Designing a marketplace for the trading and distribution of energy in the smart grid

Jesús Cerquides[†] Gauthier Picard[⋆] Juan A. Rodríguez-Aguilar[†]

[†] IIIA-CSIC, Campus UAB,
08193 Cerdanyola, Catalonia, Spain
{cerquide,jar}@iiia.csic.es

[⋆] Laboratoire Hubert Curien UMR CNRS 5516
Institut Henri Fayol, Mines Saint-Etienne,
158 cours Fauriel, 42023 Saint-Etienne cedex, France
picard@emse.fr

Abstract

Decentralized energy production is meant to reduce generation and distribution inefficiencies, leading to major economic and environmental benefits. This new model is meant to be supported by smart grids, electricity networks that can intelligently integrate the actions of all users connected to them —generators, consumers, and prosumers (those that do both)— to efficiently deliver sustainable, economic and secure electricity supplies. A major research challenge is the design of markets for prosumers in smart grids that consider distribution grid constraints. This paper introduces a novel market that allows prosumers to trade electricity while satisfying the constraints of the grid. Our market's allocation rule is implemented by means of the so-called RadPro, an efficient dynamic programming algorithm that assesses in polynomial time how much energy each prosumer trades as well as how energy must be distributed throughout the grid. Our empirical results show that RadPro significantly outperforms both CPLEX and Gurobi in solving time when computing the optimal allocation over acyclic networks. Furthermore, the message-passing nature of RadPro offers the possibility of running our market in a decentralized (peer-to-peer) manner.

1 Introduction

Our centralized model of production and transmission wastes enormous amounts of energy. According to [6], "...an astonishing two-thirds of primary energy inputs". Since power stations are generally far from centers of demand, much of the produced heat is not used, but vented up chimneys or discharged to rivers. Additional losses come about as the electricity travels along the wires of the transmission and distribution systems [6, 27]. As argued in [27], favoring the decentralized generation of energy over traditional centralized electricity generation will reduce generation and distribution inefficiencies and will facilitate increased contributions from renewables. This new model is meant to be supported by smart grids.

Following [3], a smart grid is an electricity network that can intelligently integrate the actions of all users connected to it —generators, consumers, and *prosumers* (those that do both)— to efficiently deliver sustainable, economic and secure electricity supplies. In the smart grid the consumer can be either an individual or a household, but also a community or an SME. In its more general form, a smart grid is populated by prosumers capable of both generating and consuming energy. Therefore, smart grids clearly play the central role in the integration of all these prosumers (electricity grid users) by means of the enactment of a system that satisfies a number of societal goals. Out of these goals, there is that of setting market-based prices for electricity taking into account grid system constraints. Thus, a major research challenge in the heart of several roadmaps for the Smart Grid [3, 4] is the design of markets for prosumers in smart grids that consider distribution grid constraints. This vision will allow prosumers to

directly trade over the smart grid [8]. Following [20], market operations will involve a large number of heterogeneous prosumers, distributed throughout the network (closer to the point of use of electricity), and trading much smaller amounts of energy that are nowadays traded. The distribution of electricity employs one of the three common types of network topologies: radial, ring main, and interconnected [5, 7, 25]. On the one hand, radial networks are acyclic. On the other hand, as observed in [13], though ring main and interconnected networks contain cycles, they are configured into acyclic networks by means of switches to supply power [7, 25].

The smart grid vision has spurred a wealth of research on the design of markets and trading agents for the smart grid. The state-of-the-art has mainly considered to employ different types of auctions for this endeavor. Thus, the market-based trading of energy is typically addressed by the literature by having prosumers participate in a double auction where energy is traded on a day-ahead basis [8, 9, 10, 14, 17, 23]. Submitted buy and sell orders for energy are matched either by means of either a continuous double auction [10, 17, 23] or a call market [8, 9, 14]. Exceptions to this common approach are represented by the tailored multi-unit auctions in [26] and the simultaneous combinatorial reverse auctions employed in [18] to match demand and supply.

To the best of our knowledge, none of the market mechanisms employed in the literature so far takes into account grid system constraints. Thus, the clearing of the market occurs disregarding, for instance, that the transmission of energy is carried out along capacity-constrained distribution networks (which is an actual-world constraint [25]). Therefore, trading and distribution are considered as decoupled activities. Furthermore, the bidding language offered to grid users is not expressive enough to express a prosumer's energy profile. With the exception of [18], which supports combinatorial bids, double auctions limit a grid user to submit a single price-quantity bid to either buy or sell. This does not allow a prosumer to express a full energy profile encompassing a combination of all her buy and sell offers.

Against this background, the main goal of this paper is to design a novel market that allows prosumers to trade electricity in a smart grid while satisfying the grid's distribution constraints. More precisely, we make the following contributions:

- We formally introduce the energy allocation problem (EAP) as the problem of deciding how much energy each prosumer trades as well as how energy must be distributed throughout the grid so that the overall benefit is maximized while complying with the grid constraints and the prosumers' preferences. Thus, on the one hand, we consider that the capacity of the distribution network is limited [25]. On the other hand, since a prosumer can both generate and consume energy, our formulation considers that each prosumer can encode her preferences as a combination of offers to both buy and sell energy. Solving the EAP amounts to clearing our prosumer-oriented market.
- We show how to encode the EAP as a mixed-integer program so that it can be optimally solved for any distribution network topology by means of off-the-shelf commercial solvers such as CPLEX or Gurobi.
- We propose a novel dynamic programming algorithm to efficiently solve acyclic instances of the EAP, the so-called Radial Energy Network Algorithm for Prosumer Market (RadPro). RadPro is a message-passing algorithm that manages to optimally solve the EAP on acyclic electricity networks in polynomial time. Our empirical results show that RadPro significantly outperforms both CPLEX and Gurobi in solving time when computing the optimal allocation, namely when clearing the market, as the size of the market grows, being 15.8 times faster than the runner-up in the largest scenario tested. Furthermore, they also show that it largely overcomes the limitations of message-passing algorithms such as Acyclic-Solving [2], reducing several orders of magnitude the computation thanks to its efficient message assessment.
- Finally, since the EAP defines the allocation rule of our market, we also touch upon the design of payment rules that together with our allocation rule can help design a mechanism for our prosumer-oriented market.

The rest of the paper is organized as follows. Section 2 reviews the dynamic programming algorithm on which our algorithm RadPro is based. Section 3 formally defines the allocation rule that we propose to clear prosumer-oriented electricity markets. Thereafter, section 4 shows how to implement the clear-

Algorithm 1 The Acyclic-Solving algorithm

Each vertex j of the tree executes

1: From each child k of j, receive a message $\mu_{k \to j}$.
2: **if** j is not the root **then**
3: Assess $\mu_{j \to p_j}$ using equation 1
4: Send message $\mu_{j \to p_j}$ to its parent p_j
5: Receive message $\mathbf{X}^*_{s_j}$ from its parent p_j
6: **end if**
7: Assess the best assignment $\mathbf{X}^*_j$ using equation 2
8: To each child k of j, send message $\mathbf{X}^{*\downarrow s_k}$

ing of the market as a mixed-integer program (MIP), whereas section 5 introduces RadPro, our novel computationally-efficient algorithm for the same purpose. Next, section 6 empirically analyses RadPro, section 7 touches upon how to cope with prosumers' strategic behavior, and section 8 concludes and sets paths to future research.

2 Background

We start the section by introducing constraint optimization problems and then we review Acyclic-Solving: the dynamic programming algorithm that is used as the basis for RadPro.

In the following, let $X = \langle x_1, \ldots, x_n \rangle$ be a sequence of variables, with each variable x_j taking states in a finite set $\mathcal{D}_j$ known as its *domain*. The joint domain $\mathcal{D}_X$ is the Cartesian product of the domain of each variable. We use $\mathbf{x}_j$ to refer to a possible state of x_j, that is $\mathbf{x}_j \in \mathcal{D}_j$. Given a set of variables $Y \subseteq X$, a tuple $\mathbf{X}_Y$ assigns a possible state to each variable in Y, that is $\mathbf{X}_Y \in \mathcal{D}_Y$. A utility function f with scope Y is a function $f : \mathcal{D}_Y \to \mathbb{R}$. We use $s(f)$ for the scope of f. The sum of two utility functions f and f' with scopes Y and Z respectively is a new utility function $h = f + f'$ with scope $Y \cup Z$, such that $h(\mathbf{T}) = f(\mathbf{T}^{\downarrow Y}) + f'(\mathbf{T}^{\downarrow Z})$ where $\mathbf{T}^{\downarrow Y}$ stands for the projection of tuple $\mathbf{T}$ to scope Y. The projection of a utility function f with scope Y to scope Z, is $f^{\downarrow Z}(\mathbf{Z}) = \max\{f(\mathbf{Y}) \mid \mathbf{Y} \in \mathcal{D}_Y, \mathbf{Y}^{\downarrow Z} = \mathbf{Z}\}$. Formally, a COP is an optimization problem whose input is a tuple $\langle X, \mathcal{D}, F \rangle$, where F is a set of utility functions and whose objective is to find the tuple $\mathbf{X}^*$ that maximizes the sum $g = \sum_{f \in F} f$ of the utility functions, that is to find $\mathbf{X}^* = \operatorname{argmax}_{\mathbf{X}} g(\mathbf{X})$.

Given a COP problem, the constraint network is defined as the graph that has a vertex for each constraint and an edge between two constraints if their scopes share at least one variable. When the constraint network is acyclic, the Acyclic-Solving algorithm can solve the COP problem efficiently [2], provided that the scope of the utility functions is small. Acyclic-Solving selects one vertex as the root. Each vertex j but the root is assigned a parent p_j and one utility function f_j (representing its own stake at the problem). The scope of a vertex j is the scope of f_j. The separator s_j between j and p_j is the intersection of their scopes.

Acyclic-Solving, shown in Algorithm 1, runs two phases: (1) costs are sent from the leaves up to the root; (2) optimal assignments are decided and communicated down the tree. More in detail, it determines the optimal solution using the following scheme. First, each leaf vertex starts by sending its own utility to its parent. When a vertex has received the messages from all its children, it combines them with its own utility to produce the aggregated utility of its whole subtree, and then sends it to its parent. Once the root has received messages from all its children, it assesses the aggregated utility of the whole problem and then it decides the best assignment (maximum cost tuple) for its variables. Finally, it broadcasts this assignment to its children, who assess their best assignments and send them down the tree. After the execution of the algorithm, each agent knows the optimal values for the variables in its scope. The message sent from each vertex to its parent is

$$\mu_{j \to p_j} = g_j^{\downarrow s_j} \tag{1}$$

where $g_j = f_j + \sum_k \mu_{k \to j}$ and the best assignment for the variables in each vertex is assessed as

$$\mathbf{X}_j^* \leftarrow \underset{\mathbf{X}_j}{\operatorname{argmax}}\, g_j(\mathbf{X}_{s_j}^*, \mathbf{X}_j) \tag{2}$$

Since Acyclic-Solving can be expressed as a message passing algorithm, it can be easily applied to distributed COP systems[1], where it can be seen as a particular case of DPOP [19] or Action-GDL [22]. The computational complexity of each agent in Acyclic-Solving is exponential on the number of variables in its scope, due fundamentally to the assessment of the message in equation 1. Usually bookkeeping is used during that process to make the assessment of the best assignment in equation 1 computationally inexpensive. The communication complexity of each agent j is exponential on the number of variables in the separator s_j with its parent.

3 The energy allocation problem

The aim of this section is to provide a simple mathematical model for the energy market in a prosumer network, and the allocation rule proposed for that market. We start by providing an example of an energy trading scenario that illustrates the model of prosumers and the model of energy network that we will consider. Thereafter, we provide the allocation rule for that market as the solution to an optimization problem: the energy allocation problem (EAP).

3.1 Example: energy trading scenario

Figure 1 shows an example of an energy trading scenario involving four prosumers, each one represented by a circle. Each edge connecting two prosumers means that they are physically connected. Moreover, each link is labeled with its capacity, namely with the amount of energy it can transport. For instance, prosumer 1 is connected to prosumer 2, and their link can transport up to 2 energy units. Each prosumer can offer to either buy, sell or transmit energy. The offer of each prosumer is represented as a table next to each prosumer in Figure 1, where each entry in the table is a pair $(units, price)$. As a convention, a selling offer is expressed by means of a negative number of units and a negative price, whereas a buying offer is encoded with a positive number of units and a positive price. For instance, prosumer 4 offers (among other options): to buy 2 energy units and pay 1.75c€, to sell 3 units and get paid 11c€, and to transmit energy for free (0 units at price 0). In Figure 1, we observe that prosumer 1 only sells energy, and prosumer 2 only buys energy, while prosumers 3 and 4 can either buy or sell.

There are two important aspects to consider about the semantics of a prosumer's offer. First, notice that the entries in a prosumer's offer table represent mutually exclusive offers. Thus, for instance, prosumer 4's offer indicates that it can either buy 2 energy units or sell 3 energy units, but it cannot do both at the same time. Second, a prosumer's offers may be non-linear since the prosumer fixes a specific price for each number of units.

3.2 Problem definition

Now the problem faced by the prosumers in Figure 1 is to decide how much energy to trade and with whom so that the overall benefit is maximized while the energy network's capacity constraints are fulfilled. This means that: (i) each prosumer must choose a single offer in its offer table (how much to trade); and (ii) each pair of prosumers connected by a link must agree on the amount of energy to be transferred by their link together with the direction of the transfer (with whom). In what follows we cast this problem as an optimization problem, and we put off the solution to this problem to sections 4 and 5.

Following example 1, we consider that the energy network connecting a set of prosumers P can be modeled as an undirected graph (P, E), where the vertexes stand for the prosumers and each edge in E connects a pair of prosumers. An edge $\{i, j\} \in E$ means that prosumer i and j are physically connected

[1] In those scenarios where each agent is assigned a single constraint, and the constraint network is acyclic.

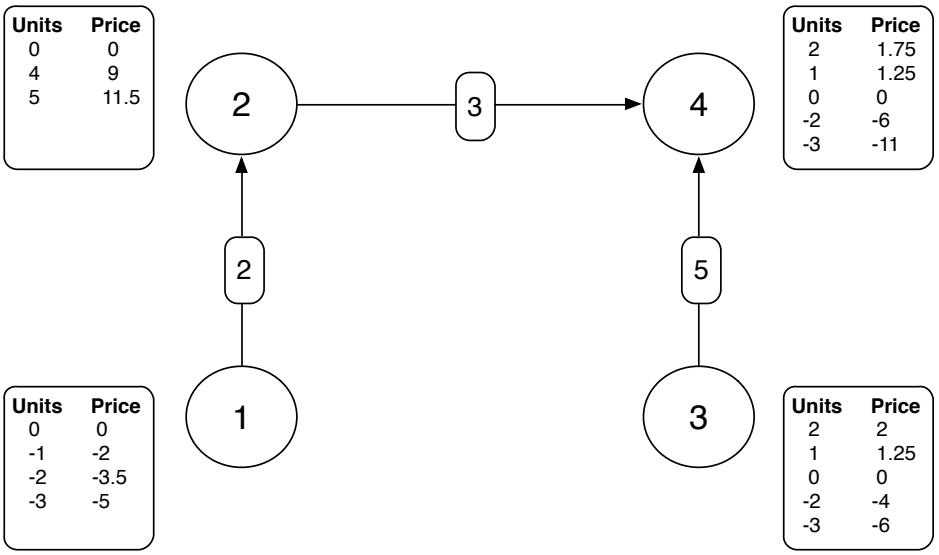

Figure 1: Energy trading scenario.

to trade energy. When $\{i, j\} \in E$, $i < j$ we say that i is an in-neighbor of j and that j is an out-neighbor of i. The set of in-neighbors (resp. out-neighbors) of j is $in(j)$ (resp. $out(j)$).

Each prosumer j expresses her offers to buy and sell energy by means of an *offer function* $o_j : \mathbb{Z} \to \mathbb{R} \cup \{-\infty\}$. For instance, $o_j(3) = 2$ indicates that prosumer j is willing to buy 3 energy units at 2c€, while $o_j(-4) = -2$ indicates that she is willing to sell 4 energy units if paid 2c€. Furthermore, we allow a prosumer to express that she is willing to let energy go through it by selling to her output prosumers as much energy as she buys from her input prosumers. Hence, we assume that $o_j(0) \neq -\infty$ for all $j \in P$. Notice that offer functions capture prosumers' constraints. To communicate her offer function, each prosumer sends a table like the ones in Figure 1 making explicit her feasible energy states and their values. If an offer for k units does not appear in the table, it means that such energy state is unfeasible for the prosumer and thus its value $o_j(k)$ is $-\infty$. Formally, the offer function is a valuation.

Definition 1. *A* valuation α *is a function* $\alpha : \mathbb{Z} \to \mathbb{R} \cup \{-\infty\}$. *It is interesting to divide the domain of* α *in two parts: the* finite valued domain $FVD(\alpha)$ *is the subset of* $\mathbb{Z}$ *in which* α *takes finite values. The size* n_α *of a valuation is the number of elements in* $FVD(\alpha)$. *We define the empty valuation* $\diamond$ *as the one that maps* 0 *to* 0 *and any other element to* $-\infty$.

Besides prosumers' offers, we also consider that the energy network is physically constrained by the capacity of the connections between prosumers. We will note as c_{ij} the capacity limit of edge $\{i, j\}$, namely the maximum number of energy units that the link between prosumers i and j can transmit. An allocation specifies the number of units that each prosumer trades with each neighboring prosumer. We will encode an allocation by means of a set of variables $Y = \{y_{ij} \mid i \in P, j \in out(i)\}$, where $\mathbf{y}_{ij}$ stands for the number of units that prosumer i sells to prosumer j and is bounded by the capacity limit c_{ij}. That is, the domain of variable y_{ij} is $D_{ij} = [-c_{ij} .. c_{ij}]$. Thus, if $\mathbf{y}_{ij}$ takes on a value k greater than 0, it means that prosumer i sells k energy units to prosumer j. Otherwise, if $\mathbf{y}_{ij}$ takes on a negative value $-k$, we say that prosumer i buys k energy units from prosumer j. From this follows that $\mathbf{y}_{ij}$ represents a trade from prosumer i's perspective.

Now we want to assess the value of a given allocation. Before that, we will define the local value of a given allocation for a single prosumer. We need to assess the amount of energy that a prosumer acquires and sells according to an allocation $\mathbf{Y}$. Prosumer j will only consider its local view of the allocation, represented by $\mathbf{Y}_j = \mathbf{y}_{.j} \cup \mathbf{y}_{j.}$. We can assess the *net energy balance* for prosumer j as

$$net(\mathbf{Y}_j) = \sum_{i \in in(j)} \mathbf{y}_{ij} - \sum_{k \in out(j)} \mathbf{y}_{jk}, \tag{3}$$

where each $\mathbf{y}_{ij}$ and $\mathbf{y}_{jk}$ are added with different signs because j takes the role of buyer in $\mathbf{y}_{ij}$ and that of seller in $\mathbf{y}_{jk}$. And therefore, the local value v_j of an allocation $\mathbf{Y}$ for prosumer j can be assessed as the value of her net energy balance by means of her offer function

$$v_j(\mathbf{Y}_j) = o_j(net(\mathbf{Y}_j)). \tag{4}$$

Therefore, the value of an allocation $\mathbf{Y}$ can be obtained by adding up the local value of the allocations for each prosumer.

$$Value(\mathbf{Y}) = \sum_{i \in P} v_j(\mathbf{Y}_j). \tag{5}$$

Now, we are ready to define the energy trading allocation as that of finding the allocation of maximum value that satisfies the capacity of the energy network.

Problem 1. *Given a set of prosumers P, their offers* $\{o_j | j \in P\}$*, and an undirected graph E where each edge is labeled with its capacity* c_{ij}*, the energy allocation problem (EAP) amounts to finding an allocation* $\mathbf{Y}$ *that maximizes* $Value(\mathbf{Y})$*. Whenever the graph E is acyclic we say that the EAP is acyclic.*

At this point we can consider again the example in Figure 1. When solving the EAP defined by Problem 1, we obtain the variable assignment shown in Figure 2. The solution indicates that prosumer 1 transfers 2 energy units to prosumer 2 ($y_{12} = 2$), prosumer 2 also receives 3 energy units from prosumer 4 ($y_{24} = -3$), and prosumer 3 transfers 3 energy units to prosumer 4. Within each offer table, we underline the offer chosen by each prosumer. For each prosumer, the underlined number of units corresponds to the net energy balance (Equation 3), while the underlined price corresponds to the local value of the allocation (Equation 4). Thus, the allocation that maximizes Equation 5 has a value of 2.

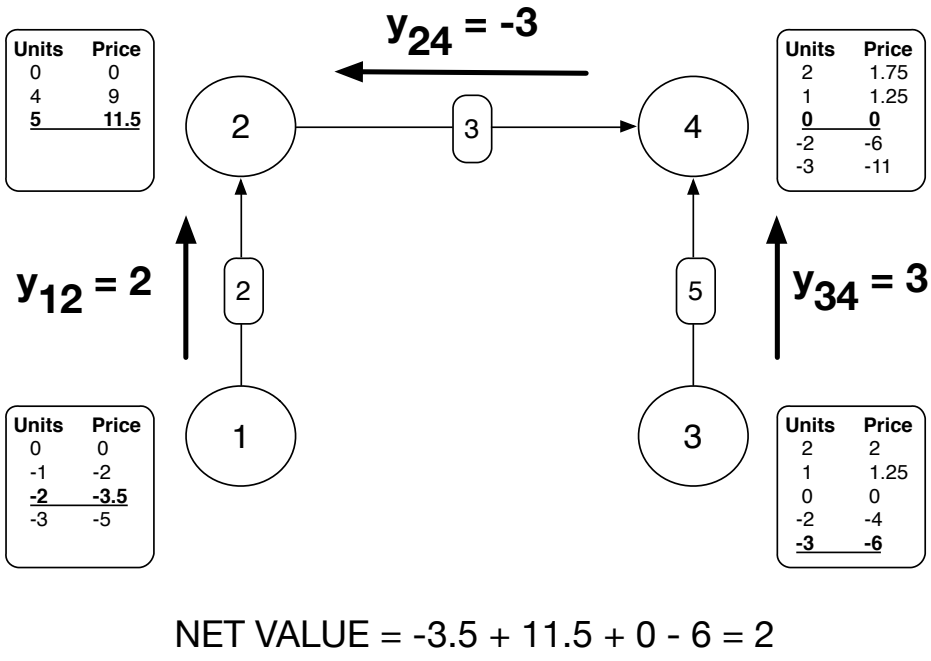

Figure 2: Solution to the EAP represented by the energy trading scenario.

Notice that prosumer 2 obtains 5 energy units by *aggregating* the energy units received from prosumers 1 and 4. However, prosumer 4 does not sell anything to prosumer 2. The role of prosumer 4 is to *relay* to prosumer 2 the energy transferred from prosumer 3, which is the one that does sell energy. In general, our model supports that each prosumer either: (i) aggregates energy received from its neighbors when buying energy; (ii) splits and distributes energy to its neighbors when selling energy; or (iii) relays energy so that other prosumers can satisfy their demand.

4 Solving the EAP through MIP

Solving optimization problems by mapping them to Mixed Integer linear Programs (MIPs) has become a standard practice whenever such a mapping can be found. Through the advance of software capabilities (including CPLEX and Gurobi), this practice turns out to be difficult to beat even for problems, such as combinatorial auctions, that have attracted a stream of research in specific algorithms [11]. Along this line, in this section we show how the EAP can be encoded as a MIP.

Before translating the EAP as a MIP, recall that the offer of prosumer j is expressed as a valuation o_j containing n_{o_j} offers. Without loss of generality we can assume that the offers in each offer function o_j are ordered. Then we will refer to the l-th offer in o_j by the pair (q_j^l, o_j^l), where q_j^l stands for the number of units offered to buy or sell, and o_j^l stands for the offered price.

To encode our optimization problem, we will consider two types of decision variables. On the one hand, as described in section 3, for each edge (i, j) in the trading energy network an integer variable y_{ij} will take on as a value the number of units that prosumer i sells to prosumer j (when $\mathbf{y}_{ij} > 0$), or that she buys from prosumer j (when $\mathbf{y}_{ij} < 0$). Notice that $\mathbf{y}_{ij}$ may also be zero if there is no trading between i and j. In general, the value of y_{ij} is within the domain D_{ij}.

Since the prosumer value $v_j(\mathbf{Y}_j)$ of equation 4 cannot be encoded as a linear function in terms of these variables, for each prosumer j we introduce a set of auxiliary binary variables $\{x_j^l |\ 1 \leq l \leq n_{o_j}\}$, where variable x_j^l indicates whether the l-th offer in o_j is taken or not. Since the offers of prosumer j are mutually exclusive, these variables are linked by a constraint that enforces one and only one of them to be active, namely $\sum_{l=1}^{n_{o_j}} x_j^l = 1$. The net energy balance $net(\mathbf{Y}_j)$ from equation 3 provides a connection between the flows of energy in and out a prosumer and the offer selected. We can express equation 3 for prosumer j by means of the constraint $\sum_{i<j} y_{ij} - \sum_{k>j} y_{jk} = \sum_{l=1}^{n_{o_j}} x_j^l \cdot q_j^l$. Finally, the prosumer value can be easily written as a linear expression in terms of these variables: $\sum_{l=1}^{n_{o_j}} x_j^l \cdot o_j^l$.

Now we are ready to define the MIP that solves the energy allocation problem introduced in the previous section.

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

Where the first two constraints enforce the correct representation of the net value in terms of x_j^l variables. Furthermore, we enforce that the quantity of energy exchanged between two prosumers cannot exceed the capacity of the edge in the third constraint – the domain constraints for variables y_{ij}. The fourth one is the domain constraint for the binary variables x_j^l. Hence, solving the EAP amounts to solve this linear program.

Let us consider again the example in Figure 1, and its solution in Figure 2. As presented in the previous section, the optimal allocation $\mathbf{Y}$ is the one where the decision variables' values are $y_{12} = 2$, $y_{24} = -3$ and $y_{34} = 3$. This allocation corresponds to set the x's as follows: $x_1^{-2} = x_2^5 = x_3^{-3} = x_4^0 = 1$ (otherwise $x_j^l = 0$). This leads to the following evaluation of the allocation:

$$Value(\mathbf{Y}) = o_1^2 + o_2^5 + o_3^{-3} + o_4^0 = -3.5 + 11.5 + 0 - 6 = 2$$

5 Message passing for acyclic EAP

This section presents RadPro, an alternative algorithm for acyclic EAP that relies on message passing, and thus can be easily implemented in a distributed way. As explained in section 2 the Acyclic-Solving

algorithm relies on message passing to solve acyclic COPs. The purpose of the section is twofold. First, section 5.1 details how to employ Acyclic-Solving directly to solve the problem. Since the assessment of messages in Acyclic-Solving is known to have exponential complexity, section 5.2 details the mathematical theory that underlies the assessment of Acyclic-Solving messages with polynomial complexity. Finally, section 5.3 introduces RadPro and compares the communication and computational complexity with that of Acyclic-Solving.

5.1 Direct message passing solution

Note that the value of an allocation, as defined in equation 5, is a sum of utility functions, one per prosumer. Thus, we can directly map the EAP into a COP. When the EAP is acyclic, so is the constraint network and we can directly apply Acyclic-Solving as it is described in section 2.

Note that the separator s_j between each node j and its parent p_j corresponds either with the single variable y_{ip_j} if p_j is an out-neighbor of j or with y_{p_jj} if p_j is an in-neighbor of j. In any case the size of the message $\mu_{j\to p_j}$ is the number of possible states of the link between j and p_j, and the message can be understood as communicating the utility of each of these states to the network composed by the prosumers in the subtree rooted at j. Thus, the amount of communication needed to run Acyclic-Solving is very small.

The expression of the message from j to its parent can be obtained by particularizing the general Acyclic-Solving equation 1. When the parent node p_j is an out-neighbor, j sends the following message:

$$\mu_{j\to p_j}(\mathbf{y}_{jp_j}) = \max_{\mathbf{Y}_{j-p_j}} \left(v_j(\mathbf{y}_{jp_j}, \mathbf{Y}_{j-p_j}) + \sum_{k\in out(j)\setminus\{p_j\}} \mu_{j\to k}(\mathbf{y}_{jk}) + \sum_{i\in in(j)} \mu_{i\to j}(\mathbf{y}_{ij}) \right), \quad (6)$$

and when the parent node p_j is an in-neighbor, j sends the following message:

$$\mu_{j\to p_j}(\mathbf{y}_{p_jj}) = \max_{\mathbf{Y}_{j-p_j}} \left(v_j(\mathbf{y}_{p_jj}, \mathbf{Y}_{j-p_j}) + \sum_{k\in out(j)} \mu_{j\to k}(\mathbf{y}_{jk}) + \sum_{i\in in(j)\setminus\{p_j\}} \mu_{i\to j}(\mathbf{y}_{ij}) \right). \quad (7)$$

where $\mathbf{Y}_{j-p_j}$ stands for an assignment of values to each variable in Y_j except y_{jp_j} (or y_{p_jj}). Note that in agreement with section 2, the assessment of messages takes time exponential in the number of variables in the scope of the constraint, which in this case corresponds with the number of neighbors of the prosumer. The computational cost for a prosumer j of assessing the message to its parent is bounded $O((2C_j+1)^{N_j})$, where C_j is the capacity of the most powerful link between i and a neighboring agent and N_j is the number of neighbors of j. Again, as explained in section 2, we can use bookkeeping during the assessment of the messages to reduce the complexity of the assessment of the best assignment according to equation 2 to constant time. Thus Acyclic-Solving will perform efficiently as long as the degree of the nodes in the energy network is small. This is likely to be so in rural scenarios [16], however in urban areas [24] the degree of nodes has been seen to follow a geometric distribution with some nodes' degree reported to be over 35. This hampers the direct application of Acyclic-Solving in such scenarios. Similar dynamic programming approaches for the problem of CO_2 reduction in energy networks have been reported functional for nodes with a branching factor up to four [13].

In the following section we develop an algebra of valuations that allows for a much more efficient computation of messages.

5.2 Efficient message computation

The assessment of messages using equations 6 and 7 takes time exponential in the number of neighbors that the prosumer is connected to. Thus, for densely connected energy networks, it can severely hinder the applicability of the algorithm. To overcome this problem, in this section we introduce an algebra of valuations that the prosumers can take advantage of when assessing their messages. As a result, the message assessment complexity is severely reduced.

Objects and Operations of the algebra of valuations. Our objective is to build an algebra that enables us to perform the assessment of messages faster. The main mathematical object of the algebra, namely the *valuation* has already been defined in definition 1.

Both the offers o_j of each prosumer and the messages $\mu_{j\to p_j}$ can be directly mapped into valuations. Valuations can be efficiently represented computationally by means of hash tables. In the following we will introduce the operations on valuations required to assess the messages.

First consider the scenario of a prosumer j with a single in-neighbor p_j. The message to send according to equation 7 will basically copy the offer o_j. However, since the possible states of $y_{p_j j}$ are limited by the capacity of the link, we will need to have an operation in the algebra that removes from valuation o_j, those values which are out of the range. This operation is called *restriction*.

Definition 2. *Given a valuation α, and a subset $D \subseteq \mathbb{Z}$ we define $\alpha[D]$, the* restriction of α to D *as*

$$\alpha[D](k) = \begin{cases} \alpha(k) & k \in D \\ -\infty & otherwise. \end{cases}$$

Thus, the message of a prosumer j to its single in-neighbor p_j can be written as $\mu_{j\to p_j} = o_j[D_{p_j j}]$.

Now consider the scenario of a prosumer j with a single out-neighbor p_j. In this case, the equation of the net energy flow for j shows that $net(\mathbf{y}_{jp_j}) = -\mathbf{y}_{jp_j}$. Thus, the message to send according to equation 6 is a symmetric version of o_j, restricted to the capacity of the link. We refer to the operation that assesses the symmetric version of a valuation as the *complement*.

Definition 3. *Given a valuation α, we define $\overline{\alpha}$, the* complement of α *as*

$$\overline{\alpha}(k) = \alpha(-k)$$

Thus, the message of a prosumer j to its single out-neighbor p_j can be written as $\mu_{j\to p_j} = \overline{o_j}[D_{p_j j}]$.

Finally take the scenario of a prosumer j such that its parent p_j is an in-neighbor and that it has an additional out-neighbor k. In that case equation 7 is reduced to

$$\mu_{j\to p_j}(\mathbf{y}_{p_j j}) = \max_{\mathbf{y}_{jk}} \left(v_j(\mathbf{y}_{p_j j}, \mathbf{y}_{jk}) + \mu_{j\to k}(\mathbf{y}_{jk}) \right) = \max_{\mathbf{y}_{jk}} \left(o_j(\mathbf{y}_{p_j j} - \mathbf{y}_{jk}) + \mu_{j\to k}(\mathbf{y}_{jk}) \right) \quad (8)$$

Note that for some of the assignments $\mathbf{y}_{p_j j}, \mathbf{y}_{jk}$, it will happen that o_j is undefined for the net energy balance $\mathbf{y}_{p_j j} - \mathbf{y}_{jk}$. We can avoid considering these assignments by introducing a new operation for valuations, the *aggregation*.

Definition 4. *Given two valuations α, β, we define $\alpha \cdot \beta$, the* aggregation of α and β *as*

$$(\alpha \cdot \beta)(k) = \max_{\substack{i,j \\ k=i+j}} \alpha(i) + \beta(j) \quad (9)$$

Now the message of a prosumer j such that its parent p_j is an in-neighbor and that it has an additional out-neighbor k, can be written as $\mu_{j\to p_j} = (o_j \cdot \mu_{k\to j})[D_{p_j j}]$.

Structure of the algebra of valuations. We have defined the mathematical objects (the valuations) and the operations (restriction, complement and aggregation) of our algebra. The following two results describe the algebraic structure we have created. First, we concentrate on the aggregation operation.

Lemma 1. *The set of valuations with the aggregation operation forms a commutative monoid with $\diamond$ as the identity element. That is, for any valuations α, β, and γ we have that*

1. $\alpha \cdot \beta = \beta \cdot \alpha$
2. $(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$
3. $\alpha \cdot \diamond = \alpha$

Proof. Commutativity follows directly from the symmetric definition of the sum in equation 9. To prove associativity, we apply the definition to $(\alpha+\beta)+\gamma$ and then we realize that we can rewrite it as $\alpha+(\beta+\gamma)$:

$$\begin{aligned}((\alpha+\beta)+\gamma)(k) &= \max_{\substack{i,j\\k=i+j}} (\alpha+\beta)(i)+\gamma(j) = \\ &= \max_{\substack{i,j\\k=i+j}} \max_{i=i'+j'} (\alpha(i')+\beta(j'))+\gamma(j) \\ &= \max_{\substack{i',j',j\\k=i'+j'+j}} \alpha(i')+\beta(j')+\gamma(j) \\ &= \max_{\substack{i',j''\\k=i'+j''}} \alpha(i') + \max_{\substack{j',j\\j''=j'+j}} (\beta(j')+\gamma(j)) \\ &= (\alpha+(\beta+\gamma))(k)\end{aligned} \tag{10}$$

Finally, we prove that $\diamond$ is the identity

$$(\alpha + \diamond)(k) = \max_{\substack{i,j\\k=i+j}} \alpha(i) + \diamond(j) = \alpha(k) + \diamond(0) = \alpha(k) \tag{11}$$

□

From a mathematical standpoint Lemma 1 enables us to write expressions such as $\alpha\cdot\beta\cdot\gamma$ that do not establish a specific order in which the aggregations should be performed. This allow us to extend the definition of aggregation from pairs of valuations to sets of valuations. We define the n-way aggregation of the valuations in $A = \{\alpha_1,\dots,\alpha_n\}$ as $(\prod_{i=1}^n \alpha_i)(k) = \max_{\sum_{i=1}^n j_i = k} \sum_{i=1}^n \alpha_i(j_i)$. Note that, as expected by the notation, we have that $\prod_{i=1}^n \alpha_i = \alpha_1\cdot\ldots\cdot\alpha_n$. Thus, the n-way aggregation of a finite set of valuations can be assessed using only the (2-way) aggregations from definition 4. From a computational perspective, Lemma 1 allow us to group the aggregation operations in the way we prefer. Thus, we can follow a sequential order, or assess them in a tree in case that is more efficient.

Now we are interested in characterizing the relations between the different operations.

Lemma 2. *The complement defines a self-inverse automorphism of the commutative monoid of valuations. That is, for any two valuations α and β we have that*

$\overline{\alpha\cdot\beta} = \overline{\alpha}\cdot\overline{\beta}$ $\quad$ $\overline{\overline{\alpha}} = \alpha$

$\overline{\diamond} = \diamond$ $\quad$ $\bar{\cdot}:\Phi\to\Phi$ *is a bijective mapping.*

Furthermore, $\overline{\alpha[D]} = (\overline{\alpha})[-D]$ *where* $-D = \{-x | x\in D\}$.

Proof. We will prove the properties in order, starting with property 1. By definition we have that

$$\overline{\alpha\cdot\beta}(k) = \max_{\substack{i,j\\-k=i+j}} \alpha(i)+\beta(j)$$

Now, changing variables to $i' = -i$ and $j' = -j$

$$\max_{\substack{i',j'\\k=i'+j'}} \alpha(-i')+\beta(-j')$$

And, complementing α and β

$$\max_{\substack{i',j'\\k=i'+j'}} \overline{\alpha}(i')+\overline{\beta}(j') = (\overline{\alpha}\cdot\overline{\beta})(k)$$

Property 2 states that $\alpha(k) = \alpha(-(-k))$, which is true because $k = -(-k)$.

Algorithm 2 Aggregation of two valuations

```
γ ← ⋄
for i ∈ FVD(α) do
    for j ∈ FVD(β) do
        γ(i + j) ← max(γ(i + j), α(i) + β(j))
    end for
end for
```

To prove Property 3 we only have to take the definition of

$$\diamond(k) = \begin{cases} 0 & \text{if } k = 0 \\ -\infty & \text{otherwise,} \end{cases}$$

assess that

$$\overline{\diamond}(k) = \begin{cases} 0 & \text{if } k = 0 \\ -\infty & \text{otherwise,} \end{cases}$$

and see that they are equal.

To prove property 4 we need to prove that $\bar{\cdot}$ is injective and surjective. We start proving that it is injective. Assume that $\overline{\alpha} = \overline{\beta}$. Since this directly implies that $\alpha = \beta$, then $\bar{\cdot}$ is injective. Now we prove that $\bar{\cdot}$ is surjective. Take any valuation β. We have to prove that there is another valuation α such that $\beta = \overline{\alpha}$. If we take $\alpha = \overline{\beta}$, then $\overline{\alpha} = \overline{\overline{\beta}} = \beta$ and we have proven that $\bar{\cdot}$ is surjective and concluded our proof of property 4 and of the lemma.

Finally, property 5 is proven as follows

$$\begin{aligned} \overline{\alpha[D]}(k) = \alpha[D](-k) &= \begin{cases} \alpha(-k) & \text{if } -k \in D \\ -\infty & \text{otherwise} \end{cases} \\ &= \begin{cases} \alpha(-k) & \text{if } k \in -D \\ -\infty & \text{otherwise} \end{cases} \\ &= \begin{cases} \overline{\alpha}(k) & \text{if } k \in -D \\ -\infty & \text{otherwise} \end{cases} \\ &= \overline{\alpha}[-D] \end{aligned}$$

□

Some of these properties will come in very handy when working with expressions that involve aggregations, complements and restrictions.

The efficiency of aggregation. Next, we are interested in determining the computational cost of aggregating a set of valuations. Algorithm 2 can be used to implement aggregation, with the following result:

Lemma 3. *The aggregation of two valuations α and β can be done in time $O(n_\alpha \times n_\beta)$. Furthermore $n_{\alpha\cdot\beta} \leq n_\alpha \times n_\beta$.*

Proof. Algorithm 2 takes time $O(n_\alpha \times n_\beta)$ and assesses the aggregation of α and β. In the worst case, every value taken by $i + j$ is different and the size of the $\alpha \cdot \beta$ is $n_{\alpha\cdot\beta} = n_\alpha \times n_\beta$. □

From here we can show that assessing the aggregation of M valuations of size N can be done in time $O(N^M)$. Our objective was to speed-up the computation by aggregating the messages. However we see that in the more general case, the aggregation of M messages takes exponential time. Next we take advantage of a particularity of the messages we need to aggregate that allow us to compute the aggregation in polynomial time. Notice that the FVD of a message $\mu_{i\to j}$ always lies in a small range $D_{ij} = [-c_{ij}..c_{ij}]$ around 0. We say that a valuation α is *of restricted capacity* C if $FVD(\alpha) \subseteq [-C..C]$. For restricted capacity valuations we get the following result.

Lemma 4. *The aggregation of two valuations α and β of restricted capacity C, can be done in time $O(C^2)$. Furthermore $\alpha \cdot \beta$ is a valuation of restricted capacity $2C$.*

Proof. Follows directly from the fact that for any two numbers $i, j \in [-C..C]$ we have that $i + j \in [-2C..2C]$. □

Note that whilst for general valuations, the size of the aggregated valuation grows quadratically, for restricted capacity valuations, it only grows linearly. This turns out to be fundamental to prove the following result, which tell us that we can aggregate valuations in polynomial time as long as they are of restricted capacity.

Lemma 5. *Given a set of M valuations A of restricted capacity C we can assess its aggregation $\prod_{\alpha \in A} \alpha$ in time $O(M^2C^2)$.*

The proof uses the associative property to divide the aggregation into two smaller aggregations each with one half of the valuations, to assess those smaller aggregations and to finally aggregate the results.

Proof. First, we prove it for the particular case when $M' = 2^K$. We perform the computation in tree. We start assessing $\frac{M'}{2}$ aggregations each taking time $O(C^2)$ and resulting in valuations of restricted capacity $2C$. We have reduced the problem to the aggregation of $\frac{M'}{2}$ valuations of restricted capacity $2C$. Next we assess $\frac{M'}{4}$ aggregations each taking time $O(4C^2)$. We continue for a total of K steps. The overall time is $O(\sum_{i=1}^{K} \frac{M'}{2^i} 2^{2i} C^2) = O(M'C^2 \sum_{i=1}^{K} 2^i) = O(M'C^2(2^{K+1} - 1)) = O(M'C^2(2M' - 1)) = O(M'^2C^2)$.

Now, for the general case with M whatsoever, we take $K = \lceil \log_2 M \rceil$ and $M' = 2^K$. Then, we can complete the M valuations we had with $M' - M$ empty valuations $\diamond$ and assess the overall aggregation in time $O(M'^2C^2)$ using the result in the previous paragraph. Since $M' < 2M$, the complexity is $O((2M)^2C^2) = O(M^2C^2)$, and this concludes the proof. □

Using the algebra of valuations to assess messages. The following theorem shows that we can provide an expression for Acyclic-Solving messages in equations 6 and 7 in terms of valuation algebra operations and that this can done in polynomial time.

Theorem 1. *The message from prosumer j to its in-neighbor parent p_j defined in equation 7 can be assessed as*

$$\mu_{j \to p_j} = \left(o_j \cdot \prod_{k \in out(j)} \mu_{j \to k} \cdot \prod_{i \in in(j) \setminus \{p_j\}} \overline{\mu_{i \to j}} \right) [D_{p_j j}] \tag{12}$$

and the message from prosumer j to its out-neighbor parent p_j defined in equation 6 can be assessed as

$$\mu_{j \to p_j} = \left(\overline{o_j} \cdot \prod_{k \in out(j) \setminus \{p_j\}} \overline{\mu_{j \to k}} \cdot \prod_{i \in in(j)} \mu_{i \to j} \right) [-D_{j p_j}]. \tag{13}$$

Furthermore, the computational complexity of assessing each message is $O(N_j C_j n_{o_j} + N_j^2 C_j^2)$ where N_j is the number of neighbors of j and C_j is the largest capacity of any of the links connected to j.

The proof relies on transforming the expressions in equations 6 and 7 into a k-way aggregation which is then mapped to 2-way aggregations and manipulated with the help of Lemma 2. The complexity result relies basically on Lemma 5.

Proof. We start proving that equation 7 is equivalent to equation 12. First notice that we only have to prove it inside of $D_{p_j j}$, since both valuations assign value $-\infty$ to any element outside of $D_{p_j j}$.

Next we will prove the following result, so we can later use it

Lemma 6. *Let α, β, and γ be valuations. Define η as*

$$\eta(i) = \max_{j,k} (\alpha(i + j - k) + \beta(k) + \gamma(j)).$$

Then $\eta = \alpha \cdot \beta \cdot \overline{\gamma}$

Proof. Define $q = i + j - k$ and $r = k - j$

$$\eta(i) = \max_{j,k} \alpha(i+j-k) + \beta(k) + \gamma(j) = \max_{\substack{q,r \\ q+r=i}} \left(\alpha(q) + \max_{\substack{j,k \\ k-j=r}} (\beta(k) + \gamma(j)) \right)$$

Now replacing $j' = -j$ we get

$$\max_{\substack{q,r \\ q+r=i}} \left(\alpha(q) + \max_{\substack{j',k \\ k+j'=r}} (\beta(k) + \gamma(-j')) \right)$$

And complementing γ,

$$\max_{\substack{q,r \\ q+r=i}} \left(\alpha(q) + \max_{\substack{j',k \\ k+j'=r}} (\beta(k) + \overline{\gamma}(j')) \right) = \max_{\substack{q,r \\ q+r=i}} \alpha(q) + (\beta \cdot \overline{\gamma})(r) = \alpha \cdot \beta \cdot \overline{\gamma}$$

□

Now, going back to equation 7 and assuming $\mathbf{y}_{p_j j}$ is inside $D_{p_j j}$

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

Using the definition of v_j in equation 4 and the definition of net energy balance in equation 3 we get

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

Introducing variables $\mathbf{z}_j = \sum_{i \in in(j) \setminus \{p_j\}} \mathbf{y}_{ij}$, and $\mathbf{t}_j = \sum_{k \in out(j)} \mathbf{y}_{jk}$, we get

$$\max_{\mathbf{z}_j, \mathbf{t}_j} \left(o_j(\mathbf{y}_{p_j j} + \mathbf{z}_j - \mathbf{t}_j) + \max_{\substack{\mathbf{Y}_j^{out} \\ \sum \mathbf{Y}_j^{out} = \mathbf{t}_j}} \sum_{k \in out(j)} \mu_{j \to k}(\mathbf{y}_{jk}) + \max_{\substack{\mathbf{Y}_{j-p_j}^{in} \\ \sum \mathbf{Y}_{j-p_j}^{in} = \mathbf{z}_j}} \sum_{i \in in(j) \setminus \{p_j\}} \mu_{i \to j}(\mathbf{y}_{ij}) \right)$$

Now we can transform the k-way aggregations into 2-way aggregations getting

$$\max_{\mathbf{z}_j, \mathbf{t}_j} \left(o_j(\mathbf{y}_{p_j j} + \mathbf{z}_j - \mathbf{t}_j) + \left(\prod_{k \in out(j)} \mu_{k \to j} \right)(\mathbf{t}_j) + \left(\prod_{i \in in(j) \setminus \{p_j\}} \mu_{i \to j} \right)(\mathbf{z}_j) \right)$$

Applying Lemma 6

$$\left(o_j \cdot \prod_{k \in out(j)} \mu_{j \to k} \cdot \overline{\prod_{i \in in(j) \setminus \{p_j\}} \mu_{i \to j}} \right)(\mathbf{y}_{p_j j})$$

By property 1 in lemma 2

$$\left(o_j \cdot \prod_{k \in out(j)} \mu_{j \to k} \cdot \prod_{i \in in(j) \setminus \{p_j\}} \overline{\mu_{i \to j}} \right)(\mathbf{y}_{p_j j})$$

And since $\mu_{j \to p_j}(\mathbf{y}_{p_j j}) = -\infty$ outside $D_{p_j j}$ we obtain that

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

as we wanted to prove.

Next, we prove the equivalence of equations 6 and 13. As it was the case in the first part of the proof we only have to prove it inside of D_{jp_j}, since both valuations assign value $-\infty$ to any element outside of D_{jp_j}.

Now, instead of working directly with the message, we will start from its complement. From the definition of complement and equation 6 we have that

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

Using the definition of prosumer value in equation 4 and the definition of net energy balance in equation 3 we get

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

Introducing variables $\mathbf{z}_j = \sum_{i\in in(j)} \mathbf{y}_{ij}$, and $\mathbf{t}_j = \sum_{k\in out(j)\setminus\{p_j\}} \mathbf{y}_{jk}$, we get

$$\max_{\mathbf{z}_j,\mathbf{t}_j} \left(o_j(-\mathbf{y}_{jp_j} + \mathbf{z}_j - \mathbf{t}_j) + \max_{\substack{\mathbf{Y}^{out}_{j-p_j} \\ \sum \mathbf{Y}^{out}_{j-p_j} = \mathbf{t}_j}} \sum_{k\in out(j)\setminus\{p_j\}} \mu_{j\to k}(\mathbf{y}_{jk}) + \max_{\substack{\mathbf{Y}^{in}_{j} \\ \sum \mathbf{Y}^{in}_{j} = \mathbf{z}_j}} \sum_{i\in in(j)} \mu_{i\to j}(\mathbf{y}_{ij}) \right)$$

Now we can transform the k-way aggregations into 2-way aggregations getting

$$\max_{\mathbf{z}_j,\mathbf{t}_j} \left(o_j(-\mathbf{y}_{p_jj} + \mathbf{z}_j - \mathbf{t}_j) + \left(\prod_{k\in out(j)\setminus\{p_j\}} \mu_{k\to j} \right)(\mathbf{t}_j) + \left(\prod_{i\in in(j)} \mu_{i\to j} \right)(\mathbf{z}_j) \right)$$

Applying Lemma 6

$$\left(o_j \cdot \prod_{k\in out(j)\setminus\{p_j\}} \mu_{k\to j} \cdot \overline{\prod_{i\in in(j)} \mu_{i\to j}} \right)(\mathbf{y}_{jp_j})$$

By property 1 from Lemma 2

$$\left(o_j \cdot \prod_{k\in out(j)\setminus\{p_j\}} \mu_{k\to j} \cdot \prod_{i\in in(j)} \prod \mu_{i\to j} \right)(\mathbf{y}_{jp_j})$$

And since $\mu_{j\to p_j}(\mathbf{y}_{jp_j}) = -\infty$ outside D_{jp_j}

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

Now, we can use property 2 from Lemma 2, substituting the previous result and applying properties 5

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

Table 1: Complexities of Acyclic-Solving and RadPro

and finally 1 from Lemma 2, we get

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

And the second part of the proof is finished.

The prove the complexity result, we can perform the aggregations of the two big products together in time $O(N_j^2C_j^2)$ by using Lemma 5. This will make a valuation of size at most N_jC_j. Thus, by virtue of Lemma 3 we can aggregate it with o_j in time $O(n_{o_j}N_jC_j)$. Since complements and restrictions can be done in linear time, the messages can be assessed in $O(n_{o_j}N_jC_j) + O(N_j^2C_j^2)$. □

5.3 RadPro message passing solution

Theorem 1 shows that Acyclic-Solving messages can be assessed in polynomial time for the EAP. We name RadPro, the new algorithm resulting from running Acyclic-Solving as described in Algorithm 1 but assessing messages using equations 12 and 13, instead of the usual Acyclic-Solving expression in equation 1. Note that we are only modifying the algorithm that we use to assess the messages. Since the messages exchanged are exactly the same for both algorithms, the number and the size of the messages exchanged is exactly the same. Both Acyclic-Solving and RadPro, send $2n$ messages, being n the number of prosumers. Furthermore, the size of each message is bounded by $2C_{max}+1$, where C_{max} is the largest capacity of any link in the network. Thus the number of values exchanged by both algorithms is $O(nC_{max})$.

Regarding computational complexity, as reported in section 5.1 in Acyclic-Solving the complexity for a prosumer j of assessing the message to its parent is $O((2C_j+1)^{N_j})$ where C_j is the largest capacity of any link connected to prosumer j and N_j is the number of neighbors of the prosumer. Thus the overall computational complexity is bounded by $O(n(2C_{max}+1)^{N_{max}})$ where N_{max} is the number of neighbors of the most connected prosumer and n is the overall number of prosumers. On the other hand, the computational complexity for assessing a message in RadPro, as shown in Theorem 1 is $O(N_jC_jn_{o_j} + N_j^2C_j^2)$. Since the number of messages assessed is equal to the number of prosumers in the network, the overall complexity of the algorithm is bounded by $O(n(N_{max}C_{max}n_{max} + N_{max}^2C_{max}^2))$ where n_{max} is the size of the largest offer of any prosumer. Assuming the common case that $n_{max} < N_{max}C_{max}$, we can simplify the expression to a final computational complexity of $O(nN_{max}^2C_{max}^2)$.

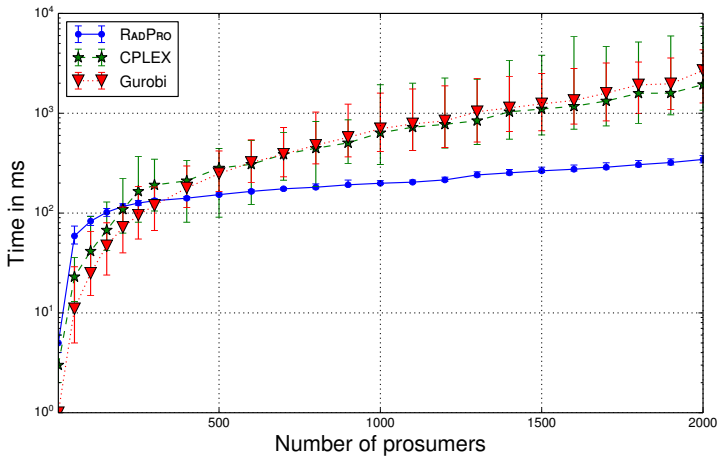

Figure 3: Runtime of RadPro in geometric distribution networks compared to that of CPLEX and Gurobi, $\kappa = 10$.

6 Experiments and Evaluation

In this section we evaluate the running time of RadPro and compare it with that of Acyclic-Solving and that of MIP solvers through the mapping proposed in section 4.

Comparison with MIP solvers. The first experiment compares RadPro performance against commercial MIP solvers (in particular against IBM CPLEX and Gurobi). To do so, we generate acyclic EAP instances whose network structure mimics that of radial networks. According to Wang [24], a good fit for the empirical distribution of node degrees in real-world power grids is obtained by using a geometric distribution. Thus, we generate trees whose node degrees follows a geometric distribution, with $p = 0.5$ to emulate the shape of acyclic urban distribution networks. Each vertex in the tree represents a prosumer and is randomly assigned to be either a producer (10% of the prosumers) or a consumer (the remaining 90%). For each consumer j, the offer o_j is

$$o_j(t) = \begin{cases} t \cdot price_j & t \in [min_j..max_j] \\ 0 & t = 0 \\ -\infty & otherwise \end{cases}$$

where min_j and max_j are the minimum and maximum number of units allowed for prosumer j, and $price_j$ is the price per energy unit for that prosumer. The maximum number of units max_j is sampled from a normal distribution $\mathcal{N}(\kappa, \frac{\kappa}{2})$ where κ is a capacity parameter. Then, the minimum number of units of the offer min_j is sampled from a uniform distribution $\mathcal{U}(1, max_j)$. We sample $price_j$ from a normal distribution $\mathcal{N}(1, 0.5)$. Producers are generated following the very same procedure, but the offer is defined in the interval $[-max_j .. -min_j]$. The capacity of a link between two nodes is determined as the maximum number of units produced or required by both prosumers. Experiments have been executed on Intel Core i7 2.66GHz, with 8GB RAM. RadPro has been implemented in Java. The code for the experiments will be made publicly available upon acceptance for the sake of reproducibility.

We analyze two different capacities $\kappa = 10$ and $\kappa = 100$. The number of prosumers in the network n varies in steps of 100 up to 2000. For each κ and n we generated 100 different random problems. Figure 3 shows the median and the 5%-95% interquartile range of the running time for $\kappa = 10$, and Figure 4 for $\kappa = 100$. In both cases, CPLEX and Gurobi are faster when the number of prosumers is small. However RadPro scales better and for $n = 2000$ it is more than one order of magnitude faster than CPLEX, which in turn outperforms Gurobi. In the more stringent scenario (networks of size $n = 2000$ and capacity $\kappa = 100$), RadPro requires in median below 2.3 seconds whilst CPLEX needs over 35.8 seconds and Gurobi more than two minutes.

Evaluating efficient message assessment. To highlight the benefit of efficient message assessment as

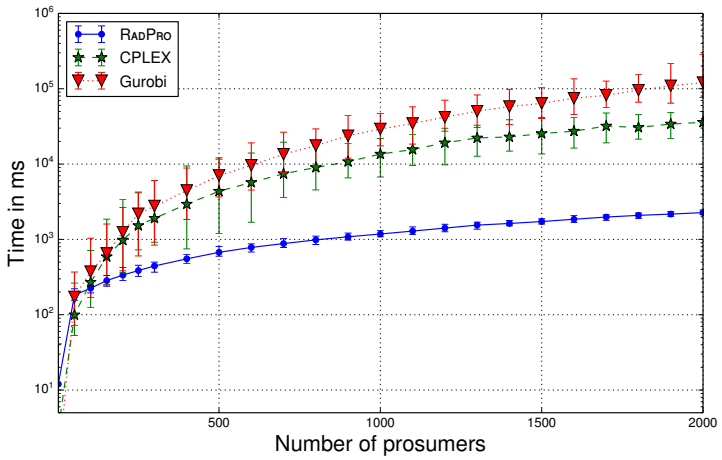

Figure 4: Runtime of RadPro in geometric distribution networks compared to that of CPLEX and Gurobi, $\kappa = 100$.

the degree of the vertex grows, we run RadPro on simpler instances based on star-shaped networks (one central vertex with several neighbors). Offers are generated as in the previous experiment, except for min_j and max_j that instead of being sampled are now fixed to $min_j = 1$, and $max_j = \kappa$. Since prosumers are bidding over every value of its capacity, this represents a worst case scenario. Furthermore, this results on fixed-edge capacities (all the links have the same maximal capacity κ), thus easing the analysis of the results.

Figure 5 shows the speedup –i.e. the ratio $\frac{time(\text{Acyclic-Solving})}{time(\text{RadPro})}$. For instance, at size $n = 7$ (one agent connected to 6 others agents), for $\kappa = 50$, RadPro is $5,000$ times faster than Acyclic-Solving. This clearly highlights how the efficient message computation presented in the previous section allows RadPro to scale up polynomially, while Acyclic-Solving runtime grows exponentially: at size $n = 7$ and $\kappa = 50$, RadPro takes approx. 1 second while Acyclic-Solving takes approx. 83 minutes to solve the problem.

Figure 6 illustrates how RadPro's runtime is affected by the capacity for much larger degrees (these scenarios are unreachable for Acyclic-Solving). There, we see that RadPro solves a problem with capacity 100 and 100 neighbors in less than 1 minute. These results make RadPro a good candidate to tackle large-scale EAP with high branching factor, as it is the case in urban areas.

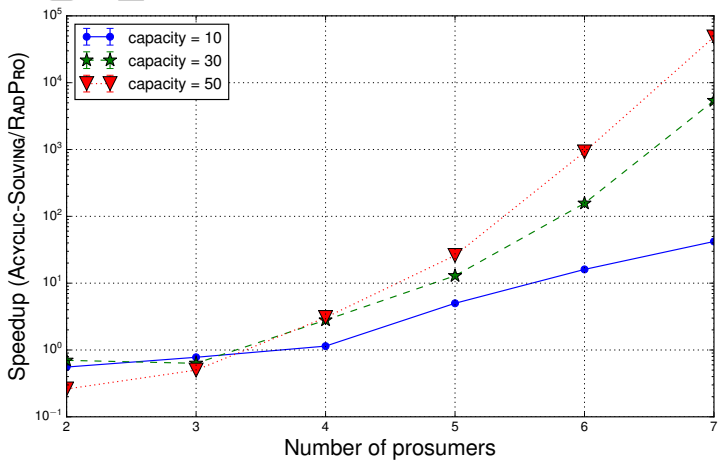

Figure 5: Speedup of RadPro (wrt Acyclic-Solving) in star-shaped trees.

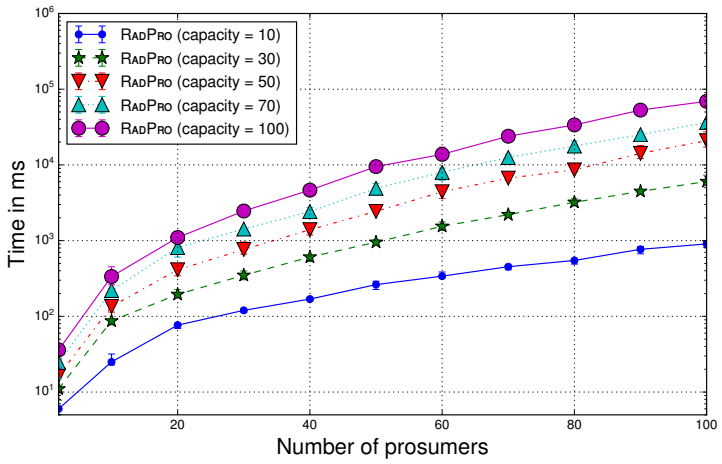

Figure 6: Runtime of RadPro in large star-shaped trees.

7 Mechanism Design

Up to now, we have concentrated on how to provide an efficient and distributed solution to the EAP, disregarding the strategic behavior of prosumers. Here we skim through some game-theoretic considerations.

Mechanisms are composed of both a choice rule and a payment rule [21]. From a mechanism design point of view, the EAP can be understood as the choice rule that selects the energy trades in our network based on the valuations provided by the prosumers. Theorem 1 shows that this choice rule can be assessed in polynomial time. However, we have not proposed any payment rule that establishes how much should each agent pay/receive afterwards.

In their classical work from 1983, Myerson and Satterthwaite [15] proved the impossibility of having an efficient, individual-rational, incentive-compatible, and budget-balanced mechanism in a simple exchange environment in which a buyer and a seller trade a single unit of a given good. This very simple case is isomorph to an energy network with two connected participants where one has available an energy unit that the other one wants to buy. Thus, the impossibility result [15] extends to our setting.

On the other hand, the central result in mechanism design, on the incentive-compatibility of the Vickrey-Clarke-Groves (VCG) mechanism, carries over to our model. Recall that the VCG mechanism allocates goods in the most efficient manner and then determines the price to be paid by each bidder by subtracting from their offer the difference of the overall value of the winning bids and the overall value that would have been attainable without that bidder taking part. That is, this "discount" reflects the contribution to the overall production of value of the bidder in question. The VCG mechanism is strategy-proof: submitting their true valuation is a (weakly) dominant strategy for each bidder. As an inspection of standard proofs of this result reveals [12], this does not depend on the internal structure of the agreements that agents make. Hence, it also applies to our model.

Furthermore, assessing the VCG payment for each prosumer only requires solving a new EAP problem where that particular prosumer is not present. Since solving an EAP with n prosumers has complexity $O(nN_{max}^2C_{max}^2)$, assessing the VCG payments for all participants can be done in $O(n^2N_{max}^2C_{max}^2)$. Thus, the complete VCG mechanism can be assessed in polynomial time.

Further studying mechanism design properties of such markets (including alternative payment rules that could lead to asymptotic efficiency along the lines of [1]) remains as future work.

8 Conclusions and future work

In this paper we have investigated how to enable energy trading in prosumer networks taking into account grid system constraints. We propose to cast the energy trading problem as an optimisation problem, the energy allocation problem (EAP). We then show that the EAP can be formulated as an MIP so that it can

be optimally solved for any network topology by commercial solvers. Given the acyclic configurations of distribution network topologies (radial networks are acyclic, and although ring main and interconnected networks contain cycles, they are configured into acyclic networks by means of switches to supply power), we design RadPro, a novel dynamic programming algorithm to optimally and efficiently solve the EAP over acyclic electricity networks. RadPro is based on the Acyclic-Solving algorithm because this relies on a tree-shaped network, which perfectly fits radial networks. However, because of the exponential message computation required by Acyclic-Solving, it cannot be employed in practice to solve the EAP. We endowed RadPro with an efficient message computation machinery based on a novel algebra of valuations. Thanks to this algebra, RadPro can efficiently compute messages in polynomial time, hence overcoming the limitations of Acyclic-Solving by reducing several orders of magnitude its computation time. Furthermore, our empirical results show that RadPro significantly outperforms both CPLEX and Gurobi in solving time when computing the optimal allocation for the EAP, namely when clearing the market, as the size of the market grows, being 15.8 times faster than the runner-up in the largest scenario tested. In general, our experiments demonstrate that the communication and computation complexities of RadPro clearly position it as a scalable, optimal algorithm for large networks.

There are several paths to future research. First, notice that the message-passing nature of RadPro offers the possibility of solving the EAP in either a centralised or a decentralised (peer-to-peer manner). This is in line with the research challenges posed in [3], and opens the possibility of investigating the economic and environmental impacts of the deployment of a peer-to-peer trading protocol. Second, since future smart grid infrastructures are likely to contain cyclic structures (e.g. mesh), we plan to investigate how to help RadPro effectively cope with cycles. Third, the speed of RadPro together with the convergence of another technologies (like smart metering) makes real-time energy awareness possible. Alternatively to centralized auctions, which operate a day-ahead basis, RadPro's speed makes us consider second- or minute-long time windows, hence allowing prosumers to carry out trades according in (quasi-)real time.

References

[1] L. Chu and Z. Shen. Truthful double auction mechanisms. *Operations research*, 56(1):102–120, 2008.

[2] R. Dechter. *Constraint processing*. Morgan Kauffman, 2003.

[3] European Technology Platform. SmartGrids SRA 2035. Strategic Research Agenda. Update of the SmartGrids SRA 2007 for the needs by the year 2035, March 2012.

[4] Federation of German Industries (BDI). The Energy Industry on the Way to the Internet Age. BDI publication No.439, 2010.

[5] T. Gonen. *Electric power distribution engineering*. CRC press, 2014.

[6] Greenpeace. Decentralising power: An energy revolution for the 21st century. http://bit.ly/1xf1RCk, 2005.

[7] L. L. Grigsby. *Electric Power Generation, Transmission, and Distribution*. CRC press, 2012.

[8] D. Ilic, P. G. Da Silva, S. Karnouskos, and M. Griesemer. An energy market for trading electricity in smart grid neighbourhoods. In *6th IEEE International Conference on Digital Ecosystems Technologies (DEST), 2012*, pages 1–6. IEEE, 2012.

[9] K. Kok, B. Roossien, P. MacDougall, O. van Pruissen, G. Venekamp, R. Kamphuis, J. Laarakkers, and C. Warmer. Dynamic pricing by scalable energy management systems—field experiences and simulation results using powermatcher. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8. IEEE, 2012.

[10] S. Lamparter, S. Becher, and J.-G. Fischer. An agent-based market platform for smart grids. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Industry track*, pages 1689–1696. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[11] K. Leyton-Brown, E. Nudelman, and Y. Shoham. Empirical hardness models. *Journal of the ACM*, 56:1–52, 2009.

[12] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic theory*. Oxford University Press, 1995.

[13] S. J. O. Miller. *Decentralised Coordination of Smart Distribution Networks using Message Passing*. PhD thesis, University of Southhampton, 2014.

[14] J. Mockus. On simulation of the nash equilibrium in the stock exchange contest. *Informatica*, 23(1):77–104, 2012.

[15] R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 1(29):165–281, 1983.

[16] B. Neagu and G. Georgescu. Optimization Possibilities for Radial Electric Energy Distribution Network Routes. *Bul. Inst. Politehnic, Iaşi, LIX (LXIII)*, (Lxiii), 2013.

[17] M. A. Olson, S. J. Rassenti, V. L. Smith, M. L. Rigdon, and M. J. Ziegler. Market design and motivated human trading behavior in electricity markets. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences, 1999. HICSS-32.*, pages 1–27. IEEE, 1999.

[18] Y. K. Penya and N. R. Jennings. Optimal combinatorial electricity markets. *Web Intelligence and Agent Systems*, 6(2):123–135, 2008.

[19] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. *IJCAI International Joint Conference on Artificial Intelligence*, pages 266–271, 2005.

[20] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Putting the 'smarts' into the smart grid: A grand challenge for artificial intelligence. *Commun. ACM*, 55(4):86–97, Apr. 2012.

[21] Y. Shoham and K. Leyton-Brown. *Multiagent systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[22] M. Vinyals, J. A. Rodríguez-Aguilar, and J. Cerquides. Constructing a unifying theory of dynamic programming DCOP algorithms via the generalized distributive law. *Autonomous Agents and Multi-Agent Systems*, 3(22):439–464, May 2011.

[23] P. Vytelingum, S. D. Ramchurn, T. D. Voice, A. Rogers, and N. R. Jennings. Trading agents for the smart electricity grid. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 897–904. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[24] Z. Wang, A. Scaglione, and R. J. Thomas. The Node Degree Distribution in Power Grid and Its Topology Robustness under Random and Selective Node Removals. *2010 IEEE International Conference on Communications Workshops*, (1):1–5, May 2010.

[25] B. M. Weedy, B. J. Cory, N. Jenkins, J. Ekanayake, and G. Strbac. *Electric power systems*. John Wiley & Sons, 2012.

[26] T. K. Wijaya, K. Larson, and K. Aberer. Matching demand with supply in the smart grid using agent-based multiunit auction. *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–6, Jan. 2013.

[27] P. Wolfe. The implications of an increasingly decentralised energy system. *Energy policy*, 36(12):4509–4513, 2008.