eCOOP: Applying Dynamic Coalition Formation to the Power Regulation Problem in Smart Grids

Radu-Casian Mihailescu

Malmö University, School of Technology
Internet of Things and People Research Center
Malmö, Sweden

Sascha Ossowski

Rey Juan Carlos University
Centre for Intelligent Information Technologies
Madrid, Spain

Matthias Klusch

German Research Center for Artificial Intelligence (DFKI)
Saarbruecken, Germany

Abstract

In this work we focus on one particular area of the smart grid, namely, the challenges faced by distribution network operators in securing the balance between supply and demand in the *intraday market*, as a growing number of load controllable devices and small-scale, intermittent generators coming from renewables are expected to pervade the system. We introduce a multi-agent design to facilitate coordinating the various actors in the grid. The underpinning of our approach consists of an online cooperation scheme, ***eCOOP***, where agents learn a prediction model regarding potential coalition partners and thus, can respond in an agile manner to situations that are occurring in the grid, by means of negotiating and formulating speculative solutions, with respect to the estimated behavior of the system. We provide a computational characterisation for our solution in terms of complexity, as well as an empirical analysis against real consumption datasets, based on the macro-model of the Australian energy market, showing a performance improvement of about 17%.

Keywords:
Multi-agent systems, Smart electricity grids, Power regulation, Dynamic coalition formation

---

Email addresses: radu.c.mihailescu@mah.se (Radu-Casian Mihailescu), sascha.ossowski@urjc.es (Sascha Ossowski), matthias.klusch@dfki.de (Matthias Klusch)

March 21, 2016

1. Introduction

Recent years have seen the advent of distributed energy resources (DERs) with particular emphasis on clean generation of electricity, predominantly based on wind and solar power [9]. Albeit representing a sustainable form of energy, renewables pose a major challenge to current electricity networks due to their stochastic behavior. DERs are essentially characterised by small-scale, intermittent and highly unpredictable output. In this context, embedding such devices in the ageing infrastructure of distribution networks requires novel approaches for managing the grid efficiently [12, 10]. Given this setting, the organization of the exchange electricity markets is also expected to change.

Currently, the majority of all power is being traded in what is known as the day-ahead spot market. Here, the following day is discretized over hourly time intervals and the market is cleared the day before, fixing the prices and volumes for the contracted amount of energy. In addition, shortages or excesses of energy are mitigated over the *intraday market*, which is cleared just before the actual power is delivered by producers. Such circumstances may include (but are not limited to) compensating for errors in renewable energy forecasts, smoothing start-up ramps of conventional power plants, correcting instantaneous mismatches between supply and demand and providing short-term contingency power in case of generator or transmission line failures.

Thus, as the network is becoming more reliant on the power generated by DERs, the role of the intraday market is expected to gain significant importance [3]. The goal is then to maximise the usage of clean energy upon its availability and maintain the delicate balance between supply and demand in real-time. In order to do so, demand should be able to adapt to the volatility in supply. This can be achieved assuming the flexibility of consumer to adapt their demand based on incentives provided by the grid operator. Moreover, the system ought to react in real-time to sudden changes of the aggregated generation profile in order to balance supply from intermittent renewable resources, while complying with consumer requirements. In this paper we address the above-identified requirements by proposing a *dynamic coalition formation* (DCF) algorithm, where agents representing consumer provide a bottom-up resolution for contingencies via a coordinated look-ahead response.

The organization of the rest of this paper is as follows. In Section 2 we review some of the related work. Section 3 introduces a new formalism for the intraday power regulation problem in terms of a dynamic coalition formation analysis. We then provide in Section 4 the $eCOOP$ control scheme for the same problem, addressing the challenges of an efficient payoff allocation and augmenting our approach in the context of privacy preservation. Finally, Section 5 provides an empirical evaluation of our scheme. Section 6 concludes.

2. Related Research

Given that the actors participating in the grid (i.e. consumer loads, distributed generators) represent different owners with particular, possibly conflicting user goals and behaviors, deploying an agent-based distributed control over the system holds as the natural approach for our scenario [27]. In this work we aim to apply the multi-agent paradigm to devise a mechanism that enables local adaptability to dynamic situations at runtime and allows coordination, as opposed to the more complex task of centralised management [16].

Similar to our work, multi-agent systems have been proposed in the smart grid domain for the task of demand-side management in a number of studies [33, 36, 28]. Critical peak pricing or spot pricing mechanisms attempt to incentivize agents to adapt their demand, by reducing consumption during peak times [22]. Of course, this may end up in situations where peaks are only temporarily flattened and then shifted to different time intervals, as some of the research has shown [32, 28]. More sophisticated solutions have proposed game-theoretic frameworks, [21, 2], for a coordinated adaptation of the agents' behavior.

Power regulation is however distinct, in that the objective of a corrective action is well defined and localized to a particular region of the grid. Peakload and contingency periods are typically handled by means of adapting the power supply, by firing expensive, carbon-intensive, peaking plant generators. Instead, here, the grid operator provides a request to consumers for a specific power regulation action that needs to be addressed in a timely fashion. While demand-side management may be regarded as a day-ahead scheduling problem, for grid regulation, the response time is constrained within minutes, or to even a couple of seconds.

Due to these challenges, the majority of current methods have been limited to propose solutions that can only be applied in the day-ahead market. For instance, in [35], the authors explore the idea of coupling generation from wind farms with storage facilities, particularly batteries from electric vehicles. Furthermore, the approach follows the assumption of a hierarchical organization, where a group leader computes an optimised schedule to maximize profit, for a fixed number of given participants. In a similar approach, in [7], the authors consider a single owner for the entire system that allows the use of centralised control, based on dynamic programming scheduling.

More relevant to our context, in [13], the authors report some preliminary work on deploying electric vehicles (EVs) for power management in the grid. However, they restrict their study to a small-scale scenario, also assuming centralized control over the set of EVs. This eludes some of the harder problems of operating within limited information environments, where the assumptions of global knowledge and top-down control of centralization no longer hold.

One of the best practices deployed so far to reduce system peak load is represented by the category of direct load control (DLC) approaches, which impose a brute-force on/off strategy to control loads. In [38], the authors report on a study in cooperation with the Taiwan Power Company, where to achieve DLC they use a multi-pass dynamic programming method to schedule the operation of air conditioners in order to reach peak reduction and maximum cost savings. Another DLC scheduling solution is given in [23], where the goal is that of increasing the profit of the utility using a linear programming algorithm. Recognizing the importance of taking user preference and comfort into account, some DLC solutions [30, 39] deploy a logic-based system to model, by means of fuzzy variables, the flexibility of interrupting the air-conditioners and electric water heaters, in an attempt to factor user satisfaction into their model.

Clearly, the above-mentioned approaches pose a series of limitations. DLC methods, although already implemented and in use by some energy companies[1], assume full control over the consumer loads, which can be exercised at will. There is still an inconclusive debate about whether such approaches are actually going to reach mass adoption, especially in the domestic sector where consumer are often reticent to comply with such energy usage violations. Centralised solutions which assume a single owner of the system that has full control over the operation of all loads is evidently not applicable to instances where participants are self-interested stakeholders. Finally, applying various pricing schemes has also been shown to deliver poor results. For example, individual consumers may unilaterally decide to shift consumption from expensive time slots to cheap time slots, thus replacing peaks from one period to another. The problem here is due to consumers *i)* not having a clear perception of the amount of energy that needs to be shifted, *ii)* having an interaction only with the grid operator, while not being aware of the constraints and consumption preferences of other consumers and *iii)* not being able to opt in/out at will, dynamically, for participation in various energy management schemes. Moving towards a decentralized, agent-based setting of the electricity grid, we identify a set of desiderata, that to best of our knowledge all current approaches fail to address. Thus, in more detail, against the existing research, the contribution of this paper is threefold:

1. Provide a new representation of the power regulation problem by formalizing it in the context of dynamic coalitional games;
2. Propose a distributed online protocol for solving this problem given its real-time constraints, where we integrate:
   (a) a cooperation scheme that on the one hand benefits from attractive economic properties and on the other hand is scalable and computationally tractable;
   (b) prediction-based learning for reasoning about future interactions and states of the grid;
   (c) privacy preservation guarantees for non-intrusive negotiations;
3. Present an empirical evaluation of the approach against datasets available from the Australian energy market.

[1] Such as Nest: https://www.nest.com/

3. A Coalitional Game Formulation for Intraday Power Regulation

Currently, the grid operator is responsible for compiling the day-ahead schedule for power generation, which is explicitly passed to the actors in the grid. However, with the advent of renewable generation, these schedules are becoming volatile in nature, as they can be influenced by a wide variety of factors (e.g. wind speed, solar irradiance, consumer patterns, etc.), though their accuracy improves as the time-to-prediction elapses.

Henceforth, we take a standpoint where the grid operator, confronted with the uncertainty regarding both generation and consumption capacities, is running a continuous prediction of both supply and demand in the near future, in order to prepare for reductions in available supply or high-peak demand. While forecasting demand at the distribution or transmission level has been widely studied in the literature and represents a current practice for network operators, with the recent deployment of smart meters and electricity sensors at the household level, utilities are enabled to extract almost real-time information about the energy consumption [11]. This assumption recognises the importance of improving the accuracy and granularity of electricity demand forecasting, however this aspect remains outside the scope of this paper.

In this work, we propose a mechanism owing to which, the grid operator can attempt to manipulate the behavior of consumers. Namely, once it determines that a *control action* needs to be executed, such that power is safely provided from the *substation level* (which delivers electric energy to the distribution grid) to the set of consumers connected to that substation, this information is published and becomes available to the consumers in the respective region of the grid. Normally, due to the small capacity of individual actors, for obtaining a meaningful impact, cooperation and coordination is required. Thus, in return for a monetary incentive, consumers can engage in a collaborative effort to shift demand according to the grid operator's request. In this paper we are only interested in considering shifting consumption, either before or after the initial starting time, however, without altering the overall daily consumption.

More formally, we represent consumers as the set of self-interested agents $\mathcal{A} = \{a_i \mid 0 < i \leq n\}$ that always aim at maximizing their incurred gains. In doing so, we associate with each consumer agent a_i the set of all its deferrable loads $\mathcal{L}^i = \{l_j \mid 0 \leq j \leq |\mathcal{L}^i|\}$, where l_j represents a unique identifier for each load. Based on this correspondence, loads can always be traced back to determine the agent associated with a certain load. In addition, loads are operated over a nonempty and finite set of distinct and successive time periods $\mathcal{T} = \{t_1, \ldots, t_m\}$, which discretize each day, by specifying their initial starting time slots s_j set by the user, their duration d_j, power rating r_j (in kW), as well as the active periods for each load φ_j. Against this background, we introduce the following definitions.

Definition 1. A *corrective action* is a tuple $\alpha_c = \langle t_i, t_j \rangle$, expressing the need to shift demand from time slot t_i to t_j, without affecting the remaining time slots.

Definition 2. A *corrective action request* is a tuple $\langle \alpha_c, p_c, \mathcal{R}, \mathcal{P} \rangle$. The grid operator solicits a set of *corrective action requests*, $\mathcal{D}$, by providing estimations that take the form of a probability distribution $\mathbb{P} : \mathcal{D} \to [0, 1]$, specifying the likelihood p_c of *corrective actions* $\alpha_c \in \mathcal{D}$ to be necessary. Additionally, functions $\mathcal{R}, \mathcal{P} : \mathbb{R} \to \mathbb{R}$ associate respectively, *monetary incentives* to be distributed amongst the members of the coalition that undertakes each task and *penalties* to be imposed for unfulfilled commitments, based on the amount of demand to be shifted.

In short, the grid operator relies on the behavioral flexibility that consumers can offer based on $\mathcal{R}$, which maps demand reductions to monetary rewards.

Definition 3. Each agent $a \in \mathcal{A}$ is characterized by its baseline preferred consumption, discretized over time slots $\mathcal{T} = \{t_1, \ldots, t_m\}$ via the *profile function* β^a that aggregates its schedule:

$$\beta^a(t_k) = \sum_{l_j \in \mathcal{L}} l_j \varphi_j(t_k), \forall t_k \in \mathcal{T}$$

$$\varphi_j(t_k) = \begin{cases} 1 & \text{if } t_k \in [s_j, s_j + d_j] \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Now, we consider that each agent a is characterized by a set of actions, which represent the shifting actions specified by the consumer, which he is willing to take.

Definition 4. An *action* is a tuple $\alpha = \langle l, \Delta \rangle$ that specifies the potential deferment Δ of load l, where $l \in \mathcal{L}$ represents the unique load identifier, while $\Delta \in \{-23, \cdots, 23\}$ specifies the positive or negative integer number of time slots for shifting l, assuming an hourly discretization of the day[2]. For each agent a we denote its *flexibility domain* as the set of possible actions $\chi_a = \cup\{\alpha_j\}$.

Essentially, an action produces an alteration to the initial profile of the agent.

Definition 5. Function $\delta : \chi_a \times \mathbb{R}^{\mathcal{T}} \to \mathbb{R}^{\mathcal{T}}$ captures the changes in consumption for each time slot, for a given profile function β^a and action α of agent a. Let β_1^a and β_2^a denote respectively the consumption of agent a before and after executing α:

$$\delta(\langle \alpha, t \rangle) = \beta_2^a(t) - \beta_1^a(t), \forall t_k \in \mathcal{T} \tag{2}$$

[2]The granularity of the daily discretization can of course be influenced by increasing or decreasing the number of time slots in $\mathcal{T}$. For instance, in case of half-hourly time slots $m = |\mathcal{T}| = 48$ resulting in $\Delta \in \{-47, \cdots, 47\}$

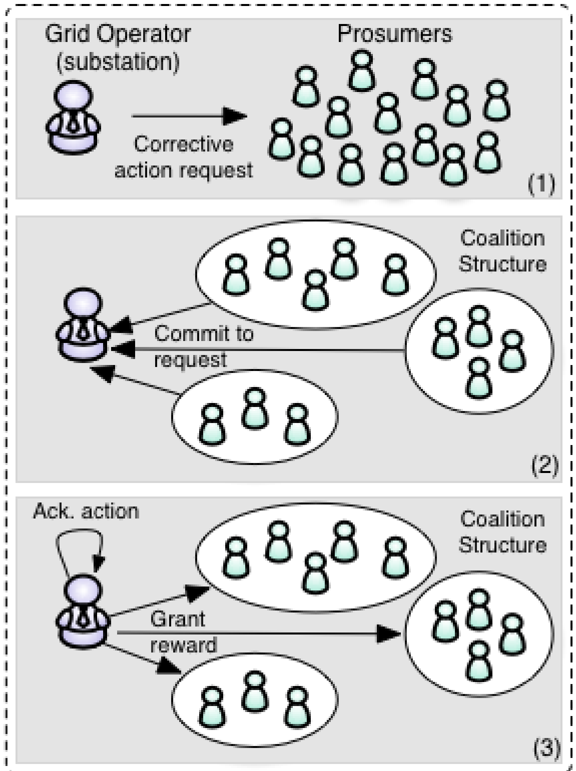

Figure 1: Structure of the coalitional game

Definition 6. Let $\alpha_c = \langle t_i, t_j \rangle$ be a corrective action. Then an action α is *relevant* for α_c if the following holds for some q:

$$\delta(\langle \alpha, t \rangle) = \begin{cases} -q & \text{if } t = t_i \\ q & \text{if } t = t_j \\ 0 & \text{otherwise }, \end{cases} \tag{3}$$

where q denotes the amount of demand shifted from t_i to t_j by α. Similarly, a set of actions $\alpha_{\mathcal{S}} = \{\alpha_1, \cdots, \alpha_m\}$ is *relevant* for α_c if each of the actions $\alpha_i \in \alpha_{\mathcal{S}}$ satisfies Equation 3.

Definition 7. Let $\chi = \bigcup_{a \in \mathcal{A}} \chi_a$. The discomfort cost $w_a : \chi \to \mathbb{R}$ quantifies the marginal loss of agent a in performing a particular action. Noticeably, $w_a(\alpha) = 0, \quad \forall \alpha \not\subset \chi_a$, meaning that actions outside its flexibility domain do not incur any discomfort to agent a.

The business model (see Fig. 1) behind this approach implements a case-by-case monetary reward for each specific corrective action requested by the grid operator to consumers willing to participate. In principle, the reliability of the agents to carry out corrective actions should be the basis in committing the agents for such tasks. Importantly, in our approach the goal is in having the grid operator be exempt from micromanaging the interactions with every agent individually. We address this issue by providing *reward* and *penalty* functions fit for this purpose. Explicitly, the reward function consists of two components (Equation 4), a *superadditive*[3] function f and a *subadditive*[4] function g. The threshold val specifies the point where increasing the amount of demand to be reduced by the agents is no longer desired by the grid operator.

$$\mathcal{R}(q) = \begin{cases} f(q) & \text{if } q < val \\ g(q) & \text{if } q \geq val \end{cases} \tag{4}$$

The penalty $\mathcal{P}$ represents a superadditive function. Noticeably, while the reward incentivizes agents to perform joint actions for a higher return, the penalty denotes a higher cost for failing to deliver these joint actions. Thus, the problem of the grid operator in assessing the agents' reliability of actually delivering their actions is now being transferred to the agents that are incentivized to police themselves, with the scope of avoiding high penalties.

This models in effect a coalition game, where upon a corrective action request of a given probability (inline with Definition 2), agents can reallocate load usage over time schedule $\mathcal{T}$, in order to fulfil the corrective action and be eligible to collect the associated reward. Coalitions are formed based on the expected reward of the coalition and the individual costs that the agents incur in performing the actions. If the corrective action takes place and a coalition delivers the action α as promised, then the reward $\mathcal{R}(\alpha)$ is awarded to coalition[5]. Contrary, if the coalition commits, but then fails to deliver action α as promised, the penalty $\mathcal{P}(\alpha)$ is to be imposed on the coalition.

Definition 8. A *coalition* is a subset of agents $\mathcal{S} \subseteq \mathcal{A}$ that agree to pursue a set of actions $\alpha_{\mathcal{S}}$ called the *joint action* of coalition $\mathcal{S}$:

$$\alpha_{\mathcal{S}} \subseteq \bigcup_{a \in \mathcal{S}} \{\chi_a\}$$

Definition 9. Let $\alpha_c = \langle t_i, t_j \rangle$ be a corrective action. Let coalition $\mathcal{S}$ *commit* joint action $\alpha_{\mathcal{S}}$ to a corrective action request α_C, producing a reduction of demand q. Then, coalition $\mathcal{S}$ is *compliant* if the following holds:

$$\sum_{\alpha_j \in \alpha_{\mathcal{S}}} \delta(\langle \alpha_j, t \rangle) = \begin{cases} -q & \text{if } t = t_i \\ q & \text{if } t = t_j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

[3]Suppose a_1 and a_2 can reduce demand with the amount q_1 and q_2 respectively. Superadditivity implies $\mathcal{R}(q_1 + q_2) > \mathcal{R}(q_1) + \mathcal{R}(q_2)$.

[4]Similarly, subadditivity implies $\mathcal{R}(q_1 + q_2) < \mathcal{R}(q_1) + \mathcal{R}(q_2)$

[5]For the sake of simplicity, hereafter, we overload notation for $\mathcal{R}(\alpha)$, denoting a reward $\mathcal{R}(q)$, where $\alpha = \langle q, \Delta \rangle$. The same holds for function $\mathcal{P}$.

Definition 10. The *cost of coalition* $\mathcal{S}$ sums up the discomfort costs for all actions performed by members a of $\mathcal{S}$:

$$\mathcal{C}(\alpha_{\mathcal{S}}) = \sum_{\alpha_j \in \alpha_{\mathcal{S}}} w_a(\alpha_j), \tag{6}$$

Evidently, an action specifies the unique identifier l of its load, which enables to relate an action α_j to the agent a_i performing it and thus, to use the appropriate discomfort cost function w_{a_i}.

Definition 11. Let $\mathcal{S}$ be a coalition with joint action α_S that is *relevant* for $\alpha_c = \langle t_i, t_j \rangle$. Once coalition $\mathcal{S}$ has *committed* to α_c, then the overall *coalition value* is computed based on whether the action α_S has actually been delivered or not, by subtracting the discomfort cost of all coalition members from the given reward $\mathcal{R}(\alpha_c)$ or penalty $\mathcal{P}(\alpha_c)$, respectively:

$$\nu(\mathcal{S}) = \begin{cases} \mathcal{R}(\alpha_{\mathcal{S}}) - \mathcal{C}(\alpha_{\mathcal{S}}) & \text{if } \alpha_{\mathcal{S}} \text{ is delivered} \\ -\mathcal{P}(\alpha_{\mathcal{S}}) - \mathcal{C}(\alpha_{\mathcal{S}}) & \text{if } \alpha_{\mathcal{S}} \text{ is } \textit{not} \text{ delivered} \end{cases} \tag{7}$$

In other words, if the coalition is compliant in terms of reducing demand to the committed amount, the reward is granted to the coalition, otherwise, if the respective amount is not met, a penalty is incurred by the coalition (see Figure 1). Note that, if the specified amount is not fully met (i.e. only some of the agents deviate from the schedule), the coalition is still penalized regardless of the fact that discomfort costs have already been incurred, therefore the result is a cumulative negative value consisting of both the penalty and the discomfort cost when an action was not delivered.

Example. Consider a 2-agent scenario, where a_1's flexibility domain is represented by the actions $\chi_{a_1} = \{\alpha^1_{a_1} = \langle l_1, \Delta_1 \rangle, \alpha^2_{a_1} = \langle l_2, \Delta_2 \rangle\}$, while for a_2 we denote χ_{a_2} as the action set $\chi_{a_2} = \{\alpha^1_{a_2} = \langle l_3, \Delta_3 \rangle\}$. Function δ determines the modifications in consumption induced by these actions: $\delta(\alpha^1_{a_1}) = \{(-q_1, t_1); (q_1, t_2)\}$; $\delta(\alpha^2_{a_1}) = \{(-q_2, t_3); (q_2, t_4)\}$; $\delta(\alpha^1_{a_2}) = \{(-q_3, t_1); (q_3, t_2)\}$. The notation captures these modifications for each of the altered time slots. For instances with $\alpha^1_{a_1}$ we denote shifting the demand q_1 from time slot t_1 to t_2. Suppose now the grid operator requires the corrective action $\alpha_c = \langle t_1, t_2 \rangle$. Consequently, the coalition of agents a_1 and a_2 could reduce consumption in t_1 with $q = q_1 + q_3$ and shift it to t_2 in compliance with the Grid's request.

4. The *eCOOP* mechanism and its implementation

We are now in the position to define a number of key requirements for our power regulation protocol. Let $\langle \alpha_c, p_c, \mathcal{R}, \mathcal{P} \rangle$ be a corrective action requested dynamically and initiated by the grid operator. Then, for the coalition formation process, the goal is to design a protocol where agents self-organize to form a coalition structure $\mathcal{CS}$, such that each coalition $\mathcal{S} \in \mathcal{CS}$ is *compliant* with the corrective action α_c. Moreover, to guarantee stability we need to divide the reward of each coalition among its members in such a way that consumers have no incentive to deviate. Specifically, the goal is to determine a *payoff distribution* $u : \mathcal{A} \rightarrow \mathbb{R}$ that is: *(i)* individually rational *iff* $\forall a \in \mathcal{S} : u(\{a\}) \geq \nu(\{a\})$, *(ii)* efficient *iff* $\sum_{a \in \mathcal{S}} u(\{a\}) = \nu(\mathcal{S})$ and *(iii)* offers coalitional stability guarantees. Given that the protocol is run distributively among agents, individual valuations, such as discomfort costs and the agents' utilities to form coalitions, need to be communicated between them, without transmitting this data to a central (trusted) site. Thus, an additional requirement is that of preserving data privacy with regard to the agent's self valuation w_{a_i} of possible shifting action in χ_{a_i} during coalition negotiation. Indeed, in Section 4.1 we address in detail the problem of payoff distribution, while in Section 4.2 we bring it all together into the ***eCOOP*** algorithm and also present an extended version[6] of this algorithm, which is able to provide privacy guarantees.

4.1. BSV-Stable Payoff Distribution for Dynamic Environments

Having described the macro dynamics of the power regulation game, we now focus on consumers and how they rationalize about joining potential coalitions. It is important to realize that agents, representing consumers in the grid, operate within significant levels of uncertainty. We model a setting where we consider the sources of uncertainty to be twofold. From the agent's perspective, on the one hand, the challenge is in accurately predicting its user's energy profile and preferences. This means that the deferment actions specified by each consumer via its flexibility domain are regarded as soft constraints, from which he may arbitrarily choose to deviate. Historical data is thus used to profile consumers and estimate the likelihood of actually executing the deferments in χ. On the other hand, in order to increase their coordination efficiency, agents need to build a similar prediction with regard to the expected behavior of potential coalition partners, allowing them to assess the probability of successfully delivering a joint action. For now, keep in mind that these aspects are captured by probability π, which can be computed for any joint action. In Section 5.2 we describe in detail how this is actually implemented in our experiments.

Definition 12. Given agent a's estimation π of the probability of a joint action α_S actually occurring, the *expected utility of agent a in coalition $\mathcal{S}$* is given by factoring in this probability:

$$\mu_a(\mathcal{S}) = \pi\mathcal{R}(\alpha_{\mathcal{S}}) - (1 - \pi)\mathcal{P}(\alpha_{\mathcal{S}}) - \mathcal{C}(\alpha_{\mathcal{S}}) \tag{8}$$

Intuitively, the utility computation considers the *expected* coalitional reward, the *expected* penalty and the cost of performing the joint action.

[6] In bold we denote the extended version of the algorithm, where privacy-preservation is enabled through cryptographic primitives during inter-agent communication.

Definition 13. Let $\mathcal{S}$ be a coalition with joint action α_S. The *expected utility* of $\mathcal{S}$ is the average over the individual expected utilities of the members $a \in \mathcal{S}$:

$$\mu(\mathcal{S}) = \frac{\sum_{a \in \mathcal{S}} \mu_a(\mathcal{S})}{|S|} \tag{9}$$

Recall that we assume the grid operator to be providing estimations that take the form of a probability distribution $\mathbb{P} : \mathcal{D} \to [0, 1]$, that specifies the likelihood p_c of a corrective action $\alpha_c \in \mathcal{D}$ to be necessary. In this context, it is important to emphasize that a corrective action will have different valuations for each agent. Agents with a non-empty flexibility domain χ will engage in a coalition formation procedure, that is, selecting a corrective action α worth pursuing, by playing the best response depending on their preferred strategy:

$$\alpha = \underset{\alpha_c \in \mathcal{D}}{\operatorname{argmax}} \ \ \mathbb{E}[\mathcal{R}(\alpha_c)] \tag{10}$$

A strategy essentially boils down to a particular interpretation of the expected reward associated with a certain corrective action. Notice now that given the fact that corrective actions can only be estimated to occur, we have used in Equation 10 the expected reward term, $\mathbb{E}[\mathcal{R}(\alpha_c)]$. Furthermore, evaluation at this stage occurs before a coalition is actually proposed and given that discomfort costs from the other coalition members are not yet available, we omit them altogether from the computation of the expected gain of the coalition. Thus, the choice for a strategy is solely based on the expected reward of the coalition. Subsequently, each agent may adopt a different strategy according to its user's exposure to risk:

i) *risk-neutral strategy*: select the solution that maximizes the expected coalition reward:

$$\alpha = \underset{\alpha_c \in \mathcal{D}}{\operatorname{argmax}} \ \ p_c \mathcal{R}(\alpha_c)$$

ii) *risk-averse strategy*: selects the solution over a restricted set of corrective actions with high probability for a given threshold h:

$$\alpha = \underset{\alpha_c \in \mathcal{D}}{\operatorname{argmax}} \ \ p_c \mathcal{R}(\alpha_c) \text{ if } p_c > h$$

iii) *risk-seeking strategy*: selects the solution by favouring corrective actions with a high monetary incentive, regardless of a low probability of occurrence:

$$\alpha = \underset{\alpha_c \in \mathcal{D}}{\operatorname{argmax}} \ \ \mathcal{R}(\alpha_c)$$

As previously detailed in Section 3 and based on their expected utilities, agents engage in a coalitional game $\mathcal{G} = (\mathcal{A}, \mu)$. In game theoretic terms, a coalitional game is constituted by a given finite, non-empty *set of agents* and a *characteristic function*, which maps each subset of agents (named a coalition) to a real number. In this particular case, the instantiation of the game pertains to the set of consumer agents $\mathcal{A}$, while the previously introduced *expected utility of a coalition* $\mathcal{S} \subseteq \mathcal{A}$, given by $\mu : 2^{\mathcal{A}} \to \mathbb{R}$, represents the characteristic function of the game. In other words, the number $\mu(\mathcal{S})$ represents the gain that is expected to be achieved by cooperation between the members of coalition $\mathcal{S}$. This is a direct result of the fact that a joint action that can comply with a corrective action request can affect the reward obtained by the individual agents in the coalition. The solution of the game is a configuration $\langle \mathcal{CS}, u \rangle$ that specifies a *payoff distribution* $u : \mathcal{A} \to \mathbb{R}$, which divides the reward of each coalition among its members and a *coalition structure* $\mathcal{CS}$, which partitions the set of agents $\mathcal{A}$ into a set of disjoint coalitions that have been formed. According to the requirements outlined at the beginning of Section 3, the payoff distribution $u(a)$ is supposed to be *individually rational, efficient and stable. Efficiency* means that the joint payoff of the coalition is distributed completely without any loss, while, *individual rationality* implies that no agent gets less than it could obtain by staying alone. *Stability*, means that another aspect needs to be addressed, namely, coming up with a payoff configuration where no agent has an incentive to leave its coalition due to its assigned payoff $u(a)$.

The payoff allocation scheme results from running a negotiation procedure, where agents reschedule loads in order to meet the required constraints. Moreover, considering the real-time requirements for generating the payoff distribution, the protocol should minimize computational and communication demands. However, it is well known that the classical *stability concepts* in coalitional game theory are of high computational complexity [25]. A solution concept identifies some preferable subset of the possible outcomes (solutions of the game). More formally, let Γ be a class of games. Associated with Γ is a set Ω of possible outcomes. Given this notation, we can model a solution concept ϕ for a class of games Γ with outcomes Ω as a function: $\phi : \Gamma \to 2^{\Omega}$, where ϕ is required to satisfy the property that $\phi(\mathcal{G}) \subseteq \Omega_{\mathcal{G}}$, with $\mathcal{G} \in \Gamma$ being a specific game. Game theorists have developed a number of solution concepts, which for every game identify some subset of the possible outcomes of the game. Solution concepts can typically be understood as strategic optimization problems, because they propose to capture some notion of optimality in a strategic setting. Often, they can be interpreted as combinatorial optimization problems, which are computationally hard, thus it becomes imperative that in practice there are ways for efficiently computing solution concepts and that they are computationally tractable.

A well-studied solution concept in coalitional games is the *Shapley value* [31], which defines a *fair* way to distribute the value obtained by a coalition. The Shapley value represents the *expected marginal contribution* that an agent brings to the set of agents preceding him in a coalition, while considering each coalition equally likely to form, as well as the size of the coalitions. The intuition behind the Shapley value is that the payment that each agent receives should be proportional to his contribution averaged over all possible orderings, or permutations, of the players. Formally, for a game $\mathcal{G} = (\mathcal{A}, \nu)$, with $|\mathcal{A}| = n$, the *Shapley value* of a agent $i \in \mathcal{A}$ is denoted by:

$$\phi_i(\mathcal{G}) = \frac{1}{n!} \sum_{P \in \Pi_{\mathcal{A}}} \nu(S_P(i) \cup \{i\}) - v(\mathcal{S}_P(i)) \tag{11}$$

where $S_P(i)$ is the set of all predecessors of i in a given a permutation P from the set of all possible permutations $\Pi_{\mathcal{A}}$ of $\mathcal{A}$.

The Shapley value is particularly appealing because it yields a *unique* payoff allocation, while satisfying a set of easily justifiable axioms: *efficiency*, *symmetry*, *dummy player* and *additivity* [25]. The challenge however is that it can be computationally hard to compute, due to its combinatorial nature. Consequently, as a solution concept for the payoff distribution, in this paper, we adopt an efficient version of the Shapley value introduced by Ketchpel in [14] and further developed in [6]:

Definition 14. The union $\mathcal{S}$ of two disjoint coalitions $\mathcal{S}_1, \mathcal{S}_2$ is called a *bilateral coalition*, with $\mathcal{S}_1$ and $\mathcal{S}_2$ called *constituent coalitions* of $\mathcal{S}$. The *bilateral Shapley value (BSV)* $\sigma(\mathcal{S}_i, \mathcal{S}, \nu)$, $i \in \{1, 2\}$ in the bilateral coalition $\mathcal{S}$ is equivalent to determining the Shapley value of constituent coalitions $\mathcal{S}_i$ in the game $(\{\mathcal{S}_1, \mathcal{S}_2\}, \nu)$:

$$\sigma(\mathcal{S}_i, \mathcal{S}, \nu) = \frac{1}{2}\nu(\mathcal{S}_i) + \frac{1}{2}(\nu(\mathcal{S}) - \nu(\mathcal{S}_k)) \tag{12}$$

with $k = \{1, 2\}, k \neq i$.

In relation to the properties displayed by the Shapley value, we characterise here the BSV solution concept in the game $(\{\mathcal{S}_1, \mathcal{S}_2\}, \nu)$, for the merger $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, according to [15, 6]:

- *efficiency:* $\nu(\mathcal{S}) = \sigma(\mathcal{S}_1, \mathcal{S}, \nu) + \sigma(\mathcal{S}_2, \mathcal{S}, \nu)$
- *symmetry:* if for all $\mathcal{C} \subset \mathcal{A}$ that $\mathcal{S}_1, \mathcal{S}_1 \not\subseteq \mathcal{C}$ we have $\nu(\mathcal{C} \cup \mathcal{S}_1) = \nu(\mathcal{C} \cup \mathcal{S}_2)$, then $\sigma(\mathcal{S}_1, \mathcal{S}, \nu) = \sigma(\mathcal{S}_2, \mathcal{S}, \nu)$
- *dummy player* (non-essential coalition entities receive no payoff): if $\nu(\mathcal{S}) = \nu(\mathcal{S} \backslash \mathcal{S}_i)$ and $\nu(\mathcal{S}_i) = 0$, then $\sigma(\mathcal{S}_i, \mathcal{S}, \nu) = 0$, $\forall \mathcal{S}_i \subset \mathcal{S}$
- for singleton coalitions, BSV equals their self-value: $\sigma(\{a\}, \{a\}, \nu) = \nu(\{a\})$, $\forall a \in \mathcal{A}$

Moreover, both of the constituent coalitions are willing to form $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, if

$$\nu(\mathcal{S}_i) \leq \sigma(\mathcal{S}_i, \mathcal{S}, \nu),\ \forall i \in \{1, 2\} \tag{13}$$

In this manner, *individual rationality* is expressed in Equation 13, while *collective rationality* is captured by Equation 12. Also, notice that Equation 12 can be rewritten[7] such that the surplus of joining S_1 and S_2 into S is distributed equally among S_1 and S_2:

$$\sigma(\mathcal{S}_i, \mathcal{S}, \nu) = \nu(\mathcal{S}_i) + \frac{1}{2}(\nu(\mathcal{S}) - \nu(\mathcal{S}_1) - \nu(\mathcal{S}_2)) \tag{14}$$

Now, given two disjunct coalitions $\mathcal{S}_1$ and $\mathcal{S}_2$, their union $\mathcal{S}$ is called a *bilateral coalition*, while $\mathcal{S}_1$, $\mathcal{S}_2$ are subcoalitions of $\mathcal{S}$. In order for a bilateral coalition $\mathcal{S}$ to be *recursively bilateral* it needs to represent the root node of a binary tree T_S for which *i)* every non-leaf node is a bilateral coalition and its subcoalitions are its children and *ii)* every leaf-node is a single-agent coalition. It follows then, that a coalition structure $\mathcal{CS}$ is *recursively bilateral* iff $\forall \mathcal{S} \in \mathcal{CS}$: $\mathcal{S}$ is recursively bilateral or $\mathcal{S} = \{a\}, a \in \mathcal{A}$.

Definition 15. Given a game $\mathcal{G} = (\mathcal{A}, \nu)$ and a recursively bilateral coalition structure $\mathcal{CS}$, a payoff distribution u is called *recursively bilateral Shapley value stable* iff for each $\mathcal{S} \in \mathcal{CS}$, every non-leaf node $\mathcal{S}^*$ in T_S: $u(\mathcal{S}_i^*) = \sigma(\mathcal{S}_i^*, \mathcal{S}^*, \nu_{\mathcal{S}^*})$, $i \in \{1, 2\}$ with $\forall \mathcal{S}^{**} \subseteq \mathcal{A}$:

$$\nu_{\mathcal{S}^*}(\mathcal{S}^{**}) = \begin{cases} \sigma(\mathcal{S}_k^p, \mathcal{S}^p, \nu_{\mathcal{S}^p}) & \text{if } \mathcal{S}^p \in T_S, \mathcal{S}^* = \mathcal{S}^{**} = \mathcal{S}_k^p, \\ & k \in \{1, 2\} \\ \nu(\mathcal{S}^{**}) & \text{otherwise} \end{cases} \tag{15}$$

Intuitively, this means that by merging two recursively bilateral coalitions, the resulting coalitional value is distributed down the coalitional tree T_S by applying the bilateral Shapley value to the actual payoffs of the respective parent coalition [1]. Note that the BSV properties previously detailed also hold for the recursive BSV payoff distribution [15]. Similar approaches based on BSV computations have been successfully applied in the context of transmission planning problems [5, 4]. Thus, we adopt the notion of a recursively bilateral Shapley value stability due to its computational efficiency and scalability, in contrast to the combinatorial nature of the Shapley value, which becomes hard to compute for coalition sizes that exceed tens of agents. Essentially, this stability concept entails that the agent's payoff configuration conforms to recursively bilateral Shapley value payoffs.

Our aim is to find a recursively bilateral coalition structure $\mathcal{CS}$ for game $\mathcal{G} = (\mathcal{A}, \mu)$, as well as a payoff distribution u that is recursively bilateral Shapley value stable. Notice that such a solution can be constructed incrementally through a bilateral merging process, where the intermediary coalition value is computed according to Equation 15.

[7] By substituting i and k with their designated values and performing arithmetic derivation.

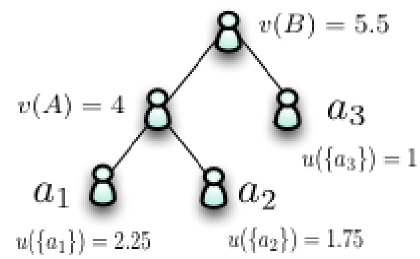

Figure 2: Example of generating payoff configuration through the bilateral Shapley value

Example. (see Fig. 2) Consider the following 3-agent scenario $(\mathcal{A}, \nu)$ with $\mathcal{A} = \{a_1, a_2, a_3\}$, where we demonstrate the calculations for the payoff distribution using the bilateral Shapley value:

- $\nu(\{a_1\}) = 1; \nu(\{a_2\}) = 0.5; \nu(\{a_3\}) = 0.5;$
- $\nu(\{a_1, a_2\}) = 4; \nu(\{a_2, a_3\}) = 2; \nu(\{a_1, a_3\}) = 2$
- $\nu(\{a_1, a_2, a_3\}) = 5.5$

It follows that merging into coalition $A = \{a_1, a_2\}$ and then into coalition $B = \{a_1, a_2, a_3\}$ yields the following payoff distributions: $\sigma(A, B, \nu) = 4 + 1/2(5.5 - 0.5 - 4) = 4.5$; $\sigma(\{a_3\}, B, \nu) = 0.5 + 1/2(5.5 - 0.5 - 4) = 1$. Similarly, the payoff of A is distributed recursively into $\sigma(\{a_1\}, A, \nu) = 2.25$ and $\sigma(\{a_2\}, A, \nu) = 1.75$.

*4.2. The **eCOOP** algorithm*

In the following, we summarize the main steps outlined in the previous sections and provide the pseudocode of our online cooperation scheme. As detailed in Section 3, the grid operator holds the task of monitoring the grid at large, in preparation for various instances of fluctuations, high-peaks, line overloads, reduced DER generation, etc. As a precautionary measure, the grid operator dynamically publishes and updates a list of corrective actions. Because the goal here is *power regulation*, the list of corrective actions is *specifically* prescribed at the *substation* level, which connects a number[8] of individual homes to the distribution grid. We set out from this typical situation in a residential area, where agents, representing households on the level of the low voltage grid, need to coalesce in order to perform the actions indicated by the grid operator. Here, we describe the ***eCOOP***-induced procedure occurring at each substation in the grid, while in Section 5 we empirically demonstrate the aggregated impact of our approach at a grid-level scale.

According to the diagram in Fig. 1, that depicts the overall structure of the game given in Section 3, the coalition formation procedure introduced hereafter

[8] We assume an average of 1000 households per substation [20].

corresponds to the second stage. It is important to point out that the BSV computation detailed in Section 4.1 is implicit to the implementation of the ***eCOOP*** algorithm. That is, generating the coalition structure follows a bilateral coalition formation procedure, kickstarted from the set of singleton coalitions, having each agent reason about the *expected utility* of a potential coalition based on its predictive model, constructed from previous encounters. During each iteration at least one new coalition is formed by merging of two coalitions, where the added value of the coalition merger is to be distributed according to the *bilateral Shapley value* (Eq. 14). Moreover, by adhering to our protocol we enable consumers to distributively converge to solutions that fulfil corrective actions, while also meeting the requirements of computing BSV-stable configurations. Given as input, for each consumer, the *flexibility domain* χ and the associated *discomfort cost* for each action within χ, this information is encapsulated by the agent. Hereafter, the agent represents the consumer for the induced game, interacting with the other agents according to the ***eCOOP*** algorithm[9].

The ***eCOOP*** algorithm is run by every agent in the system. The starting point for each agent is inspecting the global list of corrective actions provided by the grid operator, along with the associated probability of their occurrence. According to the user prescribed strategy, the agent selects a set of target events in *EventQueue* from *CorrectiveActionsList* (lines 3-4), which induce a set of goal-oriented cooperative games that are solved concurrently. Then, for each target event, the agent determines the set of relevant actions according to Definition 6, inspecting for those that shift demand in line with the respective target event. Next, for each target event the algorithm iteratively attempts to construct feasible coalitions starting from the initial set of singleton coalitions (line 6). A coalition represents an agreement between a group of agents for a successful resolution of a corrective action solicited by the grid operator. Based on the information exchange (lines 32-43), each coalition computes internally the expected utility of a bilateral merger with a potential coalition partner. The evaluation of past collaborations are captured in the computation of the utility μ of a coalition in a given coalition merger. Note that function μ is used throughout the algorithm to store and retrieve these values. Then, potential coalition formations are simulated via mergers of subcoalitions by computing the coalition value as the mean of the expected utilities of the merging coalitions (line 16). Following the assessment of potential coalition partners, for a designated candidate set where mergers provide an added reward (line 17), proposals are opportunistically advanced (lines 44-58).

Communication amongst agents assumes the use of time-outs by means of which agents place upper bounds, specifying the amount of time allocated for receiving a reply. In case no reply is received in due time, the particular agent is simply disregarded from being considered as a candidate for coalition formation. This simple request-response protocol is encapsulated by the *Send/Receive* pro-

[9]For instance by having the ***eCOOP*** functionality deployed inside the smart meter.

cedures, which specify respectively the sender and recipient agents (or coalition leaders) and the message itself. Messages are routed to the destination agent and are placed in its message queue. The *Receive* procedure examines the message queue, retrieving *null* only if the timeout expires before a desired message arrives and thus avoiding potential deadlocks during the inter-agent communication. In short, communication with other agents is parallelized by adding a non-blocking agent behavior (thread) each time commnication with another agent commences (line 12). The same principle is applied for the inter-coalition communication, where the coalition leader is responsible for aggregating the expected utilities of a potential coalition merger based on the evaluations of the members of its coalition (lines 37-38).

The simulation phase is followed by the actual coalition formation procedure, which is conducted in a distributed manner. Function $MaxValue$ is used to return the coalition with the highest expected utility from the $Candidate$ set. If proposals are bilaterally accepted, such that a coalitions $\mathcal{S}_1$ and $\mathcal{S}_2$ both evaluate the merger $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$ as a preferable outcome compared to the current configuration $\mathcal{CS}_{iter}$ or to other possible mergers, then during the next iteration of the algorithm the new configuration $\mathcal{CS}_{iter+1}$ will substitute $\mathcal{S}_1$ and $\mathcal{S}_2$ with the newly formed coalition $\mathcal{S}$ (line 25). Additionally, the $Update$ function revises the current configuration based on notifications from other coalition leaders regarding mergers that has occurred at this stage. Also, in the event of a merger, the information is broadcasted not only to the other coalition leaders, but as well, all coalition members are informed about the new configuration (line 51). The procedure terminates once the algorithm converges on a particular coalition structure, meaning that no new coalition mergers are bilaterally acceptable. Note, that the algorithm terminates after at most $|\mathcal{A}|$ rounds, since in each non-final round at least one coalition is formed.

Finally, once the corrective action has been performed by the coalition, the reward is distributed according to the BSV computation for that particular configuration (line 29), resulting in coalitions with stable payoff distributions. Specifically, once the event has elapsed, according to Equation 7, depending on the compliance or non-compliance with the corrective action, a reward or a penalty is determined respectively. The amount is then distributed down the coalitional tree based on the expected coalitional utilities μ (Definition 13) that were used in generating the tree structure. Additionally, agents update their probabilistic model (values of π) with the information inferred from the result of the coalition formation procedure.

In the following we give the agent program of a leader agent a_i in a coalition, where a leader is determined by lexicographic order. The algorithm starts from the set of singleton coalitions, thus initially each agent also plays the role of coalition leader. Once an agent becomes part of a coalition and no longer fulfils the leader role, as a coalition member his role is confined to responding requests from the coalition leader. We focus on the abovementioned tasks performed by a leader agent by providing in Algorithm 1 the pseudocode of the main thread, while Algorithm 2 and Algorithm 3 addresses respectively a number of subroutines corresponding to the communication and negotiation phases. More elaborate ways to establish a coalition leader are beyond the scope of this paper, however, in future work we intend to base this decision on additional factors such as the agents' computational resources (i.e. the agent with the greater computational power is preferred) or network properties (i.e. prioritize communication hubs).

Algorithm 1 represents the main thread of the ***eCOOP*** algorithm.

Data: χ_{a_i}, $\mu_{a_i}(S), \forall S \subset \mathcal{A}$

procedure eCOOP
  $Update(CorrectiveActionsList)$
  Select target event set $EventQueue$ from
   $CorrectiveActionsList$ according to ag. strategy in Eq.10
  for all target $T_i \in EventQueue$ **do**
    $iter = 0$; $CS_{iter} = \{\{a\} | a \in \mathcal{A}\}$;
    repeat
      if $\exists\ S \in CS_{iter}$ so that $a_i = Lead(S)$ **then**
        Det. action set $\alpha_S \subseteq \chi_S$ s.t. Eq. 3 holds $\forall \alpha \in \alpha_S$ w.r.t. T_i
        $Candidate = \emptyset$
        if α_S not null **then**
          for all $S' \in CS_{iter} \setminus \{S\}$ **do concurrently**
            $\tilde{S} = S \cup S'$
            $\mu_{S'}(\tilde{S}) =$ Communicate(S')
            if $\mu_{S'}(\tilde{S})$ not null **then**
              Compute $\mu(\tilde{S}) = \frac{\mu_S(\tilde{S}) + \mu_{S'}(\tilde{S})}{2}$
              if $\mu(\tilde{S}) > \mu(S) + \mu(S')$ **then**
                $Append(S', Candidate)$
              end if
            end if
          end for
          Bilateral Negotiation($Candidate, \mu$)
        end if
        $iter := iter + 1$
        $Update(CS_{iter})$
      else break
      end if
    until $CS_{iter} = CS_{iter-1}$ or $iter = card(\mathcal{A})$
    Compute recursively payoff vector $u(C)$ for all $C \in T_C$ as in Eq. 15
  end for
end procedure

Algorithm 2 carries out the communication with the coalition leader of $S\prime$.

32: **function** COMMUNICATE($S\prime$)
33: $Send(Lead(S), Lead(S\prime), [\alpha_S; \boldsymbol{Enc}(\mathcal{C}(\alpha_S))])$
34: $msg = Receive(Lead(S), Lead(S\prime), [\alpha_{S\prime}; \boldsymbol{Enc}(\mathcal{C}(\alpha_{S\prime}))], timeout)$
35: **if** $msg = null$ **then return** $null$;
36: **else**
37: Aggregate $\mu_S(\tilde{S})$ as in Eq. 9 based on $\mu_a(\tilde{S})$, for all $a \in S$
38: using **Homomorphic Scheme**
39: $Send(Lead(S), Lead(S\prime), \boldsymbol{Enc}(\mu_S(\tilde{S})))$
40: $val = Receive(Lead(S), Lead(S\prime), \boldsymbol{Enc}(\mu_{S\prime}(\tilde{S})), timeout)$
41: **return** $\boldsymbol{Dec}(val)$
42: **end if**
43: **end function**

Algorithm 3 attempts to establish a bilaterally accepted coalition merger.

44: **function** BILATERAL NEGOTIATION($Candidate, \mu$)
45: $S^* := MaxValue(Candidate, \mu)$
46: **if** $\nexists\ MergeProposal(S^*)$ **then**
47: $Send(Lead(S), Lead(S^*), MergeProposal)$
48: **end if**
49: **if** $Receive(Lead(S), Lead(S^*), Agree/MergeProposal, timeout)$ **then**
50: $Send(Lead(S), Lead(S^*), Agree)$
51: Inform members of S of merger with S^* and notify all coal. leaders
52: **else** $Candidate := Candidate \setminus \{S^*\}$;
53: **if** $Candidate$ not null **then**
54: BILATERAL NEGOTIATION($Candidate, \mu$)
55: **else** break
56: **end if**
57: **end if**
58: **end function**

As we have already established, we assumed that agents, representing consumers in the grid, act selfishly, therefore, during the negotiation procedure of forming coalitions, information about the consumers' profile must remain confidential. As the granularity of the data collected and transmitted over the smart grid increases, privacy preservation is becoming an imperative concern [19, 29]. In our approach, this is primarily achieved by communicating to potential coalition partners only a restricted set of actions which the agent is willing to take and their valuation, instead of its complete profile (line 33). However, as this information represents the objective of the negotiation, revealing it may expose

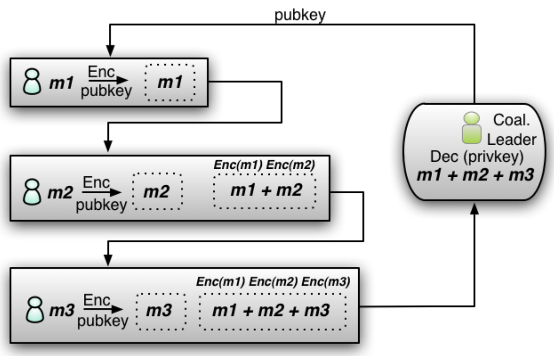

Figure 3: Example of homomorphic cryptosystem

the agent to strategic behavior, in addition to other obvious risks of sharing detailed energy profiles[10].

With these considerations in mind, we provide hereafter an *extended version*[11] of our algorithm, where we are concerned with enhancing the privacy guarantees. To this end, we employ a *homomorphic cryptosystem* that allows agents to perform data aggregation without requiring that the data is decrypted beforehand. In particular, we are interested in applying an efficient additive homomorphic encryption scheme. Let $(pubkey, privkey)$ be a pair of public and matching private keys, $Enc(pubkey, m)$ a function that encrypts message m using the public key *pubkey* and $m = Dec(privkey, Enc(pubkey, m))$ the corresponding decryption function using the private key *privkey*. Then a public key cryptosystem with homomorphic property satisfies:

$$Enc(pubkey, m_1) \cdot Enc(pubkey, m_2) = Enc(pubkey, m_1 + m_2) \qquad (16)$$

We use an efficient instantiation of such a scheme, the Paillier cryptosystem, which provides a fast encryption and decryption protocol. For a more detailed outlook on this procedure we refer the reader to [26]. In our case, this means that agents will only be able to determine the coalition value, instead of the individual agent valuations. More specifically, suppose we are interested in computing the expected utility $\mu(\mathcal{S})$ (Equation 9) of coalition $\mathcal{S}$ for a joint action $\alpha_{\mathcal{S}}$, without disclosing the individual utilities $\mu_a(\mathcal{S})$ of the coalition members $a_i \in \mathcal{S}$, which are required during the computation (line 37). The extended version of the ***eCOOP*** procedure prescribes that the *coalition leader* provides a public key used by the coalition members to encrypt $\mu_a(\mathcal{S})$. A token message is then passed further on to all agents in $\mathcal{S}$ in a lexicographical sequence, which enables the

[10]For instance, private behavior could be derived, which may reflect personal routines, when a location is occupied, work schedules, or other information regarding occupant activities and lifestyle.

[11]In the pseudocode, the additions of the *extended version* appear in bold within the function *Communicate* lines 26-42

agents to construct iteratively the final result using the additive homomorphic scheme (see Fig. 3). Once all the agents have performed this action, the result is sent to the coalition leader, who then decrypts it using the private key and makes the result available to all agents. Based on the homomorphic property, only the aggregated result is decrypted, while intermediate agents aggregate the encrypted data during data forwarding, but cannot decrypt it. Importantly, this approach is guaranteed to achieve a security level of IND-CPA[12], which is the highest security level for homomorphic schemes [18].

The complexity of the proposed DCF (dynamic coalition formation) algorithm is given in the following propositions.

Proposition 1. *The computation complexity of the algorithm is* $\mathcal{O}(pn^2m)$*, where* $n = |\mathcal{A}|$, $m = max_{S \in CS}\{|\alpha_S|\}$, $p = max\{|EventQueue|\}$.

Proof. The number of iterations that the algorithm needs to cycle through is bounded by *a)* the maximum number of events in the global queue $\mathcal{O}(p)$ (line 5); *b)* the maximum number of coalition mergers that may occur $\mathcal{O}(n)$, which corresponds to the formation of the grand coalition (line 7); *c)* $\mathcal{O}(nm)$ the maximum number of operations required in order to construct the list $Candidate$. Besides, the secure multi-party computation requires performing an encryption for every sent message, while the destination agent is needed to add the corresponding decryption. Hence, the overall complexity of the algorithm is $\mathcal{O}(p)\mathcal{O}(n)\mathcal{O}(nm) = \mathcal{O}(pn^2m)$. □

Proposition 2. *The communication complexity of the algorithm in the number of messages per agent is* $\mathcal{O}(mnp)$.

Proof. During each run of the algorithm the number of messages sent by an agent is bounded by $\mathcal{O}(n)+\mathcal{O}(m)$ for the case of coalition leaders, corresponding to inter-coalition negotiations and intra-coalition message passing respectively. Otherwise, a single message specifying μ_a is required to be sent to the coalition leader for each iteration of the algorithm. In addition to this, due to the usage of the cryptographic layer, the coalition leader is also responsible for distributing the public key to each agent, member of its coalition. Thus, given at most pn rounds of the algorithm, the overall number of messages sent by an agent is $\mathcal{O}(mnp)$. □

5. Empirical Evaluation

In this section we provide an empirical evaluation of the coalition formation mechanism introduced in Section 4. First, we explain the details of our experimental setup in Section 5.1. Then, in Section 5.2 we make several remarks about our particular choice of a prediction model. Next, we analyse our empirical results in Section 5.3.

[12] IND stands for indistinguishability and CPA for chosen plaintext attacks

5.1. Experimental Set-up

To evaluate the performance of our proposed algorithm, experiments were conducted on real datasets obtained from the Australian Energy Market Operator (AEMO)[13]. It is important to note that AEMO centrally coordinates the dispatch procedure via a *real-time pricing scheme*, onwards referred as RTP, by pooling the quantities of electricity required by consumers from available generators. Essentially, RTP is a differential pricing scheme, where the cost per unit of electricity varies periodically throughout the day (i.e. peak consumption corresponds to price increases) and it is considered to be the most efficient differential price mechanism used for demand response [8]. Through the experiments we sought to study the emerging consumption patterns induced by the ***eCOOP*** scheme w.r.t the RTP approach[14], for which archives of price and demand data for half-hourly intervals are available. The performance of ***eCOOP*** is demonstrated under the assumption of an expected elasticity in demand, which is modelled and simulated, given the lack of such data to document consumers' preference in providing power regulation services. We further assume that all messages are processed correctly and all agents work properly.

Consumer agents. Specifically, the dataset used for our first set of experiments archives price and aggregated demand, covering the month of September 2012, for each hourly slot, for the New South Wales (NSW) region. While no detailed data was available on individual consumers, we infer this information and construct the agents' profiles β^a by disaggregating the total demand. In doing so, we fix the number of agents to $N = 2.252$ million, derived from the number of households[15] in the NSW region. In Figure 4 we exemplify the real consumption profile for a typical residential household for one day, while the dataset[16] used in the experiments provides a complete year-round consumption profile. Next, based on this profile we generate stochastically, using a uniform distribution, new individual consumers that jointly match the initial *daily* aggregated demand of the AEMO. That is to say, we add random noise, for each time slot of the day, to the average load profile, generating individual consumers, whose total consumption at the end of the day matches the corresponding aggregated demand provided by the AEMO. Thus, in our scenarios, we used simulated consumption patterns for the N agents, where the consumption per agent per time slot is drawn from a uniform distribution $U(p_{min}, p_{max})$. We set the following parameters $p_{min}(t) = -0.15p(t)$ and $p_{max} = 0.15p(t)$, where $p(t)$ denotes the typical consumption at time slot t. We further assume that the strategies representing exposure to risk (Section 4.1) are equally represented in the consumer

[13] http://www.aemo.com.au/Electricity/Data/Price-and-Demand/Aggregated-Price-and-Demand-Data-Files

[14] Note that the RTP performance is not determined experimentally but provided through the AEMO datasets.

[15] http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/1338.1Dec%202010?OpenDocument

[16] Available at UC Irvine Machine Learning Repository; Individual household electric power consumption Data Set: http://archive.ics.uci.edu/ml/datasets.html

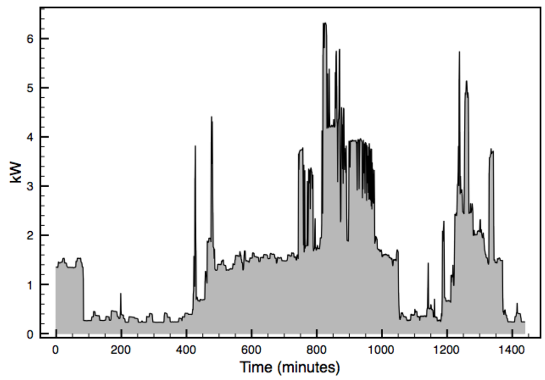

Figure 4: Example of a daily load curve for an individual household

population and that the extent to which consumers are willing to reschedule demand by shifting loads is constrained to $\Upsilon = 25\%$ of the total consumption, denoting their *elasticity of demand* as recent reports suggest [9, 17]. It is estimated that shiftable loads (i.e. washing machines, dish washers) account on average 25% of the electricity usage of a household, the remainder being largely due to entertainment and lightning purposes. For associating shiftable loads with consumer profiles we generate loads with a duration of one time-slot, which are distributed uniformly over the set of time slots $\mathcal{T} = \{1, \ldots, 24\}$, to match the given elasticity of demand. The number of time slots to which each load can be shifted is bounded to $\Delta \in [-5, 5]$, while the power ratings of these loads are uniformly distributed in the set $r \in \{1kW, 2kW, 3kW\}$. Hence, the *flexibility domain* of an agent consists of the set of possible actions, where an action links a shiftable load l to a potential deferment Δ (Def. 4).

Grid operator. A corrective action prescribes that demand needs to be shifted from a time slot t_i to another time slot t_j. Whenever the grid operator determines that the average daily consumption is expected to be exceeded by more than 5%, a corrective action is triggered, requesting that this excess demand is shifted to the following time-slot with a lower than average consumption. Corrective actions are made available to subsets of N, of a fixed sized, which is set to 1000 agents (hence preserving the local character of power regulation at the substation level). Additionally, in order to give a measure of *robustness*, we factored into our simulation random variations in the power supply, accounting for fluctuations from renewable resources, which are estimated to cover about 13% of the total generation [9]. The mean absolute percentage deviation (MAPD) is bounded to 20%. Such instances may also represent the cause for requesting corrective actions in case the abovementioned triggering condition is met. Also, we consider without loss of generality that coalitions perform joint actions successfully with a 90% probability rate, in accordance to recent consumer behavior surveys on customer acceptance retention and response to time based electricity tariffs [34]. This optimistic scenario assumes in fact that 9 out of 10 coalitions are compliant with their respective corrective action requests. Given the previously mentioned fixed maximum size of coalitions, it follows that the performance improvement is linear with this rate.

In specifying a *corrective action request*, the grid operator ought to provide the associated reward and penalty functions (Def. 2). For our scenario, the reward is based on the assumption that the approximate cost otherwise incurred by deploying expensive power plants is instead distributed to consumers willing to reduce demand (hence achieving lower emissions). Evidently, in a real scenario, the grid operator may choose to commit only a fraction of this amount for incentivising consumers. We use the following reward function in Equation 17, supposing that the desired amount of demand to be shifted is q^* and set $\theta = 0.1$ (cents). It is important to point out that besides complying with the properties of Eq. 4, quadratic functions are commonly used to capture the cost of electricity supply (see [28]). We further consider $\mathcal{P}(q) = f(q)$.

$$\mathcal{R}(q) = \begin{cases} f(q) = \theta q^2 & \text{if } q < q^* \\ 0 & \text{if } q \geq q^* \end{cases} \tag{17}$$

5.2. Predictive Model

The aspects of building an estimation model regarding potential coalition partners, based on previous encounters, as well as the agent's own estimated user behavior, have been addressed in Section 4.1. In our experiments, we approach both aspects in a unified manner by including sources of uncertainty in the form of random, uncontrollable variables with probability distributions, that each agent attempts to learn in an online fashion. Recall that for each agent $a \in \mathcal{A}$ there corresponds a set of (deferrable) loads $\mathcal{L}^a$. Essentially, the goal is to learn for a given action α^{l_j}, that shifts a load l_j, the likelihood that the shift occurs to a particular timeslot k. Suppose now that agent a wants to determine the likelihood for each of the actions that constitute its flexibility set χ_a. Let $R = \{r_1, \ldots, r_{|\chi_a|}\}$ denote the set of random variables modelling future, uncontrollable events and $\mathcal{D} = \{D_1, \ldots, D_q\}$, a set of domains for the random variables such that r_i takes values in $D_i = \mathcal{T}$. Let $\sigma : R \rightarrow \chi_a$ be a distribution function of random variables to the agent's actions. Agent a learns $P = \{\pi_1, \ldots, \pi_{|\chi_a|}\}$, which is a set of probability distributions for the random variables, where each distribution $\pi_i : D_i \rightarrow [0, 1]$ defines the probability law for random variable r_i, such that the values of π_i sum up to 1.

Also, there is uncertainty regarding the expected behavior of potential coalition partners, which in turn need to conform to their respective user demands in a timely fashion. Similarly, agent a tracks past encounters with other agents and builds a probability set P_i for each agent a_i. Consequently, we exploit the repeated game structure of the problem to learn a prediction model regarding future interactions and thus infer potential synergies between agents.

In order to compute the set of probabilities P, for the sake of clarity we adopt the *fictitious play* learning model [25][17], where agents observe other agents', as well as their own user behavior. Concretely, for the latter case, the fictitious play requires that agent a models the set of random variables r_i by keeping, for each action of its user $\alpha^{l_j} \in \chi_a$, a count $c^j_{\alpha_k}$ for each timeslot k:

$$\pi^k_{\alpha^{l_j}} = \frac{c^j_{\alpha_k}}{\sum_i c^j_{\alpha_i}} \tag{18}$$

Of note is the fact that particular actions may be enforced by the user by setting the prior counts of the distribution. By default, actions that have never been performed have an equal probability for each time slot.

The same procedure holds for tracking agents that a has been previously exposed to, during preceding runs of the algorithm. Moreover, for computing the probability of a joint action α_S, we average over the individual probabilities of each action $\alpha \in \alpha_S$:

$$\pi_{\alpha_S} = \frac{\sum_{\alpha \in \alpha_S} \pi_\alpha}{|\alpha_S|} \tag{19}$$

5.3. Results

We conducted the experiments using the *Repast* toolkit[18], which is pure Java extended portfolio for simulating distributed agent-based environments and has been previously deployed in *smart grid* scenarios [24]. At the simulation level, managing the execution of the agents' actions is done in a synchronous cyclic fashion, where at each time-step the schedule iterates through the set of agents, executing actions following the given pseudocode in Section 4. Notice, however, that the outcome of ***eCOOP*** is not affected by asynchronicity, as it has been shown for the original BSV algorithm ([6]).

We ran a comparison of the ***eCOOP*** algorithm against the existing RTP mechanism implemented in the Australian market[19]. Results from these experiments are shown in Figure 5, where we plot the average daily consumption patterns (in MW) for a one month period (September 2012). The results are obtained by averaging after repeating the experiments 100 times. Error bars are omitted, as variance between runs was negligible (relative standard deviation below 0.02) and does not improve the readability of the figure.

Based on our numerical experiments we can conclude that our coalition-based approach leads to a significant flattening of the energy consumption curve,

[17] Of course, more complex functions could be considered, but this is beyond the scope of this paper.
[18] http://repast.sourceforge.net
[19] Recall that RTP denotes the actual consumption recorded by the AEMO.

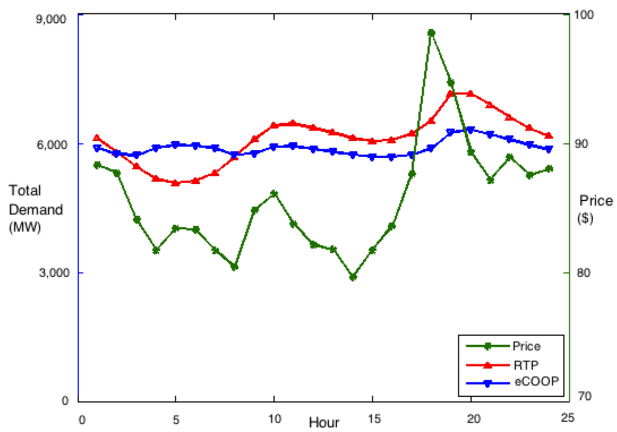

Figure 5: Comparison over aggregated demand patterns by averaging daily consumption (divided into 1h time slots) over a one month period.

as opposed to the RTP solution, although the overall consumption is maintained the same. Intuitively, Figure 5 clearly shows that by applying our proposed algorithm, ahead of critical peak periods, demand can efficiently adapt so that such instances are being prevented from occurring. In order to give a more quantitative measure for our results we consider the *load factor* metric [37], which represents the ratio of average power demand to the maximum (peak) demand. Let β denote the daily total load profile across all consumers over time-schedule $\mathcal{T}$:

$$\beta(t) = \sum_{a \in \mathcal{A}} \beta^a(t), \ \forall t \in \mathcal{T}$$

$$\beta_{peak} = \max_{t \in \mathcal{T}} \ \beta(t) \quad \text{and} \quad \beta_{aveg} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \beta(t) \tag{20}$$

Then the load factor L_f is calculated as:

$$L_f = \frac{\beta_{aveg}}{\beta_{peak}} \tag{21}$$

One of the key challenges behind bringing about the *smart grid* vision is particularly related to the improvements of load factors. Using this metric as an indicator of operational efficiency, we can measure the disparity of the peak from average usage. Thus, the flattening of the demand curve corresponds to an increase of the load factor toward unity. For the one-month interval we have considered in our experiments, our approach produces a 14% increase of the load factor from 0.77 for the RTP scheme to 0.91 when applying the ***eCOOP*** algorithm. Through our approach, we aim that the aggregated load is more evenly distributed across the time schedule $\mathcal{T}$. In contrast, traditional approaches, that aim to nudge user consumption using price signals broadcasted to all, in effect, attempt that each user individually achieves a more balanced load, which need not necessarily be the case for ***eCOOP***. Here, we move the focus from the bilateral interaction between the grid operator and consumers, to a setting where we enable direct interactions between consumers, incentivized by the grid operator, such that their *coordinated* effort achieves a improved L_f value.

The second set of experiments are designed to speculate about future scenarios, when due to a wide adoption of plug-in electric vehicles (PEV) as well as electrification of heating, the proportion of shiftable consumption may increase significantly. Specifically, we are interested in evaluating the impact of our mechanism for variations of the *elasticity of demand* Υ, which denotes the percentage of energy a consumer is willing to defer upon an incoming request from the grid operator. In Figure 6 (a) we plot the effect of different values of Υ upon the load factor of the system. We have already seen from our initial setting very promising results regarding this parameter even for a moderate elasticity of 25%. A considerable increase in elasticity above this value is thus expected to provide negligible increases of the load factor, below 10%. Importantly though, the experiments show that against a consumer with zero elasticity, even a small increase can produce a large impact on the load factor, which is a very encouraging result.

Next, we investigate another criterion for evaluating the energy efficiency of the system, in close relation to the previous case. A high value of the load factor means a decrease in the usage of peaking plants and as a result, a lower carbon footprint. In other words, a reduced consumption during peak intervals and an overall flatter demand means that energy can be generated from less polluting sources. The amount of emissions is in direct correspondence to the energy mix required in order to satisfy the aggregated demand. We use the AEMO dataset[20] to determine the correspondence between a load factor value and the induced CO_2 emissions. We use the Pearson coefficient, r, as a measure of the strength of the linear relationship between these two variables. For the period under consideration we obtain a statistically significant result with $r = 0.81$. Based on these findings, Figure 6 (b) shows the decrease in the amount of carbon emission per kWh by applying the ***eCOOP*** algorithm for various degrees of elasticity in demand. Not surprisingly, a similar pattern can be observed, representing a steep reduction of emissions for small increases in elasticity for up to an approximate 20%.

Lastly, avoiding the need to deploy peaking plants can be directly translated

[20]http://www.aemo.com.au/Electricity/Settlements/Carbon-Dioxide-Equivalent-Intensity-Index

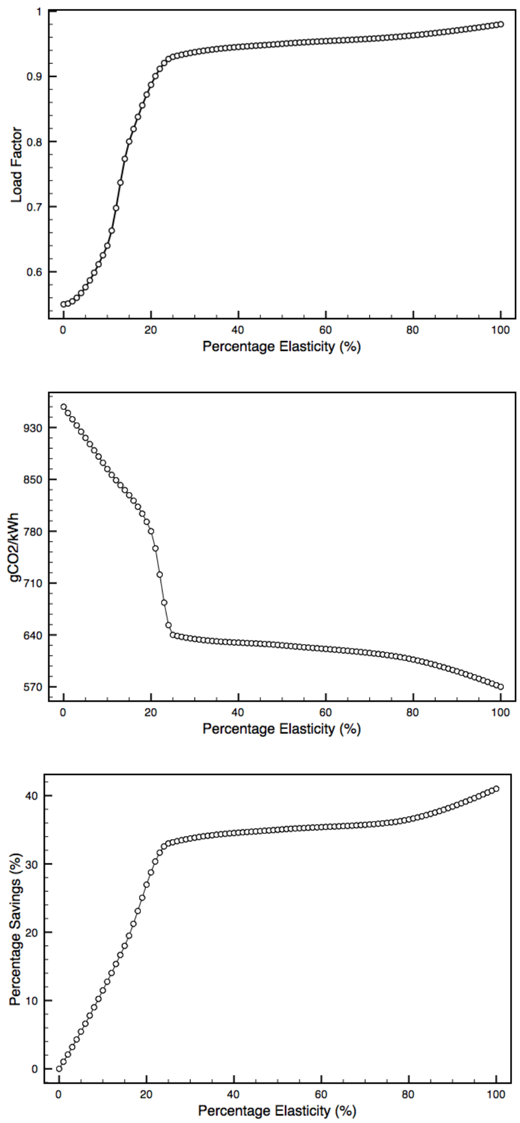

Figure 6: Average evolution of *(a) load factor*, *(b) carbon emissions* and *(c) percetage savings* for different degrees of elasticity of demand

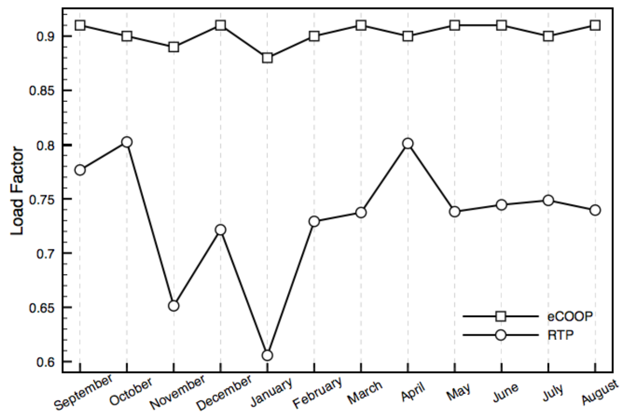

Figure 7: Simulation results for year-round load factor comparison averaged monthly

into consumer savings. In Figure 5 we have also represented the evolution of real-time pricing according to the given aggregated demand, as provided by the AEMO dataset. Observably, off-peak intervals are correlated with lower prices, while higher prices correspond to peak periods. As we have seen, applying our algorithm produces on aggregate a modified consumption pattern. We now determine the average cost savings perceived by consumers, assuming that the cost of electricity (per kWh) is given by the total demand in the system according to the RTP pricing in our dataset. We plot the results in Figure 6 (c), where we give an estimation of percent savings incurred by consumers during the time slot with the highest consumption during the day, for varying degrees of elasticity. This is again an encouraging result, showing an approximate 30% reduction of kWh cost in return for an elasticity of 20%. Further flexibility in consumption can lead to a reduction of up to 40%.

Finally, we provide conclusive results for the performance of our algorithm, demonstrating how ***eCOOP*** outperforms the existing RTP scheme, evaluated over an extended year-round scenario (Sept. 2012 - Sept. 2013). It is important to note that consumption patterns vary throughout the year. Specifically, winter and summer months are known to exhibit increased high-peak intervals due to an intense usage of electricity. We started our experiments investigating an average consumption month. For generality, we now give in Figure 7 the year-round results based on the load factor computation for the two approaches under consideration. It is interesting to observe that RTP produces different outcomes depending on the particular period of the year, highly correlated to the expected consumption usage. For instance, on the one hand, the lowest efficiency is observed during January with a load factor value of 0.6 and on the other hand, April and November represent the highest efficiency months. In contrast, ***eCOOP*** consistently manages to attain a higher efficiency of an approximately 0.9 load factor value, invariantly of the period under consideration, while producing an average 17% improvement.

6. Conclusions

In this paper we are interested in a mechanism that can cope with an increasing amount of intermittent energy generated via renewable resources. We introduced the ***eCOOP*** agent-based algorithm, where look-ahead coalitional negotiations are run within minimal information environments in order to address the dynamism and uncertainty of the energy system. Furthermore, our protocol provides for computing an efficient payoff allocation scheme that guarantees stable coalitions, while the *extended version* offers strong guarantees for satisfying privacy-preservation of sensitive data. We have also provided an empirical evaluation of our approach based on real datasets and have shown the advantages of using it in terms of increased grid efficiency.

It is important to point out that, by design, the intervention of the grid operator addresses explicitly the shifting actions that consumer need to perform in order to collect the reward. In contrast to traditional pricing schemes, this allows us to impose the necessary constraints such that by removing peaks we are not replacing them by new ones, which is also known as the *herding* effect. Moreover, the design of the reward function allows us to transfer the responsibility of determining the reliability of the agents to carry out corrective actions, from the grid operator to the agents themselves.

In this work we have used a standard approach for computing the prediction model, namely fictitious play. In future work we plan to look into more complex models and assess their performance. Also, providing an extensive study on the impact of the communication infrastructure is another interesting future line of work, which does not make the scope of this paper, hence we do not involve with it deeply. We hypothesize that a more efficient way to determine coalition leaders would be to account for the agents' communication and computational resources. Additionally, we assume that ***eCOOP*** is not affected by any data loss or noise, while messages are sent, received and processed exactly in the order prescribed in the pseudocode. However, enhancing the protocol with synchronization procedures initiated by coalition leaders, in order to check for correct message passing along the execution of the algorithm, is one way that could guarantee robustness.

Moreover, we are interested to evaluate our model in scenarios where consumers are not only willing to shift loads to different time intervals given monetary incentives, but may additionally be considering to reduce their total consumption given that a certain revenue could be attained. Expectedly, this ought to further flatten demand and thus, increase the overall efficiency of the grid, especially during periods when generation from renewables is highly fluctuating. Unfortunately, specifying these sorts of parameters, such as the threshold in revenue to which consumers may react and the extent to which their consumption behavior may be altered, remains an open question. With more pilot programs led by utility companies surfacing in this area, we expect that in the future more of this type of data will become available. For the same considerations, given the lack of data required to quantitatively assess how consumers would price potential shifting actions in a realistic setting, in this work, we feel more confident in presenting the result with respect to the elasticity in demand of consumers and only provide an indication of maximum user savings, based on the electricity price decrease corresponding to the levels of demand. In this regard, we aim to dwell on this type of evaluation more deeply in future work subject to the availability of relevant data.

Acknowledgements. This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through grants CSD2007-00022 ("Agreement Technologies", CONSOLIDER-INGENIO2010), TIN2012-36586-C03-02 ("iHAS") and TIN2015-65515-C4-4-R (SURF) as well as by the by the Autonomous Region of Madrid through grant P2013/ICE-3019 ("MOSI-AGIL-CM", co-funded by EU Structural Funds FSE and FEDER") and by the Knowledge Foundation through the Internet of Things and People research profile.

References

[1] Blankenburg, B. and Klusch, M. (2005). BSCA-P: Privacy preserving coalition formation. In Eymann, T., Klugl, F., Lamersdorf, W., Klusch, M., and Huhns, M., editors, *Multiagent System Technologies*, volume 3550 of *Lecture Notes in Computer Science*, pages 47–58. Springer Berlin Heidelberg.

[2] Chalkiadakis, G., Robu, V., Kota, R., Rogers, A., and Jennings, N. R. (2011). Cooperatives of distributed energy resources for efficient virtual power plants. In *Proc. of The 10th International Conference on Autonomous Agents and Multiagent Systems AAMAS*, pages 787–794.

[3] Chris, H. (2008). Electricity markets: Pricing, structures & economics. *European Journal of Control*, 14(4).

[4] Contreras, J., Klusch, M., Vielhak, T., Yen, J., and Wu, F. (1999). Multi-agent coalition formation in transmission planning bilateral Shapley value and kernel approaches. In *Proc. of the 13th Power Systems Computation Conference*, pages 777–787.

[5] Contreras, J., Klusch, M., and Yen, J. (1998). Multiagent coalition formation in power transmission planning: a bilateral Shapley value approach. In *Proc. of the 4th International Conference on Artificial Intelligence Planning Systems*, pages 19–26.

[6] Contreras, J., Wu, F., Klusch, M., and Shehory, O. (1997). Coalition formation in a power transmission planning environment. In *Proc. of 2nd Intl. Conference on Practical Applications of Multi-Agent Systems, PAAM*, pages 21–23.

[7] Costa, L., Bourry, F., Juban, J., and Kariniotakis, G. (2008). Management of energy storage coordinated with wind power under electricity market conditions. In *Probabilistic Methods Applied to Power Systems, 2008. PMAPS '08. Proceedings of the 10th International Conference on*, pages 1–8.

[8] Edward J. Bloustein School of Planning and Public Policy (2006). Assessment of customer response to real time pricing. Technical report. http://ceeep.rutgers.edu/wp-content/uploads/2013/11/customerresponse.pdf.

[9] European SmartGrids Technology Platform (2006). Vision and strategy for european electricity networks of the future. Technical report, European Commission. http://ec.europa.eu/research/energy/pdf/smartgrids_en.pdf.

[10] Hammerstrom, D. J., Brous, J., Chassin, D. P., Horst, G. R., Kajfasz, R., Michie, P., Oliver, T. V., Carlon, T. A., Eustis, C., and Jarvegren, O. M. (2007). Pacific northwest gridwise testbed demonstration projects; part II. grid friendly appliance project. Technical Report PNNL-17079, Pacific Northwest National Laboratory.

[11] Humeau, S. F. R. J., Wijaya, T. K., Vasirani, M., and Aberer, K. (2013). Electricity Load Forecasting for Residential Customers: Exploiting Aggregation and Correlation between Households. In *Sustainable Internet and ICT for Sustainability (SustainIT)*, pages 1–6.

[12] Infield, D., Short, J., Home, C., and Freris, L. (2007). Potential for domestic dynamic demand-side management in the UK. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1 –6.

[13] Kamboj, S., Kempton, W., and Decker, K. S. (2011). Deploying power grid-integrated electric vehicles as a multi-agent system. *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 13–20.

[14] Ketchpel, S. P. (1993). Coalition formation among autonomous agents. In *Proc. of Modelling Autonomous Agents in a Multi-Agent World (MAAMAW)*, pages 73–88.

[15] Klusch, M. (1998). *Kooperative Informationsagenten im Internet*. Dr Kovacz Verlag. ISBN: 3-86064-746-6.

[16] Klusch, M. and Gerber, A. (2002). Dynamic coalition formation among rational agents. *IEEE Intelligent Systems*, 17:42–47.

[17] MacKay, D. (2009). *Sustainable Energy without the hot air*. UIT Cambridge.

[18] Mao, W. (2003). *Modern Cryptography: Theory and Practice*. Prentice Hall Professional Technical Reference.

[19] McDaniel, P. and McLaughlin, S. (2009). Security and privacy challenges in the smart grid. *Security Privacy, IEEE*, 7(3):75 –77.

[20] McDonald, J., Wojszczyk, B., Flynn, B., and Voloh, I. (2013). Distribution systems, substations, and integration of distributed generation. In Begovic, M. M., editor, *Electrical Transmission Systems and Smart Grids*, pages 7–68. Springer New York.

[21] Mihailescu, R.-C., Vasirani, M., and Ossowski, S. (2011). A game-theoretic coordination framework for energy-efficient demand-side management in the smart grid. In *Proceedings of the Ninth European Workshop on Multi-agent Systems (EUMAS)*. http://webshare.mah.se/af4299/Files/EUMAS11/paper.pdf.

[22] Newsham, G. R. and Bowker, B. G. (2010). The effect of utility time-varying pricing and load control strategies on residential summer peak electricity use: A review. *Energy Policy*, 38(7):3289 – 3296.

[23] Ng, K.-H. and Sheble, G. (1998). Direct load control-a profit-based load management using linear programming. *Power Systems, IEEE Transactions on*, 13(2):688–694.

[24] North, M. J., Conzelmann, V., Koritarov, C., Macal, C., Thimmapuram, P., and Veselka, T. (2002). E-Laboratories: agent-based modeling of electricity markets. In *American Power Conference*.

[25] Osborne, M. and Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.

[26] Paillier, P. and Pointcheval, D. (1999). Efficient public-key cryptosystems provably secure against active adversaries. In *ASIACRYPT*, pages 165–179.

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

[28] Ramchurn, S. D., Vytelingum, P., Rogers, A., and Jennings, N. R. (2011). Agent-based control for decentralised demand side management in the smart grid. In *Proc. of The 10th International Conference on Autonomous Agents and Multiagent Systems AAMAS*, pages 5–12.

[29] Rashvand, H., Salah, K., Calero, J., and Harn, L. (2010). Distributed security for multi-agent systems - review and applications. *Information Security, IET*, 4(4):188 –201.

[30] Salehfar, H., Noll, P. J., LaMeres, B., Nehrir, M., and Gerez, V. (1999). Fuzzy logic-based direct load control of residential electric water heaters and air conditioners recognizing customer preferences in a deregulated environment. In *Power Engineering Society Summer Meeting, 1999. IEEE*, volume 2, pages 1055–1060 vol.2.

[31] Shapley, L. S. (1953). A value for n-person games. *Contributions to the theory of games*, 2:307–317.

[32] Strbac, G. (2008). Demand side management: Benefits and challenges. *Energy Policy*, 36(12):4419–4426.

[33] Tompros, S. L., Mouratidis, N. P., Draaijer, M., Foglar, A., and Hrasnica, H. (2009). Enabling applicability of energy saving applications on the appliances of the home environment. *IEEE Network*, 23(6):8–16.

[34] U.S. Department of Energy (2015). Interim report on impacts from the consumer behavior studies. Technical report. http://energy.gov/sites/prod/files/2015/06/f24/ARRA-CBS_interim_program_impact_report_June2015.pdf.

[35] Vasirani, M., Ossowski, S., Kota, R., Cavalcante, R. L. G., and Jennings, N. R. (2011). Using coalitions of wind generators and electric vehicles for effective energy market participation. In *Proc. of the 10th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS, pages 1099–1100.

[36] Vytelingum, P., Voice, T. D., Ramchurn, S. D., Rogers, A., and Jennings, N. R. (2010). Agent-based micro-strorage management for the smart grid. In *Proc. of the 8th International Conerence on Autonomous Agents And Multi-Agent Systems, AAMAS*, pages 39–46.

[37] Weedy, B. M. and Cory, B. J. (2004). *Electric Power Systems*. John Wiley Sons. 4th edition.

[38] Wei, D.-C. and Chen, N. (1995). Air conditioner direct load control by multi-pass dynamic programming. *Power Systems, IEEE Transactions on*, 10(1):307–313.

[39] Yang, H.-T. and Huang, K.-Y. (1999). Direct load control using fuzzy dynamic programming. *Generation, Transmission and Distribution, IEE Proceedings-*, 146(3):294–300.