Efficient Customer Selection for Sustainable Demand Response in Smart Grids

Vasileios Zois[†], Marc Frincu[‡], Charalampos Chelmis[‡], Muhammad Rizwan Saeed[‡], Viktor Prasanna[‡]
University of Southern California
Email: {vzois,frincu,chelmis,saeedm,prasanna}@usc.edu
Department of Computer Science[†], Department of Electrical Engineering[‡]

***Abstract*—Regulating the power consumption to avoid peaks in demand is a known method. Demand Response is used as tool by utility providers to minimize costs and avoid network overload during peaks in demand. Although it has been used extensively there is a shortage of solutions dealing with real-time scheduling of DR events. Past attempts focus on minimizing the load demand while not dealing with the uncertainty induced by customer intervention which hinder sustainability of the reduced load. In this paper we describe a smart selection algorithm that solves the problem of scheduling DR events in a broad spectrum of customers observed in common Smart Grid Infrastructures. We deal both with the problem of real-time operation and sustainability of the reduced load while factoring customer comfort levels. Real Data were used from the USC campus micro grid in our experiments. On the overall achievable reduction the results produced a maximum average approximation error $\approx 0.7\%$. Sustainability of the targeted load was achieved with maximum average error of less than 3%. It is also shown that our solution fulfils the requirements for Dynamic Demand Response providing a solution in a reasonable amount of time.**

Keywords: smart grid, scheduling, optimization, sustainability, demand response, selection algorithm, real time, change making, customer comfort

I. Introduction

As demand for power increases so does the complexity involving safe and reliable [20] energy distribution. Recent innovations in power grids have provided smart tools, to help utility providers monitor and predict [3] [4] power demand. All together an automated power distribution network was created on top of the old power grid infrastructure, now know as a smart grid [13] cyberphysical system. It consists of different components [20] addressing reliability, load management and security. Smart meters [6], capable of bi-directional communication, are a vital part of the smart grid. They are used for real time monitoring of power consumption helping utility providers predict future demand. Fulfilling the necessary energy requirements is based on these predictions. However installing additional power generation capability to meet peak demands sometimes is neither feasible nor sufficient. Demand Response(DR) [2] is a well known method employed by energy providers to control demand. It is used to find the equilibrium between energy production and demand through load control.

Energy providers use different paradigms in their attempt to control customer load including: direct control [10], price incentives [19] as well as voluntary participation. Although these techniques have been broadly used in practice they still are unable to produce good results while dealing with uncertainty induced by the customer behaviour. Real-time techniques need to be employed to cope with unexpected peak-demands or situations where adaptation during a DR event is needed(**dynamic DR**) to sustain a consistent power reduction under a defined safe threshold. A DR event is defined as a schedule consisting of customers and their corresponding strategies for a specific period of the day. It is initiated by energy providers to achieve a combined load reduction from the participating customers.

A DR event is said to be sustainable if it achieves consistent load reduction for the scheduled time frame. A consistent load reduction is considered to be a reduction of the typical power consumption under a specified threshold. This threshold is usually defined by the utility providers. The goal can be to ensure reliability in power distribution, protect the equipment on the grid or simply maximize profit. Sustainable load reduction is a hard problem. Customers employed on top of the power grid are inherently unpredictable. This uncertainty hinders attempts on achieving sustainability. Customer comfort [23] is defined as the ability of users in the power grid to decide independently about the time and way to consume power. Each one has a different footprint characterized by their behaviour and different reactions to exogenous events(e.g rise in temperature). When scheduling a DR event all of this has to be taken into consideration. It is imperative to deal with uncertainty [9], instead of relying too much on predictable loads in order to sustain a consistent reduction. Hence it is important also to quickly(dynamic DR) react and adapt to changes induced by uncertainty from the behaviour of individual customers. If the above points are to be considered then sophisticated scheduling algorithm is needed. This algorithm should produce a selection of customers to participate in a DR event aiming to minimize uncertainty while maximizing comfort. These two notions are complementary since a comfortable customer is more likely to comply with a DR event. This paper addresses the above gap by introducing an algorithm that fits the aforementioned properties.

The contribution of this paper can be summarized in the following points:

- The proposed algorithm deals with customers as indivisible entities. Detailed description of the consumption of individual devices is not needed. We just focus on the consumption patterns initiated by different DR strategies.

- Dealing with customer comfort is not included as an extra variable. We argue that large deviation in power reduction throughout a DR event represent customer discomfort. A selection of unique strategies define different levels of intrusiveness.
- Sustainable reduction throughout the DR event was our initial goal. We achieve that by analysing in a coarse grained manner the potential reduction(fixed intervals) of each customer.
- The complexity of the algorithm confirms the efficiency claim. We achieve polynomial complexity in the number of customers. Our experimental results show a linear and polynomial increase in the relative execution time consistent with the complexity analysis.

We start by describing in section II the related work. Next we continue by formulating our problem in section III. An analytical description for the algorithm developed is provided in IV. We conclude with the analysis of the conducted experiments in section V.

II. Related Work

Some of the earliest work focused on directly controlling device schedules to minimize power consumption. In [11] [10] dynamic programming was employed to minimize the controllable load. In [21] linear programming in combination with customer grouping was used based on a profit based approach to minimize load consumption.

A recent trend on the field of load manipulation is factoring customer comfort. Arguments in favour of it are the uncertainty induced by users who may override the system and cause peaks in demand. Different approaches emerged based on modelling consumption of specific devices according to their usage patterns. In [23] particle swarm optimization was used to control demand by focusing on water heaters which accounted for 30% of the overall load. The same technique was employed in [12] and [18]. The former focused on the special case of Plug-in Hybrid Electronic Vehicles (PHEVs). The latter examined residential cases where a schedule was provided based on the optimal time to operate domestic appliances.

In [17] load manipulation was studied conforming to constraints induced by dynamic pricing policies. An heuristic approach was proposed providing a solution to the scheduling problem in $O(ATN^2 \log TN)$ time. Other approaches include a game theoretic analysis on the minimization of load demand. This was addressed in [9] using a residential environment with real-time pricing as the case of study. Also in [19] a similar technique was employed by relying on smart pricing as an incentive to achieve the necessary load reduction. These approaches relay on distributed calculation to provide a real time solution to the scheduling of power reduction to different customers. Our solution also considers the need for real time execution.

Other proposed solutions with similar goals include [22]. There a multi-objective optimization problem was formulated adhering to specific constraints. Those are induced by the need to minimize consumption and maximize utility. Moreover evolutionary algorithm was used to solve the constrained problem. In [16] an office environment was the use case. Power consumption was minimized based on dynamic pricing and production of electricity from renewable energy sources. The ultimate goal was to leave productivity unaffected. Finally in [30], a fine grained description of a smart home is used to schedule usage of individual appliances in respect to residential needs.

It is important at this point to note that many of the research done so far deals with residential cases [14] [19]. Also many approaches focus on specific devices [15] [18] [25] [8] [7]. This scenario is unrealistic for energy providers as they cannot sustain information for all the different appliances and their consumption patterns. A similar problem statement to ours has been made in [24]. There the authors study the case of a feeder failure in the distribution network. They made a mixed-integer programming formulation of the scheduling problem and proposed three approximate methods to solve it. Contrary to ours their work assumes customer compliance during the DR event. Also it deals with distributing a portion of the load demand on other transformers. We deal only with finding a schedule of customers to reduce the load demand.

Our solution considers the abstract notion of a customer associated with available strategies. These strategies are the different plans available by utility providers. Each one defines a specific level of intrusiveness. Comfort is implied by the overall achievable reduction and the observed inconsistencies of the reduced load across the whole event. A dynamic pricing policy associated with each strategy can be used as incentive for participation in curtailment. Similar incentives have been assumed in [5]. We avoided a device based analysis since it is heavily dependent on their individual characteristics. It is a realistic scenario since in many cases utility providers do not have access to this kind of information. Our goal was to capture the overall customer behaviour during a DR event and use it as reference to schedule future ones.

III. Problem Definition

We formulate the problem as follows: We are given a set of n customers. Each one is associated with m available strategies. As we mentioned before each strategy represents an individual plan provided by an energy provider. The strategy provides some possible incentives in exchange for voluntary participation in a DR event. Achievable reduction is strongly related to the incentives provided. For each customer-strategy combination there exists a curtailment vector $< r_1, r_2, ..., r_k >$. Each value $r_k \in \mathcal{R}$ is calculated at fixed time intervals. It represents the curtailment level which is the difference between the predicted baseline consumption and the predicted consumption during a DR event. A positive number indicates a successful power reduction while a negative number indicates an increase in consumption. The baseline is based on the observed consumption of the customer during normal operation. The accuracy of the values indicating the level of curtailment, depends on the predictions of the methods used. Discussing in more detail the different kind of prediction methods is out of the scope of this paper.

Let R be the targeted overall reduction throughout the whole DR event. Our goal is to find a subset of the available customers which should accumulatively curtail $\frac{R}{t}$ per interval, where t is the number of intervals in a specific DR event. Each customer must conform to a single strategy from the available ones through the whole DR time frame. Based on the above definition, our problem is defined as finding a set A of customer-strategy combinations that collectively achieves $\frac{R}{t}$ reduction per interval. This is described more formally in (1) where $r_{i,k}$ is the reduction achieved by the i-th customer-strategy combination $\in A$ at the k-th interval from the curtailment vector.

$$\sum_{i=1}^{|A|} r_{i,k} = \frac{R}{t} \tag{1}$$

Our problem can also be expressed as Integer Linear Programming [28] Problem(ILP). It is known that ILP is $\mathcal{NP}$-hard [28]. This means that for a small number of customers the solution is achievable in reasonable time but it is not the same case for a large number of customers. We can also deal with it as a knapsack problem [29]. However dynamic programming does not fit well to our definition of the problem. A real time solution independent of the targeted reduction is needed. Moreover it important to use the minimum number of participating customers. Also we are dealing with real numbers for weights and we need to solve an 0-1 knapsack problem. In the next section we present our approximative solution which is based on the change making [27] problem.

IV. Proposed Algorithm

In this section we focus on describing the developed algorithm. At the beginning we discuss the motivation behind our choice to implement an approximative solution. An overview of the major steps which constitute the algorithm is presented. Finally we focus in discussing further some parts which affect the accuracy, sustainability and the complexity of our approach. In the previous section we made a strong case in favour of sustainability. Also we emphasized the need for a dynamic demand response scheduling to deal with uncertainty. These are two of the strongest points which impelled us to provide an approximative solution. Real Time scheduling of DR events is important. As it was mentioned before solutions provided by formulation of our problem as a 0-1 knapsack or as an ILP are feasible but not efficient. Moreover we need a way to incorporate in the selection procedure the uncertainty induced by the customers. This can be realized when considering the control of an AC unit during a warm summer day. The discomfort induced can force a customer to override the DR event causing unpredictable peaks in demand. Finally we can see that eliminating discomfort provides us with predictable load reduction. This makes it easier to sustain a stable reduction throughout the event time frame. In the next section the proposed solution is described in more detail following the properties defined above.

A. Notation

Before we begin describing the algorithm in detail we need to define some common notation used:
U : set of representatives of the n customers.
u_j : *j*-th estimate from set U.
x_i : Number of customers in *i*-th bin.
$\tilde{C}$: set of coins
$\tilde{c}_i$: *i*-th coin from US domination.
C : set of bin values.
c_i : *i*-th bin value from C, $c_i = \tilde{c}_i \cdot v$
v : unit value used to define the bins ranges.
M : reduction per interval.
$\tilde{M}$: Amount to be payed using change making.
B_i : *i*-th building in a bin.
B_{ij} : *j*-th strategy of *i*-th customer.
r_k : reduction value of k-th interval.
BN : set of bin ranges.

B. Algorithm Description

Our procedure involves formulating the solution based on the change making problem. The initial step is to distribute the customers into bins. Each bin has a specific range determined by a quantity we call unit value(v). The upper value of each bin is defined as the bin value. Customers are distributed into specific bins according to an estimated reduction value. We refer to this value with the term **representative** of a customer.

$$\min_{\forall c_i \in C, \forall B_i \in (c_{i-1}, c_i]} \sum_{r_k \in B_{ij}} (c_i - r_k)^2 \tag{2}$$

After distribution we conform each customer to a specific strategy. We decide on this by choosing the strategy which minimizes the accumulated error produced by the difference between the bin value and the reduction per interval values(2). This is done for all available strategies to each customer. The last step is to greedily iterate over the bins and select the customer-strategy combinations to *pay* for the needed reduction. The algorithm for change making is used at this point to index the necessary bins.

We describe here more formally the above procedure. Consider that we need a reduction of M KWh per interval across a DR event of size k ($M = \frac{R}{k}$). Assuming a known unit value we get $\tilde{M} = \frac{M}{v}$ as the amount needed to be *paid* in order for the necessary reduction to be achieved. By using the change making approach we get (3). Starting from the biggest coin which is equal or less than $\tilde{M}$ we use zero or more coins of value $\tilde{c}_i$ to pay for the amount $\tilde{M}$. The set of coins $\tilde{C} = \{1, 2, 5, 10, 25, 50, 100\}$ is based on the US denominations which always gives an optimal solution using the minimum number of coins. If we multiply (3) by v which is the unit value we get (4). This way we can construct seven bins using the values from $\tilde{C}$ multiplied by v. If we do this we get the bin ranges in (5). Every customer with reduction per interval in the ranges defined by the boundaries of the i-th bin from (5) belongs in that bin. The estimate for the reduction is the customer's representative. A pseudo code describing the

$ChangeMakingScheduler(buildings, M)$

Data: List of building-strategy reduction vectors, Reduction needed

Result: List of building-strategy

```
representatives = buildings.representatives();
u = calc_unit_value(representatives);
for i ← 1 to c.size do
  | c[i] = c̃[i] * u
end
M̃ = M/u;
bins = distribute(c̃, buildings);
for i ← 1 to buckets.size do
  | sort(bins[i])
end
for i ← c.size to 1 do
  | j = 0
  | while M − c̃[i] ≥ 0 do
  |   | while c[j] − bins[i].building[j].reduction ≥
  |   | 0 and j ≤ bins[i].length do
  |   |   | result.add(bins[i].building[j])
  |   |   | c[j] = c[j] − bins.building[j].reduction
  |   |   | j = j + 1
  |   | end
  |   | M̃ = M̃ − c[i]
  | end
end
```

Algorithm 1: Change Making Scheduler

major steps of our change making scheduler is presented in Algorithm 1.

$$\tilde{M} = \sum_{1}^{7} c_i \cdot x_i \ , \ x_i \in \mathbb{N} \tag{3}$$

$$M = \sum_{1}^{7} c_i \cdot v \cdot x_i \ , \ x_i \in \mathbb{N} \tag{4}$$

$$BN = \{(0, v], (v, 2v], (2v, 5v], \ldots (50v, 100v]\} \tag{5}$$

The most crucial step of the mapping described above is selecting a suitable unit value. A unit value is suitable if we can find at least one customer-strategy combination in each bin which achieves the bin value per interval. An inaccurate decision at this point will limit the true potential reduction of a customer. Although we can have another case which is omitting a customer-strategy by placing it in a bigger bin. Since customer-strategies are sorted in each bin before the bins are indexed, choices which deviate much from the bin value are not chosen. These limitations will always provide us with a suboptimal solution that does not achieve the target set. This affects both sustainability and the overall achievable reduction through the whole DR event. Much of this paper is devoted in inventing sophisticated procedures for choosing the unit value. In the next subsection we describe them in detail.

C. Unit Value

Finding a suitable unit value must be a compromise between accuracy and speed. From the equality $\tilde{M} = \frac{M}{v}$ described before we only know M. So we must find a way to assume an approximate value for one of the variables in the right hand side. We start by assuming that $v = M$ and built the bins accordingly. This technique is defined as the greedy approach. All the customers under consideration will be limited in the first bin since bins of larger value will overshoot our target and won't be considered in the final solution. Although simple and efficient this technique does not achieve the best accuracy. This is because iterating greedily through the bins, provides us with a solution containing customer-strategy combinations of the highest reduction level less than M. It may be the case that there is a selection of intermediate customer-strategy combinations that are able to achieve the targeted reduction. For that reason we need a solution adaptable to the specific reduction patterns of each customer during a DR event.

Taking into account customer behavioural patterns we assumed that there must be a suitable unit value from the set of representatives for each customer. In this case we developed three procedures to select from this set a suitable value. Our solution begins by examining each individual representative separately. This is done by using it as potential unit value to built the bin ranges. Then assuming those bin range an error measure is employed to decide which representative to select. This error measure estimates the effect of our choice to the final solution. It selects as the unit value the representative that minimizes this corresponding error measure.

The first technique which we call Minimum Goal Accumulated Bin Error(MGABE), focuses on minimizing two quantities (6) connected to the choice of the unit value. The first quantity is the accumulated error produced by the absolute difference of the bin value from the reduction level estimated by the representatives of that bin. The second quantity is the absolute difference between the target reduction M and the sum of the bin values produced and being used from the solution provided by the change making algorithm. This approach is expected to produce relative good results in the subject of sustainability since it considers minimizing the error from the bin values. However i will have a hard time matching the overall reduction needed since it focuses on two targets independently. It will be argued from the experiments that minimizing the latter quantity does not produce better results. However we can see that the maximum error produced from that quantity will not exceed 0.5 due to rounding of $\frac{M}{v}$.

$$\min_{u_j \in U} \sum_{c_i \in C} (c_i - (\max_{c_{i-1} \leq u_k \leq c_i} u_k)) + \min_{u_j \in U} (M - \sum_{c_i \in CM} (c_i)) \tag{6}$$

A simpler technique is to consider only minimizing the quantity connected with the bin values. The second technique we developed, which is known as Minimum Accumulated Average Bin Error(MAABE) does exactly that. It considers the average reduction of the representatives that belong in each bin and tries to minimize their difference from the bin value. Again this is done for all the representatives from U. We describe this minimization problem by defining equation (7). We expect to get better results here in respect to the overall reduction since we don't have the independent quantities of (6). Also in the matter of sustainability since we are dealing with reduction per interval we will manage to acquire a solution achieving a sustainable reduction.

$$\min_{u_j \in U} \sum_{v_i \in v} (c_i - (\frac{1}{x_i} \cdot \sum_{v_{i-1} \leq u_k \leq v_i} (u_k))) \tag{7}$$

Both of the described techniques consider all bins when calculating the overall error. It would be more advantageous to focus only on bins that are going to be used in the end result. That is because we want to focus on minimizing less goals to get a better result. It is important to consider a good heuristic in order to get a real estimate on the bins used. Our approach was to estimate $\tilde{M}$ since we already have a potential v. Using this information we can decide on which bins are being used in the end result. Then we select v from the representatives to minimize the accumulated error induced by these bins. This technique Minimum Coin Error(MCE), is similar to (6) although we only deal with choosing the representative that minimizes the accumulative bin error. Also another difference is that we consider $\tilde{M}$ and not M. We describe MCE using (8). It is expected our results to be strongly dependent to the estimate we have for the bins being used.

$$\min_{u_j \in U} \sum_{c_i \in C} (c_i - (\max_{c_{i-1} \leq u_k \leq c_i} u_k)) \tag{8}$$

The presented methods focused on searching the unit value from the representative set. The major drawback in this is that it assumes a suitable unit value is present in that set. It is clear that there might be cases where that is not true. This is the reason behind the development of our last technique which is called Unit Data Trend(UDT). Our reasoning behind this method is that we need to try and fit the patterns in the dataset to the constructed builds. Only this way we can achieve sustainability which will also ensure achieving the overall load reduction. We focus on using as bin values the initial values of the coins from the change making problem. Then we state that a unit value is needed to fit the dataset and provide corresponding bin ranges. We initially distribute the buildings into the corresponding bins. Based on this distribution we calculate the weighted average(9) which serves as a unit value. The weight is the number of customers in a bin and the value is the max reduction from the representatives in that bin. We expect to get good results in respect to sustainability as we try to match the patterns existing in our dataset. The overall achievable reduction might not be achieved since sustainability might produce reduction close to the needed one but less or more than that.

$$v = \frac{\sum_{i=1}^{7} (x_i \cdot \max_{\forall u_j \in U, c_{i-1} < u_j \leq c_i} u_j)}{\sum_{i=1}^{7} x_i} \tag{9}$$

D. *Representatives*

The previous section made the role of the representatives clear. They are used as estimates for the achievable reduction of each customer. Based on them we created heuristic methods to calculate a suitable unit value. Also we use them to decide on how buildings are divided into bins. It is clear that their role is important and affects the robustness of the scheduling algorithm. Making a wrong estimate will give an unsuitable unit value which in turn limits the potential reduction of each customer. This is because we conform each customer to the only strategy that provides the lowest deviation from the bin value (2). Realizing this we decided to test different ways of calculating representatives. Our choices we are motivating by the need to show estimates that produce good results as well as bad ones. It was our to goal to give a clear understanding on the importance of these selections to the end result. Moreover we focused on finding simple and efficient solutions which should not add much to the overall complexity. We present briefly the three methods used and get into a detailed discussion of their affect in the experiments section. The first method(MAX) calculated the maximum value from all intervals of all possible strategies for a customer. The second method(AVG) calculated the average reduction from all intervals of each available strategy for a customer. Finally the third methomd (MAVG) used as a representative the maximum average reduction from each individual strategy available to a customer. In general we expect to get the worst results by AVG since large deviations between strategies is going to provide an inaccurate estimation. It is important to note that the representatives affect strategies which are heavily based on them. Those are all the strategies which use them to calculate an approximate unit value. Between the other two methods (MAX and MAVG) we expect better results from the former. This is connected to the bin values which constraint the selection of strategies. MAVG uses an averaging estimation for the whole interval giving a bad estimation in cases where large deviations in the reduction exist. So our safest choice will be MAX since it is not so restricted on the max bin values constructed.

E. *Complexity*

For the complexity part we can divide the algorithm into two sections. We first have the section where some common steps are executed (e.g distribution of customers into bins, representative calculation e.t.c). This section is common for all methods and provides a standard complexity. In the second section we deal with the individual methods used to calculate the unit value. There the complexity differs among each method.

In general our algorithm consists of some common steps independent from the technique used to calculate the unit value. These include calculating the representatives, distributing the customers into bins and conforming them to a specific strategy, sorting the customer-strategy combinations in each bin according to their corresponding error.Finally we iterate greedily over the bin and produce the indices used to select which customers are going to participate in the DR event. The size of the input is defined as $n{\cdot}m{\cdot}k$ where n is the number of customers m is the number of available strategies and k is the number of intervals in the DR event. In practice we have 3-10 strategies to consider and at most 96 intervals of 15 minute granularity, exist in a day. These choices were made during our experiments and present a realistic scenario. We conclude that the size of the input is linear in the number of customers. The common steps we described before have a complexity of $O(n)$, $O(n)$, $O(n\ log\ n)$ and $O(n)$ respectively considering an input size of n customers. So in conclusion the overall complexity is $O(n\ log\ n)$ for the first part.

Except for the greedy technique, calculating the unit value adds an extra computational cost. In MGABE and MCE, there is an extra computational cost of $O(n\ log\ n)$. This is because we consider n representatives at most and for each we need to consider the one closer to the bin value. In MAABE the extra cost is $O(n^2)$ because we calculate for each of the n representatives the average of the customers reduction that belong in each bin. The final method(UDT) calculates the weighted average. Here we need to iterate over all the representatives which adds $O(n)$ additional complexity.

It can be argued that the devised algorithm fulfils the requirement for an efficient solution. In any case we need at most polynomial time to provide a solution. It will be shown in the experiments that the above complexities are verified. Also it is noted that the bottleneck in the execution is loading the data from the hard disk. Finally some common operations like calculating the representatives can be executed as a preprocessing step.

V. Experiments & Results

The experiments we conducted were designed to test the accuracy of the algorithm in terms of the targeted reduction and the sustainability of the reduced load. Also we measured the number of buildings used in the solution to argue about the level of intrusiveness of our solution. Finally we simulated consecutive DR events and measured the collective execution time for a solution to be provided.

The algorithm was implemented using Java and was executed on Windows operating system. The experiments where executed in single quad core CPU system(Intel Core i7 3632QM @ 2.20 GHz) with system memory(8,00 GB Dual-Channel DDR3 @ 665 MHz).

A. Dataset Used & Experiment Categories

To test our implementation we used measurements from meters in the USC campus micro grid. The dataset is populated with values representing the average power reduction(KWh) for a fixed periods of time(15 minute granularity).In total 33 buildings participated in 380 DR tests by employing specific reduction strategies or a combination of them.The time frame of the DR events is from 1:00 - 5:00 PM. We chose this time frame because it is the time of the day where peak demand is observed. The available strategies for each building consist of equipment groupings which are controlled directly during a DR event. Measurements from past DR events conducted on campus buildings were used as input to the ARIMA [26] prediction model. We needed to predict the actual building consumption during past DR events. Southern California Edison(CASCE) [1] was used as the baseline to define the actual consumption on a regular day for each building. As it was mentioned before we measured the curtailment level by taking the difference between the baseline and the predicted reduction during a DR event. CASCE was used because we found from previous work [3] that it gave the most accurate results in terms of the predicted consumption during a normal day.

We consider two cases in our experiments: The first case includes all the buildings and strategies available in our dataset. We do this to evaluate the accuracy of each technique developed in terms of overall load reduction and sustainable load during a DR event. Absolute Percentage error was used as the metric for our evaluation. The second case randomly selects buildings and strategies to be eliminated. Every building or strategy has a probability of 50 % to be selected. This decision was made in order to simulate a harder instance of the problem being solved. The results from these experiments were used to evaluate the robustness of each technique. In this case we run a thousand requests and used Mean Absolute Percentage Error(MAPE) as the metric for the evaluation of each individual technique.

The overall performance for the developed methods is evaluated against two random selection approaches used as baselines. The first baseline is a uniform random selection of buildings and their corresponding strategies. The second baseline implements a probabilistic selection. It is based on the distribution of buildings into bins of fixed length according to their reduction estimated by the representatives. Bins with more buildings in them have greater probability to be selected. The ranges of the bins are built in increments of 10 beginning from 0. There were some buildings with negative reduction but we ignored them for the baselines.

B. Overall Reduction

Achieving an optimal reduction is strongly related to the method used for calculating the unit value. Since many of the developed methods are based on the representatives it is imperative to have a good initial estimate for them. As it was stated a good estimate is the one that does not limit the potential reduction of each customer. Limitation exists if the bin value conforms a customer to a strategy with lower reduction than the maximum achievable. Each individual technique was tested against the different estimations we presented in section IV. As it was expected when using AVG as the estimate on the representatives we got the worst result(maximum average error $\approx$ 2.5%)(Fig.1). Although compared to MAX and MAVG estimates (maximum average error $\approx$ 0.7% and $\approx$ 1.3% respectively) AVG seems to produce the worst results, it still outperforms the baseline we set by much. In general the estimate that gives us the best result is MAX. This correlated to the need we have not to bin limited by the bin value when choosing a strategy. So if we build our bins using this estimate we can always fit the maximum reduction provided by a specific strategy. In the case of MAVG we are more restricted from the bin value in comparison to MAX. However we are less restricted than in the case of AVG. So it was expected to get results in between MAX and AVG.In (Fig.2)we chose to present the results produced by the MAX estimate. This decision was made because we get the best results out of it.

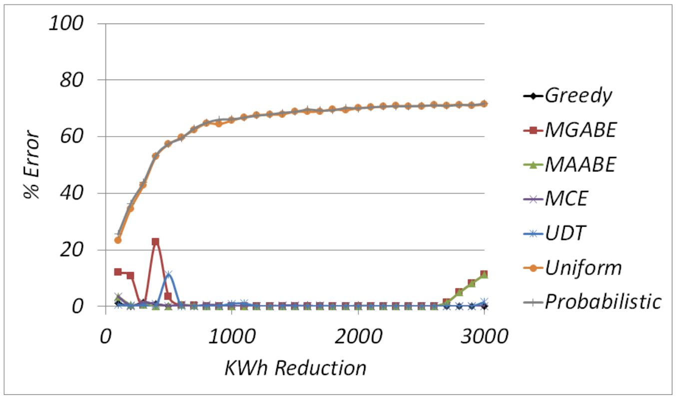

Fig. 1: Case 1 - Overall Targeted Reduction(AVG).

In order to compare the effectiveness of the techniques developed we plotted the Cumulative Distribution Function (CDF) of the normalized error. In (Fig.3) we present the result we got using the MAX estimate for the representatives. Although we argued that against the greedy technique it seems that we are getting better results in comparison to the other approaches. This can be explained by the nature of the dataset. In the campus there exist many buildings similar to each other that have a low deviation on their achievable reduction. Taking this into account we can realize that a greedy choice of building-strategy combination will approximate the target reduction accurately most of the time. However this is not the case for reduction of higher value. There are a few buildings with larger reduction that need to be included into the final solution. So the relative error increases for these values. A clearer view of the robustness of each approach can be realized in the second part of the experiments we conducted.

In general the results showed that MAABE is the technique which provides the best accuracy and stability. This is because we just focus only in minimizing one quantity when deciding on the unit value to be selected. Also by using an averaging method to estimate the achievable bin value, we can always get the best result given a good representative estimate as it was explained before. An average closer to the bin value always considers the maximum potential reduction of the customers in a bin. In MGABE the results produce the worst error from the targeted reduction. Given the reasoning we just presented this was expected. The independent minimization of the two quantities is responsible for this performance. Selecting a suitable unit value is based in the specific targeted reduction. Nevertheless we consider all the bins for the accumulated error which results in conflicting objectives. MCE was developed on that reasoning and focused to minimize the accumulated error on specific bins indexed by the change making algorithm. As the results show, it produces lower approximation error outperforming MGABE. It keeps up with MAABE but shows less stability and eventually performs worst. The problem in this case is that we consider each bin as having the necessary number of buildings needed. This in practice is not the case as sometimes the construction of bin ranges produce empty ones. Finally we consider the solution provided by UDT. In this case the results are follow a similar pattern to that of MCE. This approach is strongly correlated with the data distribution. A pitfall exists when the deviation between the reduction provided by each one of the buildings is large. The unit value selected fits the buildings with a small reduction and restricts the ones with a large reduction. This happens because the bin values are small considering the unit value chosen. In our case the deviation between the buildings is small that is why such a case is not clearly visible. It will become more clear in the next part of the experiments.

The second round of experiments aimed at testing the robustness of each method. In (Fig.4) we present the results we got using AVG for the calculation of representatives. The results are consistent with our claims about the importance of choosing a good estimate for the representatives. In this case(AVG) none of the methods we developed can match the

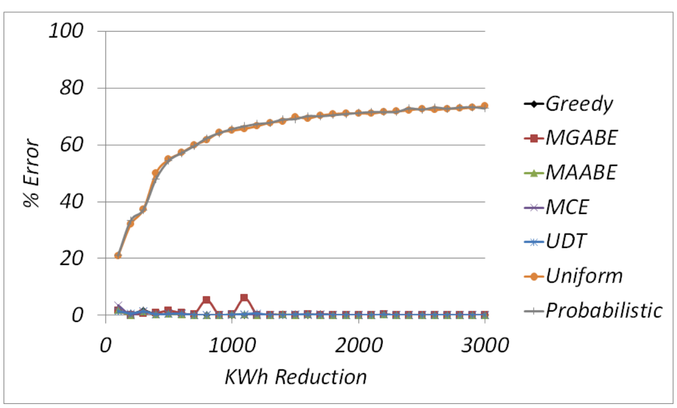

Fig. 2: Case 1 - Overall Targeted Reduction(MAX).

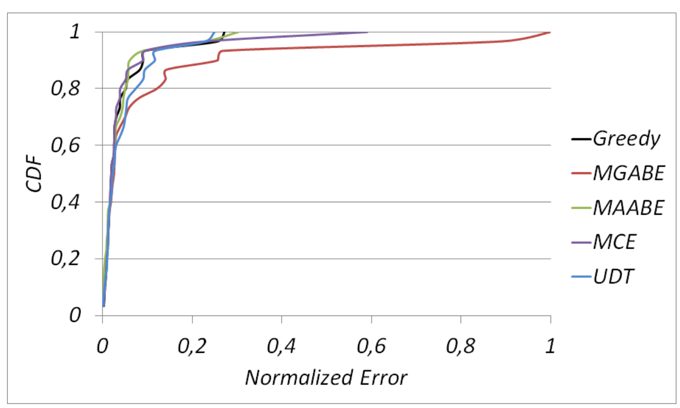

Fig. 3: Case 1 - Accuracy of Unit Value Techniques(MAX).

performance of the greedy technique. In (Fig.5) we chose to present the results for the case of MAX since they are similar to that of MAVG although a little bit worst. We realize that the results follow the same pattern as in the first round of the experiments. Our claim in favour of MAABE is supported since it manages to match and outperform every other method. The greedy approach ranks second along with MCE. Finally we see that MGABE has an unstable performance in terms of accuracy. It manages to follow at some cases the accuracy of other methods but it is unpredictable for the most part. Finally UDT manages to match the error produced by the greedy approach as expected. A point we need to make is that the overall error in this round of results has increased significantly. Since we discard randomly selected buildings from the original set there might not be a solution that fulfils our request. The error curve presented in Fig.5 is expected. In campus there are many buildings of low reduction which can be combined to achieve an overall low reduction. However to achieve a bigger reduction we need to include buildings with high reduction levels. Those are eliminated more easily from the final set since few of them exist.

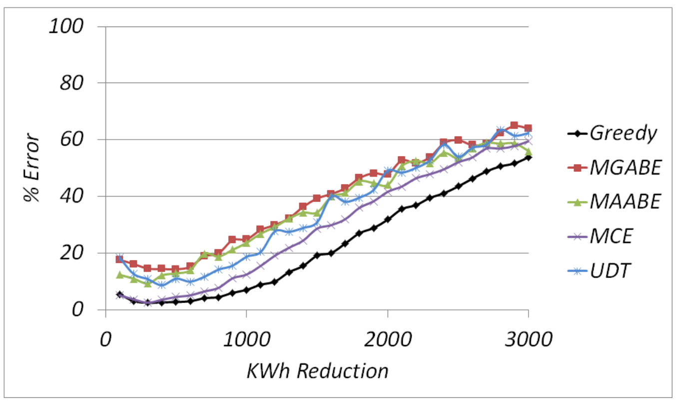

Fig. 4: Case 2 - Overall Targeted Reduction(AVG).

C. Reduction Sustainability

A strong asset of the selection algorithm we developed is achieving sustainability. We argue in favour of this by

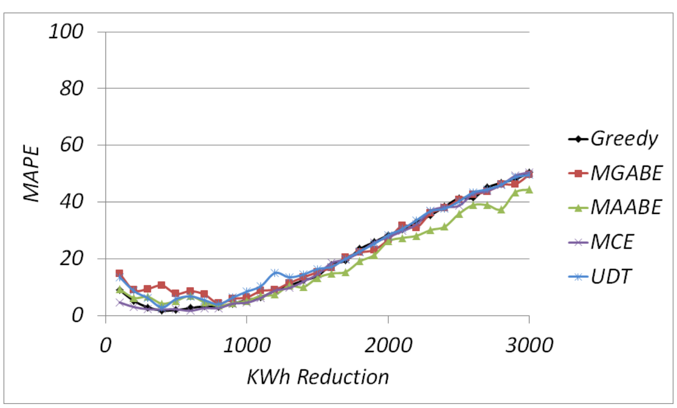

Fig. 5: Case 2 - Overall Targeted Reduction(MAX).

presenting the results for the first case of our experiments. We focused on the highest overall reduction we could achieve in our experiments(3000 KWh). We present the deviation from the optimal reduction per interval using the absolute percentage error as a metric. Our choice is based on the collective number of buildings used in the DR event. The goal behind this decision is to show how the behaviour of different buildings hinders our ability to achieve sustainability.

It is important to point out that sustainability it is not connected directly to achieving the overall load reduction across a DR event. A large average deviation from the target per interval might again provide the overall reduction we want by establishing lower reduction in the beginning of the event and higher close to the end.

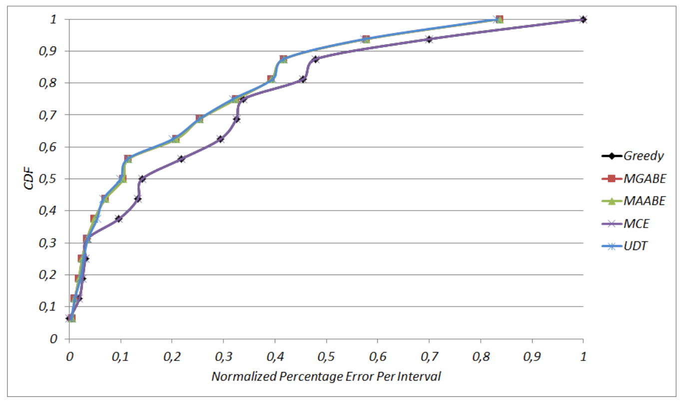

Fig. 6: Case 1 - Deviation of Sustainable Load(MAX)

The average deviation from the optimal sustainable reduction was less than $\approx 3\%$. The results(Fig.6) showed that we achieved the lowest deviation in the cases of MGABE, MAABE and UDT. Higher deviation has been observed for the case of the greedy method and MCE. The results are related to the method used to calculate the unit value. Although MGABE gives a high approximation error for the overall load reduction it is among the best to achieve sustainability of the load. Similar results are observed for MAABE and UDT. A common property of these methods is that they are trying to construct bin values that fit the patterns in the dataset. At this point our decision to present the results of an overall reduction achievable by the largest number of buildings is justified.In our attempt to provide a solution, we index all the bins and choose at least one building from each one. Creating a sustainable schedule requires us to minimize the error from the bin values. The bin values are calculated based on the unit value. So every approach that considers the existence of curtailment vectors that fit the reduction defined by the bin values can achieve the best sustainability. That is why the MAABE, UDT and MGABE achieve the best results. Given this statement one would expect to get the same results for the case of MCE. However this is not verified by the results. It is also not expected to be verified. MCE is set to minimize the error considering only specific bins. Also those bins are assumed to have the needed amount of buildings which is not

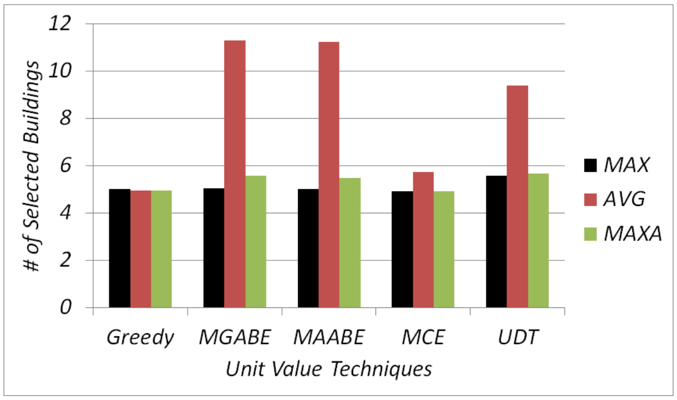

Fig. 7: Case 1 - Average Number of Building Selected.

the case. So MCE makes a wrong estimation that results in a unit value that poorly fits the dataset. In this way sustainability of the curtailed load is hindered. Similar reasoning can be applied for the greedy method, although in this case it is more obvious since there is no consideration of customers reduction patterns from the start.

D. Comfort Level

Since we used the change making approach to index the bins we expect to get the minimum number of building-strategy combinations to achieve the necessary reduction. In (Fig. 7) we see that for different techniques of calculating the representatives we used $\approx 15-34\%$ of the campus buildings to achieve the overall targeted reduction. The increase in building selection when using AVG is caused by the choice of the unit value. As it was noted before the unit value limits the potential reduction of each building from the resulting bin values. In that case the overall reduction is achievable through selection of more buildings. It was stated that the comfort levels are implied through the intrusiveness of each strategy. This is induced into the potential sustainable reduction patterns of each building associated with a specific strategy. It is the goal of the algorithm to detect those patterns and discard the ones that do not achieve a consistent reduction. In (Fig. 8) a heat map was created including the buildings of the provided solutions. Each building has an available strategy depicted with different code in the second column. To construct the heat map strategies of the same building were grouped together. Higher and lower reduction per interval is depicted by different levels of green and red respectively. Reduction level in the middle is drawn with yellow and orange. It can be seen that our solution selects buildings with consistent high reduction. We can associate the curtailment vectors which produce inconsistent load reduction to intrusive strategies. In that case we would have consistent low reduction or inconsistent patterns with reduction of high deviation among intervals. It is important to note that the algorithm return the optimal selection of building-strategy combinations in respect to the ones available.

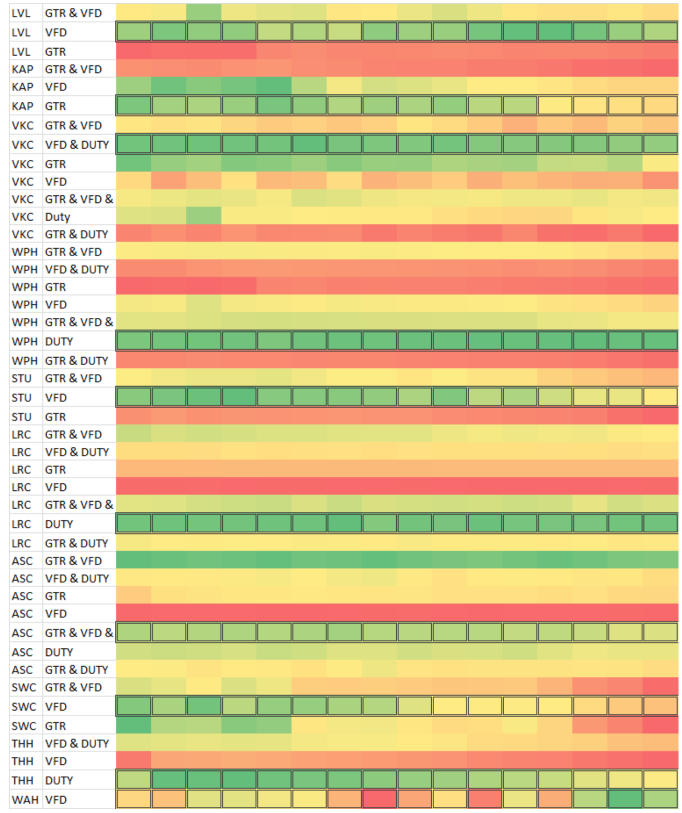

Fig. 8: Case 1 - Heat Map of Building - Strategy.

E. Execution Time

Evaluation of the efficiency was done by using synthetic data. A simulation of 1000 DR requests provided us with the maximum average execution time. We present the results in (Fig. 9). The number of customers ranged from 1000 to 32000 each one having 10 available strategies. The DR event time frame was 4 hours (16 intervals). In our results we observed an almost linear increase in the execution time for all methods excepts in MAABE. There we got a polynomial increase in the execution time. All results were expected and derive directly from the complexity analysis in the subsection IV-E. It is important to note that the bottleneck in the execution was the calculation of the unit value. Since the algorithm inherently does not have any data dependencies a distributed election algorithm can be used to decide on the unit value.

VI. Conclusion & Future

In this paper we focused on solving the problem of Dynamic DR scheduling. Our goals where to achieve a sustainable targeted reduction while factoring uncertainty induced by customer discomfort. It was shown that our proposed solution achieves a sustainable load reduction in respect to the targeted one. It detects behavioural patterns implied in the reduction levels of customers associated with specific strategies. The provided solution fulfils the dynamic requirements as it provides a schedule in a reasonable amount of time in respec to the number of customers. In our future work we

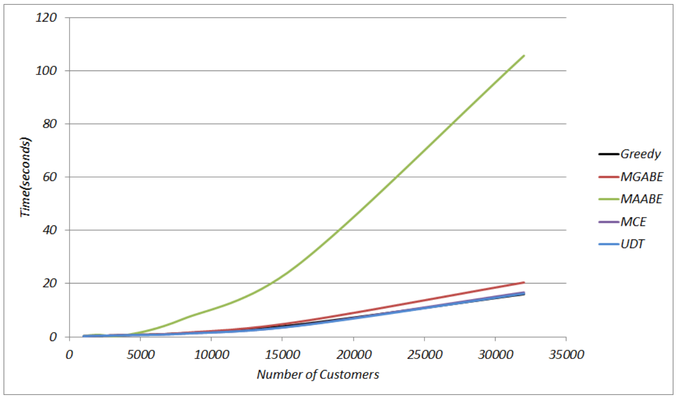

Fig. 9: Timings of Consecutive DR events.

will be focusing in two parts. Although the algorithm has a low complexity it presents a bottleneck when retrieving the customer information. It is imperative to provide a distributed solution to overcome this pitfall. Moreover we need to deal with uncertainty induced by changes in the customer behaviour through multiple DR events. This again affects the dynamic nature of the algorithm. It is important to factor on demand update of dataset to be accurate in our scheduling.

VII. Acknowledgment

This material is based upon work supported by the United States Department of Energy under Award Number number DE-OE0000192, and the Los Angeles Department of Water and Power (LA DWP). The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, the LA DWP, nor any of their employees.

References

[1] 10-day average baseline and day-off adjustment. Technical report, Southern California Edison, 2011.

[2] M. Albadi and E. El-Saadany. Demand response in electricity markets: An overview. In *IEEE Power Engineering Society General Meeting*, volume 2007, pages 1–5, 2007.

[3] S. Aman, M. Frincu, C. Charalampos, U. Noor, Y. Simmhan, and V. Prasanna. Empirical comparison of prediction methods for electricity consumption forecasting. Technical Report 14-942, University of Southern California, 2014. URL: http://www.cs.usc.edu/assets/007/89887.pdf (accessed Mar 6, 2014).

[4] S. Aman, Y. Simmhan, and V. K. Prasanna. Improving energy use forecast for campus micro-grids using indirect indicators. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 389–397. IEEE, 2011.

[5] S. Bahrami, M. Parniani, and A. Vafaeimehr. A modified approach for residential load scheduling using smart meters. In *Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on*, pages 1–8, Oct 2012.

[6] F. Benzi, N. Anglani, E. Bassi, and L. Frosini. Electricity smart meters interfacing the households. *Industrial Electronics, IEEE Transactions on*, 58(10):4487–4494, 2011.

[7] B. Chai, Z. Yang, and J. Chen. Optimal residential load scheduling in smart grid: A comprehensive approach. In *Control Conference (ASCC), 2013 9th Asian*, pages 1–6, June 2013.

[8] C. Chen, J. Wang, Y. Heo, and S. Kishore. Mpc-based appliance scheduling for residential building energy management controller. *Smart Grid, IEEE Transactions on*, 4(3):1401–1410, Sept 2013.

[9] J. Chen, B. Yang, and X. Guan. Optimal demand response scheduling with stackelberg game approach under load uncertainty for smart grid. In *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*, pages 546–551, Nov 2012.

[10] W.-C. Chu, B.-K. Chen, and C.-K. Fu. Scheduling of direct load control to minimize load reduction for a utility suffering from generation shortage. *Power Systems, IEEE Transactions on*, 8(4):1525–1530, Nov 1993.

[11] A. I. Cohen and C. Wang. An optimization method for load management scheduling. *Power Systems, IEEE Transactions on*, 3(2):612–618, May 1988.

[12] J. Ding, R. Yu, and Y. Liu. Optimal phevs charging management by tradeoff between utility cost and user satisfaction. In *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, pages 936–941, Aug 2013.

[13] X. Fang, S. Misra, G. Xue, and D. Yang. Smart gridthe new and improved power grid: a survey. *Communications Surveys & Tutorials, IEEE*, 14(4):944–980, 2012.

[14] N. Gatsis and G. Giannakis. Cooperative multi-residence demand response scheduling. In *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, pages 1–6, March 2011.

[15] N. Gatsis and G. Giannakis. Residential demand response with interruptible tasks: Duality and algorithms. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 1–6, Dec 2011.

[16] I. Georgievski, V. Degeler, G. Pagani, T. A. Nguyen, A. Lazovik, and M. Aiello. Optimizing energy costs for offices connected to the smart grid. *Smart Grid, IEEE Transactions on*, 3(4):2273–2285, Dec 2012.

[17] H. Goudarzi, S. Hatami, and M. Pedram. Demand-side load scheduling incentivized by dynamic energy prices. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 351–356, Oct 2011.

[18] F. Mangiatordi, E. Pallotti, P. Del Vecchio, and F. Leccese. Power consumption scheduling for residential buildings. In *Environment and Electrical Engineering (EEEIC), 2012 11th International Conference on*, pages 926–930, May 2012.

[19] A.-H. Mohsenian-Rad, V. 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*, 1(3):320–331, Dec 2010.

[20] K. Moslehi and R. Kumar. A reliability perspective of the smart grid. *Smart Grid, IEEE Transactions on*, 1(1):57–64, 2010.

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

[22] S. Salinas, M. Li, and P. Li. Multi-objective optimal energy consumption scheduling in smart grids. *Smart Grid, IEEE Transactions on*, 4(1):341–348, March 2013.

[23] A. Sepulveda, L. Paull, W. Morsi, H. Li, C. Diduch, and L. Chang. A novel demand side management program using water heaters and particle swarm optimization. In *Electric Power and Energy Conference (EPEC), 2010 IEEE*, pages 1–5, Aug 2010.

[24] H. Simão, H. Jeong, B. Defourny, W. Powell, A. Boulanger, A. Gagneja, L. Wu, and R. Anderson. A robust solution to the load curtailment problem.

[25] S. Tang, Q. Huang, X. yang Li, and D. Wu. Smoothing the energy consumption: Peak demand reduction in smart grid. In *INFOCOM, 2013 Proceedings IEEE*, pages 1133–1141, April 2013.

[26] Wikipedia. Autoregressive integrated moving average(arima) — Wikipedia, the free encyclopedia, 2014. [Online; accessed 15-March-2014].

[27] Wikipedia. Change-making problem — Wikipedia, the free encyclopedia, 2014. [Online; accessed 15-March-2014].

[28] Wikipedia. Integer programming — Wikipedia, the free encyclopedia, 2014. [Online; accessed 15-March-2014].

[29] Wikipedia. Knapsack problem — Wikipedia, the free encyclopedia, 2014. [Online; accessed 15-March-2014].

[30] D. Zhang, L. G. Papageorgiou, N. J. Samsatli, and N. Shah. Optimal scheduling of smart homes energy consumption with microgrid. In *ENERGY 2011, The First International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*, pages 70–75, 2011.