Self-Organising Algorithms for Residential Demand Response

Adam Taylor, Ivana Dusparic, Colin Harris, Andrei Marinescu,
Edgar Galván-López, Fatemeh Golpayegani, Siobhán Clarke, Vinny Cahill
Distributed Systems Group,
School of Computer Science and Statistics,
Trinity College Dublin
Email: {tayloral, ivana.dusparic, colin.harris, marinesa, edgar.galvan, golpayef, siobhan.clarke, vinny.cahill}@scss.tcd.ie

Abstract—Residential Demand Response has shown promising results in smart grid applications. It can be achieved manually or autonomously. The variety of algorithms applied to achieve autonomous Demand Response have lacked a common baseline. The selection of testing environment is typically skewed by the desire to find one in which a particular algorithm performs well.

This work evaluates several algorithms in a common evaluation environment. Which has been designed to encompass the range of conditions in a typical Residential Demand Response application. The environments described exhibit the following characteristics: non-stationary, dynamic, multi-actor, multi-objective. These characteristics will then be used to provide heuristics for algorithm selection.

The algorithms used were selected to cover the spectrum of possible approaches to Demand Response. Some are centralised, others distributed. There are collaborative approaches and non-collaborative ones. Some are learning based, others require no training. This work provides criteria for which particular algorithms should be applied to a given application.

I. Introduction

The smart grid involves applying computational intelligence to the electrical grid [1]. Intelligent control of residential devices allows them to be more flexible. This means that they can change their operational schedules to meet the grid's requirements. This is the essence of Demand Response. Electrical demand is modulated by autonomous control to match supply (or other concerns) [2]. The amount of flexibility that a particular device can exhibit depends on its functional requirements. For example, an Electric Vehicle (EV) can be significantly rescheduled as long as it is sufficiently charged to meet user requirements. On the other hand, electric lighting must operate when requested, there can be no rescheduling. There are several different constraints which devices can exhibit that limit their flexibility. Some devices can be stopped and started frequently (e.g., an EV) –called unconstrained–, some must run completely once started (e.g., washing machine) –called continuous–, some device can be set to different output levels operating at some percentage of maximum power –called variable–, this makes them much more flexible as they can fill whatever energy surplus there is (e.g., water heater, EV). In general, devices that have some sort of energy storage will be most flexible. This storage can be thermal (e.g., water heating or fridge), electrical (e.g., EV) or mechanical (e.g., potentially flywheels or compressed air). With such devices,

energy can be stored when there is a surplus –perhaps due to low demand or abundant renewable generation– by turning them on. Effectively, demand is increased to fill valleys in the energy usage. During periods of high demand the stored energy can be used, allowing devices to keep operating without drawing power from the grid, resulting in reduced peak load.

Demand response can be applied to smooth out the overall energy drawn during the day (see Figure 2). Rather than producing a smooth load profile, it can be used to match intermittent renewables. For example, wind generation is typically high in the evening (see Figure 1), at this time residential demand is low. If Demand Response was applied, energy could be stored or used during evening and night wind production to offset the demand during low wind periods.

In customer behaviour trials [3], [?], users were incentivised to reduce electricity usage at peak times, with increases in prices ranging from 150% to 450% compared to off peak times. Peak reductions ranged from 0% to 22% (3 trials out of 21 recorded no change). Over the remaining trials there was a noticeable reduction of peak demand of between 7% and 12%, with price increases between 143%-231% during peak times. A separate groups of users were incentivised to reduce electricity usage only during critical events, which on average occurred 12 times a year, with the event dates provided on short notice (day ahead). These were considered critical peaks, with cost increases compared to normal prices between 500% and 4000%. Here peak reductions ranged between 5-38%. This shows the potential of Demand Response in residential environments. Another trial [4] that used automated Demand Response on central air conditioning and electric heating, showed a 34% reduction in summertime peak load. Aside from being more effective, automation of Demand Response also overcomes some potential problems. Users can lose motivation to participate in such programs over time and they can misunderstand what they are required to do particularly when Demand Response is applied because of energy mix or energy security.

The rest of the paper is organised as follows, Section II will describe Demand Response in general and algorithms that have been applied to Demand Response. Section III will describe the smart grid's characteristics and challenges and how they can be encapsulated in scenarios. Section V
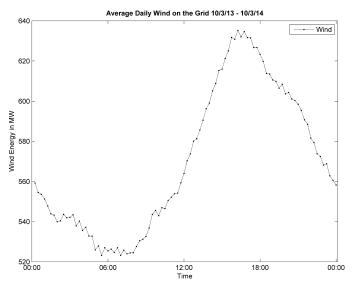
Fig. 1: Wind energy generation in Ireland's grid per day (average figures for a year).
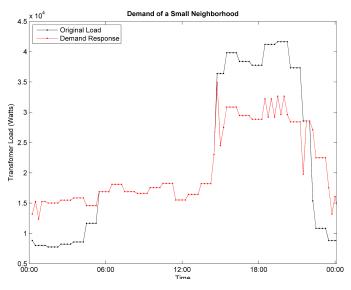

Fig. 2: The load of a small neighbourhood both with and without Demand Response.

will provide details of the experimentation and its results. Section VI will give analysis and design recommendations and finally conclusions are in Section VII.

II. BACKGROUND

The categories of algorithms that can be applied to Demand Response are listed in the Table I. They are compared on the basis of key criteria for Demand Response [5], which are as follows:

- Type of load - Which types of loads can the algorithms currently manage.
- Training - What initial data does the algorithm need to function well.

- Objectives - Can the algorithm accommodate several objectives at the same, time or just one.
- Communications - What are the communications requirements of the algorithms.

A. Probabilistic

The Probabilistic category in the table is comprised of set point algorithms. The central point broadcasts a set point and devices react to this probabilistically. Devices can choose to override this if meeting the user requirements is more important. In aggregate this can increase or reduce the load depending on the broadcast. These algorithms have minimal communications requirements and take very little processing at the device level.

One such algorithm is Set Point Control [6]. In this a probability is broadcast, devices then use this probability to determine if they should turn on. If the aggregated load is too high the probability is reduced, otherwise it is increased.

Another algorithm in this category is additive increase, multiplicative decrease (AIMD) [7]. It is classed as probabilistic as it operates based on a broadcast, with devices responding according to a set scheme. This scheme aims to have some fraction of devices charging while others not, so in aggregate they balance load. The devices additively increase their power draw until the broadcast warns them to stop, in which case they drop down. It is the same scheme as TCP uses. The sharp drops when load is shed will lead to times of underutilised capacity.

B. Reinforcement Learning

The basic pattern underlying Reinforcement Learning (RL) [8] is to observe the environment, choose and execute an action to affect the environment and see how good that action was. An agent (a process that implements RL) can learn to perform optimally for a given state of the environment by executing this process multiple times. When multiple objectives and agents are in a system the optimality guarantee is lost, but generally good performance is achievable. The major advantage of RL is that it does not require a model of the environment, so it can be quickly configured for new applications.

Distributed W-Learning (DWL) is a multi-actor, multi-objective algorithm based on Q-Learning and W-Learning. In this approach devices pass messages about what state of the environment they are in and how good it is (calculated based on their goals). The devices learn what is best according to their own objectives and how their actions affect neighbouring devices[9]. The degree to which agents care about their neighbours objectives can be easily changed.

There is an extension to reduce learning time in RL which involves the application of Transfer Learning. This is called Parallel Transfer Learning (PTL) [10]. The addition of PTL increases the amount of knowledge shared and make the algorithm more collaborative.

A collaborative RL approach using negotiation and planning is found to achieve 95% of the efficiency of a quadratic programming approach [11]. The approach requires some

Algorithm	Type of Load	Training	Objectives	Communications
Distributed Algorithms				
RL Non-collaborative	Unconstrained	On-line	Multiple	None
RL Collaborative	Unconstrained	On-line	Multiple	Between Neighbours
Centralised Algorithms				
Probabilistic	Variable/Unconstrained	✗	Single	Single Broadcast
Scheduling	Unconstrained/Continuous	Off-line	Multiple	Initial Schedule

TABLE I: Features of the centralised/decentralised algorithms used for Demand Response.

knowledge of future uncontrollable load so that the RL agents can negotiate when to operate. This would perform poorly when the base load is dynamic, but achieves good smoothing when it is predictable.

Another agent-based method uses Widrow-Hoff learning and intermittent re-optimisation to reduce peak load by up to 17% [12]. Intermittent re-optimisation is used to prevent many agents finding the same optima at the same time. If many agents use the same load shifting solution it can lead to new peaks, so there needs to be an element of randomness (or other collision avoidance method). This method also requires future knowledge to optimise device scheduling, but it can use the day ahead electricity price, which is readily available regardless of dynamism in most markets.

Predicting base load can be challenging particularly when it is done at small scales. When predicting load across the whole grid, errors and individual variation tend to average out so accuracy of sub 2% is regularly achievable [13]. However, when applying similar methods to small sections of the grid they can fail to obtain the same level of performance as there is a sparsity of data and individuals have a greater influence. For example, an 8MW university campus prediction had 5.15% error [14] and a village with energy usage of 15kW had errors of 13.8% [15]. By combining prediction methods performance can be improved. This is because some methods estimate peaks well and others provide good overall estimates but tend to underestimate peaks. As peak estimation is particularly important having a method just for that is logical. One such system obtains 3.22% accuracy on a 75kW residential neighbourhood [16].

C. Scheduling

Evolutionary Algorithms (EA) are in the Scheduling category. A set of candidate solutions are created, these are then evaluated. The solutions are changed, then re-evaluated. The way they are changed depends on the particular implementation, but it usually involves preferentially selecting good solutions and changing them slightly. This approach allows the greediness to be calibrated. A particular part of the search space can be focused on or a wider array of potential solutions can be used [17].

Another scheduling approach is Approximate Dynamic Programming, which has been applied to several smart grid problems [18]. It is flexible in the types of applications to which it can be applied. When used for device control, it requires foreknowledge to perform well, so much like other planning based approaches it is not suited to dynamic environments.

Centralised scheduling approaches have significant issues when applied to Residential Demand Response. They are unlikely to scale to networks of potentially hundreds of thousands of devices which will eventually constitute a Residential Demand Response aggregation unit. There is also no guarantee that a user will follow the schedule. If users ignore the schedule such approaches will be unable to respond without recalculating the entire schedule. Moreover, any time a new request comes from the grid, a new schedule will need to recalculated and resent. These approaches have serious implications for privacy and data ownership, as potentially information on user behaviour and intentions will be factored into calculations. Such issues are avoided in decentralised approaches or centralised approaches that function by apportioning requests from the grid to devices in the system.

III. SCENARIO DESIGN CRITERIA

To comprehensively test algorithms for Demand Response, the scenarios will need to exhibit all of the challenges found in the smart grid. They will need to have multiple objectives, heterogeneous devices and uncontrollable load. In addition, the environment will be non-stationary and experience different types of change. Change in the environment can be cyclical (e.g., day/night or seasons), transient (e.g., weather) or permanent (e.g., introduction of a new device). Change can occur gradually or suddenly. The way change happens is particularly important to the algorithms. For example, a planning or learning based algorithm will typically react poorly to unexpected sudden change, while gradual change can potentially be tracked and predicted. If patterns repeat then algorithms capable of reusing old knowledge gain an advantage. This is why scenarios can be carefully chosen to play to the strengths of a particular approach.

IV. EXPERIMENTAL SET UP

A. Common Set Up

The algorithms' evaluation is done using a dynamic, multi-objective scenario. The scenario is designed to be realistic and challenging. The simulation was run using GridLAB-D [19], which is an open-source, electrical grid simulator developed by the US Department of Energy. A scenario with 90 houses is used. The base load is derived from a smart meter trial in Ireland [3] (see Figure 3). In addition to this load, each house has a controllable water heater and electric vehicle charger. A year will be simulated so the algorithms tested will need
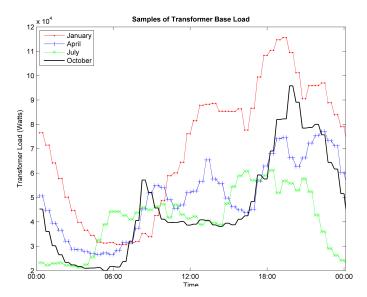
Fig. 3: The total base load for 90 houses excluding EVs and water heating. Samples from four different times of the year.

to account for seasonal variation in demand due to weather. Each devices has a typical usage pattern. EVs are away from home (unavailable to charge) for 9 hours, there is variation in their arrival and departure times, but all EVs are away at the same time for at least 7 hours. When they return they require approximately 11kWh or 30% of their capacity to be fully charged. They charge at a rate of 1.7kW. The water heaters are required to provide a large amount of water in the morning and intermittent small draws throughout the day. They have a 4.5kW heating element. Neither device is capable of drawing variable power, so only fully on or off are possible.

Each device aims to meet its requirements. For the EVs that means being sufficiently charged for the daily journey and for the water heater that is maintaining tank temperature throughout the day. There is also a shared Demand Response goal of smoothing transformer load, peak shaving and valley filling. These goals will be evaluated using the following metrics:

- Peak-to-average-ratio (PAR) [20] will be used for transformer load reduction.
- δ, the average change in load between samples will be used to assess smoothing performance.
- Average battery state of charge and number of complete discharges will be used for EV evaluation.
- Average water temperature will be used for water heater evaluation. It should be at between 40°C and 60°C.

B. Algorithm Set Up

1) DWL: When DWL is applied to this scenario, an agent is attached to each device. Each agent has two policies, one representing the user's requirements for a device and the other representing the transformer requirements. The exploration phase lasts for 11 months[1]. Once the learning is complete, exploitation begins and the agent selects the best action for its current state.

2) DWL with Prediction: When prediction is coupled with DWL a third policy is added. This policy represents predicted future load. For example, if the current load was high but expected to be low imminently, the policy would suggest that the device turns off until the low load occurred. Overall this policy aims to improve the load shifting performance. The predictions are generated off line and provided for the next hour.

3) DWL with PTL: When Parallel Transfer Learning is used each device has a set of ten neighbours. These are the agents that information can be transferred to and from. Transfer occurs only during the exploration phase as this is when there is the greatest benefit. Transfer can be particularly effective when applied to devices like those used here. This is because when exploring, action selection is at least partially random, which means that the physical parameter being affected will tend to stay around the same value, this leads to the agent only experiencing a small section of the state space and it taking a particularly long time to learn. By transferring information this slow progress though the state space can be turned to advantage.

4) Probabilistic: This approach broadcasts a probability with which devices should switch on. The probability is calculated based on what the grid wants. For these experiments the aim was to maintain an even load of 350kW. If the load was above this the probability decreased, if not it increased. There is a user aware version of this approach which does not switch on regardless of probability, unless the user requirements are currently not satisfied by the device. For example, if an EV was above its minimum required battery charge (known for this scenario, but would have to be estimated in actuality) and its charging probability told it to charge, it would not.

V. RESULTS

The results in Table II show average values for the various metrics. For the algorithms that required training, the results represent the exploitation phase. A sample of the transformer load is shown in Figure 4.

Method	Water Temperature Average	State of Charge Average	δ (kW)	PAR
Base Line	45.0244°C	81.5%	1.934	2.13
DWL	52.7411°C	39.71%	3.9162	2.12
DWL + Prediction	54.0535°C	10.43%	7.2869	1.63
DWL +PTL	52.32°C	24.5%	2.5799	1.24
Set Point	60.7399°C	43.71%	4.8198	1.65
User Aware Set Point	51.7411°C	44.98%	2.3114	2.07

TABLE II: Averaged results of the different algorithms.

The water temperature is kept in the desired range with all algorithms. By this metric, all algorithms fall approximately into the middle of the range as would be expected when

---

[1]This length of time is to allow the algorithm to experience the full range of seasonal variation in the scenario. Without variation approximately 1 month would be sufficient training see [21].
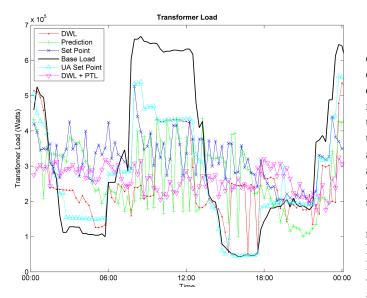
Fig. 4: Transformer load for the different algorithms.

averaging device meeting the temperature goal. The Set Point approach, however, sits at the top of the range as at periods of low load from other devices, it operates if at all possible. This is apparent on the graph during the inter-peak periods where it maintains a high average load. The largest water demand is in the morning, this is what causes the high load between mid-morning and midday. Due to the tanks' insulation, little energy is expended maintaining temperature once the tank is reheated after being filled.

The EVs? state of charge shows one of the risks with learning based methods. Both DWL + Prediction and DWL + PTL maintain a comparatively low average charge, this is because they have learned that a full journey uses approximately 30% battery. This affords them the opportunity to only charge EVs at night when there is little other load. While this means they have a relativity constant energy draw across the day, it runs the risk of an EV being completely discharged due to abnormal use.

The average δ is interesting. For most of the approaches that achieve demand response, the δ is quite high. This is because when several devices change action at the same time, there is a considerable effect on transformer load. This is particularly true with prediction as devices are over reacting to predicted low load. When low load is forecast many switch on creating a spike. The increased collaboration in PTL reduces the spiking profile as each set of neighbours (devices that share information) reach slightly different solutions. This means there is more variation in device behaviour, so some collisions are avoided.

The PARs are generally low. This indicates that the magnitude of the peaks is acceptable and from the graph considerable peak shaving can be seen. Both of the valleys have been filled by most approaches.

VI. Algorithm Analysis

The selection of algorithm for a particular application will depend both on the environment and what it is required to do. If, for example, an algorithm's main aim is to maintain a quality of service [22], reactiveness will be prioritised. Alternately, if minimising electricity price is the goal, considerable time can be put into prediction which enables the lowest price to be used. The set point algorithms can be suitably reactive as they take very little computation. Generally learning based algorithms will not be responsive enough, as they take time to adapt to change. However, with prediction they can be made suitable.

The structure of the system and guarantees from the algorithms are also important. If a set of devices implementing Demand Response are being factored into the dispatch calculations of an electricity market, they will be required to achieve what they promise. This means there is much less flexibility in the algorithm as there are requirements that have to be met.

The algorithms should also fit to the structure of the grid, as it is inefficient to run a single centralised algorithm for the whole grid. The delay in gathering data and propagating control is unacceptable. This is why distributed solutions will be preferable, they also allow individual devices' to manage their own concerns. This will be particularly attractive to network operators as they are then not responsible for devices which fail to meet their owners' objectives. Distributed approaches like DWL naturally fit into the structure of the grid, as agents can be attached to any device.

VII. Conclusions

This work has evaluated several representative Demand Response algorithms in a challenging scenario. Residential Demand Response algorithms should fit into the structure of the grid, so that both data acquisition and dissemination of control can happen with minimal latency. Given the number of devices that will necessarily participate in a residential Demand Response, there will need to be some mechanism to avoid overreaction to grid requests. This is particularity true when there is a large percentage of flexible devices (such as EVs) available, as these will more readily respond to requests. Collaborative algorithms can achieve this sort of collision avoidance by having devices stagger their operation. Non-collaborative algorithms could be made to exhibit these same behaviours, if devices implementing them were put into small groups and grid request where distributed in a manner that resulted in each group getting a slightly different version of the request, in aggregate then the ramp up of load would be smoothed.

Future work involves extending these algorithms to even larger scenarios and investigating the issues this entails. Developing methods to increase diversity and variation in device behaviour to avoid over reacting to grid requests.

"Self-organizing Architectures for Autonomic Management of Smart Cities".

REFERENCES

[1] A. Dimeas and N. Hatziargyriou, "Multi-agent reinforcement learning for microgrids," in *Power and Energy Society General Meeting, 2010 IEEE*. IEEE, 2010, pp. 1–8.

[2] S. Lu, N. Samaan, R. Diao, M. Elizondo, C. Jin, E. Mayhorn, Y. Zhang, and H. Kirkham, "Centralized and decentralized control for demand response," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*. IEEE, 2011, pp. 1–8.

[3] Comission for Energy Regulation, Ireland. (2011) Smart meter trial data. [Online]. Available: www.ucd.ie/issda/data/commissionforenergyregulationcer/

[4] eMeter Strategic Consulting for the Smart Meter Pilot Program Inc. Powercentsdc program, fianl report.

[5] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley, "A survey of multi-objective sequential decision-making," *Journal of Artificial Intelligence Research*, vol. 48, pp. 67–113, 2013.

[6] C. Harris, I. Dusparic, E. Galván-López, A. Marinescu, V. Cahill, and S. Clarke, "Set point control for charging of electric vehicles on the distribution network," in *IEEE PES Innovative Smart Grid Technologies (ISGT)*. IEEE, 2014.

[7] S. Stüdli, E. Crisostomi, R. Middleton, and R. Shorten, "A flexible distributed framework for realising electric and plug-in hybrid vehicle charging policies," *International Journal of Control*, vol. 85, no. 8, pp. 1130–1145, 2012.

[8] R. Sutton and A. Barto, *Reinforcement learning: An introduction*, 1999.

[9] I. Dusparic, C. Harris, A. Marinescu, V. Cahill, and S. Clarke, "Multi-agent residential demand response based on load forecasting," in *Technologies for Sustainability (SusTech), 2013 1st IEEE Conference on*. IEEE, 2013, pp. 90–96.

[10] A. Taylor, E. Galván-López, S. Clarke, and V. Cahill, "Accelerating Learning in Multi-Objective Systems through Transfer Learning." in *In a Special Session on Learning and Optimization in Multi-Criteria Dynamic and Uncertain Environments at the International Joint Conference on Neural Network 2014 (IEEE IJCNN 2014*. IEEE, 2014.

[11] S. Vandael, N. Boucké, T. Holvoet, and G. Deconinck, "Decentralized demand side management of plug-in hybrid vehicles in a smart grid," in *Proceedings of the First International Workshop on Agent Technologies for Energy Systems (ATES 2010)*, 2010, pp. 67–74.

[12] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. Jennings, "Agent-based control for decentralised demand side management in the smart grid," in *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, ser. AAMAS '11. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 5–12.

[13] H. K. Alfares and M. Nazeeruddin, "Electric load forecasting: literature survey and classification of methods," *International Journal of Systems Science*, vol. 33, no. 1, pp. 23–34, 2002.

[14] M. Irzaq bin Khamis, Z. bin Baharudin, N. Hisham bin Hamid, M. F. bin Abdullah, and S. S. M. Yunus, "Electricity forecasting for small scale power system using fuzzy logic," in *IPEC, 2010 Conference Proceedings*. IEEE, 2010, pp. 1040–1045.

[15] J. Llanos, D. Saez, R. Palma-Behnke, A. Nunez, and G. Jimenez-Estevez, "Load profile generator and load forecasting for a renewable based microgrid using self organizing maps and neural networks," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–8.

[16] A. Marinescu, I. Dusparic, C. Harris, S. Clarke, and V. Cahill, "A hybrid approach to very small scale electrical demand forecasting," in *Innovative Smart Grid Technologies (ISGT)*. IEEE PES, 2014.

[17] E. Galván-López, A. Taylor, S. Clarke, and V. Cahill, "Design of an Automatic Demand-Side Management System Based on Evolutionary Algorithms," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM, 2014.

[18] R. N. Anderson, A. Boulanger, W. B. Powell, and W. Scott, "Adaptive stochastic control for the smart grid," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 1098–1115, 2011.

[19] D. P. Chassin, K. Schneider, and C. Gerkensmeyer, "Gridlab-d: An open-source power systems modeling and simulation environment," in *Transmission and Distribution Conference and Exposition, 2008. T&D. IEEE/PES*. IEEE, 2008, pp. 1–5.

[20] A. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *Smart Grid, IEEE Transactions on*, vol. 1, no. 3, pp. 320–331, 2010.

[21] A. Taylor, I. Dusparic, E. Galván-López, S. Clarke, and V. Cahill, "Transfer Learning in Multi-Agent Systems Through Parallel Transfer," in *Workshop on Theoretically Grounded Transfer Learning at the 30th International Conference on Machine Learning (Poster)*, vol. 28. Omnipress, 2013.

[22] A. Molina-García, F. Bouffard, and D. S. Kirschen, "Decentralized demand-side contribution to primary frequency control," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 411–419, 2011.