Load Scheduling for Residential Demand Response on Smart Grids[☆]

Miguel F. Anjos[a,b,*], Luce Brotcorne[a], Martine Labbé[a,c], María I. Restrepo[a,d]

[a] *Inria Lille-Nord Europe, Team Inocs, 40 Avenue Halley, 59650 Villeneuve d'Ascq, France*
[b] *GERAD and Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal, Québec H3C 3A7, Canada*
[c] *Département d'Informatique, Université Libre de Bruxelles, CP 212, Boulevard du Tromphé, 1050 Bruxelles, Belgique*
[d] *CIRRELT and Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal, Québec H3C 3A7, Canada*

Abstract

The residential load scheduling problem is concerned with finding an optimal schedule for the operation of residential loads so as to minimize the total cost of energy while aiming to respect a prescribed limit on the power level of the residence. We propose a mixed integer linear programming formulation of this problem that accounts for the consumption of appliances, generation from a photovoltaic system, and the availability of energy storage. A distinctive feature of our model is the use of operational patterns that capture the individual operational characteristics of each load, giving the model the capability to accommodate a wide range of possible operating patterns for the flexible loads. The proposed formulation optimizes the choice of operational pattern for each load over a given planning horizon. In this way, it generates a schedule that is optimal for a given planning horizon, unlike many alternatives based on controllers. The formulation can be incorporated into a variety of demand response systems, in particular because it can account for different

---

[☆] The authors are listed in alphabetical order.

[*] Corresponding author

Email addresses: anjos@stanfordalumni.org (Miguel F. Anjos), luce.brotcorne@inria.fr (Luce Brotcorne), mlabbe@ulb.ac.be (Martine Labbé), maria-isabel.restrepo-ruiz@polymtl.ca (María I. Restrepo)

aspects of the cost of energy, such as the cost of power capacity violations, to reflect the needs or requirements of the grid. Our computational results show that the proposed formulation is able to achieve electricity costs savings and to reduce peaks in the power consumption, by shifting the demand and by efficiently using a battery.
Keywords: Residential load scheduling, Demand response, Energy storage, optimization

---

1. Introduction

Electric power systems in many countries are experiencing major changes in their operations. One of the fundamental developments is the increasing need for the power grid to obtain sufficient *flexibility* from the demand side to be able to balance supply and demand. This is due to a combination of various factors, including the general trend for systems to operate more frequently near their maximum power production levels, the integration of greater quantities of distributed generation whose output is mostly intermittent, and the increasing penetration of electric vehicles. The collection of means available to procure flexibility from the demand side of the balance is commonly referred to as *demand response (DR)*.

While it is challenging to harness, as we discuss below, the potential for buildings to provide DR is significant. Worldwide, the power consumption of buildings accounts for an estimated 40% of global energy consumption [19]. In the United States, a 2012 survey by the US Federal Energy Regulatory Commission (FERC) reports a total potential for peak reduction of over 66 GW [4]. Residential and commercial buildings represent around 70% of the total energy demand and the potential of peak DR was estimated in 2015 at 8.7 GW in the United States [3]. In Canada, space heating is responsible for more than 60% of the total residential energy consumption. On the other hand, the province of Ontario is a summer-peaking system with a high penetration of air-conditioning systems [2, 1].

The main goal of residential DR programs is to motivate customers to adapt their power consumption habits via price incentives [7]. For instance, DR can be related to shifting electric heating away from electricity price peaks, charging (resp.

discharging) an energy storage device (e.g., battery, electric vehicle) at times of lower (resp. higher) prices, or delaying the use of a dishwasher.

Although DR is frequently cast in terms of reducing the peaks in demand, more generally the objective is to reduce the fluctuation of net demand, which is equal to the total electric demand in the system minus the contribution of intermittent generation. For this reason we make use of the concept of (power) *capacity limit*: this is a prespecified target for the maximum amount of electric power that the building uses at any given time. By committing in advance to such a target towards the power grid operating entity, the building contributes to a decrease in the fluctuations of demand. We do not address here the matter of selecting a suitable capacity limit for a given building, but we refer the reader to the recent papers [10, 11] that provide an in-depth study of this question.

Assuming that the capacity limit is set to a level of power that requires the building to make adjustements to the timing and the level of its power demands, it is necessary to manage the operation of the power-consuming devices in the building in order to respect the power capacity limit. Any practical home energy management system must work within the parameters provided by the user, such as ensuring an adequate comfort level for the temperature, and the availability of an electric vehicle when needed. On the other hand, it is possible to take advantage of the characteristics of the individual loads, such as the possibility to adjust their operating power level and/or to postpone them, or the ability to pause (and later resume) a load that is currently operating [13]. Load scheduling becomes even more important with the integration of energy storage units into buildings; by its very nature, storage can be scheduled to store and release energy in the right quantities and at the right times to help respect the power capacity limit. The management and optimization of residential energy consumption becomes much more complex when all these elements are considered, and it can thus benefit from the use of optimization techniques. We refer the reader to [6] for more details on the applications of optimization to this area.

In this paper we propose a mixed integer linear programming (MILP) formulation of the residential load scheduling problem. The proposed formulation optimizes the

choice of operational pattern for each load over a given planning horizon (assumed to be 24 hours, without loss of generality), and can be used as part of the system architecture for demand side load management presented in [9] or any similar system. A distinctive feature of our model is the capability to accommodate a wide range of possible operating patterns for the flexible loads, thanks to the use of operational patterns. Moreover, we can in principle incorporate different cost functions (e.g., surcharge for power capacity violations) to reflect the needs or requirements of the grid.

2. Related Work

There has been a significant increase in research studies on improving DR systems, see the surveys [5, 18]. These studies have been reviewed and categorized according to several criteria, including: the DR program control mechanism (centralized or distributed); the incentives offered to the customer to reduce power consumption; and the DR optimization techniques used. An extensive comparative analysis of the methods proposed in the literature for modelling different aspects of residential energy management is presented in [7], where the authors also discuss the challenges concerning different modelling approaches, and examine the computational considerations associated with home energy management systems

Load scheduling for DR has been formulated as an optimization problem and solved via mathematical optimization or meta-heuristic methods. The mathematical optimization approaches used include linear programming (LP), quadratic programming (QP), mixed integer linear programming (MILP), and mixed integer non-linear programming (MINLP). Recent studies using these approaches can also be categorized depending on the planning horizons considered, these being in some cases multiple or rolling horizons [12, 8, 17] or in other cases a single horizon [16, 15, 20, 21].

A two-stage optimization model for load scheduling in a building energy management problem is introduced in [12]. The first stage solves a medium-term MILP problem, defined over a time horizon of one day. The results of the first stage model are used to solve a short-term MILP problem that determines the optimal energy management for the next hour. Similarly, [8] propose a two-horizon algorithm

to obtain load schedules with a reasonable energy cost, while limiting the computational time. By dividing the objectives into a low-resolution component and a high-resolution component, the algorithm is able to reduce the impact of forecast errors on the schedule, as well as the computational burden. A residential energy management problem including uncertainty in real-time prices, weather conditions, and occupant behaviour is solved in [17]. The authors propose an online stochastic optimization algorithm using a rolling horizon approach with the time horizon divided into time periods of possibly different durations. Using a 16-hour horizon with the first two time periods of 15 minutes, then 30-minute periods for the rest of the horizon, the authors report important monetary and comfort cost savings, when compared to control-based approaches that can over-react.

The contributions most closely related to the approach presented in this paper are listed in Table 1 where we indicate for each reference if the model considers energy storage (*Storage*), task preemptions (*Preemptions*), and a surcharge for the excess in power consumption (*Surcharge*). The last column (*Time Period*) presents the length of the time periods.

Reference	*Storage*	*Preemptions*	*Surcharge*	*Time Period*
[15]	Yes	Yes	No	10 min
[16]	Yes	Yes	No	60 min
[20]	Yes	No	No	60 min
[14]	Yes (in EVs)	No	No	60 min
[21]	Yes	No	Yes	60 min

Table 1: Characterization of related works.

We briefly summarize these contributions. The first two papers both consider storage and preemptions, but not the possibility of exceeding the power limit, and they differ in the length of the time periods used. Specifically, [15] proposes a MILP that incorporates the appliances' operation priorities in an energy management system with the objective of scheduling the household demand for the next day so as to achieve the minimum cost. The alternative load scheduling algorithm proposed

in [16] controls the operation time and energy consumption level of each appliance so as to maximize the net utility of the residence, while satisfying its budget limit. This approach is formulated as a MINLP and solved via Benders decomposition.

The paper [20] introduces a MILP that also seeks to minimize the electricity costs (under a time-varying electricity price). While it does not allow preemptions, it includes different energy consumption patterns of home appliances, a photovoltaic system, and storage. The paper [14] also uses a MILP formulation but with the objective of minimizing the electricity generated from conventional sources in the context of a microgrid of residential buildings, and also considering energy consumption patterns of home appliances, photovoltaic generation, and storage in the forms of EVs in the homes.

Finally, the authors of [21] present a MILP that minimizes the peak hourly load to achieve a balanced daily load schedule. The proposed model schedules the optimal power, as well as the optimal operation time for appliances classified into two groups: power-shiftable and time-shiftable. The latter, also known as deferrable loads with deadlines, are tasks that require consuming a given total amount of energy to finish, but their operation can be scheduled very flexibly before the given deadline.

The approach that we propose in this paper takes into account all of energy storage, power consumption excess surcharge, and fully flexible preemptions. This last feature is in contrast with most of the literature where task preemptions, if considered, apply only to appliances such as battery chargers or vacuum cleaners, or only in situations where the minimum time a task must be running before being preempted is not controlled.

Moreover, most of the literature uses time periods of 60 minutes, reducing the flexibility that the energy management system can provide. We report computational results using time periods of 15 minutes which is more detailed than most of the literature, including the approaches in Table 1. The ability to use shorter time periods in the scheduling can be particularly useful for the provision of DR because it supports a more targeted response by the home energy system to the requests of the grid or a third party such as an aggregator or virtual power plant.

This paper is structured as follows. In section 3, we present the mathematical

formulation of our proposed load scheduler. In Section 4, we report and discuss the computational results of our preliminary implementation. Finally, Section 5 concludes the paper and outlines possible future research directions.

3. Load Scheduling Model

In this section, we present the approach to address the residential load scheduling problem. First, we categorize the tasks according to their attributes and present the procedure to generate load patterns for each task type. Second, we present the proposed mathematical model to find the optimal load schedule.

3.1. Task Types and Pattern Generation

We assume that all time periods have the same length and that Δ_t represents the length of the time periods.

For each task j, let S_j be the set of possible operational patterns (schedules) over the planning horizon. These patterns depend on the attributes of each task and on the individual user preferences. This leads to the following categorization of the tasks into three types:

- **Type A tasks** are *thermal or regular loads* that correspond to devices ensuring that specific temperatures are maintained. Examples of such devices are heaters, air conditioners, refrigerators, and water heaters. Because their power consumption can be adjusted (within given limits), and in some cases interrupted for a limited time, their operation can be managed.

  The choice of the operational patterns for these loads must account for factors such as user preferences, and the interaction between the external and internal temperatures of the building. We assume that these patterns are computed in advance using available knowledge about the loads, such as would be collected by the forecaster module of the architecture in [9].

- **Type B tasks** are *activity-based loads*, also called burst loads in [9]. These tasks arise from devices whose operation has a fixed duration and power consumption,

and each task must start and finish within a given time window. Examples of type B tasks are dishwashers, washing machines, and dryers.

Each of these tasks j has an associated operation time window $[l_j, u_j]$ and a number of subtasks ζ_j, where each subtask $st \in \zeta_j$ corresponds to a different phase of the operation of device j. For instance, the operation of a washing machine consists of three phases, each with a different duration and power consumption:

- During the first phase, water is pumped into the drum and a sequence of drum rotations is performed.
- In the second phase, fresh water is pumped into the drum and the rinsing process is carried out.

Therefore, each $st \in \zeta_j$ is associated with a fixed duration (number of time periods) and a (possibly varying) power consumption per time period. It follows that the associated number of possible preemptions is $\rho_j = \zeta_j - 1$. We assume that each preemption has an associated minimum and a maximum duration (in time periods).

The operational patterns for this type of tasks are generated with the following procedure. First, we create all possible *sequences of preemptions* for each task $j \in J_B$. These sequences are built with a *recursion tree*, including the number of preemptions with their minimum and maximum duration. An example of such a tree is presented in Figure A.10. Second, we include each subtask $st \in \zeta_j$ in each sequence of preemptions to generate a set of *potential operational patterns*, denoted as $\tilde{S}_j$. Third, for each time period t in time window $[l_j, u_j]$, we verify if $s \in \tilde{S}_j$ can be entirely contained in $[t, u_j]$. If so, we add pattern s, with start time t, to the set of feasible operational patterns S_j. Otherwise s starting at t is rejected. An example of this procedure is presented in AppendixA.

- **Type C tasks** are *highly flexible loads* that allow different power usage at each time period so long as a predefined amount of energy is allocated to them during the planning horizon. Garage heaters, pools, and plug-in electric vehicles are examples of such tasks.

These tasks have the same characteristics as type B tasks, except that the power consumption at each time period is not specified but instead can vary between a lower limit $P_{jt}^{\min}$ and an upper limit $P_{jt}^{\max}$. In addition, the total energy consumption θ_j has to be completed during the operation of the task.

The generation of all possible operational patterns is done in the same way as for tasks of type B.

Figure 1 shows an example of different operational patterns for tasks type A, B, and C. In this example, the planning horizon is twelve hours divided into 1-hour time periods. The first three rows contain the patterns for the operation of a space heater (SH). The following rows, labeled as DW, contain all possible schedules for a dishwasher that should operate between 10:00 and 12:00. The operation of this appliance consists of two subtasks, therefore there is one preemption. The length and power consumption of the first subtask is one hour and 1.8 kW, respectively. The length and power consumption of the second subtask is one hour and 1 kW, respectively. The maximum length of the preemption is one hour. The last row shows the only feasible pattern for a garage heater (GH), representing a task of type C. This garage heater can be turned on between 16:00 and 20:00. A total amount of energy of 8 kWh must be consumed within this time window and the values for the minimum and maximum power consumption at each time period are 0 kW and 3 kW, respectively.

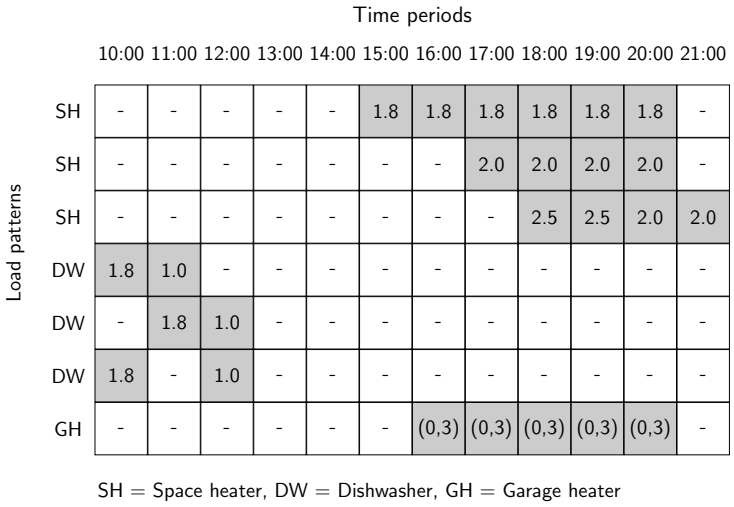

Figure 1: Load patterns for three tasks.

3.2. Mathematical Model

We now present the mixed integer programming model that calculates an optimal schedule for the operation of loads. The data required for the model is given in Table 2.

Sets	
T	Set of time periods in the planning horizon;
B	Set of batteries;
J	Set of tasks (loads);
J_A	Set of tasks of type A;
J_B	Set of tasks of type B;
J_C	Set of tasks of type C;
Parameters	
Δ_t	Length of the time periods (h);
E_t	Electricity cost at time period t (\$/kWh);
M	Excess power cost (\$);
C_t	Contracted amount of power at time period t (kW);
Γ_b	Energy capacity of battery b (kWh);
E_{bt}	Total energy available in battery b at time period t (kWh);
κ_b	Initial energy in battery b (kWh);
η_b	Auto discharge rate of battery b (%);
ϕ_b	Maximum number of cycles for battery b ;
c_{bt}, d_{bt}	Charging and discharging efficiencies of battery b at time period t (%);
p^c_{bt}, p^d_{bt}	Charging and discharging power of battery b at time period t (kW);
δ^s_{jt}	Equals 1 if task j is covered at time period t by schedule s, 0 otherwise;
P^s_{jt}	Power consumption of task $j \in J_A \cup J_B$ at time period t for schedule s (kW);
F^s_j	Operation cost of task $j \in J_A \cup J_B$ for schedule s ($F^s_j = \sum_{t \in T} \Delta_t P^s_{jt} E_t$) (\$);
$P^{\min}_{jt}, P^{\max}_{jt}$	Lower and upper limits for the power consumption of task $j \in J_C$ at time period t (kW);
θ_j	Total energy consumption of task $j \in J_C$ (kWh).

Table 2: Load scheduling problem notation.

The formulation of the load scheduling model, denoted as LS, is as follows:

$$f(LS) = \min \sum_{j \in J_A \cup J_B} \sum_{s \in S_j} F_j^s x_j^s + \sum_{j \in J_C} \sum_{t \in T} \Delta_t E_t p_{jt} + \sum_{b \in B} \sum_{t \in T} E_t q_{bt} + M \sum_{t \in T} y_t \quad (1)$$

$$\text{s.t.} \sum_{j \in J_A \cup J_B} \sum_{s \in S_j} P_{jt}^s x_j^s + \sum_{j \in J_C} p_{jt} + \sum_{b \in B} \frac{q_{bt}}{\Delta_t} \leq C_t + y_t, \; \forall \, t \in T, \quad (2)$$

$$\sum_{s \in S_j} x_j^s = 1, \; \forall \, j \in J, \quad (3)$$

$$\sum_{t \in T} \Delta_t p_{jt} = \theta_j, \; \forall \, j \in J_C, \quad (4)$$

$$P_{jt}^{\min} \sum_{s \in S_j} \delta_{jt}^s x_j^s \leq p_{jt} \leq P_{jt}^{\max} \sum_{s \in S_j} \delta_{jt}^s x_j^s, \forall \, j \in J_C, \; t \in T, \quad (5)$$

$$\Delta_t d_{bt} p_{bt}^d (z_{bt} - 1) \leq q_{bt} \leq \Delta_t c_{bt} p_{bt}^c z_{bt}, \; \forall \, b \in B, \; t \in T, \quad (6)$$

$$z_{bt+1} - z_{bt} \leq w_{bt}, \; \forall \, b \in B, \; t \in T | t < |T|, \quad (7)$$

$$\sum_{t \in T} w_{bt} \leq \phi_b, \; \forall \, b \in B, \quad (8)$$

$$E_{bt+1} = \eta_b E_{bt} + q_{bt}, \; \forall \, b \in B, \; t \in T | t < |T|, \quad (9)$$

$$0 \leq E_{bt} \leq \Gamma_b, \; \forall \, b \in B, \; t = \{2, ..., |T| - 1\}, \quad (10)$$

$$E_{bt} = \kappa_b, \; \forall \, b \in B, \; t = \{1, |T|\}, \quad (11)$$

$$x_j^s \in \{0, 1\}, \; \forall \, j \in J, \; s \in S_j, \quad (12)$$

$$p_{jt} \geq 0, \; \forall \, j \in J_C, \; t \in T, \quad (13)$$

$$y_t \geq 0, \; \forall \, t \in T, \quad (14)$$

$$z_{bt} \in \{0, 1\}, w_{bt} \geq 0, \; \forall \, b \in B, \; t \in T. \quad (15)$$

We next explain the structure of the model. Instead of strictly following the order of the equations above, we structure our explanation around the various components of the model, concluding with the description of how the objective function collects the impact of all the components.

We begin with the tasks of type A or B. For these tasks, the scheduling decisions are represented by the binary variables x_j^s such that $x_j^s = 1$ if scheduling pattern $s \in S_j$ is chosen for task $j \in J$. The constraints (3) guarantee that precisely one

pattern is selected per task.

For tasks of type C, there are no pre-defined schedules, and therefore the power consumption of task $j \in J_C$ at time period $t \in T$ is represented by the continuous variables p_{jt}. Constraints (4) require that the total energy consumption of task j be met during the planning horizon. The variables are p_{jt} linked to the scheduling decisions by constraints (5) that ensure p_{jt} is between the prescribed lower limit $P_{jt}^{\min}$ and upper limit $P_{jt}^{\max}$, according to the values of x_j^s. In particular for a given task j, if $x_j^s = 0$ for all $s \in S_j$ then (5) sets $p_{jt} = 0$.

There are three sets of variables associated with the operation of the battery. The first set of variables q_{bt} represent the amount of energy injected into/extracted from battery b at time period t. Specifically, for each time period $t \in T$, if $q_{bt} > 0$ then battery b is charging, if $q_{bt} < 0$ then battery b is discharging, and if $q_{bt} = 0$ then battery b is inactive. The energy conservation for battery $b \in B$ is accordingly enforced by constraints (9) that update the energy stored in the battery at time period $t+1$ taking into account the value of q_{bt}. In addition, constraints (10) require that the energy in the battery remain non-negative and no greater than the energy capacity of the battery, and constraints (11) ensure that the energy level of the battery at the end of the planning horizon matches the level at the start.

The second and third set of battery-related variables are the binary variables z_{bt} and w_{bt}. The meaning of z_{bt} is that it equals 1 if battery b is charging at time period t (and 0 otherwise), and the meaning of w_{bt} is that it equals 1 if the operation of the battery changes from discharging or inactive to charging (and 0 otherwise). These variables are linked to the energy variables q_{bt} by constraints (6) that ensure q_{bt} is between the minimum and maximum power levels allowed for discharging/charging battery $b \in B$ at time period $t \in T$. Note that these levels depend on the values of z_{bt}; in particular for time period t, if $z_{bt} = 1$ then battery b can only charge, and if $z_{bt} = 0$ then it can only discharge.

An additional set of variables and constraints enforce a maximum number of operational cycles for the battery. A battery cycle occurs when the battery charges and then discharges (or conversely discharges and then charges) during a certain number of time periods (time block). Note that this time block might include some

periods where the battery is inactive. It is common to restrict the number of cycles allowed as a means to limit the degradation of the battery. For this purpose, the variables w_{bt} track the changes in z_{bt} via constraints (7), and constraint (8) enforces the desired maximum number of cycles. Note that $w_{bt} = 1$ is ensured only when the battery goes from discharge to charge. For the other cases, w_{bt} can take the value 0 or 1, but because we are minimizing the total cost w_{bt} will always equal 0 (for the model to allow more battery cycles). The remaining constraints (12 - 15) defines the type of each set of variables as required.

The objective function (1) is the sum of four different terms. The first two terms account for the energy consumption cost of the tasks according to the task type. The first term represents the energy consumption cost for tasks of types A and B over the planning horizon, where the cost of executing task j according to schedule s is computed in advance as F_j^s. The second term represents the energy consumption cost for tasks of type C, for which there are no pre-determined schedules and therefore the cost of a task j is directly calculated according to its power consumption during each time period t.

The third term represents the net cost of the energy consumed by the battery, where the energy discharged at time period t is valued equally to energy purchased during that time period, and independently from when it was paid for and stored into the battery. (Should the user wish to calculate this cost differently, it is straightforward to adjust our model accordingly.)

The fourth term represents a cost surcharge for exceeding the pre-determined power capacity C_t during a time period t. Specifically, variable y_t denotes the amount of power beyond C_t used at time period t, and if y_t is positive, then a surcharge of M per unit of power is added to the objective function. Assuming that the cost M is substantial, the minimization will keep the value of y_t as close to 0 as possible, but it will be positive if according to constraint (2), the total power consumption at time period t is strictly greater than C_t.

4. Experimental Results

In this section, we present our computational experiments to test the performance and behaviour of the load scheduling model LS. First, we describe the set of instances used. Second, we report and discuss the results.

4.1. Instances

The instances were generated over a one-day planning horizon divided into 96 time periods of 15 min. We considered a battery with a capacity of 6.4 kWh and with an initial amount of stored energy of 3 kWh. The efficiency and power capacity for charging and discharging the battery were set to 100% and to 3.3 kW, respectively. We set the electricity prices according to a time-of-use pricing scheme with three levels:

- The first level (L_1) starts at 19:00 and ends at 7:00 of the next day, and the electricity cost is $E_t = \$0.087$/kWh for time periods t in L_1.
- The second level (L_2) covers two periods of the day, namely 7:00 to 11:00 and 17:00 to 19:00, and the electricity cost is $E_t = \$0.180$/kWh for time periods t in L_2.
- The third level (L_3) starts at 11:00 and ends at 17:00, and the electricity cost is $E_t = \$0.132$/kWh for time periods t in L_3.

We assume that the contracted amount of power depends on the electricity prices. Specifically we compute it with the formula

$$C_t = 3.2 + \lceil ((0.18 - E_t)/0.087) * 2 \rceil , \forall t \in T.$$

In addition the surcharge cost per extra unit of power above C_t was set to $M = \$6.20$/kWh.

Table 3 presents a list of the appliances considered for the generation of the instances. This list includes the appliance's name (*Load*), its type (*Type*), a column that indicates if the task allows preemptions or not (*Preem.*), the maximum number

of preemptions allowed (*N.Preem.*), and the total operation time, in hours, for tasks of type B or C (*Oper.time*). A detailed description of the characteristics and operation requirements of each appliance is given in AppendixB.

Load	*Type*	*Preem.*	*N.Preem.*	*Oper.time*
Water heater	A	No	-	-
Space heater	A	No	-	-
Fridge	A	No	-	-
Dishwasher	B	Yes	2	1.5
Washing Machine	B	Yes	1	0.75
Dryer	B	No	0	1
TV	B	No	0	2
Oven	B	No	0	1
Garage heating	C	Yes	6	-

Table 3: Appliances attributes.

We define 11 scenarios to evaluate the impact in the total cost by allowing task preemptions, by including a battery, by changing the number of cycles in the battery, and by surcharging the excess in power consumption. Table 4 presents the parameters of each scenario. *Surcharge* indicates if the excess power cost M is included in the objective function (1) of problem LS. *Preem.* indicates if tasks type B are allowed to have preemptions. *Battery* indicates if a battery is used, and *Cycles* denotes the maximum number of battery cycles during the planning horizon. “No" indicates that no limit was set on the number of cycles, and therefore variables w_{bt} and constraints (7) and (8) are removed from the model.

Scen.	*Surcharge*	*Preem.*	*Battery*	*Cycles*
1	No	No	No	-
2	Yes	No	No	-
3	Yes	No	Yes	No
4	Yes	Yes	No	-
5	Yes	Yes	Yes	No
6	Yes	Yes	Yes	1
7	Yes	Yes	Yes	2
8	Yes	Yes	Yes	4
9	Yes	No	Yes	1
10	Yes	No	Yes	2
11	Yes	No	Yes	4

Table 4: Scenario properties

4.2. Results

Tables 5 and 6 present the computational effort and the results for the eleven scenarios described in Table 4.

For each scenario (*Scen.*), Table 5 reports the time to solve the load scheduling model (*Time (s.)*), the time to generate the operational patterns (*Time.P (s.)*), and the total number of patterns generated for type B tasks ($N.PatternsB$) and for type C tasks ($N.PatternsC$). Table 6 presents the total energy cost (*E.Cost ($)*), the excess power surcharge (*P.Surch ($)*), the total energy consumed by the loads (*Energy (kWh)*), the total power peak (*T.Peak (kW)*) over the planning horizon, and the time periods at which the power peaks occur (*T.Periods Peak*).

Scen.	*Time (s.)*	*Time.P (s.)*	*N.PatternsB*	*N.PatternsC*
1	2.46	3.83	25	11,440
2	4.71	3.83	25	11,440
3	4.47	3.83	25	11,440
4	4.85	4.2	59	11,440
5	6.79	4.2	59	11,440
6	23.85	4.2	59	11,440
7	69.16	4.2	59	11,440
8	5.47	4.2	59	11,440
9	23.69	3.83	25	11,440
10	134.81	3.83	25	11,440
11	4.75	3.83	25	11,440

Table 5: Computational effort and number of operational patterns under different scenarios.

Scen.	*E.Cost ($)*	*P.Surch ($)*	*Energy (kWh)*	*T.Peak (kW)*	*T.Periods Peak*
1	9.9	0	91.45	66.5	[9,10,19,20,25-34,69-77,81-84]
2	11.19	233	97.85	37.52	[29,33,34,37-44,69-77,81,86-89]
3	9.41	7.45	91.45	1.2	[74-76]
4	11.19	233	97.85	37.52	[29,33,34,37-44,69-77,81,86-89]
5	9.41	1.74	91.45	0.28	[74-75]
6	11.07	88.55	97.85	14.26	[29,33,34,37-44,74,75,81,86-89]
7	10.32	44.46	95.95	7.16	[74,75,81,86,89]
8	9.42	1.74	91.45	0.28	[74-75]
9	11.12	91.81	97.85	14.78	[29,33,34,37-44,72,75,84-87,96]
10	10.37	47.72	95.91	7.68	[72,75,84-87,96]
11	9.43	7.45	91.45	1.2	[74-76]

Table 6: Costs, energy consumption, and power peak under different scenarios.

The results in Table 5 suggest that the scenario tested affects the CPU time to solve model LS. For instance, this CPU time is shorter when no surcharge for the

peak in power consumption is considered (Scenario 1). On the contrary, this time is significantly larger when a power surcharge $M > 0$ and a battery are included in the model. Specifically, when the number of battery cycles is restricted to two (Scenarios 7 and 10), we can observe that the values for *Time (s.)* are 28 and 55 times larger than for Scenario 1.

In Table 5 we also see that the use of preemptions in tasks of type B appears to not have a significant effect in the CPU time (see Scenario 3 versus Scenario 5, and Scenario 6 versus Scenario 9), as the task that has the largest number of patterns is the task "Garage heating" of type C.

We observe in Table 6 that the largest values for the four aspects evaluated (total energy cost, total surcharge cost, total energy consumed by the tasks, and total peak in power consumption) occur for the scenarios with no battery or with the number of battery cycles limited to one (Scenarios 2, 4, 6, and 9). Conversely, the energy and power surcharge costs significantly decrease for scenarios that include preemptions and allow more battery cycles (Scenarios 3, 5, 8, and 11).

4.2.1. Excess power charge, preemptions, and battery impact

Tables 7 and 8 report the results of comparisons between certain pairs of scenarios. For each table we present the scenarios compared (*Scen.C.*), as well as the percentage decrease (or increase if the value is negative) in the total power peak (*Peak.D.*), in the total cost (*T.Cost.D.*), and in the total energy cost (*E.Cost.D.*). These values were computed as

$$D = \frac{(x_j - x_i)}{x_j} \times 100,$$

so that D corresponds to the percentage decrease when scenario i is compared against scenario j, and x_j, x_i denote the values for the total peak (*T.Peak (kW)*), the total cost (*E.Cost ($) + P.Cost ($)*), and the energy cost (*E.Cost ($)*) for scenarios j and i, respectively.

Scen.C.	*Peak.D.*	*T.Cost.D.*	*E.Cost.D.*
2 vs 1	43.58%	-	-0.13%
3 vs 2	96.8%	93.09%	0.16%
4 vs 2	0.00%	0.00%	0.00%
5 vs 3	76.67%	33.9%	0.00%

Table 7: Comparison of results to measure the excess power charge, preemptions, and battery impact.

Scen.C.	*Peak.D.*	*T.Cost.D.*	*E.Cost.D.*
5 vs 6	98.04%	88.81%	0.15%
5 vs 7	96.09%	79.65%	0.09%
5 vs 8	0.00%	0.11%	0.00%
3 vs 9	91.88%	83.61%	0.15%
3 vs 10	84.38%	70.97%	0.09%
3 vs 11	0.00%	0.08%	0.00%

Table 8: Comparison of results to measure the battery cycles impact.

The results in Table 7 indicate the positive impact, in terms of the power peak reduction and the total cost decrease, of including a surcharge in power consumption, a battery, and preemptions for some tasks. In particular, a power peak decrease of 43.58% is observed when Scenario 1 is compared against Scenario 2. This decrease is obtained by including a surcharge $M > 0$ in the objective function of the model. Because some tasks can be shifted, i.e., start at different times, or have a different power consumption at each time period (e.g., type C tasks), a positive value of M leads to a more "balanced" power consumption along the planning horizon. This is shown by the *P.Cons* line in Figures 2 and 3.

The effect of using a battery is observed when Scenario 2 is compared against Scenario 3. The results from columns *Peak.D.* and *T.Cost.D.* suggest that storing energy when there is a surplus of power capacity and discharging energy when there

is not enough power capacity, leads to a large decrease in the power peak and the total energy costs (see Figures 3 and 4).

We did not observe any positive impact in the peak power and total cost reduction when type B task preemptions are allowed and there is not a battery (Scenario 2 versus Scenario 4). However, when a battery is included (Scenario 5), we observed a reduction of 76.67% and 33.9% in the peak power consumption and in the total cost, respectively.

Note that the comparison between Figures 4 and 6 show a change (shift) in the power consumption ($P.Cons$) between 11:00 and 19:00.

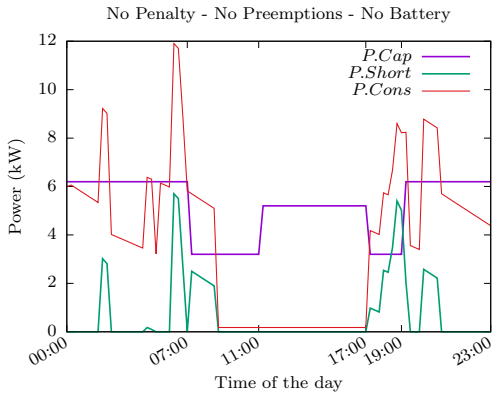

Figure 2: Scenario 1.

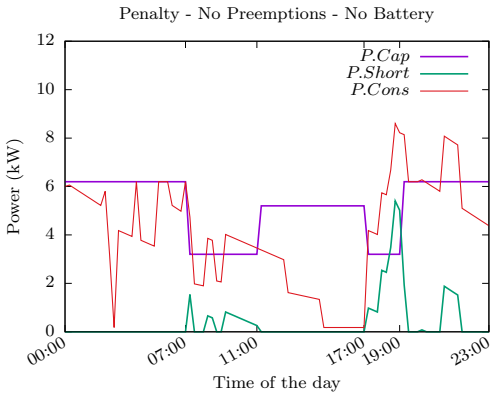

Figure 3: Scenario 2.

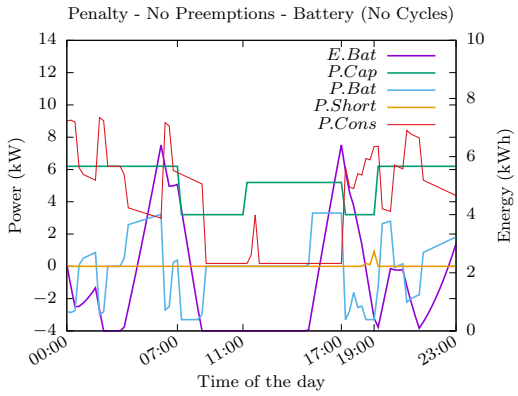

Figure 4: Scenario 3.

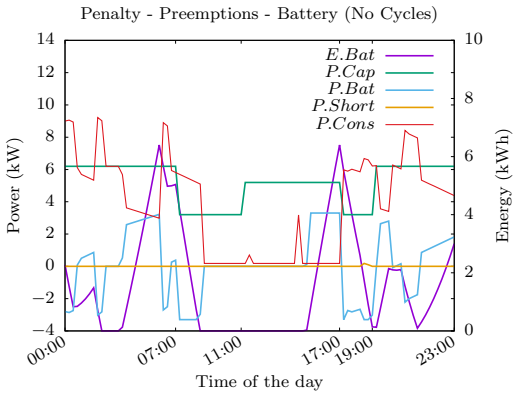

Figure 5: Scenario 5.

4.2.2. Impact of the battery cycles

We now consider the impact of the number of cycles allowed in the operation of the battery. The results in Table 8 indicate a positive impact of an increase in the number of battery cycles on the total cost, in the energy cost, and in the power consumption peak. More precisely, when the number of battery cycles changes from 1 or 2 cycles to an unlimited number of cycles, ($Scen.C.$ = 5 vs 6, 5 vs 7, 3 vs 9, and 3 vs 10), the reduction in $Peak.D.$, $T.Cost.D.$, and $E.Cost.D.$ is always positive and often large, as is the case for $Peak.D.$ and $T.Cost.D.$

We did not observe any significant change when the number of battery cycles changes from 4 to unlimited ($Scen.C. = 5$ vs 8 and 3 vs 11). For the given prices, power capacities and load demands, this indicates that 4 cycles are enough to guarantee an optimal utilization of the battery. In other words, allowing more than 4 cycles has no impact on the total cost. The behaviour of the state of the battery ($E.Bat$), as well as the evolution of the power charged or discharged from the battery ($P.Bat$) is presented in Figures 6 - 9.

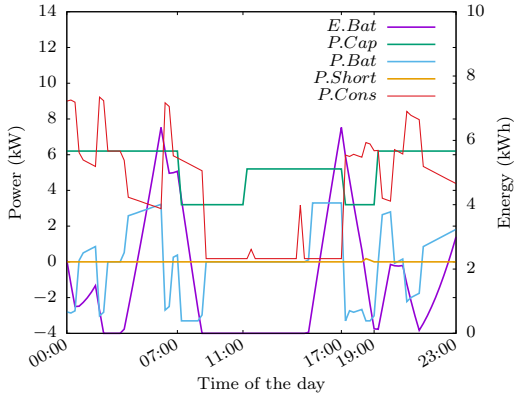

Figure 6: Scenario 5.

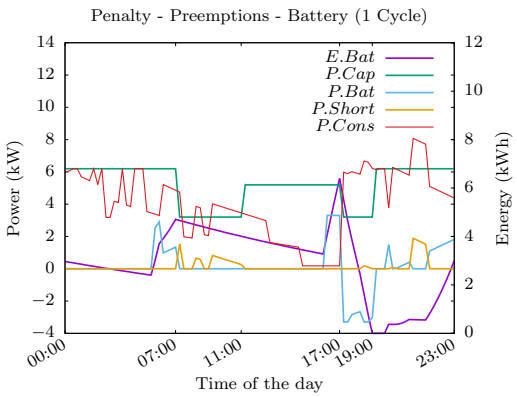

Figure 7: Scenario 6.

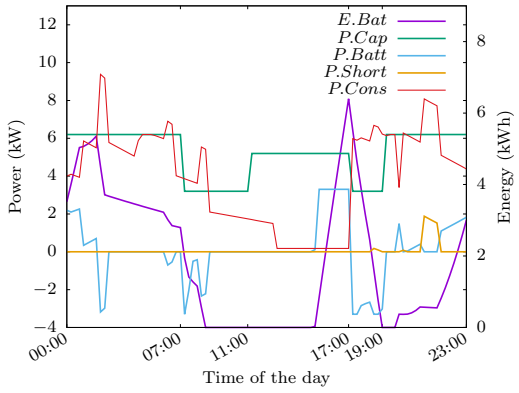

Figure 8: Scenario 7.

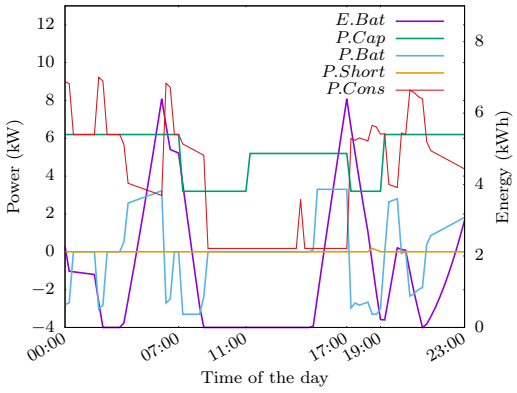

Figure 9: Scenario 8.

5. Concluding Remarks

In this paper we presented a flexible load scheduling MILP formulation for residential demand response on smart grids. The computational results show the positive impact on the reduction of total costs, as well as a decrease of peaks in power consumption, by allowing task preemptions, by using a battery, and by surcharging the excess of power. Future research avenues include the incorporation of local renewable energy generation (e.g., solar PV power systems, small wind turbines), the extension of the model to multiple users (e.g., a building or a residential complex), and the coordination between the load scheduling model with a real-time demand response model.

AppendixA. Recursion Tree to Generate Operational Patterns

We present an example on how to generate all possible operational patterns for a task containing four subtasks denoted as st_1, st_2, st_3, st_4, and three preemptions denoted as p_1, p_2, p_3. In this example, each subtask has a duration of 1 time period. Preemptions p_1, p_2, and p_3 have a minimum duration of zero time periods (i.e. it is possible not to have a preemption between subtasks). Preemptions p_1 and p_3 have a maximum duration of one time period, while preemption p_2 have a maximum duration of two time periods. The time window for operation of the task is equal to $[1, 7]$.

Figure A.10 presents the recursion tree to generate the sequences of preemptions. This tree has as many layers as the number of preemptions allowed in the task plus one initial layer containing the root node. The number of nodes at each layer of the tree is computed as the product of the number of nodes in the previous layer and the value of the difference between the maximum and the minimum duration of the corresponding preemption plus one. In Figure A.10, the number of nodes in the first layer is equal to $1 \times (1 - 0 + 1) = 2$, the number of nodes in the second layer is equal to $2 \times (3 - 0 + 1) = 6$, and the number of nodes in the third layer is equal to $6 \times (1 - 0 + 1) = 12$. Each node of the tree is labeled with the id of the preemption and with its corresponding length (e.g. node $(p_2, 1)$ represents the

second preemption with a length of 1 time period). Each path in the recursion tree corresponds to a different sequence of preemptions that might or might not be feasible depending on the operation time window of the task. For instance, the dashed path $(p_1, 1) - (p_2, 2) - (p_3, 1)$ is not feasible because when subtasks are included to generate $st_1 - p_1 - st_2 - p_2 - p_2 - st_3 - p_3 - st_4$, its total length (eight time periods) exceeds the number of time periods allowed in the time window (seven time periods).

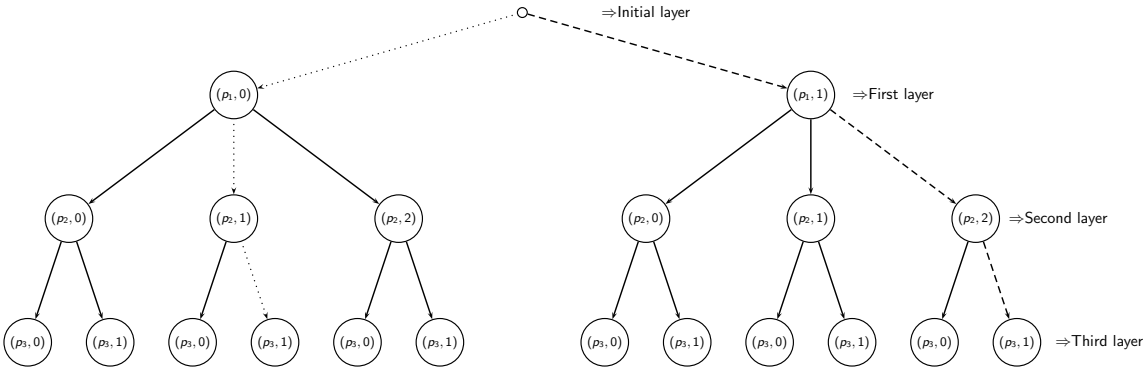

Figure A.10: Recursion tree.

The generation of the set potential operational patterns $\tilde{S}_j$ is done by including each subtask, with its corresponding duration, to the sequences of preemptions. For instance, the pattern $st_1 - st_2 - p_2 - st_3 - p_3 - st_4$ is obtained after inserting subtasks st_1, st_2, st_3 and st_4 in the dotted path $(p_1, 0) - (p_2, 1) - (p_3, 1)$. As mentioned in Section 3.1, for each time period t in time window $[1, 7]$, we verify if each potential operational pattern can be entirely contained in $[t, 7]$. If so, we add the pattern, with start time t, to the set of feasible operational patterns S_j. Otherwise the potential operational pattern with start time t is rejected. For instance, patterns with length four $(st_1 - st_2 - st_3 - st_4)$ can start at any time period from one to four. On the contrary, patterns with length seven $(st_1 - st_2 - p_2 - p_2 - st_3 - p_3 - st_4$, $st_1 - p_1 - st_2 - p_2 - st_3 - p_3 - st_4$ and $st_1 - p_1 - st_2 - p_2 - p_2 - st_3 - st_4)$ can only start in time period one.

AppendixB. Attributes of Appliances

Tables B.9 and B.10 present the characteristics and operation requirements of the appliances used in the computational experiments presented in Section 4. Table B.9 shows, for each task of type A, the type of load (*Load*), the start time of operation (*Start period*), the duration in hours (*Length Task (h)*) and the lower and upper bounds for the power consumption at each time period (*Power C. (kW)*).

Load	*Start period*	*Length Task (h)*	*Power C. (kW)*
Water heater Morning	0:00	5	(2.80,4)
	1:00	4.75	(2.72,4)
	2:00	4.5	(2.64,4)
	3:00	4.25	(2.56,4)
	4:00	4	(2.48,4)
Water heater Evening	17:00	0.75	(3.12,4)
Space heater Night	0:00	2.5	(1.64,2)
	1:00	3.5	(1.48,2)
	2:00	4.5	(1.32,2)
	3:00	5.5	(1.16,2)
Space heater Morning	6:00	2.5	(1.64,2)
	7:00	3.5	(1.48,2)
	8:00	4.5	(1.32,2)
	9:00	5.5	(1.16,2)
Space heater Evening	16:00	4.5	(1.32,2)
	17:00	5	(1.24,2)
	18:00	5	(1.24,2)
	20:00	4	(1.40,2)
Fridge	0:00	6	(0.18,0.18)

Table B.9: Type A task attributes.

Table B.10 shows, for each task of type B and one task of type C (Garage heating), the type of load (*Load*), the number of preemptions (*Pre.*), the earliest operation time

(EOT) and the latest operation time (LOT). In addition, Table B.10 presents for each subtask, its length (*Length SubT. (h)*), its power consumption at each time period (*Power C. (kW)*) and the length of the following preemption (*Length Pre. (h)*). Observe that, for the Garage heating load (type C task), there is not a fixed value for the power consumption at each time period (e.g. the power consumption can take any value from zero to three). Instead there is a value of six for the total power consumption θ.

Load	*Pre.*	*EOT*	*LOT*	*SubT.*	*Length SubT. (h)*	*Power C. (kW)*	*Length Pre. (h)*
Dishwasher	2	17:00	18:00	1	0.5	1.8	1
		17:00	18:00	2	0.5	0.9	1
		17:00	18:00	3	0.5	1.8	0
Washing Machine	1	17:00	18:00	1	0.25	2	1
		17:00	18:00	2	0.5	0.8	0
Dryer	0	20:00	21:00	1	1	2.5	0
TV	0	19:00	20:00	1	2	0.1	0
Oven	0	17:00	18:00	1	1	2	0
Garage heating	6	00:00	23:00	1 - 7	3	[0 - 3] , $\theta = 6$	4

Table B.10: Type B and type C task attributes.

References

[1] Survey of Household Energy Use. Technical report, Natural Resources Canada, 2011.

[2] 2014 Yearbook of Electricity Distributors. Technical report, Ontario Energy Board, 2015.

[3] Electric Power Annual 2015. Technical report, U.S Energy Information Administration, 2016.

[4] Assessment of Demand Response and Advanced Metering. Technical report, Federal Energy Regulatory Commission, Dic 2012.

[5] A. Ahmad, N. Javaid, U. Qasim, and Z. A. Khan. Demand response: From classification to optimization techniques in smart grid. In *Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on*, pages 229–235. IEEE, 2015.

[6] Miguel F. Anjos and Juan A. Gómez. *Operations Research Approaches for Building Demand Response in a Smart Grid*, chapter 7, pages 131–152.

[7] M. Beaudin and H. Zareipour. Home energy management systems: A review of modelling and complexity. *Renewable and Sustainable Energy Reviews*, 45: 318–335, 2015.

[8] M. Beaudin, H. Zareipour, A. K. Bejestani, and A. Schellenberg. Residential energy management using a two-horizon algorithm. *IEEE Transactions on Smart Grid*, 5(4):1712–1723, 2014.

[9] G. T. Costanzo, G. Zhu, M. F. Anjos, and G. Savard. A system architecture for autonomous demand side load management in smart buildings. *IEEE Transactions on Smart Grid*, 3(4):2157–2165, 2012.

[10] Juan A Gomez and Miguel F Anjos. Power capacity profile estimation for building heating and cooling in demand-side management. *Applied Energy*, 191: 492–501, 2017.

[11] Juan A. Gómez and Miguel F. Anjos. Power capacity profile estimation for activity-based residential loads. Technical Report G-2017-39, GERAD, May 2017.

[12] J. K. Gruber, F. Huerta, P. Matatagui, and M. Prodanović. Advanced building energy management based on a two-stage receding horizon optimization. *Applied Energy*, 160:194–205, 2015.

[13] C. Li, D. Srinivasan, and T. Reindl. Real-time scheduling of time-shiftable loads in smart grid with dynamic pricing and photovoltaic power generation. In *Smart Grid Technologies-Asia (ISGT ASIA), 2015 IEEE Innovative*, pages 1–6. IEEE, 2015.

[14] Petra Mesari and Slavko Krajcar. Home demand side management integrated with electric vehicles and renewable energy sources. *Energy and Buildings*, 108: 1–9, 2015. ISSN 03787788. URL `http://dx.doi.org/10.1016/j.enbuild.2015.09.001`.

[15] M. Rastegar, M. Fotuhi-Firuzabad, and H. Zareipour. Home energy management incorporating operational priority of appliances. *International Journal of Electrical Power & Energy Systems*, 74:286–292, 2016.

[16] H.-T. Roh and J.-W. Lee. Residential demand response scheduling with multiclass appliances in the smart grid. *IEEE Transactions on Smart Grid*, 7(1): 94–104, 2016.

[17] P. Scott, S. Thiébaux, M. Van Den Briel, and P. Van Hentenryck. Residential demand response under uncertainty. In *International Conference on Principles and Practice of Constraint Programming*, pages 645–660. Springer, 2013.

[18] J. S. Vardakas, N. Zorba, and C. V. Verikoukis. A survey on demand response programs in smart grids: pricing methods and optimization algorithms. *IEEE Communications Surveys & Tutorials*, 17(1):152–178, 2015.

[19] World Business Council for Sustainable Development. Transforming the market: Energy efficiency in the buildings. Technical report, WBCSD, June 2009.

[20] T. Yu, D. S. Kim, and S.-Y. Son. Optimization of scheduling for home appliances in conjunction with renewable and energy storage resources. *International Journal of Smart Home*, 7(4):261–272, 2013.

[21] Z. Zhu, J. Tang, S. Lambotharan, W. H. Chin, and Z. Fan. An integer linear programming based optimization for home demand-side management in smart grid. In *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–5. IEEE, 2012.