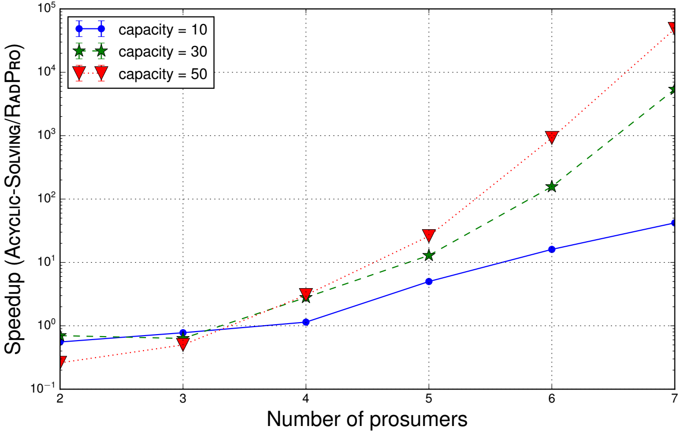
RadPro outperforms CPLEX & Gurobi
(more than one order of magnitude faster than CPLEX!)



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

Efficiency of message assessment

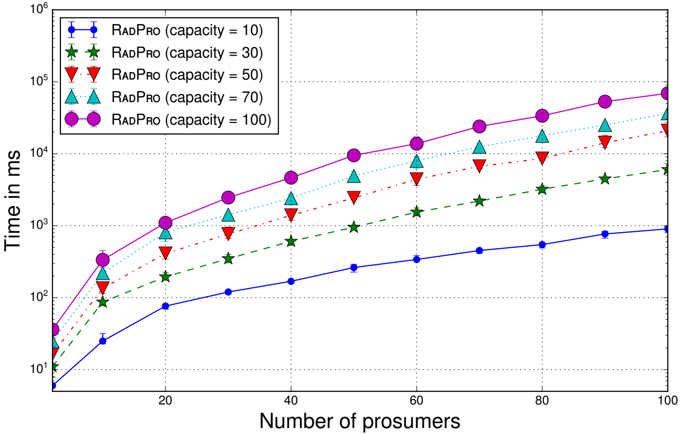
RADPRO outperforms ACYCLIC-SOLVING
(more than three orders of magnitude faster than ACYCLIC-SOLVING!)



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

Scalabity

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

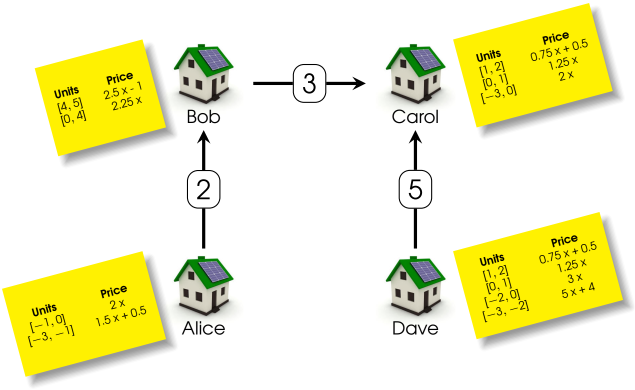


Star-shaped networks (hubs) with $C_j \in \{10, 30, 50, 70, 100\}$ and $N_{max} \in [1..99]$

RadPro Limitations

- Offers are discrete: a prosumer can offer to buy either 3 KW for 6 EUR or 2 KW for 4 EUR, but not any amount of energy between 2 KW and 3 KW and pay 2 EUR per KW.
- Such offers provide a better representation of prosumers' preferences.
- **Our next goal is to extend the EAP to allow prosumers to communicate continuous (piecewise linear) utility functions.**

Example: continuous energy trading scenario



- *Prosumers* ($j \in P$)
- *Offers* as **piecewise linear valuations**
- *Links* ($\{i, j\}$) w/ some max capacity (c_{ij})

Definition: continuous energy allocation problem

Given a set of prosumers P whose offers are piecewise linear valuations, the **continuous energy allocation problem** (*CEAP*) 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})

Mapping the CEAP to a Linear Program

Decision variables per prosumer:

- interval valuation to select within a piecewise linear valuation
- amount of energy to trade within the chosen interval

Mapping the CEAP to a Linear Program

Decision variables per prosumer:

- interval valuation to select within a piecewise linear valuation
- amount of energy to trade within the chosen interval

Decision variable per link:

- amount of energy to trade between prosumers

Mapping the CEAP to a Linear Program

Decision variables per prosumer:

- interval valuation to select within a piecewise linear valuation
- amount of energy to trade within the chosen interval

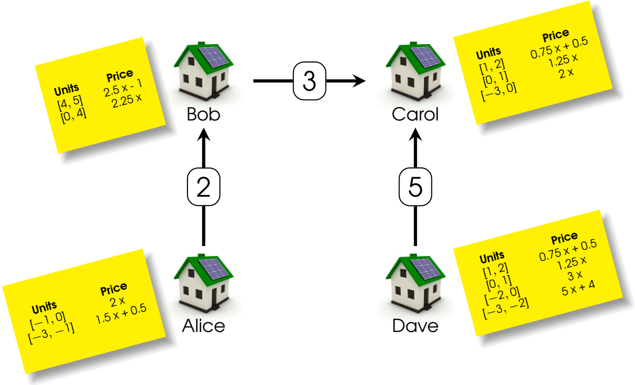
Decision variable per link:

- amount of energy to trade between prosumers

Constraints:

- *Mutually exclusive intervals:* only an interval valuation per piecewise linear valuation
- *Energy balance:* amount of energy to trade per prosumer equals difference between input and output energy
- *Network capacity:* energy traded between prosumers respects links' capacities

Example: continuous energy trading scenario



- *Prosumers* ($j \in P$)
- *Offers* as **piecewise linear valuations**
- *Links* ($\{i, j\}$) w/ some max capacity (c_{ij})

Mapping the CEAP to a Linear Program

LP that solves the continuous energy allocation problem:

$$\textbf{maximize} \quad \sum_{j=1}^{|P|} \sum_{k=1}^{|W_j|} a_{o_j^k} \cdot x_j^k + b_{o_j^k} \cdot z_j^k$$

$$\textbf{subject to} \quad z_j^k \cdot l_{o_j^k} \leq x_j^k \leq z_j^k \cdot u_{o_j^k}$$

$$\sum_{l=k}^{|W_j|} z_j^k = 1$$

$$\sum_{i<j} y_{ij} - \sum_{q>j} y_{jq} = \sum_{k=1}^{|W_j|} x_j^k$$

$$y_{ij} \in D_{ij}$$

$$z_j^k \in \{0, 1\}$$

$$x_j^k \in \mathbb{R}$$

$$\forall j \in P, 1 \leq k \leq |W_j|, \forall (i, j) \in E$$

CEAP: Current state

- Completed
  - Implementation of a MIP solver for the CEAP based on our LP mapping.
  - Extension of RadPro to provide a decentralised solver for the acyclic CEAP. This hinges on a valuation algebra whose valuations are piecewise linear functions.
- Ongoing
  - Empirical evaluation of both the centralised and decentralised solvers for CEAP.

Summary of contributions (I)

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

Summary of contributions (II)

- **Continuous Energy Allocation Problem**
  - CEAP offers more expressiveness to prosumers: **offers as piecewise linear functions**
  - **CEAP can be cast as a Linear Program** and hence solved by commercial solvers like CPLEX or Gurobi.
  - **CEAP's decentralised solver as an extension of RadPro.**

Perspectives

- **RADPRO's next steps**
  1. Cope with **cyclic** networks
  2. Mechanism design issues (VCG paymens are feasible!)

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.

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}