

Multi-stage Stochastic Programming for Demand Response Optimization

Munise Kübra Şahin, Özlem Çavuş, Hande Yaman

Department of Industrial Engineering, Bilkent University, 06800 Ankara, Turkey

kubra.sahin, ozlem.cavus, hyaman@bilkent.edu.tr

Abstract

The increase in the energy consumption puts pressure on natural resources and environment and results in a rise in the price of energy. This motivates residents to schedule their energy consumption through demand response mechanism. We propose a multi-stage stochastic programming model to schedule different kinds of electrical appliances under uncertain weather conditions and availability of renewable energy. We incorporate appliances with internal batteries to better utilize the renewable energy sources. Our aim is to minimize the electricity cost and the residents' dissatisfaction. We use a scenario groupwise decomposition (group subproblem) approach to compute lower and upper bounds for instances with a large number of scenarios. The results of our computational experiments show that the approach is very effective in finding high quality solutions in small computation times. We provide insights about how optimization and renewable energy combined with batteries for storage result in peak demand reduction, savings in electricity cost and more pleasant schedules for residents with different levels of price sensitivity.

Key words: smart grid, demand response, multi-stage stochastic programming, scenario groupwise decomposition

1 Introduction

World energy consumption is expected to grow by 48% between 2012 and 2040 [1]. The increase in the energy consumption results in a rise in the price of energy by putting pressure on natural resources and environment. Residential energy consumption grows with the population growth accompanied by fewer occupants living in larger houses, the change in people's life style and a higher rate of utilization of electrical appliances. Since some of the electrical appliances, like plug-in hybrid vehicles, have a potential to double residential energy consumption, demand side management becomes more crucial. Due to the increase in the energy consumption and the price of electricity, residents are motivated to schedule their energy consumption to reduce their cost of electricity and their negative impact on the environment. Managing the energy consumption by providing information to residents who can change their consumption patterns with respect to the given information is called "demand response" and is one of the demand side management approaches.

Smart grids can play a significant role to provide a sustainable future by integrating residents into the energy saving system through demand response mechanism. With an installation of home energy management system (HEMS) including demand response program, residents can participate in the smart grid system. They can monitor the price information through the smart meters and control their own appliances. By the end of 2016, 47% of all residential consumers in U.S. have a smart meter in their homes [2]. Since smart meters provide information about the fluctuation of electricity price between low-demand and high-demand hours, with the help of HEMS, residents can shift their demands to the off-peak hours or hours with high renewable energy levels. Thus, the main objective of HEMS is to reduce the electricity bill of the resident both by scheduling the electrical appliances and increasing the use of renewable energy. To increase the benefits of smart meters, the EU member countries decided to provide smart meters to at least 72% of all residential consumers by 2020 [3]. The increase in the number of residents with smart meters demonstrates the importance of demand response management and the consequent need to develop a method for efficient energy consumption scheduling.

There are two main approaches to schedule smart home appliances. While one approach is to minimize the

electricity cost [4] - [7], the most popular approach is to use the concept of disutility function to obtain a desired trade-off between minimizing the electricity cost and minimizing the residents' dissatisfaction [8] - [15].

The existing studies vary in the types of electrical appliances they consider. Based on the energy consumption characteristics, the appliances can be classified under two major types: appliances with continuous level energy consumption and appliances with discrete level energy consumption. While the energy consumption amount of the appliances with continuous level energy consumption can vary during the operation, this amount is fixed for the appliances with discrete level energy consumption. The studies [4] - [6], [9] and [12] consider the coordination of appliances with continuous level energy consumption. Scheduling of appliances with discrete level energy consumption is studied in [11] and [14]. [7], [10], [13] and [15] consider the scheduling of appliances both with continuous and discrete level energy consumption. The energy consumption of appliances that control the temperature of the environment comprised 40% of total residential energy consumption [16]. This points out the need of scheduling this type of appliances as done by [7], [8], [12] and [13]. However, these studies ignore the randomness in weather conditions and model the problem in a deterministic setting.

Although one of the main purposes of HEMS is to better utilize the renewable energy, to the best of our knowledge, the only study that considers renewable energy is [13]. Renewable energy must be consumed at the moment it is produced unless it is stored. Appliances with internal batteries can be used to increase the use of renewable energy, as done in [13]. Further, they can be used to reduce the peak load as done in [12].

In this study, we propose a multi-stage stochastic nonlinear mixed integer programming model that coordinates all types of appliances mentioned above in the presence of weather uncertainty. In particular, our major contributions are the following:

- We propose a novel model to schedule all types of electrical appliances considered in the literature in the presence of renewable energy and batteries.
- We use multi-stage stochastic programming to hedge against the uncertainty in weather conditions, which is critical to model accurately both the energy consumption of the appliances that control the temperature

and the availability of the renewable energy.

- Our model is a multi-stage stochastic nonlinear mixed integer programming model. We use a scenario groupwise decomposition (group subproblem) approach to obtain tight lower and upper bounds for instances with large number of scenarios.
- Our computational experiments provide valuable insights on the benefits of optimization in demand response and the savings due to renewable energy and battery for residents with different price sensitivity.

The rest of the paper is organized as follows. In Section 2, we describe the energy consumption characteristics and disutility functions for different appliance types and the electricity pricing scheme. In Section 3, we present our model. In Section 4, we describe the scenario groupwise decomposition approach and present two different ways to construct group subproblems. We present the results of our computational experiments in Section 5. We conclude in Section 6.

2 Problem Setting

We consider a smart home equipped with electrical appliances that are networked together and controlled by HEMS. The communication of the price information between the energy provider and the resident is provided by the smart meter.

We consider a discrete-time model with a finite horizon, where scheduling horizon is divided into time slots of equal duration. The energy consumption scheduling problem aims to achieve a trade-off between minimizing the electricity cost and minimizing the residents' dissatisfaction due to loss of comfort.

2.1 Types of Appliances

Let \mathcal{A} denote the set of appliances networked in this residential unit. This set includes appliances with both continuous and discrete level energy consumption. An appliance a with discrete level energy consumption only

operates in on or off state and it consumes a fixed energy level of \bar{e}_a and \underline{e}_a in on and off modes, respectively. Consequently, HEMS decides only when this appliance should be in on mode. On the other hand, for an appliance with continuous level energy consumption, HEMS needs to decide how much energy it consumes in each time period. Each appliance a operates within a user's preferred time interval $T_a = [\underline{t}_a, \bar{t}_a]$.

We classify the appliances with continuous level energy consumption under three categories as follows:

- 1) *Type 1* : This type of appliances are equipped with internal batteries. Since they have an ability to store or dispatch energy, they better utilize the renewable energy sources. We denote the set of appliances with battery by \mathcal{A}_B .
- 2) *Type 2* : This type of appliances control the temperature of the environment. An example is air conditioner. We denote the set of appliances that control the temperature of environment by \mathcal{A}_T .
- 3) *Type 3* : This type includes the appliances with continuous level energy consumption that are not of *Type 1* and *Type 2*. Let \mathcal{A}_C denote the set of appliances of *Type 3*. An example of this type is refrigerator.

We classify the appliances with discrete level energy consumption as follows:

- 1) *Type 4* : This type of appliances can be shut down during operation. This means that their loads are interruptible. An example is a hair dryer. We denote the set of appliances with interruptible load by \mathcal{A}_I .
- 2) *Type 5* : For this type of appliances, once the operation starts, it should be run to completion. An example is a washing machine. We denote the set of appliances with uninterruptible load by \mathcal{A}_U .

2.2 Disutility Functions

The disutility can have different causes depending on the types of appliances. For an appliance that controls the temperature, the disutility is a result of the discomfort due to low or high inside temperature, whereas for an appliance with discrete consumption, the disutility is a function of earliness of the starting time or the lateness of the finishing time.

The disutility functions for different types of appliances are as follows:

- **Appliances with discrete level energy consumption (types 4 and 5):** For each appliance $a \in \mathcal{A}_{\mathcal{I}} \cup \mathcal{A}_{\mathcal{U}}$ with discrete level energy consumption, the important decisions are starting and finishing times. The disutility for appliance a if it starts operating at time \hat{t}_a and finishes at time \bar{t}_a is given by $\gamma_a(\hat{t}_a, \bar{t}_a)$, which is computed as:

$$\gamma_a(\hat{t}_a, \bar{t}_a) = \begin{cases} (\hat{t}_a - t_a^{des})^2 & , \text{ if } \hat{t}_a < t_a^{des} \\ 0 & , \text{ if } t_a^{des} \leq \hat{t}_a \text{ \& } \bar{t}_a \leq \bar{t}_a^{des} \\ (\bar{t}_a - \bar{t}_a^{des})^2 & , \text{ if } \bar{t}_a > \bar{t}_a^{des} \end{cases}$$

where t_a^{des} and \bar{t}_a^{des} , respectively, are the most desirable starting and finishing times. The function $\gamma_a(\hat{t}_a, \bar{t}_a)$ penalizes starting before or ending after the desired times.

- **Appliances that control temperature (type 2):** Let h_a^{comf} be the most comfortable temperature and $h_{a,t}^{in}$ be the inside temperature at the location of appliance $a \in \mathcal{A}_{\mathcal{T}}$ in time period $t \in T_a$ and $\tau_{a,t}(h_{a,t}^{in})$ be the disutility at time t for appliance a . We use the disutility function used in [12] and [13] (We ignore the constant term since it does not affect the optimal solution.):

$$\tau_{a,t}(h_{a,t}^{in}) = (h_{a,t}^{in} - h_a^{comf})^2.$$

- **Appliances with battery (type 1):** As in [12] and [13], the disutility $\psi_{a,t}(r_{a,t}, b_{a,t})$ for appliance $a \in \mathcal{A}_{\mathcal{B}}$ with battery depends on $r_{a,t}$, which is the power charged to (when $r_{a,t} \geq 0$) or discharged from (when $r_{a,t} \leq 0$) the appliance in period $t \in T_a$ and $b_{a,t}$, which is the energy level of the appliance a in period t . The disutility is computed as

$$\psi_{a,t}(r_{a,t}, b_{a,t}) = \eta_1 (r_{a,t})^2 - \eta_2 r_{a,t} r_{a,t+1} + \eta_3 (\max\{\delta \bar{b}_a - b_{a,t}, 0\})^2,$$

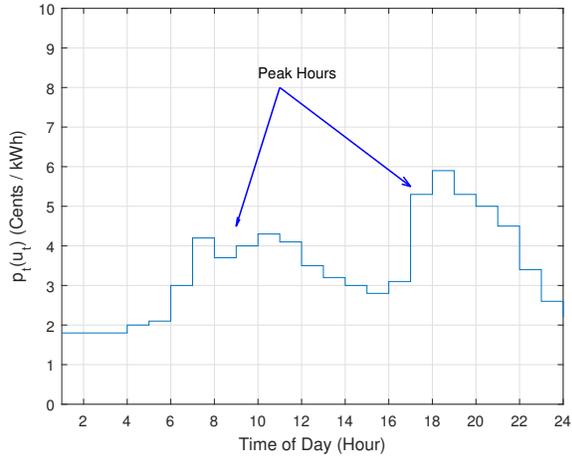
where η_1, η_2, η_3 , and δ are positive constants and \bar{b}_a is the capacity of battery for appliance a . The first term penalizes the damaging effect of charging and discharging, the second term penalizes charging-discharging cycles and the third term penalizes deep discharging, which happens when the energy level of the battery drops below $\delta\bar{b}_a$. The value of δ may vary according to the type of battery.

2.3 Electricity Price

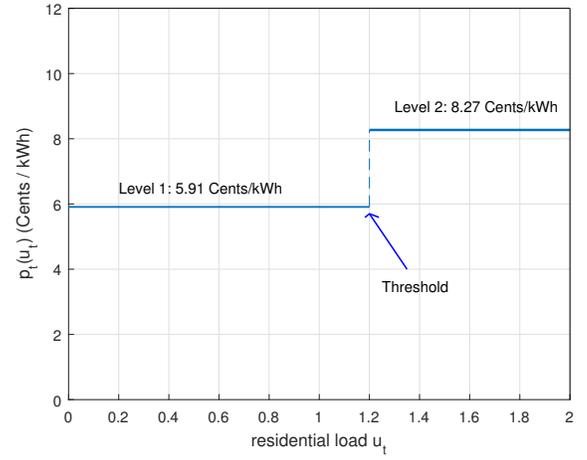
Since electricity is perishable, the prices set by the electricity retailers fluctuate between low-demand hours and high-demand hours. The most common pricing schemes are *real-time pricing* (RTP), *day-ahead pricing* (DAP), *time-of-use pricing* (TOUP), *critical-peak pricing* (CPP) and *inclining block rates* (IBR). ([10]; see for further references). In our study, as in [10] and [14], we consider a general hourly pricing function that combines RTP and IBR schemes. It also has a DAP structure as we assume that the future price parameters are known by the residents ahead of time. In our price function, prices vary every hour as in RTP, and beyond a certain threshold for total hourly residential load, the electricity price increases to a higher value as in IBR. The cost $\rho_t(u_t)$ of consuming u_t units of electricity in period t depends on the low cost c_t^l , the high cost c_t^h and the threshold \bar{u} as follows:

$$\rho_t(u_t) = \begin{cases} c_t^l & , \text{ if } 0 \leq u_t \leq \bar{u} \\ c_t^h & , \text{ if } u_t > \bar{u}. \end{cases}$$

In numerical studies, price data of Illinois Power Company and British Columbia (BC) Hydro Company are used (Figure 1).



(a) Real-time prices set by Illinois Power Company



(b) Two-level inclining block rates set by BC Hydro

Figure 1: Examples of two-non flat pricing models. The real time prices are used by Illinois Power Company on December 15 [17]. The inclining block rates are used by BC Hydro Company in December [18]. (This figure is reproduced from [10].)

3 Multi-stage Stochastic Programming Model

In this section, we formulate the scheduling problem for smart home appliances as a multi-stage stochastic nonlinear mixed integer program. We assume that the changes in the temperature and in the power of renewable energy are uncertain and this uncertainty is realized gradually at specific time periods called as stages. We also assume that the random temperature and random renewable energy in each stage have discrete distributions with finite number of realizations. Therefore, the uncertainty in the decision process can be represented by a scenario tree. A scenario is defined as a unique path from the root node to a terminal node. Consequently, the number of scenarios is equal to the number of terminal nodes.

We denote the set of all scenarios by \mathcal{S} and the set of scenarios having the same history as scenario s at stage t by $\mathcal{S}_{s,t}$. The probability of scenario s is represented by $p(s)$ and the scheduling horizon is represented by T . We use the following notation:

Decision Variables

- $u_t^l(s)$: the net energy consumption at low price in period $t \in T$ under scenario $s \in \mathcal{S}$
- $u_t^h(s)$: the net energy consumption at high price in period $t \in T$ under scenario $s \in \mathcal{S}$
- $b_{a,t}(s)$: the energy level of appliance $a \in \mathcal{A}_B$ in period $t \in T_a$ under scenario $s \in \mathcal{S}$
- $r_{a,t}(s)$: the power charged/discharged for appliance $a \in \mathcal{A}_B$ in period $t \in T_a$ under scenario $s \in \mathcal{S}$
- $e_{a,t}(s)$: the energy consumption of appliance $a \in \mathcal{A} \setminus \mathcal{A}_B$ in period $t \in T_a$ under scenario $s \in \mathcal{S}$
- $e_{a,t}^+(s)$: the energy consumption for heating for appliance $a \in \mathcal{A}_T$ in period $t \in T_a$ under scenario $s \in \mathcal{S}$
- $e_{a,t}^-(s)$: the energy consumption for cooling for appliance $a \in \mathcal{A}_T$ in period $t \in T_a$ under scenario $s \in \mathcal{S}$
- $x_{a,t}(s) : \begin{cases} 1, & \text{if appliance } a \in \mathcal{A}_U \cup \mathcal{A}_I \text{ starts operation in period } t \in T_a \text{ under scenario } s \in \mathcal{S} \\ 0, & \text{otherwise} \end{cases}$
- $y_{a,t}(s) : \begin{cases} 1, & \text{if appliance } a \in \mathcal{A}_I \text{ is in on statue in period } t \in T_a \text{ under scenario } s \in \mathcal{S} \\ 0, & \text{otherwise} \end{cases}$
- $z_{a,t}(s) : \begin{cases} 1, & \text{if appliance } a \in \mathcal{A}_I \text{ completes operation in period } t \in T_a \text{ under scenario } s \in \mathcal{S} \\ 0, & \text{otherwise} \end{cases}$
- $w_t(s) : \begin{cases} 1, & \text{if energy usage is charged at low price in period } t \in T \text{ under scenario } s \in \mathcal{S} \\ 0, & \text{otherwise} \end{cases}$
- $d_{a,t}(s) : (\delta \bar{b}_a - b_{a,t})^+$ deep discharging amount for appliance $a \in \mathcal{A}_B$ in period $t \in T_a$ under scenario $s \in \mathcal{S}$
- $h_{a,t}^{in}(s)$: the temperature inside the place of appliance $a \in \mathcal{A}_T$ in period $t \in T_a$ under scenario $s \in \mathcal{S}$

Parameters

- e_a^{tot} : the total energy required for appliance $a \in \mathcal{A}_C$
- $b_{a,0}$: the initial energy level of the battery for appliance $a \in \mathcal{A}_B$
- \bar{r}_a : the maximum charging amount for appliance $a \in \mathcal{A}_B$
- \underline{r}_a : the maximum discharging amount for appliance $a \in \mathcal{A}_B$
- \underline{e}_a : the minimum energy level required for appliance $a \in \mathcal{A} \setminus \mathcal{A}_B$
- \bar{e}_a : the maximum energy level required for appliance $a \in \mathcal{A} \setminus \mathcal{A}_B$
- \bar{b}_a : the battery capacity for appliance $a \in \mathcal{A}_B$
- \underline{h}_a : the minimum comfortable temperature level for appliance $a \in \mathcal{A}_T$
- \bar{h}_a : the maximum comfortable temperature level for appliance $a \in \mathcal{A}_T$
- h_a^{comf} : the most comfortable temperature level for appliance $a \in \mathcal{A}_T$
- $v_t(s)$: the power of renewable energy source in period $t \in T$ under scenario $s \in \mathcal{S}$
- n_a : the total number of time slots required for appliance $a \in \mathcal{A}_I \cup \mathcal{A}_U$ to complete its task
- $h_{a,t}^{out}(s)$: the temperature outside the location of appliance $a \in \mathcal{A}_T$ in period $t \in T_a$ under scenario $s \in \mathcal{S}$
- \underline{t}_a^{des} : the most desirable starting time for appliance $a \in \mathcal{A}_I \cup \mathcal{A}_U$
- \bar{t}_a^{des} : the most desirable finishing time for appliance $a \in \mathcal{A}_I \cup \mathcal{A}_U$
- $h_{a,0}^{in}(s)$: the initial room temperature for appliance $a \in \mathcal{A}_T$ under scenario $s \in \mathcal{S}$

- α_a, β_a : the thermal characteristics of appliance $a \in \mathcal{A}_{\mathcal{T}}$ and the environment in which appliance operates

($\beta_a > 0$ if appliance is a heater and $\beta_a < 0$ if appliance is a cooler)

- ϕ : the weight of disutility function
- ϕ_D : the weight of disutility function for appliances in $\mathcal{A}_{\mathcal{I}} \cup \mathcal{A}_{\mathcal{U}}$
- ϕ_T : the weight of disutility function for appliances in $\mathcal{A}_{\mathcal{T}}$
- ϕ_B : the weight of disutility function for appliances in $\mathcal{A}_{\mathcal{B}}$

An efficient energy consumption schedule can be obtained by solving the following model:

$$\begin{aligned}
\min \sum_{s \in \mathcal{S}} p(s) & \left(\sum_{t \in T} \left(c_t^l u_t^l(s) + c_t^h u_t^h(s) \right) + \phi \left(\phi_D \sum_{a \in \mathcal{A}_{\mathcal{I}} \cup \mathcal{A}_{\mathcal{U}}} \left(\sum_{t=\underline{t}_a}^{\bar{t}_a^{des}} (t - \underline{t}_a^{des})^2 x_{a,t}(s) + \sum_{t=\bar{t}_a^{des}}^{\bar{t}_a} (t - \bar{t}_a^{des})^2 z_{a,t}(s) \right) \right. \right. \\
& \left. \left. + \phi_T \sum_{a \in \mathcal{A}_{\mathcal{T}}} \sum_{t \in T_a} \left(h_{a,t}^{in}(s) - h_a^{comf} \right)^2 + \phi_B \sum_{a \in \mathcal{A}_{\mathcal{B}}} \sum_{t \in T_a} \left(\eta_1 (r_{a,t}(s))^2 - \eta_2 r_{a,t}(s) r_{a,t+1}(s) + \eta_3 (d_{a,t}(s))^2 \right) \right) \right)
\end{aligned} \tag{1}$$

s.t.

$$\underline{e}_a \leq e_{a,t}(s) \leq \bar{e}_a, \quad \forall a \in \mathcal{A}_{\mathcal{C}}, t \in T_a, s \in \mathcal{S} \tag{2}$$

$$\sum_{t \in T_a} e_{a,t}(s) = e_a^{tot}, \quad \forall a \in \mathcal{A}_{\mathcal{C}}, s \in \mathcal{S} \tag{3}$$

$$\underline{e}_a \leq e_{a,t}^+(s) \leq \bar{e}_a, \quad \forall a \in \mathcal{A}_{\mathcal{T}}, t \in T_a, s \in \mathcal{S} \tag{4}$$

$$\underline{e}_a \leq e_{a,t}^-(s) \leq \bar{e}_a, \quad \forall a \in \mathcal{A}_{\mathcal{T}}, t \in T_a, s \in \mathcal{S} \quad (5)$$

$$e_{a,t}(s) = e_{a,t}^-(s) + e_{a,t}^+(s), \quad \forall a \in \mathcal{A}_{\mathcal{T}}, t \in T_a, s \in \mathcal{S} \quad (6)$$

$$h_{a,t}^{in}(s) = h_{a,t-1}^{in}(s) + \alpha(h_{a,t}^{out}(s) - h_{a,t-1}^{in}(s)) + \beta(e_{a,t}^+(s) - e_{a,t}^-(s)), \quad \forall a \in \mathcal{A}_{\mathcal{T}}, t \in T_a, s \in \mathcal{S} \quad (7)$$

$$\underline{h}_a \leq h_{a,t}^{in}(s) \leq \bar{h}_a, \quad \forall a \in \mathcal{A}_{\mathcal{T}}, t \in T_a, s \in \mathcal{S} \quad (8)$$

$$b_{a,t}(s) = \sum_{u=1}^t r_{a,u}(s) + b_{a,0}, \quad \forall a \in \mathcal{A}_{\mathcal{B}}, t \in T_a, s \in \mathcal{S} \quad (9)$$

$$0 \leq b_{a,t}(s) \leq \bar{b}_a, \quad \forall a \in \mathcal{A}_{\mathcal{B}}, t \in T_a, s \in \mathcal{S} \quad (10)$$

$$\underline{r}_a \leq r_{a,t}(s) \leq \bar{r}_a, \quad \forall a \in \mathcal{A}_{\mathcal{B}}, t \in T_a, s \in \mathcal{S} \quad (11)$$

$$d_{a,t}(s) \geq \delta \bar{b}_a - b_{a,t}(s), \quad \forall a \in \mathcal{A}_{\mathcal{B}}, t \in T_a, s \in \mathcal{S} \quad (12)$$

$$d_{a,t}(s) \geq 0, \quad \forall a \in \mathcal{A}_{\mathcal{B}}, t \in T_a, s \in \mathcal{S} \quad (13)$$

$$r_{a,\bar{t}_a+1}(s) = 0, \quad \forall a \in \mathcal{A}_{\mathcal{B}}, s \in \mathcal{S} \quad (14)$$

$$e_{a,t}(s) = \underline{e}_a + (\bar{e}_a - \underline{e}_a)y_{a,t}(s), \quad \forall a \in \mathcal{A}_{\mathcal{I}}, t \in T_a, s \in \mathcal{S} \quad (15)$$

$$\sum_{t \in T_a} y_{a,t}(s) = n_a, \quad \forall a \in \mathcal{A}_I, s \in \mathcal{S} \quad (16)$$

$$\sum_{u=t+1}^{\bar{t}_a - n_a + 1} x_{a,u}(s) + y_{a,t}(s) \leq 1, \quad \forall a \in \mathcal{A}_I, t \in [\underline{t}_a, \bar{t}_a - n_a], s \in \mathcal{S} \quad (17)$$

$$\sum_{u=\underline{t}_a + n_a - 1}^{t-1} z_{a,u}(s) + y_{a,t}(s) \leq 1, \quad \forall a \in \mathcal{A}_I, t \in [\underline{t}_a + n_a, \bar{t}_a], s \in \mathcal{S} \quad (18)$$

$$\sum_{t=\underline{t}_a + n_a - 1}^{\bar{t}_a} z_{a,t}(s) = 1, \quad \forall a \in \mathcal{A}_I, s \in \mathcal{S} \quad (19)$$

$$\sum_{t=\underline{t}_a}^{\bar{t}_a - n_a + 1} x_{a,t}(s) = 1, \quad \forall a \in \mathcal{A}_I \cup \mathcal{A}_U, s \in \mathcal{S} \quad (20)$$

$$e_{a,t}(s) = \underline{e}_a + (\bar{e}_a - \underline{e}_a) \left(\sum_{u=\max\{\underline{t}_a, t-n_a+1\}}^t x_{a,u}(s) \right) \quad \forall a \in \mathcal{A}_U, t \in T_a, s \in \mathcal{S} \quad (21)$$

$$u_t^l(s) + u_t^h(s) \geq \sum_{a \in \mathcal{A} \setminus \mathcal{A}_B} e_{a,t}(s) + \sum_{a \in \mathcal{A}_T} e_{a,t}^-(s) + \sum_{a \in \mathcal{A}_B} r_{a,t}(s) - v_t(s), \quad \forall t \in T, s \in \mathcal{S} \quad (22)$$

$$0 \leq u_t^l(s) \leq \bar{u} w_t(s), \quad \forall t \in T, s \in \mathcal{S} \quad (23)$$

$$0 \leq u_t^h(s) \leq M(1 - w_t(s)), \quad \forall t \in T, s \in \mathcal{S} \quad (24)$$

$$e_{a,t}(s') = e_{a,t}(s), \quad \forall a \in \mathcal{A} \setminus \mathcal{A}_B, s \in \mathcal{S}, s' \in \mathcal{S}_{s,t}, t \in T_a \quad (25)$$

$$e_{a,t}^+(s') = e_{a,t}^+(s), \quad \forall a \in \mathcal{A}_{\mathcal{T}}, s \in \mathcal{S}, s' \in \mathcal{S}_{s,t}, t \in T_a \quad (26)$$

$$e_{a,t}^-(s') = e_{a,t}^-(s), \quad \forall a \in \mathcal{A}_{\mathcal{T}}, s \in \mathcal{S}, s' \in \mathcal{S}_{s,t}, t \in T_a \quad (27)$$

$$r_{a,t}(s') = r_{a,t}(s), \quad \forall a \in \mathcal{A}_{\mathcal{B}}, s \in \mathcal{S}, s' \in \mathcal{S}_{s,t}, t \in T_a \quad (28)$$

$$x_{a,t}(s) \in \{0, 1\}, \quad \forall a \in \mathcal{A}_{\mathcal{I}} \cup \mathcal{A}_{\mathcal{U}}, t \in T_a, s \in \mathcal{S} \quad (29)$$

$$y_{a,t}(s) \in \{0, 1\}, \quad \forall a \in \mathcal{A}_{\mathcal{I}}, t \in T_a, s \in \mathcal{S} \quad (30)$$

$$z_{a,t}(s) \in \{0, 1\}, \quad \forall a \in \mathcal{A}_{\mathcal{I}}, t \in T_a, s \in \mathcal{S} \quad (31)$$

$$w_t(s) \in \{0, 1\} \quad \forall t \in T, s \in \mathcal{S} \quad (32)$$

The objective function (1) is the sum of expected electricity cost and expected dissatisfaction.

For appliances of *Type 3* (appliances with continuous consumption), constraints (2) and (3) ensure that energy consumption is between minimum standby power level and maximum power level and that the total energy requirement is provided within the specified time interval, respectively.

Constraints (4) - (8) are given for appliances of *Type 2* (appliances that control temperature). As above, constraints (4) - (6) ensure that energy consumption is between minimum and maximum levels. Constraints (7) are the balance equations for inside temperature and energy consumption. Constraints (8) ensure that the temperature is in the range that user defines as comfortable.

Constraints (9) - (14) are for the appliances of *Type 1* (appliances with battery). Constraints (9) compute

the battery energy level at each period. Constraints (10) ensure that the total charge stored in the battery does not exceed its capacity. Constraints (11) bound the battery charging/discharging amount. Constraints (12) and (13) compute the amount of deep discharging. Constraints (14) ensure that disutility does not occur outside the specified time interval.

Constraints (15) - (20) are for appliances of *Type 4* (interruptible appliances with discrete consumption). Constraints (15) compute the energy consumption: if the appliance is off, the energy consumption is equal to the minimum standby power level and if it is on, the energy consumption amount equals the maximum power level. Constraints (16) ensure that the appliance is on for the number of periods required to fully complete its task. Constraints (17) and (18) allow operation of an appliance only between its starting and finishing time. Constraints (19) and (20) ensure that there is one starting and one finishing time for each appliance.

Constraints (20) and (21) are for appliances of *Type 5* (uninterruptible appliances with discrete consumption). One starting time is chosen for each appliance due to constraints (20). Once an appliance starts, it operates consecutively for the required number of periods. Constraints (21) compute the energy consumption amount.

Constraints (22) - (24) compute the energy consumption amount at low and high prices. Constraints (22) give the relation between the net energy request, energy consumption amount, and renewable energy amount. Constraints (23) and (24) decide whether the energy usage is charged at low price with regard to threshold value \bar{u} .

Constraints (25) and (28) are non-anticipativity constraints to ensure that for any stage t , decision variables that have a common history of uncertainty till stage t must have the same value at that stage.

Finally, constraints (29) - (32) are variable restrictions.

4 Solution Method

Multi-stage stochastic programming problems are in the framework of extremely challenging problems since their size grows exponentially with the number of stages. Our problem has additional difficulties due to nonlinear

objective function and integer variables. In the literature, some exact solution techniques exist for the multi-stage stochastic programming problems such as stochastic dual dynamic programming (SDDP) [19] and Lagrangian relaxation of non-anticipativity constraints [20]. Since both techniques require convexity to obtain an exact solution, they are not applicable for the problems with integer variables. However, a lower bound can be obtained for multi-stage stochastic integer problems using these methods. An extension of SDDP method proposed by [21] makes the method applicable for problems with integer variables by enforcing the integrality requirements only in the forward steps and relaxing in the backward steps. If the state variables are binary and the cuts satisfy some sufficient conditions, [22] proves that SDDP can provide an exact solution for the mixed integer multi-stage stochastic problems. However, even so, SDDP cannot be used in our problem since it also requires the assumption of stagewise independency in the stochastic process. [23] demonstrates that Lagrangian relaxation of non-anticipativity constraints within a branch and bound procedure can be used to obtain an exact solution for multi-stage stochastic integer problems. However, since complete recourse assumption is a requirement for the branch and bound procedure, this method is also not applicable for our problem. Besides these techniques, scenario groupwise decomposition approach which is recently proposed by [24] can be used to obtain bounds for multi-stage stochastic integer problems. In this approach, the problem is divided into smaller problems called as group subproblems. Group subproblems are defined over a reduced number of scenarios and each of them has the same number of stages as the original problem. The problem is solved for each group subproblem separately rather than being solved for all scenarios simultaneously. Besides easy implementation, since it does not rely on restrictive assumptions such as stagewise independency, complete recourse and convexity, scenario groupwise decomposition is applicable for a wide range of problems including our problem. Group subproblem approach is used in the literature to calculate lower and upper bounds for multistage stochastic programs [25] and [26].

4.1 A Lower Bound

In this section, we provide a lower bound on the optimal value of our problem using the scenario groupwise decomposition approach of [24]. In this method, a lower bound is obtained by solving the problem separately for subsets of the scenarios called as groups. We define the set of groups as a partition of \mathcal{S} . Note that \mathcal{S} denotes the set of all scenarios. Let $\hat{\mathcal{S}}$ be the set of groups and $I_{\hat{\mathcal{S}}}$ be the set of indices of groups in $\hat{\mathcal{S}}$. A subset $\hat{\mathcal{S}}_i \subseteq \mathcal{S}$ is called as a group for all $i \in I_{\hat{\mathcal{S}}}$ and the collection of these groups $\hat{\mathcal{S}} = \{\hat{\mathcal{S}}_i\}_{i \in I_{\hat{\mathcal{S}}}}$ is called as a scenario partition if it satisfies $\cup_{i \in I_{\hat{\mathcal{S}}}} \hat{\mathcal{S}}_i = \mathcal{S}$ and $\hat{\mathcal{S}}_i \cap \hat{\mathcal{S}}_j = \emptyset$ for all $\{i, j\} \in I_{\hat{\mathcal{S}}}$ such that $i \neq j$. Note that groups do not have to be disjoint to obtain valid bounds, as mentioned in [24]. However, for ease of representation, we consider disjoint groups. The problem defined over a group is called as group subproblem.

After a partition is determined, the probability of each scenario $s \in \mathcal{S}$ is adjusted to achieve a lower bound. The probability of each group $\hat{\mathcal{S}}_i$ is the summation of the probabilities of scenarios in that group, that is, $p(\hat{\mathcal{S}}_i) = \sum_{s \in \hat{\mathcal{S}}_i} p(s)$ for all $i \in I_{\hat{\mathcal{S}}}$. The adjusted probability of each scenario s is $\hat{p}(s) = p(s)/p(\hat{\mathcal{S}}_i)$ for all $s \in \hat{\mathcal{S}}_i$ and $i \in I_{\hat{\mathcal{S}}}$, which is the ratio of the probability of the scenario to the probability of its group. Once the adjusted probabilities are calculated, the weighted summation of the optimal values of the group subproblems provides a lower bound for our minimization problem as proved in [24]. Let $z^*(\mathcal{S})$ be the optimal value of our problem with scenario set \mathcal{S} and $\underline{z}(\hat{\mathcal{S}})$ be a lower bound on $z^*(\mathcal{S})$ obtained using scenario partition $\hat{\mathcal{S}}$. The following inequality is satisfied by all $\hat{\mathcal{S}}$ selections as proved in Proposition 4.4 in [24]:

$$\underline{z}(\hat{\mathcal{S}}) = \sum_{i \in I_{\hat{\mathcal{S}}}} p(\hat{\mathcal{S}}_i) \cdot z^*(\hat{\mathcal{S}}_i) \leq z^*(\mathcal{S}). \quad (33)$$

The quality of the lower bound depends on the selection and size of groups as demonstrated in [24]. To see the effect of different partition selections, we construct the group subproblems with size $|\hat{\mathcal{S}}_i| = |\mathcal{S}|/|I_{\hat{\mathcal{S}}}|$ for all $i \in I_{\hat{\mathcal{S}}}$ with following two partition methods:

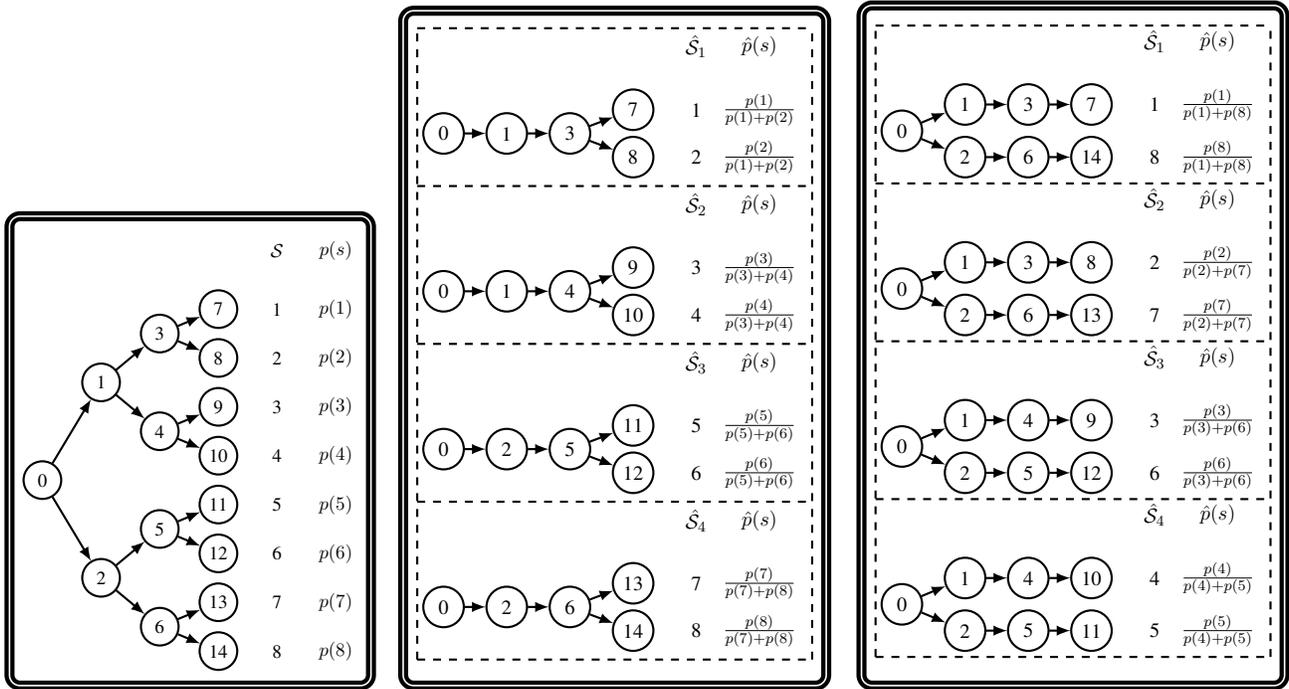
1) *Consecutive*: We select $|S|/|I_{\hat{S}}|$ consecutive scenarios using the order that they appear in the scenario tree.

This collection constructs the first group subproblem. We repeat this procedure for the rest of the scenarios till all of them are selected.

2) *Half-and-half*: We select $|S|/|I_{\hat{S}}|$ scenarios half of which is taken from the leaf nodes with the smallest indices and half is taken from the largest indices. This collection constructs the first group subproblem.

We repeat this procedure for the rest of the scenarios till all of them are selected.

Since the *consecutive* method constructs the groups with scenarios having more common nodes, it is expected that *consecutive* method has an advantage in computation time over the *half-and-half* method per group. On the other hand, grouping scenarios with less common nodes, as in the *half-and-half* method, shortens the computation time to compute an upper bound as will be explained in detail in Section 4.2.



(a) The original problem

(b) Group subproblems of the *consecutive* method

(c) Group subproblems of the *half-and-half* method

Figure 2: Group subproblems of the *consecutive* and *half-and-half* methods for groups of two scenarios for a four stage problem

These partition strategies are explained in Figure 2 over a four stage scenario tree with two branches in each

stage. The scenario tree is provided in (a). The construction of groups of size two and the adjusted probabilities of each scenario under this construction are demonstrated in (b) and (c) for the *consecutive* and *half-and-half* strategies, respectively.

4.2 An Upper Bound

We use an optimal solution of each group subproblem to generate a feasible solution to the original problem as in [24]. Let $x^*(S, t)$ be the restriction of an optimal solution of our problem with scenario set S to the first t stages. For a given scenario partition $\hat{\mathcal{S}}$, to obtain an upper bound for our problem, the optimal solution $x^*(\hat{\mathcal{S}}, t)$ is substituted to the original problem one by one for each group of scenarios $\hat{\mathcal{S}}_i, i \in I_{\hat{\mathcal{S}}}$. Each resulting problem is called as residual problem and denoted by $\mathcal{P}(\hat{\mathcal{S}}_i, t)$. Then, the residual problems are solved for the remaining decision variables. This procedure is explained in Figure 3 for the four stage problem provided in Figure 2. Residual problems obtained after the substitution of $x^*(\hat{\mathcal{S}}_1, 4)$ into the original problem are demonstrated in (a) and (b) for *consecutive* and *half-and-half* methods, respectively. Note that, residual problems consist only of decision variables and constraints associated with the non-colored nodes.

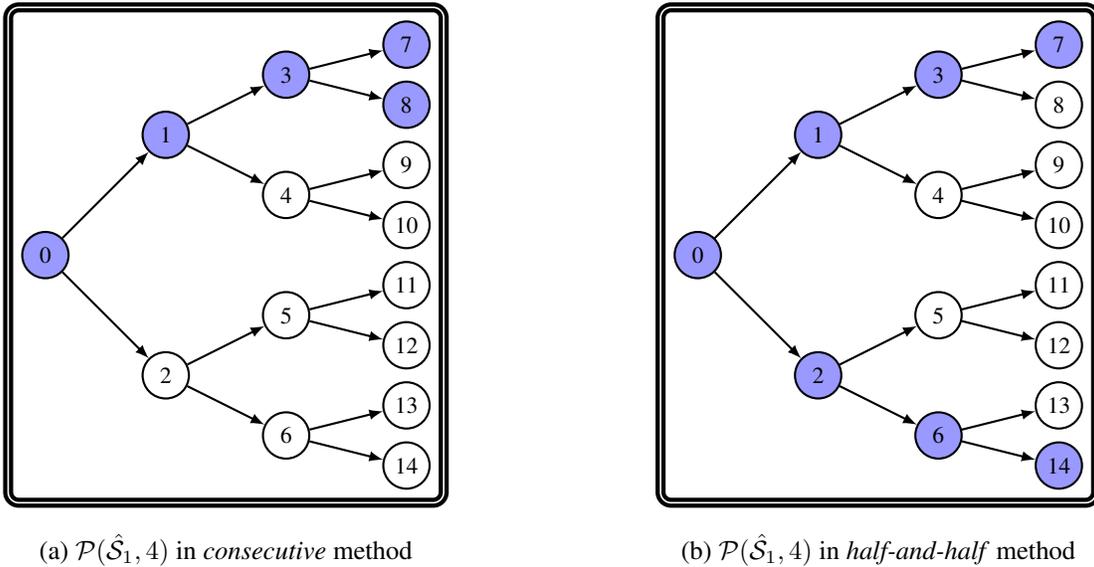


Figure 3: Problems solved to compute an upper bound in different selection strategies

After substitution, if $\mathcal{P}(\hat{\mathcal{S}}_i, t)$ is feasible, its optimal value represented by $z^*(\hat{\mathcal{S}}_i, t)$ is a valid upper bound on

$z^*(\mathcal{S})$. If $\mathcal{P}(\hat{\mathcal{S}}_i, t)$ is infeasible, then we set $z^*(\hat{\mathcal{S}}_i, t)$ to ∞ . Note that, to reduce the possibility of infeasibility, small t values can be used. Additionally, since the number of fixed variables in $\mathcal{P}(\hat{\mathcal{S}}_i, t)$ reduces as t decreases, the resulting bound may get stronger. After solving $\mathcal{P}(\hat{\mathcal{S}}_i, t)$ for all $\hat{\mathcal{S}}_i, i \in I_{\hat{\mathcal{S}}}$, the minimum of the obtained upper bounds is selected: $z^*(\hat{\mathcal{S}}, t) = \min_{i \in I_{\hat{\mathcal{S}}}} z^*(\hat{\mathcal{S}}_i, t)$ and $z^*(\hat{\mathcal{S}}, t)$ is the upper bound provided by scenario partition $\hat{\mathcal{S}}$.

The residual problems have less decision variables and constraints compared to the original problem, furthermore, they are decomposable into smaller problems. Therefore, each residual problem may require far less computation time than the original problem. Figure 4 provides the scenario decompositions of the residual problems given in Figure 3. As seen in Figure 4, the *consecutive* and *half-and-half* methods differ with regard to the number and sizes of the problems obtained after decomposition. In the *consecutive* method, since the scenarios in the group subproblems have more common nodes, the sizes of the problems after decomposition are greater. Consequently, we expect that solving the residual problem in the *half-and-half* method takes less time compared to the *consecutive* method.

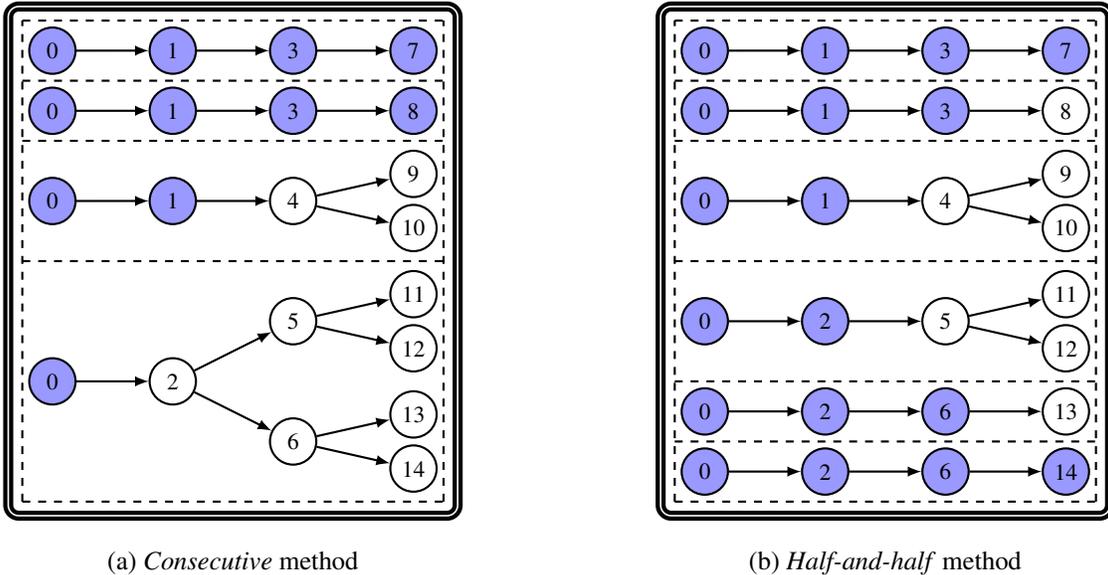


Figure 4: Scenario decomposition of $\mathcal{P}(\hat{\mathcal{S}}_1, 4)$ for different selection strategies

5 Computational Results

In this section, we present the results of our computational experiments. We first investigate the sizes of the instances that are solved to optimality using a general purpose solver and the quality of bounds obtained with the scenario groupwise decomposition for large instances. Second, we investigate the gains of demand response optimization for residents with different levels of price sensitivity. Third, we analyze the savings due to the use of renewable energy and batteries.

5.1 Scenario generation

We consider a time period of 24 hours and use equal number of periods in each stage. For instance, if we have eight stages, each stage consists of three hours. The random parameters are the temperature and the power of renewable energy. In our study, we consider the solar energy as our renewable energy source. The power of renewable energy can be calculated using global horizontal irradiance (GHI) data (total amount of solar irradiance on the horizontal surface of the ground [27]) and the features of the solar panel such as area, tilt angle and the conversion efficiency.

To generate the data of temperature and the power of renewable energy randomly, we use the hourly temperature and GHI data obtained from the dataset of National Renewable Energy Laboratory [28]. The correlation between GHI and temperature is calculated as 40% when considering the daytime data of the year 2005. This correlation can also be observed in Figure 5.

Despite the positive correlation, high tempera-

ture and low amount of solar radiation can be observed at the same time because of the solar absorption by

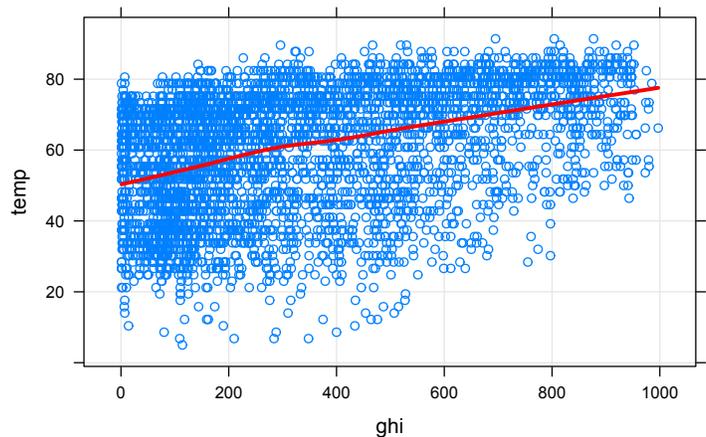


Figure 5: The relation between temperature and GHI

clouds, as mentioned in [29] and [30]. We use the real temperature and GHI data for each hour to take into account both the solar absorption and the positive correlation. We provide two data sets generated by the following two methods:

- *Data set-1*: This data set represents the cases with a good forecast for temperature and GHI. During the data generation process, the data of one month of a specific year are used. Firstly, a random day in the given month is chosen. For each hour of the first stage, corresponding temperature and GHI values of this day are used. When the stage is over, two branches emanate: one branch continues to use the data of this day and for the other branch, the temperature of the last hour of previous stage is compared with the temperature of the same hour of each day in the given month and the data of the day having closest temperature are used. The same procedure is applied for the rest of the stages.
- *Data set-2*: This data set represents the cases with a relatively bad forecast for temperature and GHI. Different from *Data set-1*, the data of a given month for different years are used. Data generation process is almost the same as *Data set-1*. As explained above, two branches emanate from each node, one branch continues to use the same year data of previous stage and for the other branch, the data of a randomly selected day of a randomly selected year are assigned.

For computational experiments, we consider a smart home equipped with five electrical appliances, each from different types, where HEMS is used to schedule their operations in a day in July.

Our data is available at <https://github.com/DemandResponseOptimization/datasets>.

5.2 Computation times and quality of bounds

All experiments are carried out on a 64-bit machine with Intel Xeon E5-2630 v2 processor at 2.60 GHz and 96 GB of RAM using Java and CPLEX 12.6.

To measure the limits of a general purpose solver and to see the quality of bounds obtained by the scenario groupwise decomposition approach, we first consider small instances. First, we analyze the impact of group

size and the number of stages for which the variables are fixed on the gap between lower and upper bounds. In Table 1, we report the averages of results for six instances from dataset 1 with six stages (32 scenarios). The first column gives the number of scenarios in each group and the second column gives the number of stages for which we fix the variables to compute an upper bound. We give the computation times and the gaps obtained by the *half-and-half* and *consecutive* methods. We set time limit of one hour to compute the lower bound and one hour to compute the upper bound. Since each bound computation requires solution of several problems, we distribute the hour equally for each problem. For instance, when each group subproblem contains two scenarios, we need to solve 16 subproblems to compute a lower bound. We set a time limit of 3600/16 seconds to each subproblem. To compute an upper bound, we set different time limits to each subproblem directly proportional to their sizes. If subproblem is not solved to optimality within the time limit, the best bounds are used.

Table 1: Comparison of gaps and solution times for instances with six stages

group size	number of stages fixed	<i>half-and-half</i> method		<i>consecutive</i> method	
		time	opt gap	time	opt gap
		(sec)	(%)	(sec)	(%)
2	1	2537.47	0.57	2269.42	0.68
	2	63.41	0.61	1063.85	0.63
	3	53.39	0.62	1057.05	0.63
	4	50.97	0.62	1054.83	0.63
	5	49.27	0.62		
4	1	2204.91	0.49	1711.04	0.60
	2	122.69	0.56	842.57	0.60
	3	115.6	0.58	793.80	0.60
	4	115.39	0.58		
8	1	2860.53	0.63	1722.61	0.41
	2	1274.68	0.71	880.56	0.45
	3	1270.76	0.73		
16	1	4259.95	1.92	4096.77	2.23
	2	3244.75	2.01		

If the *half-and-half* method is used, the smallest gap is observed when we solve subproblems of four scenarios and fix the variables in the first stage. However, it takes a long time to compute the bounds. The next best case arises again with subproblems of four scenarios but this time we fix the variables in the first two stages and this saves a significant amount of computation time. A better gap is obtained by the *consecutive* method when the group size is eight and the variables in the first two stages are fixed. However, this again requires a longer computation time. Based on this analysis, we decided to look closely on different instances when we take subproblems of size four and fix the first two stages. To compute upper bounds, we use the solutions from two randomly selected group subproblems. We report the results in Table 2 for 15 instances. For each instance, we report the lower and upper bounds computed using both selection methods as well as the time required to compute both bounds and the gaps. We also report the optimality gaps given by the solver in the same amount of time. Additionally, in the last column, we report the gaps for the solver with a time limit of one hour.

We see that the quality of bounds computed using both selection methods are very similar. However, the *half-and-half* selection requires a smaller amount of time to compute the bounds. The average gap for the *half-and-half* method is 0.51% and the average computation time is 82 seconds. The solver stops with an average gap of 12.37% in the same amount of time. When given the computation times of the *consecutive* method, the gap of the solver decreases to 10.92%. After one hour, this gap drops to 5.93%, which is significantly larger than the gap of the scenario groupwise decomposition approaches.

Table 2: Comparison of bounds, gaps and solution times for instances with six stages, when groups have four scenarios and the first two stages are fixed in computing upper bounds

<i>half-and-half method</i>					<i>consecutive method</i>					CPLEX
lower	upper	sol time	opt gap	CPLEX	lower	upper	sol time	opt gap	CPLEX	opt gap
bound	bound	(sec)	(%)	opt gap (%)	bound	bound	(sec)	(%)	opt gap (%)	in 3600 sec (%)
131.65	131.82	27.16	0.13	6.66	131.71	131.78	355.3	0.05	6.56	4.87
158.61	159.69	53.37	0.68	9.64	158.36	159.37	165.64	0.64	9.64	7.64
142.39	143.9	303.93	1.05	8.49	142.3	143.11	1855.11	0.57	6.85	6.74
133.58	133.81	27.43	0.17	7.24	133.65	133.8	379.24	0.12	5.81	4.69
153.84	154.77	26.36	0.60	7.82	153.85	154.34	296.09	0.32	7.82	6.64
161.45	161.95	16.02	0.31	18.28	161.42	161.78	56.32	0.22	13.42	4.26
150.17	150.39	17.01	0.15	7.36	150.09	150.83	18.07	0.49	7.36	4.96
158.84	160.15	268.08	0.82	12.60	158.51	159.96	214.33	0.90	12.6	8.65
148.00	148.83	88.27	0.56	19.41	147.98	148.82	486.02	0.56	9.25	5.87
143.69	144.41	58.23	0.50	9.24	143.68	144.24	254.6	0.39	9.1	5.54
211.03	211.94	25.35	0.43	21.52	210.45	211.92	281.38	0.69	19.73	6.99
189.36	190.15	12.78	0.41	22.45	189.47	190.2	71.79	0.39	22.45	1.85
151.37	151.96	17.66	0.39	7.97	151.32	151.9	59.16	0.39	7.94	6.08
178.42	178.89	15.53	0.26	16.69	178.53	178.82	38.17	0.16	16.69	5.83
157.82	159.74	273.73	1.20	10.22	157.47	159.49	1169.08	1.27	8.52	8.39
avg		82.06	0.51	12.37			380.02	0.48	10.92	5.93

As the *half-and-half* method gives smaller computation times, we use it to compute bounds for larger instances. In Table 3, we report the results for instances from datasets 1 and 2, respectively, with eight stages (128 scenarios) under different threshold values when each group contains four scenarios, the decisions in the first three stages are fixed and optimal solutions of four randomly selected group subproblems are used in computing upper bounds.

Table 3: Results for instances with eight stages

\bar{u}	dataset 1				dataset 2			
	sol time	opt gap	CPLEX opt	CPLEX opt	sol time	opt gap	CPLEX opt	CPLEX opt
	(sec)	(%)	gap (%)	gap in 3600 sec (%)	(sec)	(%)	gap (%)	gap in 3600 sec (%)
1.8	1962.65	0.58	9.87	8.39	2056.24	1.80	12.53	8.6
	2128.24	0.34	7.95	7.80	2491.96	1.43	7.8	7.62
	2129.39	0.72	8.96	8.41	2126.88	0.60	9.47	9.13
	2210.99	0.96	10.68	9.77	2206.33	0.85	8.08	7.78
	2012.08	0.33	9.22	9.16	3594.08	0.98	7.14	7.14
	1092.17	0.51	11.56	9.63	2247.10	0.92	7.62	5.55
avg	1922.59	0.57	9.71	8.86	2453.76	1.10	8.77	7.64
2.2	618.54	0.40	6.38	4.53	807.89	1.71	8.91	7.24
	53.95	0.10	1.63	0.96	721.89	0.78	16.07	7.30
	51.32	0.23	1.59	1.08	1468.1	1.11	15.02	10.85
	86.14	0.59	2.34	1.32	1156.85	0.85	8.83	6.6
	44.23	0.10	1.67	1.02	120.59	0.39	2.19	1.05
	1163.25	0.63	10.52	7.81	1009.09	0.89	15.13	12.11
avg	336.24	0.34	4.02	2.79	880.73	0.95	11.02	7.52

The results show that the decomposition approach is able to compute high quality solutions in reasonable computation times whereas the solver stops with significantly larger gaps after one hour of computation. We also observe that instances from dataset 2 with a smaller threshold are more challenging.

Finally, we present results for 12 stage problems (2048 scenarios) in Table 4. Here, we use groups of size eight. To compute upper bounds, optimal solutions of four random group subproblems are used and the decisions in the first four stages are fixed. As the problem is very large, we increase the time limit to two hours to compute

the lower bound and two hours to compute the upper bound.

Table 4: Results for instances with 12 stages

\bar{u}	dataset 1			dataset 2		
	sol time	opt gap	CPLEX opt	sol time	opt gap	CPLEX opt
	(sec)	(%)	gap (%)	(sec)	(%)	gap (%)
1.8	14431.62	12.70	28.65	14343.22	10.77	25.59
	14156.04	11.99	25.57	12533.56	12.8	23.49
	14353.66	8.92	26.93	14142.96	12.03	22.99
	10332.87	10.95	22.7	14099.57	14.6	25.06
	14239.2	16.38	28.74	13749.75	13.2	23.18
	14430.57	10.75	29.76	13520.34	17.41	25.82
avg	13657.30	11.95	27.06	13731.57	13.50	24.36
2.2	9380.30	1.98	8.26	12602.52	13.60	30.03
	13660.67	4.31	29.57	14046.66	8.92	28.25
	12851.99	0.71	30.45	12397.66	5.16	27.71
	13800.13	7.23	27.53	13448.18	10.13	19.47
	6933.12	0.75	2.05	11017.74	6.11	27.92
	6639.77	0.42	1.42	8368.98	3.6	29.74
avg	10544.33	2.57	16.55	11980.29	7.92	27.19

With the increase in the number of stages, we observe that the quality of bounds degrade. However, these instances are very challenging and the bounds of the scenario groupwise decomposition approach are significantly better than those of the solver in the same amount of time. The observation on the difficulty of instances from dataset 2 with a smaller threshold remains valid with 12 stages.

Based on the results of this experiment, we can conclude that the stochastic demand response optimization

problem is very difficult to solve exactly using a general purpose solver. Even the instances with six stages are not solved to optimality in an hour. The decomposition approach, on the other hand, gives very good quality bounds for instances with up to eight stages. However, the gap increases as the number of stages increases. Finally, instances with higher variability are more difficult to solve.

5.3 Gains due to demand response optimization

In our second experiment, we investigate the gains of demand response optimization for residents with different price sensitivity. To this end, we use instances with four stages (eight scenarios), which are all solved to optimality. We set the threshold value to 1.8 for the rest of the experiments. To model different levels of price sensitivity, we vary the disutility weight ϕ as 0.1, 0.5, 1, 5. The value 0.1 models a resident who is very sensitive to price whereas the value 5 models a resident who is not sensitive to price and values his/her comfort.

In Table 5, we report the improvements in the electricity cost and the weighted sum of electricity cost and disutility (our objective function) when we use optimization for four types of residents. For the case without optimization, we schedule the appliances in two ways. First we minimize the discomfort: The continuous appliances use electricity uniformly over the day. The temperature is set to its comfortable value and interruptible and uninterruptible appliances are used in the middle of their desired time intervals. No battery is used, hence the renewable energy is utilized when available. In the case of optimization, we solve our model. We report the average results over ten instances.

Table 5: The average gains of optimization compared to a schedule that minimizes disutility

	very price sensitive	price sensitive	less price sensitive	price insensitive
	resident	resident	resident	resident
gain in cost (%)	41.25	26.58	22.42	17.05
loss in disutility	18.65	9.49	6.72	1.73
total gain (%)	32.45	22.15	19.24	16.22

We observe that there is significant gain for all types of residents. There is clearly a big difference between the improvements in the electricity cost for residents that are very sensitive and insensitive to price. In the case of price insensitive residents, optimization still reduces the electricity cost by 17% with a little increase in the disutility.

In the second approach, we schedule the appliances in a greedy way to minimize cost. The continuous appliances operate at maximum energy level during the required number of periods with the lowest electricity price and operate at the remaining energy level at the period with the next lowest electricity price to provide total energy requirement. They operate at minimum energy level for the rest of the day. If the inside temperature is in the allowed range without heating or cooling, the temperature appliance does not operate. Otherwise, the temperature is set to its minimum or maximum comfortable value depending on the outside temperature. Interruptible and uninterruptible appliances are used in time periods that provide the lowest total electricity cost to complete their tasks in their desired interval. No battery is used. We report the average results over ten instances.

Table 6: The average gains of optimization compared to a greedy schedule with cost objective

	very price sensitive resident	price sensitive resident	less price sensitive resident	price insensitive resident
loss in cost (%)	3.08	22.38	26.64	32.89
gain in disutility (%)	87.91	98.60	99.43	99.98
total gain (%)	49.17	81.88	89.85	97.74

We observe that even if there is an increase in the electricity cost, the overall gain is significant for all types of residents as in the previous case. While improvement in disutility is more than 85% for all types of residents, it is above 99% for less price sensitive and price insensitive residents.

These results show that important gains are possible using optimization in demand response for all types of residents.

Next, we are interested in the tradeoff between the electricity cost and the disutility as well as the amount of electricity consumption exceeding threshold value for different types of residents. In Table 7, we report the average results over ten instances. Since keeping the temperature close to the comfortable value requires more energy than keeping it in the comfortable interval, total electricity consumption increases as price sensitivity of residents decreases. When appliances are used in their most desirable time intervals, the threshold value is exceeded more often and the electricity is charged at the higher price. Consequently, residents who are insensitive to price pay 43.37% more than the ones who are very sensitive to price while decreasing their disutilities by 86.78%. Almost half of the energy consumption of the price insensitive residents is charged at the higher price whereas the percentage is less than 18% for residents who are very sensitive to price.

Table 7: Results on electricity consumption above the threshold, cost and disutility

	very price sensitive	price sensitive	less price sensitive	price insensitive
	resident	resident	resident	resident
electricity consumption exceeding the threshold value	6.81	14.63	16.80	21.10
total net electricity consumption	37.87	43.79	44.77	45.67
total cost	117.71	146.77	155.61	168.76
total disutility	17.77	10.34	8.12	2.35

5.4 Gains due to battery and renewable energy

In our final experiment, we investigate the impact of battery and renewable energy on the electricity costs and disutilities of residents.

We present average results over ten instances in Table 8. Without the renewable energy, the use of batteries already results in significant improvements in the objective function for all types of residents. The gain due to

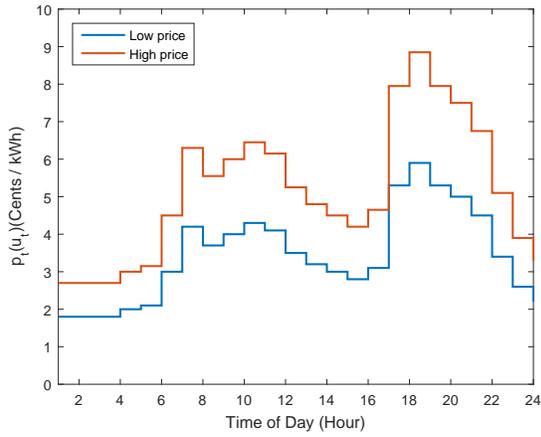
renewable energy is similar for all resident types and it is a remarkable amount. But even further improvements are possible when the renewable energy is combined with batteries.

Table 8: Results of gains due to battery and renewable energy

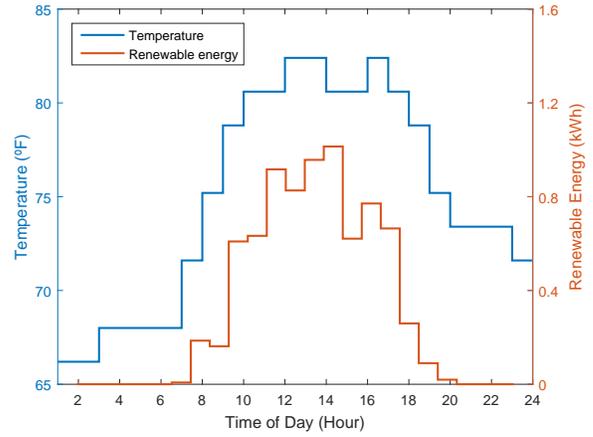
	very price sensitive	price sensitive	less price sensitive	price insensitive
	resident	resident	resident	resident
gain due to battery (%)	3.16	3.48	3.67	4.21
gain due to renewable energy (%)	25.31	25.10	24.82	24.84
gain due to battery and renewable energy (%)	27.11	28.34	28.66	28.74

Finally, to see the impact of the renewable energy and the battery on electricity consumption during a day, we plot the consumption due to different types of appliances for an instance over a scenario in Figure 6. Low and high price values and the temperature and available renewable energy over a day are depicted in (a) and (b), respectively. From (c) and (d), it can be deduced that using a battery decreases the electricity consumption in time periods with higher prices (between 5 and 10 P.M.). In (c), battery stores electricity when the price is low. While the battery feeds the system when interruptible appliance operates, it stores electricity when it does not operate or when the amount of renewable energy is high. This flexibility enables the appliances to work during their desirable time intervals while keeping total consumption under the threshold value. When the battery is not used, in (d), although the uninterruptible appliance moves away from its desirable time interval, total consumption exceeds the threshold value in a time period with a relatively higher price. The comparison of (c) and (e) shows that when renewable energy is not available, consumption shifts to hours with cheaper price. In addition to the increase in the number of time periods in which electricity is charged at the higher price, the amount charged at this price also increases without renewable energy. Figures (e) and (f) show that the battery provides savings even when renewable energy is not available. While the periods in which the threshold is exceeded are generally those with lower electricity price in (e), when the battery is removed, we encounter also

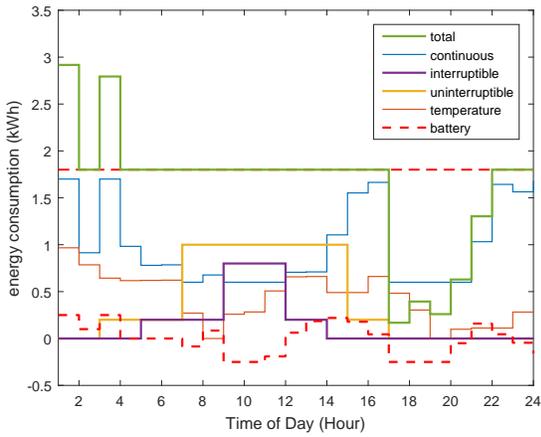
periods with high consumption at relatively higher prices in (f).



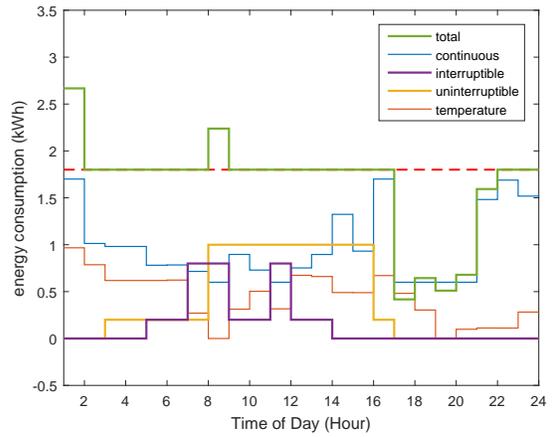
(a) Low and high price values



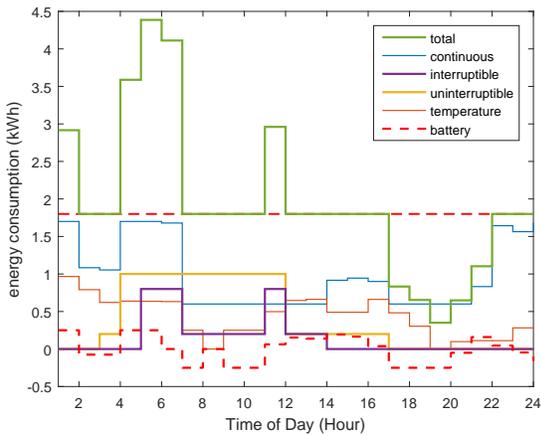
(b) Temperature and renewable energy



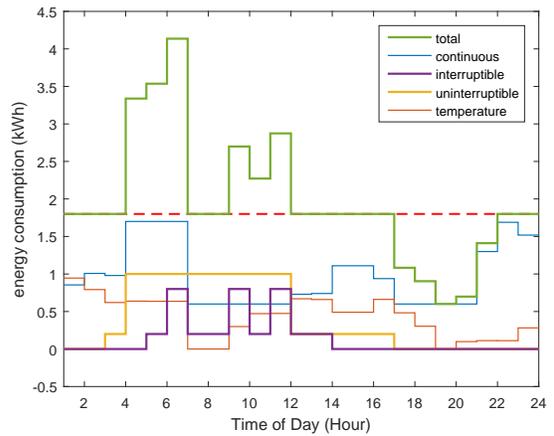
(c) Energy consumption



(d) Energy consumption without battery



(e) Energy consumption without renewable energy



(f) Energy consumption without battery and renewable energy

Figure 6: Effect of battery and renewable energy on the daily consumption

6 Conclusion

In this study, we presented a model for demand response optimization in which we considered all types of appliances already mentioned in the literature as well as renewable energy and batteries. We also incorporated the uncertainty in temperature and the availability of renewable energy using multi-stage stochastic programming. We provided a decomposition scheme that produced good quality bounds for large instances. We also conducted a detailed experiment that provided insights about the gains in cost and utility for residents with different price sensitivity. We observed that optimization can provide significant improvements both in cost and utility for all types of residents. The same is also true for renewable energy and batteries. With these three tools combined, the peak demand can be reduced significantly, providing savings for residents, flexibility for the electricity providers and less damage to the environment.

Our model is a multi-stage stochastic mixed integer nonlinear program and is quite challenging to solve exactly. The 0-1 variables are used to model the nonlinear structure of the price function and the schedules of interruptible and uninterruptible appliances. The uninterruptible appliances are modeled easily as the only decision is about their start times. However the interruptible appliances have a more interesting structure; we need to keep track of their start and finish times as well as the periods in which they function. A future research is to work on a strong formulation for the schedule of a single interruptible appliance. The inequalities for this structure can be used to strengthen the overall model and shorten the computation times. We find it also interesting to investigate how such strong formulations can be used in other decomposition approaches such as SDDP. Since SDDP is a stagewise decomposition approach, it would perform better for problems with large number of stages. However, since it depends on convexity and stagewise independency assumptions, it cannot be applied to our problem directly, therefore, solving our problem with SDDP is also another future research direction.

References

- [1] J. Conti, P. Holtberg, J. Diefenderfer, A. LaRose, J. T. Turnure, and L. Westfall, “International energy outlook 2016 with projections to 2040,” tech. rep., USDOE Energy Information Administration (EIA), Washington, DC (United States). Office of Energy Analysis, 2016.
- [2] U. EIA, “Electric power sales, revenue, and energy efficiency form eia-861 detailed data files,” 2017.
- [3] C. F. Covrig, M. Ardelean, J. Vasiljevska, A. Mengolini, G. Fulli, E. Amoiralis, M. Jiménez, and C. Filioiu, “Smart grid projects outlook 2014,” *Joint Research Centre of the European Commission: Petten, The Netherlands*, 2014.
- [4] A.-H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, and R. Schober, “Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid,” in *Innovative Smart Grid Technologies (ISGT), 2010*, pp. 1–6, IEEE, 2010.
- [5] A.-H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, “Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid,” *IEEE transactions on Smart Grid*, vol. 1, no. 3, pp. 320–331, 2010.
- [6] N. Gatsis and G. B. 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*, pp. 1–6, IEEE, 2011.
- [7] S. Althaher, P. Mancarella, and J. Mutale, “Automated demand response from home energy management system under dynamic pricing and power and comfort constraints,” *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1874–1883, 2015.

- [8] S. Tiptipakorn and W.-J. Lee, "A residential consumer-centered load control strategy in real-time electricity pricing environment," in *Power Symposium, 2007. NAPS'07. 39th North American*, pp. 505–510, IEEE, 2007.
- [9] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V. W. Wong, and J. Jatskevich, "Optimal real-time pricing algorithm based on utility maximization for smart grid," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 415–420, IEEE, 2010.
- [10] A.-H. Mohsenian-Rad and A. Leon-Gracia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Transactions on Smart Grid*, vol. 1, pp. 120 – 133, 2010.
- [11] 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*, pp. 351–356, IEEE, 2011.
- [12] N. Li, L. Chen, and S. H. Low, "Optimal demand response based on utility maximization in power networks," in *Power and Energy Society General Meeting, 2011 IEEE*, pp. 1–8, IEEE, 2011.
- [13] K. M. Tsui and S.-C. Chan, "Demand response optimization for smart home scheduling under real-time pricing," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1812–1821, 2012.
- [14] Z. Zhao, W. C. Lee, Y. Shin, and K.-B. Song, "An optimal power scheduling method for demand response in home energy management system," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1391–1400, 2013.
- [15] K. Ma, T. Yao, J. Yang, and X. Guan, "Residential power scheduling for demand response in smart grid," *International Journal of Electrical Power & Energy Systems*, vol. 78, pp. 320–325, 2016.
- [16] R. Energy, "Energy efficiency trends in residential and commercial buildings," 2010.

- [17] A. Illinois, “Real-time pricing for residential customers.” http://www.ameren.com/Residential/ADC_RTP_Res.asp, 2009.
- [18] B. Hydro, *BC Hydro Annual Report 2009*. BC Hydro, 2009.
- [19] M. V. Pereira and L. M. Pinto, “Multi-stage stochastic optimization applied to energy planning,” *Mathematical programming*, vol. 52, no. 1-3, pp. 359–375, 1991.
- [20] R. A. Collado, D. Papp, and A. Ruszczyński, “Scenario decomposition of risk-averse multistage stochastic programming problems,” *Annals of Operations Research*, vol. 200, no. 1, pp. 147–170, 2012.
- [21] J. F. Bonnans, Z. Cen, and T. Christel, “Energy contracts management by stochastic programming techniques,” *Annals of Operations Research*, vol. 200, no. 1, pp. 199–222, 2012.
- [22] J. Zou, S. Ahmed, and X. A. Sun, “Nested decomposition of multistage stochastic integer programs with binary state variables,” *Optimization Online*, vol. 5436, 2016.
- [23] R. Schultz, “Stochastic programming with integer variables,” *Mathematical Programming*, vol. 97, no. 1-2, pp. 285–309, 2003.
- [24] B. Sandikçi and O. Y. Özaltın, “A scalable bounding method for multistage stochastic programs,” *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1772–1800, 2017.
- [25] F. Maggioni, E. Allevi, and M. Bertocchi, “Monotonic bounds in multistage mixed-integer stochastic programming,” *Computational Management Science*, vol. 13, no. 3, pp. 423–457, 2016.
- [26] F. Maggioni and G. C. Pflug, “Bounds and approximations for multistage stochastic programs,” *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 831–855, 2016.
- [27] M. J. Reno, C. W. Hansen, and J. S. Stein, “Global horizontal irradiance clear sky models: Implementation and analysis,” *SANDIA report SAND2012-2389*, 2012.

- [28] NREL, “Nsrd b data viewer,” 2018. Retrieved February 8, 2018, <https://maps.nrel.gov/nsrdb-viewer>.
- [29] R. D. Cess, M. Zhang, P. Minnis, L. Corsetti, E. Dutton, B. Forgan, D. Garber, W. Gates, J. Hack, E. Harrison, *et al.*, “Absorption of solar radiation by clouds: Observations versus models,” *Science*, vol. 267, no. 5197, pp. 496–499, 1995.
- [30] S.-C. Tsay, M. D. King, R. F. Cahalan, and W. K.-M. Lau, “Absorption of solar radiation by clouds: A second look at irradiance measurements,” 2001.