Renewable and Sustainable Energy Reviews 80 (2017) 1528–1536

Contents lists available at ScienceDirect

Renewable and Sustainable Energy Reviews

journal homepage: www.elsevier.com/locate/rser

Residential demand response: Experimental evaluation and comparison of self-organizing techniques

Ivana Dusparic*, Adam Taylor, Andrei Marinescu, Fatemeh Golpayegani, Siobhan Clarke

Future Cities, Distributed Systems Group, Trinity College Dublin, Ireland

ARTICLE INFO

Keywords:
Smart grid
Intelligent systems
Energy demand response

ABSTRACT

Residential demand response (DR) has gained a significant increase in interest from industrial and academic communities as a means to contribute to more efficient operation of smart grids, with numerous techniques proposed to implement residential DR programmes. However, the proposed techniques have been evaluated in scenarios addressing different types of electrical devices with different energy requirements, on different scales, and have compared technique performance to different baselines. Furthermore, numerous review papers have been published comparing various characteristics of DR systems, but without comparing their performance. No existing work provides an experimental evaluation of residential DR techniques in a common scenario, side-by-side comparison of their properties and requirements derived from their behaviour in such a scenario and analysis of their suitability to various domain requirements. To address this gap, in this paper we present four self-organizing intelligent algorithms for residential DR, which we evaluate both quantitatively and qualitatively in a number of common residential DR scenarios, providing a performance comparison as well as a benchmark for further investigations of DR algorithms. The approaches implemented are: set-point, reinforcement learning, evolutionary computation, and Monte Carlo tree search. We compare the performance of approaches with regards to energy-use patterns (such as reduction in peak-time energy use), adaptivity to changes in the environment and device behaviour, communication requirements, computational complexity, scalability, and flexibility with respect to type of electric load to which it can be applied, and provide guidelines on their suitability based on specific DR requirements.

1. Introduction and background

Due to steady urbanization, the electrical energy grid is facing significant changes in the supply of resources as well as in the type, scale, and patterns of residential user demand. Renewable energy is increasingly used but several forms of it (e.g., wind, solar) are much more variable and intermittent than traditional supply as they depend on the changing weather conditions. Electricity demand is estimated to significantly increase due to the increasing penetration of electric vehicles and electrification of heating. To optimize residential energy usage in this new set of circumstances, numerous residential demand response (DR) techniques have been proposed to shift device usage to the periods of low demand and to coordinate device usage to avoid peaks. The proposed techniques include a wide range of algorithms, including but not limited to centralized linear programming [1], quadratic programming [2], AIMD framework [3], broadcast connection rule [4], market-based approach [5], particle swarm optimization [6], evolutionary algorithms [7], reinforcement learning [8], variable charging/connection rate algorithms [9], expert systems theory [10], and game theory [11].

However, proposed techniques have been evaluated on scenarios addressing different types of domestic electrical devices (e.g., some use only electric vehicles while others are agnostic with respect to type of schedulable device used), on a different scale (e.g., 10 customers in [2], 20,000 in [6]), have assumed different user constraints (e.g., different electric vehicle charging and departure times), and have compared technique performance to different baselines. Multiple review papers have been published comparing subsets of characteristics of DR systems, but without comparing their performance. No existing work provides experimental evaluation of multiple algorithms in common scenarios, using the same simulation environment and same parameters, to directly compare the algorithm performance and behaviour and based on this derive a side-by-side comparison of their properties, requirements or suitability to various domain and device requirements.

However, no single work can provide a comparison of performance of all of the proposed solutions, due to such a vast range of algorithms, and expertise required to design and implement each. To start addressing this gap, in this paper we select a subset of the proposed

* Corresponding author.
E-mail address: ivana.dusparic@scss.tcd.ie (I. Dusparic).

http://dx.doi.org/10.1016/j.rser.2017.07.033
Received 2 December 2016; Received in revised form 27 January 2017; Accepted 9 July 2017
Available online 15 July 2017
1364-0321/

techniques characterised by the use of self-organizing intelligent algorithms, and provide such a comparison. We report on our experiences with their implementation as well as provide quantitative and qualitative analysis of their performance in a common scenario. We have selected four commonly used self-organizing DR algorithms, covering both ends of the spectrum with respect to their architecture (i.e., both centralized and decentralized approaches are included) and requiring both one-way or two-way grid communication, or communication between devices themselves. The different DR techniques presented are analyzed with respect to multiple perspectives: their ability to reduce and shift electrical load to off-peak periods, user utility, scalability, communication requirements, whether intelligence is situated at device end or at the centralized point (e.g., at the transformer level), support for multiple performance policies, energy load data required (current vs. predicted vs. historical), ability to synchronize energy demand of multiple devices, flexibility with respect to changing user schedules, and user privacy.

The rest of this paper is organized as follows. Section 2 presents the details of the algorithms: set-point probabilistic approach, decentralized reinforcement learning approach, evolutionary computation centralized scheduling, and heuristic search algorithm Monte Carlo tree search. Section 3 presents the design of the residential DR scenario in which algorithms have been evaluated. Section 4 presents the performance of the four algorithms as well as the analysis of their characteristics, while Section 5 concludes the paper.

2. Residential DR algorithms

2.1. Background

Increasing market penetration of EVs and renewable energy has resulted in an increased need for flexible demand side management techniques and DR in particular. Numerous research communities are undertaking work addressing various aspects of DR, from social sciences investigating incentives for user uptake, economists addressing the dynamic pricing models, engineering community addressing aspects of grid operation and control and ICT community addressing control algorithms. This work has resulted in a vast number of publications, both presenting individual algorithms specific to certain parts of grid architecture as well as classification and survey papers. For the extensive background of the DR techniques used an interested reader can refer to [12] for DR research challenges and opportunities, [13] and [14] for surveys and classification of architectural components required for DR, [15] for the detailed review of different DR scheduling techniques, communication technologies, and the role of IoT in DR, and [16] for a particularly extensive review of EV charging strategies, which is the most commonly used DR use case in literature. The goal of our paper is not to provide another comprehensive review paper but to complement existing ones by taking the comparison of a subset of DR algorithms further by evaluating their performance in a number of common evaluation scenarios. In this section we provide only a brief summary of the algorithms evaluated, further detail can be found in relevant papers.

2.2. Selected algorithms

When selecting algorithms for evaluation we were conscious that evaluating all different algorithms and their flavours is neither feasible nor necessary, as long as different broad categories of algorithms were represented to enable comparison of their main features. We have analyzed the characteristics of the algorithms presented in the literature and observed two main category distinctions. First is whether DR decisions in a community of households are made centrally or on each individual device/household. The second main difference between algorithms is whether devices are optimizing only their own utility, or there is a coordination between devices ensuring utility of other devices and a community as a whole. For this evaluation, we have implemented and evaluated one algorithm from each of the combinations of the above two categories, as shown in Table 1. We compare centralized and collaborative implementation of evolutionary algorithm (EA), decentralized and collaborative reinforcement learning-based (RL) algorithm, centralized non-collaborative set-point control (SPC), and decentralized non-collaborative Monte Carlo tree search (MCTS). Multiple other intelligent algorithms fall into these categories and share some of the characteristics of the selected ones, for example, particle swarm optimisation can be implemented as a centralized collaborative algorithm similarly to EA, and other game theory approaches can be implemented as decentralized non-collaborative algorithms same as MCTS. We here briefly present specific flavours of the algorithms we implemented before presenting the results of the evaluation and the analysis. As the most frequently-used example of DR in literature is synchronisation of charging on electric vehicles (EVs), we evaluate DR approaches by using a community of EVs, but also discuss if and how each algorithm can be adapted to different device types.

Table 1
Residential DR algorithms evaluated.

	Centralized	Decentralized
Non-Collaborative	Set-Point Control	MCTS
Collaborative	EA	RL

2.2.1. Broadcast-based set-point control

A SPC algorithm is based on one-way broadcast communication, executed centrally (e.g., at a transformer level). SPC calculates the difference between current load, and the maximum available capacity, compares it with the number of devices requiring energy use, and calculates the percentage P of the devices that can be turned on, either for the full required charging duration [4], or for the duration of the next time step, such is in [17], whose implementation we base our algorithm on. The percentage P is broadcast to all devices, and each device turns on with probability P, ensuring that the sum of the resulting load stays around the desired target load.

2.2.2. Evolutionary algorithms

Evolutionary algorithms (EA) are a family of optimization algorithms inspired by biological evolution. At each optimization step, a number of potential solutions is evaluated and those with higher quality (fitness) are propagated to the next generation. The process repeats until a suitable/optimal solution is found. EA are used in DR as centralized scheduling approaches to calculate optimal schedule for a device or multiple devices (e.g., [7,18]). Our solution is inspired by the solution presented in [19] and aims to find the optimal schedule to a number of EVs charging while both ensuring their utility (i.e., achieving desired battery charge), and keeping transformer load under specified limits. The solution is deemed collaborative, as when the schedule for each device is calculated it takes into account the schedule of other devices and their joint impact on the overall load. A schedule is calculated once a day, and requires full knowledge of EV arrival/departure times, battery charge, as well as an estimate of the load for each time step in the charging window.

2.2.3. Monte Carlo tree search

Monte Carlo tree search (MCTS) is a heuristic search algorithm widely applied in game playing. At each time step, for every available action, the game is played-out until the end by selecting random moves; the quality of the final result of every sequence of actions is then used to assign weights to the actions taken along the way, to increase the chance of better moves being selected. Other similar game theoretic approaches have been applied in DR, for example in [20] and in [11], while MCTS itself has been applied in [21] and [22]. Our implementation is based on Parallel MCTS as presented in [23], where each device implements its own MCTS process, effectively competing with other devices for energy usage. We use MCTS as an example of a centralized non-collaborative approach. The quality of every move is evaluated with respect to both device utility (i.e., EV battery charge), as well as overall transformer load in the community. No historical information is needed for MCTS, as decisions are made by estimating the quality of actions in the future, but accurate prediction of future baseload is required.

2.2.4. Reinforcement learning

Reinforcement learning (RL) is an unsupervised learning technique in which an agent repeatedly tries out actions, receives feedback (in a form of a reward) on suitability of those actions, and over time learns the suitability of each available action in each particular state of the environment. RL is used widely in DR to enable devices to learn the most suitable time to use the energy [24,25,8]. In order to learn useful information, extensive interaction with the environment is required, so RL either needs to learn online, while devices turn on and off during their operation, or train on historical data. Numerous variations of RL exist enabling centralized optimization of all devices, decentralized optimization of individual devices (multi-agent approach), learning for all policies in a single process (single policy RL) and learning for multiple policies separately (multi-policy RL). In this paper we evaluate multi-agent multi-policy RL (and specifically, DWL [26]), and use RL as an example of a decentralized collaborative approach. We implement DWL with three system policies: minimize transformer load (where reward is inversely proportional to transformer load), maximize individual battery charge (where reward is proportional to battery charge), and charge during low price period, where agents are rewarded for charging during low price/load period and punished for charging during high price periods.

3. Evaluation scenario

In this section we present an optimal EV battery charging scheduling algorithm, which we use as a baseline for comparison of the selected algorithms, as well as outline the design and details of scenarios in which we compare them.

3.1. Valley-filling algorithm

As a baseline for the evaluation of the surveyed algorithms, we implemented a centralized optimal solution using a valley-filling approach as presented in [27]. While such a centralized scheduling solution is not feasible in large scale implementations due its computational complexity (it is NP-complete [28]), it is guaranteed to be optimal with respect to a defined set of constraints. We aim to optimize transformer load (while fully meeting vehicle charging requirements), by minimizing the charging cost function, where the cost is directly proportional to transformer load. The resulting constrained optimization function is defined as follows:

$$\min F(x) = \min \sum_{j=1}^{m} \left[\sum_{i=1}^{n} (x_{ij} + C_j) \right] x_{ij} \tag{1}$$

where $F(x)$ is the cost function, n the total number of EVs, m the total hours available for charging (assuming the same availability schedules for EVs), x_{ij} the charging decision of vehicle i at time j (0 for not charging & 1 for charging), and C_j the cost of energy at time j. The problem can be solved by dividing the available time into charging slots, computing the minimum amount of charging slots required for each EV, and allocating these charging slots in the periods of low demand. Each EV incrementally updates the overall demand until all EV charging slots are allocated.

3.2. Simulation set up and parameters

All experiments were performed in an open-source energy simulator GridLAB-D [29], developed by the U.S. Department of Energy at Pacific Northwest National Laboratory. Experiments were performed using 90 EVs, and two different values of a required daily mileage for EVs: 30 and 50 miles. Batteries of EVs simulated have a capacity of 30 kWh and charge at a rate of 1.4kWh. Each household also has other energy devices forming the base energy usage, so-called baseload, which ranges from 0.8 kW (during the night) to 3 kW (at peak time), and is taken from the data recorded in Smart Metering Electricity Customer Behaviour Trials in Ireland [30].

3.3. Scenarios

The algorithms were evaluated in three different scenarios to assess their performance in standard conditions, where EV owners allow full control of their charging to the algorithm and where baseload energy use is accurately predicted, as well as in conditions where the original baseload and EV use assumptions do not hold.

Standard Performance Scenario In this basic scenario we evaluated the performance of the algorithms under standard conditions, where the baseload is accurately predicted and all EVs fully respect the algorithms' charging decisions, i.e., are fully automatically controlled.

Changes in Environment Scenario In this scenario, instead of a baseload assumed by the algorithm during training the load pattern is shifted forward by 2 h, and load is increased by 15% throughout. This could, for example, be a result of a sudden cold spell and snowfall, where residents arrive home later than usual due to road and traffic conditions, as well as having an increase in the energy consumption once they arrive home, due to increased heating requirements. This scenario has been designed to evaluate the behaviour of the algorithms in the unpredicted situations; some of the algorithms are heavily reliant on predicted load matching the actual load, and we assess whether they can adjust to baseload changes in real time.

Changes in Device Behaviour In this scenario, a certain number of EV owners manually override the charging decisions; we simulate a scenario where 10 out of 90 EVs do not respect assigned charging slots. This scenario has been simulated to address the fact that not all EV owners will sign up to the automatic DR, or not necessarily always adhere to algorithm's decisions in case of emergencies or sudden daily routine changes. This allows us to evaluate the resilience and adaptivity of algorithms to changes in EV usage patterns, and resulting changes in the uncontrollable load.

4. Analysis

In this section we present both quantitative and qualitative analysis of the algorithms and discuss their performance with respect to shifting load to off-peak times, but also various implementation and deployment issues.

4.1. Performance evaluation

We have conducted performance evaluation of the selected algorithms in standard conditions, as well as in conditions in which underlying conditions (environment and the behaviour of other devices) suddenly changes Fig. 1.

4.1.1. Standard conditions

We evaluate the standard conditions performance of all algorithms in two sub-scenarios: where EVs have a 30 mile round trip to complete during the day (Fig. 2), and a 50 mile round trip (Fig. 3). This varies the duration of afternoon/nighttime charging that EVs need to achieve. The goal of the algorithms is to postpone/shift as much of the energy usage to the nighttime valley period, i.e., to avoid vehicles charging as

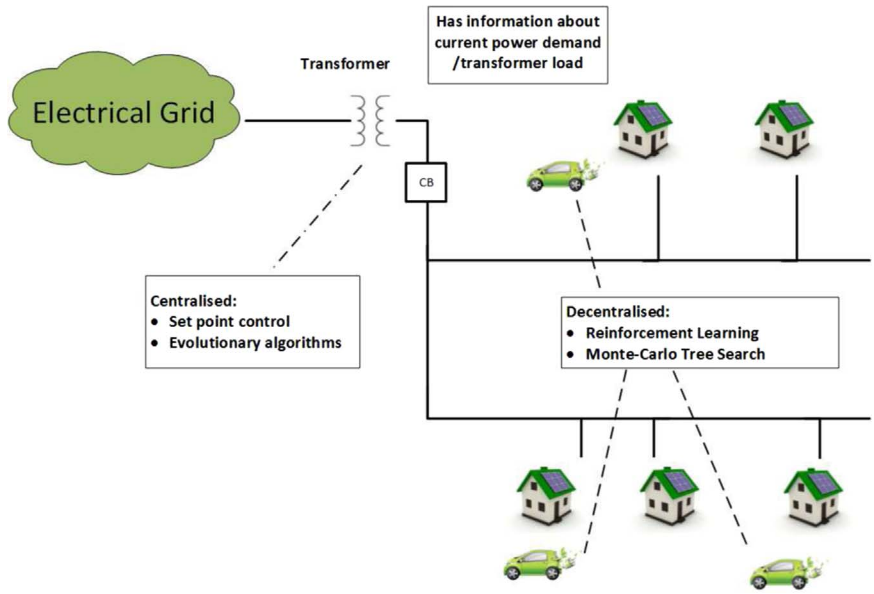

Fig. 1. Demand response system architecture.

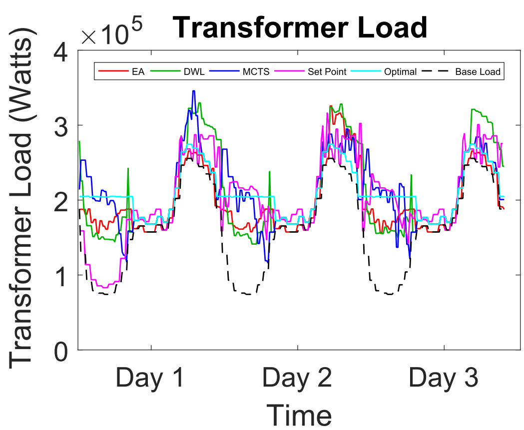

Fig. 2. Transformer load: 90EVs, 30 miles round trip.

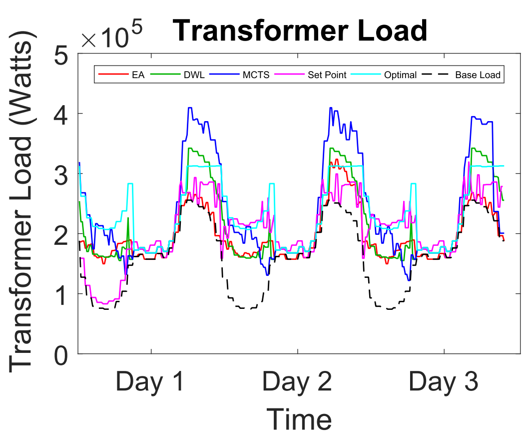

Fig. 3. Transformer load: 90EVs, 50 miles round trip.

soon as they arrive home. The optimal centralized valley-charging algorithm (as described in Section 3.1) has been implemented and shown on the graphs as an indication of how much energy use shifting is possible while still achieving required battery charge. In the 30-mile scenario (Fig. 2), we see that all of the energy use for all 90 EVs can be postponed until the nighttime valley, while in 50-mile scenario (Fig. 3), peak time charging is required as well, adding about 70 kW to the peak load.

All of the algorithms, in both 30-mile and 50-mile scenario, filled the nighttime gap. However, most of them added some unnecessary charging at the peak time too. In particular, MCTS overcharged the most at peak-time, resulting in the highest load increase, as well as failing to utilize the final part of the off-peak period, as EVs were fully charged by then. EA was the best at achieving the smoothness of the curve, i.e., the difference between peak-time load and off-peak load was the smallest. SPC added the least amount of load to the peak period, as the enforced maximum limit was set at cca 300 kW. Downside of this however, is that EVs might not end up fully charged especially after a few days of undercharging, if the specified limit is not sufficient for required charges for all EVs.

4.1.2. Adaptivity to changes in environment conditions

Fig. 4 shows the performance of the algorithms where charging actions have been scheduled or learnt assuming a certain baseload, while a different baseload actually occurred on the specific days. The graphs show two days with default baseload that behaviour has been scheduled/learnt based on, while on day three new a baseload has been introduced. We allowed EA to recalculate full daily schedule based on new baseload prediction, so it performed as well on day three as on other days, fully utilizing new shifted off-peak period. MCTS, in contrast, was not given new baseload prediction (in order to compare the accuracy of scheduling approaches with respect to quality of prediction), and, as expected, following its previous fixed schedule

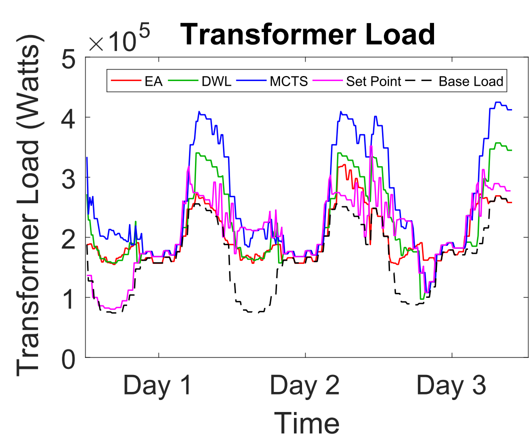

Fig. 4. Transformer load: 90EVs, adapting to baseload changes.

Table 2
Adaptivity of algorithms.

Changes in:	Environment	Device behaviour
SPC	gradually adjust	seamless adapt
RL/DWL	re-learn based on new data	seamless adapt
MCTS	recalc. schedule with new info	recal. schedule at each step
EA	recalc. schedule with new info	recal. schedule at each step

extended a peak-period while failing to utilize at all about one third of the off-peak period. This difference shows that scheduling algorithms can perform as well by recalculating schedules when new accurate environment information is given, however, if they operate under incorrect predictions, the performance severely deteriorates. Similarly, algorithms which learn based on historical conditions, do not adjust as well once current conditions change; DWL learnt that additional charging needs to be done during peak period, and therefore extended the peak period, failing to utilize later available off-peak energy. SPC also resulted in an additional spike at peak-time; it did eventually adjust to the new baseload, but as it only adjusts the broadcast percentage of vehicles to be turned on by a fixed amount at every time step, the change was gradual rather than instantaneous.

Summary of the behaviour of all algorithms in the presence of the changing environment (baseload) conditions is shown in Table 2. Algorithms that pre-schedule the behaviour based on predicted baseload might perform worse when the baseload changes; recalculating schedules is possible only if such prediction exists (e.g., in our scenario in case of EA), which might not always be the case (e.g., in our scenario in case of MCTS). The algorithms that make decisions based only on the current load, are able to adjust to new baseloads, i.e., to account for inaccurate baseload predictions, but that adaptation might not be instantaneous. For example, SPC is set to only increase/decrease the broadcast number of vehicles to charge by a fixed amount at each timestep; therefore the change is gradual rather than instantaneous, still resulting in additional peak-time increases. Similarly, DWL has, based on historical data, learnt that for vehicles with high daily mileage some of the charging needs to be done during the peak-period; as peak-period in the new environment was longer, DWL charged during the peak for longer than necessary. Off-peak period was utilized for charging but not fully, and utilizing it fully would require gradual re-learning of the new baseload conditions.

4.1.3. Adaptivity to changes in device behaviour

Fig. 5 shows the performance of the algorithms when 10 out of 90 EVs do not respect the schedules or learnt actions. The graph shows the first two days where all devices obey the instructions, and changes are introduced on day three. In MCTS, a schedule is calculated for all EVs,

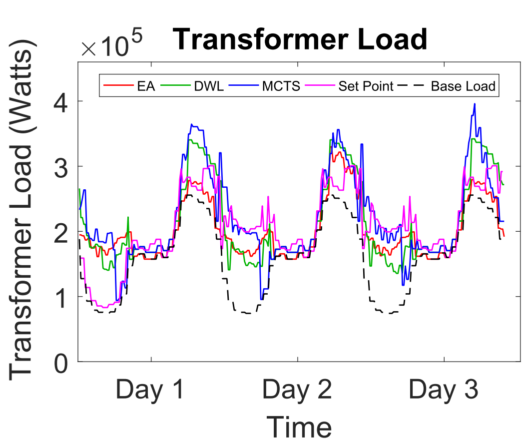

Fig. 5. Transformer load: 90EVs, 10 not respecting schedules.

but only some of them respect it. When EVs which acted independently from MCTS decisions started charging out of sync with the remaining devices, peak load increased, as the remaining EVs were not able to re-adjust their schedule. As a result, an additional ˜35 kW was added at peak time on top of what MCTS added on days one and two. DWL showed no significant change in behaviour as behaviours are not scheduled but react only to existing and predicted load, so the remaining EVs were able to adjust for the behaviour of devices acting independently of DWL decisions. Similarly, SPC showed no significant changes in behaviour as charging decisions are made in reaction to current load only; independently-acting vehicles were simply treated in the same way as pre-existing baseload. EA behaviour was similar to that on day one;˜20 kW were added to the peak load, but it cannot be determined whether that is a result of devices not following the schedule, or because the algorithm scheduled the charging during the peak to allow for full battery charge.

Summary of the behaviour of all algorithms in the presence of the unplanned and unscheduled changes in behaviour of some of the devices is shown in Table 2. Algorithms where a predefined schedule was calculated were not able to adjust to changes in the number of EVs participating in the program and can only do so if the schedule is newly calculated at each time step in which at least one device deviates from the schedule (i.e., most likely all steps). Schedules can, of course, be recalculated when required, and this might be feasible in certain circumstances when changes are not frequent. Algorithms that react to current conditions, such SPC and DWL only were able to seamlessly compensate for independently-acting EVs.

4.2. Characteristics of the algorithms

In the previous section we compared the selected algorithms with respect to their ability to shift the load to off peak times. However, a decision to implement a certain algorithm does not only depend on its performance in a given set of conditions, but also on its additional characteristics such as ease (i.e., cost) of its implementation and deployment, its scalability, communication capabilities available in a given requirement, the amount of device usage data available, its flexibility with respect to the number of policies that devices implement, types of devices, and privacy requirements of the users. We compare those characteristics of the algorithms in this section.

Scalability The size of the residential DR system will have a critical impact on the selection of the algorithm. As households consume very little energy compared to the large industrial consumers currently taking part in DR programmes, energy usage for a significant number (thousands or tens of thousands) of households will need to be aggregated and synchronized in order to make an impact on the overall energy network. For example, to qualify for enrolment in a commercial DR programmes [31], a user needs to have a minimum of 5 MW peak consumption, while an average residential consumer in Ireland has a peak load of ~3 kW [30]. However, in a DR system, a design choice could be made that such a large system is not managed from a single point, but could be divided into a fully decentralized solution, or in a hierarchical solution, where, for example, load on each local transformer (i.e., roughly 230 households) is managed separately.

By definition, decentralized algorithms will have a greater availability [32] as the amount of computation per device does not increase with the number of devices. In RL and MCTS algorithms there is no central component, each device performs its own calculation and decision making, so the system can function exactly the same regardless of the number of devices. In centralized scenarios, computational requirements on the central component increase with the number of devices. The increased complexity is not just the result of more computationally complex fitness function calculations, but also based on the data requirements of algorithms. In the EA case, a central component must have information about daily plans for each of the EVs, which needs to be transferred to it, as well as taken into account as a constraint in the calculation. Scalability becomes increasingly important if the load and EV travel plans are dynamic; recalculation might need to be done every few minutes, rendering such approaches infeasible. In the case of SPC, even though it is a centralized algorithm, the complexity does not increase with the number of devices, as the only calculation that is done is dividing the available capacity with the number of devices.

Communication Requirements. The suitability of an algorithm also depends on its communication requirements, both to account for the infrastructure that is physically available, and also to reduce time requirements and potential security and privacy issues resulting from increased communication. There are three communication links DR algorithms might require: communication from the energy provider (e.g., a transformer or a similar centralized aggregation unit) to the end-user device (or a household), communication from the end-user device to the energy provider, and communication between end-user devices (or households). Requirements for all algorithms with respect these three communication types are summarized in Table 3. There are also three main categories of frequency of this communication: once-off (at the start of the DR scheduling), periodically (e.g., every time there is a change in baseload, EV requirements, or a number of EVs participating), or at every decision time-step.

All algorithms surveyed require communication from the transformer to the end-user, as all algorithms base their decisions on the current energy usage load (baseload) in the community. In addition in SPC, that communication is also required to broadcast instructions to EVs (i.e., charge or do not charge), and in EA it is required to send out a detailed schedule of charging to the end device (once-off or at every time step). For RL and MCTS it is only required to inform agents at the end-user device of the current load in the system.

Communication from the end-user device to the energy provider is required only for SPC and EA algorithms. In SPC, the centralized component residing at the transformer needs to know the number of end-user devices in order to calculate the percentage of them that can be turned on at the same time, so every new device needs to notify the provider that they have joined (e.g., EV arriving home) or that they are about to leave. In EA, the centralized component needs to know additional information about each end-user device, e.g., for an EV it needs to know current battery charge as well as desired departure time and the planned trip duration, in order to calculate required charging time. As RL and MCTS are decentralized algorithms, no calculation happens at the energy provider/transformer end, so no information is sent from the device.

Communication between end-user devices (not necessarily a direct one, but potentially through a local network, or a web service etc.) is only required by RL in order to ensure synchronization between end-device usage, to enable collaboration which prevents all devices turning on at the same time. However, this communication is optional (and therefore marked in brackets in Table 3, as RL can also operate in a non-collaborative mode. In the case of centralized algorithms (SPC and EA), this synchronization is done by a centralized component, and in the case of decentralized MCTS, no direct synchronization is performed. Therefore, RL is the only algorithm which requires additional household to household link. However, this capability allows RL to provide more fine-grained and adaptive synchronization of end user device usage, as shown in Section 4.1.

Location of intelligent DR device add-ons. Closely related to architecture and communication requirements of the algorithms is their complexity in terms of location and number of additional specialized DR-enabled devices that need to be installed in order to enable implementation of residential DR programmes. These devices range from those hosting intelligent algorithms which do schedule calculation and learning, to simple devices which can take only on/off orders remotely in order to turn the participating device on or off. Table 4 summarizes these requirements for each of the algorithms evaluated. In the case of centralized implementations, i.e., SPC and EA, a device capable of running intelligent algorithms (and having communication capabilities as outlined in the previous section) needs to be installed and operating at the transformer/community level. In the case of EA, end-user devices need to be equipped only with on/off switches capable of being remotely controlled by the instructions from the centralized component. In the case of SPC, end-user devices, instead of requiring remotely-controlled on/off capability, need to be able to perform simple calculation which will determine at each timestep whether device should be turned on or off, based on the broadcast percentage of the devices that should be turned on at that time step. Decentralized algorithms DWL and MCTS do not require any additional equipment on the transformer/community level, but each participating devices needs to have capabilities to run the intelligent algorithm which can learn/calculate the schedule for that device.

Financial implications on DR programmes. The discussed complexity and the number of additional hardware devices required for the implementation of a DR programme, as well as communication links required, directly influence the cost of its installation and maintenance. However, estimation of a cost of any DR programme is non-trivial, and apart from hardware costs, depends on multiple additional factors, balancing out both the implementation costs and benefits. Economic analysis of various types of DR programmes has been a subject of extensive research, alongside the research into technical aspects of such implementations reviewed here. Such analysis is outside of scope of this paper, as we focus on technical aspects, but an interested reader can, for example, refer to [33] for a discussion on DR incentives, to [34] for sample of analysis of how benefits of different models differ per region (based on e.g., climate), to [35] for econometrics of real-time pricing in DR systems, and [36] for an extensive review of economical impact of DR systems.

Support for Multiple System Goals. Another consideration when selecting a DR algorithm is how flexible it is with respect to balancing multiple goals of the DR system and the end user. For example, a device/household needs to respect a maximum transformer load set by the energy provider, minimize its own energy cost (by using energy at off-peak times), and ensure EV battery is sufficiently charged. The relative priority of these policies might differ at different times and for different end users. Table 5 summarizes how each of the evaluated algorithms addresses the possibility of multiple existing in the system. SPC algorithm is the simplest in that it only aims to implement the

Table 3
Communication requirements.

	SPC	Rl/DWL	MCTS	EA
Transformer to device	✓	✓	✓	✓
Device to transformer	✓	X	X	✓
Device to device	X	(✓)	X	X

Table 4
Additional DR device requirements.

	SPC	RL/DWL	MCTS	EA
Transformer	Algorithm	X	X	Algorithm
Device	On/off device	Algorithm	Algorithm	On/off device

Table 5
Algorithm support for multiple goals.

	SPC	RL/DWL	MCTS	EA
Single objective function	X	✓	✓	✓
Separate policies	X	✓	X	X

energy provider policy of not exceeding transformer load, regardless of end user policies. As such, multiple policy implementations are not enabled at all. EA and MCTS evaluate each potential next action/schedule against a fitness function, which encodes the importance of the multiple goals and their relative priorities into a single function. The downside of this approach is that the relative priorities between policies are fixed at design time and are the same at each time step, which might not always be the case (e.g., a battery policy might increase in priority as departure time is approaching). MCTS is implemented separately on every device, so relative priorities of policies could differ per individual end-user, while in EA, due its centralized nature, the exact same fitness function is used for all devices. In multi-policy RL, as implemented as described in Section 2.2.4, relative priorities of policies are specific per end-user and can change dynamically if policies become "neglected" over time, enabling additional flexibility. In addition, RL also supports encoding multiple policies into a single objective function, should that be sufficient for the desired implementation, matching the functionality of EA and MCTS.

Support for Multiple Device Types. All algorithms presented have been evaluated on a scenario of charging EVs. This influenced several specific characteristics to the algorithm - the devices generally required extended energy use (e.g., it might take 6–8 h in total to charge the battery), but frequent switching on and off was possible. Other types of devices that have these characteristics, and that all algorithms could easily port to without changes, are for example space heating and cooling, and water heating. Other large household devices, e.g., washing machines, dryers and dishwasher, have a different set of constraints - they require less overall energy (e.g., a total of 1–2 h), but generally should not be interrupted during their operation. All of the presented algorithms could be applied to these types of devices as well, by changing action granularity, e.g., instead of taking actions every 15 min, actions could be taken every 1.5 h. Scheduling algorithms (EA and MCTS) will be affected as the granularity of action will be different at each device, and this additional information will need to be taken into account when calculating the schedule, increasing the complexity of calculation. RL learning time will be affected, as each action taken is used as a learning experience; by having less frequent actions, sufficient learning will be spread over more days than with more frequent actions. SPC would require the least modification to the algorithm; each device which cannot be turned off at the particular timestep can be treated as a baseload and it can skip the decision making, without affecting further timesteps.

Data Requirements. The algorithms presented also differ with respect to the amount of data they require - some require historical data to learn on, some base their decisions only on the current information, and some rely on prediction of the future energy use. Summary of these requirements is presented in Table 6. SPC makes the decisions only on the current load and the number of devices which are interested in energy use. Even if present, due to simplicity of SPC algorithm, historical or future predicted data cannot be utilized. MCTS and EA, as scheduling algorithms, require accurate predictions of the future load (up until the departure time of each EV) to find the most suitable slots for each EV. Similarly, even if historical data was present, MCTS and EA in their current form are not able to utilize it. RL needs historical data (or extensive online learning on current data) to learn the suitable charging actions based on energy use and baseload patterns. Predicted baseload information is not required for RL, but it has been shown that RL can further benefit if accurate predictions are available [26,37].

Table 6
Data requirements per algorithm.

	SPC	RL/DWL	MCTS	EA
Historical	X	✓	X	X
Current	✓	✓	✓	✓
Predicted	X	optional	✓	✓

Table 7
Collaboration between DR devices.

	SPC	RL/DWL	MCTS	EA
Architecture	Centralized	Decentralized	Decentralized	Centralized
Direct	X	✓	X	X
Indirect	✓	✓	✓	✓

Collaboration and synchronisation between device behaviours. Focus of DR programmes is often to even out the energy usage throughout the course of the day - reducing the afternoon/early evening peak, and rescheduling non-critical loads to off-peak nighttime period instead. Achieving this requires a coordination between energy usage by multiple devices, to ensure that not all of them are turned on at the same time creating the peak, but evenly spread out in response to non-reschedulable baseload and usage of energy by other devices. This coordination can be direct, by devices directly cooperating with each other and some deciding to defer their use until off-peak period, or coordinated indirectly by responding to a signal/baseload, or as directed by the central algorithm. Table 7 summarizes how each of the evaluated algorithms achieves device use coordination. In SPC, coordination is achieved by the central controller evaluates the ratio between devices which require the energy and the available energy capacity on top of non-reschedulable baseload, and based on this broadcast the direction to all devices. In EA, exact schedule for all devices is calculated centrally, so constraints and priorities of all devices can be taken into account. In MCTS, each device determines its own schedule, but indirect coordination is achieved by taking baseload into account (where energy usage of other devices is incorporated into the baseload, as seen by each MCTS agent). Only DWL enables direct collaboration between devices, if the communication link between devices is available, as discussed previously. This enables devices to learn the importance of energy use for other devices at each particular time, and yield to them if they have a higher priority or are nearer to their hard time constraints (e.g., departure time approaching).

Privacy. All of the algorithms presented require the end user to share some level of information with the energy provider. Numerous techniques are being proposed in literature to ensure collection and analysis of energy usage data does not infringe on users' privacy (e.g., [38,39]). The privacy characteristics of the algorithm are closely linked with the amount of data they require for their operation as well as their communication requirements; the greater both of these are, the greater the risk for privacy infringement. SPC requires minimal knowledge about the end-user devices, i.e., it only needs to know when the device joins and leaves the system. EA and MCTS require information about EV arrival time and departure time, as well as current battery charge and required daily commute charge. However, as MCTS is decentralized that data is not shared with anyone but only used on the local device, while in EA data for all EVs is sent to the energy provider. RL does not need information on EV schedules as it bases the decisions on current battery charge and load, and it does not share any information with the energy provider, but if it is coordinating usage with other end-user devices it does need to share some information with them. However, only non-identifying information on the learning process (e.g., rewards received at a particular time-step) is shared between a small number of neighbouring households/devices.

5. Conclusion

This paper presented several residential demand response algorithms recently proposed in the literature, experimentally evaluated them in a number of common scenarios, and based on their performance derived and discussed algorithm properties and characteristics.

Algorithms were broadly categorized into centralized vs decentralized and collaborative vs non-collaborative ones, and one algorithm from each combination of categories was selected. Experiments have showed that all algorithms are able to shift load from peak to off-peak hours, with different success rates. Adaptivity of algorithms to changes in the environment and to behaviours of other devices had been evaluated, showing that scheduling algorithms are not as adaptive as those that only react to current environment conditions. However, in cases of discrepancies between historical and current behaviours, scheduling algorithms can adapt quicker if given accurate environment predictions, than those requiring extensive historical episodes to learn on. Numerous other characteristics of the algorithms were discussed, showing that it is not only performance with respect to energy use that is of importance when selecting algorithm for a particular DR implementation, but also considerations addressing communication requirements, data requirements, type of devices supported, and end-user privacy. A summary of the conclusions is presented in the Appendix A in Table 8 which can be used as a guide for selecting an algorithm. Selection of an algorithm suitable for a particular implementation will likely involve several trade-offs. For example, as shown in the table, more sophisticated algorithms require more extensive training data; simple performance optimization can be achieved by SPC based only on the current load information, but in order to satisfy multiple policies and preserve privacy requirements (such as with RL), historic data is also required to enable learning over time. MCTS and EA can provide improved performance in terms of shifting the peak-load, as shown in Section 4.1, however they require baseload prediction information which might not be available. RL and MCTS enable coordination between devices, which improves the performance, however, in order for that to be achieved there needs to be a means of devices communicating with each other (either directly or indirectly), which adds another implementation requirement.

However, advantages and disadvantages of a particular algorithm are highly dependant on the specific circumstance in which DR system is about to be deployed, so we cannot offer a definitive guide on the most suitable algorithm for specific conditions. For example, the choice of algorithm first and foremost depends on the goal of a DR programme - whether it is, for example, to reduce daily peak-usage, to prevent occasional extremely high peaks which jeopardize grid operation, or whether it is to respond to availability of renewable energy in real-time to prevent the need for curtailing. Another factor is how quickly does the algorithm need to respond to DR requests - whether up-front daily/ 24-h notice is available based on energy generation and energy use prediction, or whether they need to be responsive instantaneously. The availability of underlying intelligent and communication infrastructure is also a determining factor, together whether DR actions are provider-actuated or actuated by the end-user. As already discussed, DR programmes can also come with different pricing models, need to be tailored to particular energy patterns, which can, for example, depend on the climate, and obey given local regulatory rules, all of which should be taken into account when selecting the best technology for implementation.

Furthermore, a DR implementation does not necessarily need to utilize only a single algorithm/approach; multiple algorithms can be combined to meet the requirements of the particular implementation, or particular algorithm implementations modified to address specific shortcomings identified in this paper.

Acknowledgement

This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero, and under the Principal Investigator research programme 10/IN.1/I2980 'Self-organising Architectures for Autonomic Management of Smart Cities'.

Appendix A. Summary of the Evaluated Self-Organizing Residential DR algorithm Properties

Please see Table 8.

Table 8
Characteristics of Self-Organizing algorithms for residential DR.

	SPC	RL/DWL	MCTS	EA
Architecture	Centralized	Decentralized	Decentralized	Centralized
Communication	Transformer to device two-way	Transformer to device one-way, device to device two-way	Transformer to device one-way	Transformer to device two-way
Algorithm location	Transformer	Device	Device	Transformer
Type of devices	Frequent on/off possible	Frequent on/off possible	Frequent on/off possible	Frequent on/off possible
Multiple policies	No	Yes, independent multiple policies possible	Yes, only in a single objective function	Yes, only in a single objective function
Data required	Current only	Historic, Current	Current, Predicted	Current, Predicted
Collaboration	Indirect via transformer	Indirect via transformer; direct device collaboration	Indirect via transformer	Indirect via transformer
User interference	Yes	Yes	No, new schedule require	No, new schedule required
Privacy	Partial, no specific info sent to transformer but whether device in use or not	Yes, no device info sent to transformer	Yes, no device info sent to transformer	No, device info has to be sent to transformer

References

[1] Richardson P, Flynn D, Keane A. Optimal charging of electric vehicles in low-voltage distribution systems. IEEE Trans Power Syst 2012;27(1):268–79. http://dx.doi.org/10.1109/TPWRS.2011.2158247.

[2] McNamara P, McLoone S, Efficient predictive demand response using laguerre functions. in: Power and Energy Society General Meeting (PES), 2013 IEEE, 2013, pp. 1–5. http://dx.doi.org/10.1109/PESMG.2013.6672963.

[3] Studli S, Crisostomi E, Middleton R, Shorten R. A flexible distributed framework for realising electric and plug-in hybrid vehicle charging policies. Int J Control Ahead–p 2012:1–16. http://dx.doi.org/10.1080/00207179.2012.679970.

[4] Turitsyn K, Sinitsyn N, Backhaus S, Chertkov M, Robust broadcast-communication control of electric vehicle charging. in: SmartGridComm, 2010, pp. 203–207. http://dx.doi.org/10.1109/SMARTGRID.2010.5622044.

[5] Weckx S, D'hulst R, Claessens B, Driesensam J. Multiagent charging of electric vehicles respecting distribution transformer loading and voltage limits. IEEE Trans Smart Grid 2014;5(6):2857–67. http://dx.doi.org/10.1109/TSG.2014.2345886.

[6] Faria P, Soares JP, Vale ZA, Morais H, Sousa TM. Modified particle swarm optimization applied to integrated demand response and DG resources scheduling. IEEE Trans Smart Grid 2013;4(1):606–16. http://dx.doi.org/10.1109/TSG.2012.2235866.

[7] Logenthiran T, Srinivasan D, Shun TZ, Demand side management in smart grid using heuristic optimization, Smart Grid, IEEE Transactions on 3 (3). http://dx.doi.org/10.1109/TSG.2012.2195686.

[8] Wen Z, Maei HR, ONeill D, Optimal demand response using device based reinforcement learning, Tech. rep., Stanford University (2012).

[9] Harris C, Dusparic I, Galván-López E, Marinescu A, Cahill V, Clarke S, Set point control for charging of electric vehicles on the distribution network, IEEE PES ISGT.

[10] Dam Q, Mohagheghi S, Stoupis J, Intelligent demand response scheme for customer side load management. in: Energy 2030 Conference, 2008. ENERGY 2008. IEEE, 2008, pp. 1–7. http://dx.doi.org/10.1109/ENERGY.2008.4781013.

[11] Maharjan S, Zhu Q, Zhang Y, Gjessing S, Basar T. Dependable demand response management in the smart grid: A stackelberg game approach. IEEE Trans Smart Grid 2013;4(1):120–32. http://dx.doi.org/10.1109/TSG.2012.2223766.

[12] Law YW, Alpcan T, Lee V, Lo A, Marusic S, Palaniswami M, Demand response architectures and load management algorithms for energy-efficient power grids: A survey. in: KICSS, 2012, pp. 134–141. http://dx.doi.org/10.1109/KICSS.2012.45.

[13] Siano P. Demand response and smart grids: a survey. Renew Sustain Energy Rev 2014;30:461–78, [C].

[14] Gelazanskas L, Gamage K. Demand side management in smart grid: a review and proposals for future direction. Sustain Cities Soc 2014;11(2):22–30.

[15] Haider HT, See OH, Elmenreich W. A review of residential demand response of smart grid. Renew Sustain Energy Rev 2016;59:166–78, [C].

[16] Emmanouil Rigas NB, Ramchurn Sarvapali D, Managing electric vehicles in the smart grid using artificial intelligence: A survey, IEEE Transactions on Intelligent Transportation Systems.

[17] Harris C, Dusparic I, López EG, A. Marinescu, Cahill V, Clarke S, Set point control for charging of electric vehicles on the distribution network. in: IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2014, Washington, DC, USA, February 19-22, 2014, 2014, pp. 1–5. http://dx.doi.org/10.1109/ISGT.2014.6816415.

[18] Gomes A, Antunes C, Martins A. A multiple objective approach to direct load control using an interactive evolutionary algorithm. IEEE Trans Power Syst 2007;22(3):1004–11. http://dx.doi.org/10.1109/TPWRS.2007.901468.

[19] Galván-López E, Taylor A, Clarke S, Cahill V, Design of an automatic demand-side management system based on evolutionary algorithms. in: SAC, 2014.

[20] Mohsenian-Rad A-H, Wong V, Jatskevich J, Schober R, Leon-Garcia A. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. IEEE Trans Smart Grid 2010;1(3):320–31. http://dx.doi.org/10.1109/TSG.2010.2089069.

[21] Wijaya TK, Papaioannou TG, Liu X, Aberer K, Effective consumption scheduling for demand-side management in the smart grid using non-uniform participation rate. in: Sustainable Internet and ICT for Sustainability, SustainIT 2013, 2013, pp. 1–8. http://dx.doi.org/10.1109/SustainIT.2013.6685188.

[22] Galvan-Lopez E, Harris C, Trujillo L, Rodriguez-Vazquez K, Clarke S, Cahill V, Autonomous demand-side management system based on monte carlo tree search. in: Energy Conference (ENERGYCON), 2014 IEEE International, 2014, pp. 1263–1270. http://dx.doi.org/10.1109/ENERGYCON.2014.6850585.

[23] Golpayegani F, Dusparic I, Clarke S, Collaborative, parallel monte carlo tree search for autonomous electricity demand management. in: Sustainable Internet and ICT for Sustainability (SustainIT), 2015, 2015, pp. 1–8. http://dx.doi.org/10.1109/SustainIT.2015.7101360.

[24] O'Neill D, Levorato M, Goldsmith A, Mitra U, Residential demand response using reinforcement learning, in: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, 2010, pp. 409–414. http://dx.doi.org/10.1109/SMARTGRID.2010.5622078.

[25] Dimeas A, Hatziargyriou N, Multi-agent reinforcement learning for microgrids. in: PES General Meeting, 2010 IEEE, 2010. http://dx.doi.org/10.1109/PES.2010.5589633.

[26] Dusparic I, Harris C, Marinescu A, Cahill V, Clarke S, Multi-agent residential demand response based on load forecasting, in: SusTech 2013, IEEE, 2013, pp. 90–96.

[27] Chen N, Quek TQS, Tan CW, Optimal charging of electric vehicles in smart grid: Characterization and valley-filling algorithms. In: IEEE Proceedings of the Third International Conference on Smart Grid Communications, SmartGridComm 2012, Tainan, Taiwan, November 5-8, 2012, 2012, pp. 13–18. http://dx.doi.org/10.1109/SmartGridComm.2012.6485952. URL ⟨http://dx.doi.org/10.1109/SmartGridComm.2012.6485952⟩.

[28] Zhu M, Liu XY, Kong L, Shen R, Shu W, Wu MY, The charging-scheduling problem for electric vehicle networks. in: 2014 IEEE Wireless Communications and Networking Conference (WCNC), 2014, pp. 3178–3183. http://dx.doi.org/10.1109/WCNC.2014.6953026.

[29] GridLAB-D. U.S. Department of Energy at Pacific Northwest National Laboratory. ⟨http://www.gridlabd.org/⟩.

[30] Irish Social Science Data Archieve. Comission for Energy Regulation, CER Smart Metering Project, Smart Metering Electricity Customer Behaviour Trials 2009–2011, ⟨http://www.ucd.ie/issda/data/commissionforenergyregulationcer/⟩, last [accessed July 2014].

[31] The Independent Electricity Systems Operator, Ontario, Canada, Industrial Conservation Initiative Eligibility Worksheet for Distributors, ⟨http://www.ieso.ca⟩.

[32] Sun Q, Han R, Zhang H, Zhou J, Guerrero JM, A multi-agent-based consensus algorithm for distributed coordinated control of distributed generators in the energy internet, IEEE Transactions on Smart Grid.

[33] Wijaya TK, Vasirani M, Villumsen JC, Aberer K, An Economic Analysis of Pervasive, Incentive-Based Demand Response. in: IEEE International Conference on Smart Grid Communications (IEEE SmartGridComm), 2015.

[34] Honeywell International Inc., Structuring a residential demand response program for the future, Tech. rep. (2011).

[35] Horowitz SR, Topics in residential electric demand response, Horowitz SR, Topics in residential electric demand response, Ph.D. Thesis, Carnegie Mellon University (2012)., Carnegie Mellon University (2012).

[36] Conchado A, Linares P, The Economic Impact of Demand-Response Programs on Power Systems. A Survey of the State of the Art, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 281–301.

[37] Marinescu A, Dusparic I, Taylor A, Cahill V, Clarke S, P-marl: Prediction-based multi-agent reinforcement learning for non-stationary environments. in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1897–1898.

[38] Cardenas A, Amin S, Schwartz GA, Privacy-aware sampling for residential demand response programs. in: HiCoNS, 2012.

[39] Li H, Lin X, Yang H, Liang X, Lu R, Shen X, Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid, Parallel and Distributed Systems, IEEE Transactions on 25 (8). http://dx.doi.org/10.1109/TPDS.2013.124.