Priority-Based Scheduling of Smart Appliances
With a Renewable Energy Source

Ashvini Vyavhare, Surya Tejaswini, and Shrisha Rao*

International Institute of Information Technology - Bangalore
Bangalore 560 100, India
http://www.iiitb.ac.in

July 2014

*Corresponding author, shrao@ieee.org
Abstract

The problem of scheduling electrical appliances according to priorities given and a variable power supply (such as from a renewable energy source) is considered. Present approaches to variable energy supply from a given source often require a spinning reserve or secondary source, and do not allow the use of appliances in a manner that fulfills users’ needs while taking account of the time-varying nature of the supply from such a source. We formulate this as a linear programming optimization problem, give a scheduling algorithm, and discuss its correctness. This algorithm also handles the case of a tie where more than one appliance has the same priority. In such a case, the most possible number of appliances are scheduled from that priority class. Simulation results with real data about appliances and the power supply from a wind power plant shows how the available power budget and power consumption of appliances is balanced at all times. The approach presented can become a fundamental concept for scheduling of complex appliances in the smart home environment or even in certain industry settings.

Keywords: scheduling, algorithms, priority, smart appliances, linear programming, renewable energy


This material is made available under the Creative Commons Attribution-Noncommercial-Share Alike License.
See http://creativecommons.org/licenses/by-nc-sa/3.0/ for details.
List of Figures

5.1 Priorities of appliances at time instance t1, t2, t3 . . . . . . . 19
5.2 Higher priority appliances getting scheduled at time instance t1, t2, t3 20
5.3 Eligible Appliances With Same Priorities At Time Instances t1,t2,t3 21
5.4 Maximum possible appliances getting power at time instance t1, t2 and t3 22
5.5 Power Budget and Power Consumption Vs Time . . . . . . . . . 23
List of Tables

3.1 Notations ................................................. 11
4.1 Notation in Algorithms ............................... 13
5.1 Power Budget Input to the Algorithm ............... 18
5.2 Generated Sample Input Data ......................... 18
5.3 Power Required By Some Appliances ................. 18
5.4 Data For Priority Tie Case ......................... 20
Chapter 1

Introduction

Smart building technologies currently being developed will allow the control of appliances in homes and industrial settings to meet desired goals including demand response [1] and demand-side management. At the same time, the increasing prevalence of renewable energy sources worldwide, and efforts to integrate the same with domestic settings, means that sooner rather than later, homes and other power consumers will need to account for the characteristics of variable energy generation, and use the same effectively with minimal need for intermediation [2]. Though there are many products dealing with building automation and smart homes, and there is a lot of research in the same as well (e.g., Jiang et al. [3] and Ricquebourg et al. [4] discuss the concept of a smart home in detail), there is yet a dearth of an approach that would permit an automated scheduling of appliances taking note of their relative priorities (given their specific relevances at particular times). There is also no approach that shows how to schedule loads in a smart building supplied by a variable energy source.

In this report, a household load priority scheduling algorithm for these appliances is given, which takes note of variable availability from the source, time-varying priorities of appliances, and scheduling constraints. As well known, the availability of power from renewable energy sources is generally not constant; e.g. solar energy supply itself is inherently variable [5], with the also true of wind energy supply [6]. The conventional way of using these sources requires the availability of “control power” on a grid from spinning reserves [7] to lessen the effects of variability. However, we propose here that appliances themselves should be controlled in such a way as to perform their tasks in a satisfactory manner in spite of variability in the power supply.

To evaluate the system, a household with a air conditioner, a washing machine, a refrigerator, a dryer, a toaster, and a dishwasher is simulated and analyzed. In this report, these devices are referred to as “household appli-
ances” or just as “appliances.” A user assigns priorities to these appliances according to current needs. For example, on a hot summer day, the air conditioner can be given higher priority compared to other appliances. Similarly, a toaster’s priority in the morning is possibly higher than that of a washing machine. Thus, priorities assigned to appliances change according to time. Taking note of the priorities given by users for the appliances, an attempt is made to maximize the count of active appliances, subject to the constraint of the power budget available at that time.

The hourly power budget from a renewable source is considered given historical data. Based on the allocated priorities of appliances, they are scheduled within the available power budget from the renewable source. The aim of the system is to schedule the consumption of appliances in the users’ residence taking into consideration the priority given by the user so that the total consumption does not exceed a certain limit. The purpose of the system is not to reduce the electricity consumption of the household per se, but rather to effectively schedule the appliances within the available power budget. The aim is to avoid a mismatch between demand and supply, rather than to effect savings directly.

Thus, an algorithm for scheduling smart electrical appliances, considering their priorities and the available power supply, is proposed in this report, in order to maintain the balance between demand and supply while satisfying user needs. Because of this, the time-varying electricity supply available from a renewable source can be effectively used, given the scheduling that takes into account the power budget availability. The approach can become a fundamental aspect of scheduling of complex appliances in the smart home environment, or even in the context of certain industrial systems.

The problem of matching supply and demand is mathematically formulated into a linear programming optimization problem, where we have a renewable energy source, and appliances to be scheduled. For our simulations, we consider a wind power plant as the renewable source. Appliances to be scheduled are specified by parameters like the power required, priority, and task specifications. The power required by the appliance is the rated working power of the appliance taken as a constant; the priority of the appliance is a user-defined vector giving the time-varying value of the priority; and the task specification gives the earliest start time, the latest end time, and the time required to complete the task effectively.

More than one appliance can be assigned the same priority and then the concept of priority class can be used. A priority class is a set of appliances having the same priority. If there is a tie among different appliances of a given priority class, then the algorithm chooses to maximize the number of appliances scheduled. In the linear programming problem, there is a maxi-
mization function and a constraint function which satisfy the required problem statement. The maximization function sees to it that power is scheduled to higher-priority appliances and maximization of appliances is done in case of a priority tie. The constraint function sees to it that the total power required by all scheduled appliances is less than or equal to the power budget available at that particular time instant.

Priority-based scheduling is of course a well-known concept, particularly so in the context of operating systems [8]. However, it does not appear to have been used to any notable extent in conjunction with energy usage, or in smart building systems.

The rest of the report is organized as follows. Chapter 2 considers the existing literature and related work. Chapter 3 describes the system model with problem formulation and its explanation. Chapter 4 explains the algorithm in detail and gives the proof of correctness for the same. Experimentation and results are presented in Chapter 5. Finally, conclusions and future work are considered in Chapter 6.
Chapter 2

Related Work

The literature relevant to our work is presented in three different sections, each presenting a different set of methods: demand-side management methods, load shifting methods, and other specific methods.

2.1 Demand Side Management Methods

The problem of scheduling smart appliances is often studied in the context of demand side management (DSM) [9], which in the power engineering context is often called demand response (DR), and refers to a set of related actions that enables electricity customers to modify the shapes and magnitudes of their electricity load profiles, in response to electricity supplier requirements or incentives. In a general sense, the philosophy of DSM is to encourage customers to use less energy during periods of peak demand when the power grid is likely to be strained, or to shift the use of energy to off-peak hours when the grid may be at risk of an unwanted surplus. While DSM often does not directly lead to a decrease in total energy use, it is considered to reduce the need for investments in grid expansions and maintenance, which tend to be capital intensive. Logenthiran et al. [10], Fadlullah et al. [11], Nguyen et al. [12], Fadlullah et al. [13] consider appliance DSM scheduling problem using game theoretic approaches, with real-time pricing.

Li et al. [14] consider households operating different appliances including batteries and PHEVs, and propose a DR approach on utility maximization. Each appliance is considered to make possible a certain benefit by changing its pattern of power consumption, which in turn effects individual optimality, and results in social optimality, i.e., when households selfishly optimize their own benefits, they automatically also maximize the social welfare. In the model of Li et al. [14], different appliances are coordinated indirectly in real
time, which reduces variation in demand and also increases the load factor, integration of battery which reduces peak load. It also amplifies benefit and reduces the demand variation. But in that model, appliances have to change their power consumption patterns, and the user’s preference is not considered at all. Such an approach to generating an optimal schedule of power consumption can be very disconcerting for the user, and may limit its practical value.

Ranade et al. [15] give a “ColoredPower Algorithm” designed to provide collaborative electricity demand shaping for both residential and small business customers, where customers participate by “coloring” their appliances with a qualitative priority. Demand shaping for this system must be scalable to millions of appliances, operate quickly and fairly across customers, and act on any given appliance infrequently. The ColoredPower algorithm addresses these challenges using randomized local actions. According to this algorithm, a controller shuts off some appliances in order to adjust the aggregate energy consumption to match global demand shaping. By contrast, in our approach, appliance scheduling is non-preemptive. Also, Ranade et al. [15] consider a varying aggregate demand from a user, while in our system the changing power budget from a renewable energy source is considered.

2.2 Scheduling Of Appliances To Shift Load

It is well known that there is a tendency to consume electricity in a regular manner during the same times of the day, leading to demand peaks. It is possible to smooth out such demand peaks by making users’ appliances consume electricity in a more temporally distributed way.

Busquet et al. [16] give a method for reducing demand peaks by suitable scheduling of appliances. The main concept behind their approach is the aggregation of home appliances into priority classes, and the definition of a maximum power consumption limit, which is not allowed to be exceeded during peak hours. In order to keep the total consumption of the household under the pre-determined limit, an event-driven scheduling algorithm is used. However, Busquet et al. [16] consider only a stable, conventional electric grid, i.e., the power supply cannot be variable.

The objective of Briel et al. [17] is to try and achieve the ideal shiftable load. A simple distributed approach for scheduling smart appliances is proposed. This schedules the start times of appliances that are programmed to run within a specified time window, so that in the aggregate one achieves a given ideal load. The main idea is to schedule smart appliances based on a probability distribution function that is derived from the ideal shift-
able load. However, Briel et al. [17] have not considered priorities for the appliances and operational constraints of the underlying power distribution system which depend on voltage regulation, or the fluctuations of renewable energy generation.

2.3 Other Methods

Pathak et al. [18] give a scheduling strategy for DSM. This has gained attention in the domain of smart-grid technology because of advantages for the reliable functioning of power distribution systems and for desirable outcomes on electricity prices. They propose a particle swarm optimization (PSO) method applied to residential load management. This approach manages load consumption when the consumption is above an imposed and fixed power consumption limit. But in our case we seek to schedule appliances according to the available power budget limit at that instant.

Another approach by Srikanth and Samunuri [19] is in the context of computing systems, and uses a real time operating system (RTOS). Their main focus is on execution of different power applications using priority dynamic scheduling during run-time. In this approach, the scheduling of tasks or power appliances is through direct memory access (DMA) peripherals to optimize CPU utilization. This is mostly a hardware-oriented embedded solution for scheduling. In our work, we propose an algorithm that can also work in software, rather than requiring dedicated, expensive hardware.
Chapter 3

System Model

In this work, the problem of scheduling appliances according to user priorities is considered. A written description of the appliance scheduling problem is provided, which is mathematically formulated as a linear programming problem.

We make the following assumptions as part of the model:

- We consider one electricity source provider. (This is not a serious limitation, insofar as multiple sources can be aggregated and mathematically dealt with as one.)

- \( n \) is the number of appliances to be scheduled, as mentioned in Table 3.1.

- For each appliance in consideration, the start time, end time and required time to complete the work are given.

- Priority \( P_i \) for appliance is given by a user for particular time instance \( t \) (the priority of an appliance may change over time).

- For priorities from 1 to \( P \), \( P \) is considered the highest and 1 the lowest priority.

- More than one appliance can be given the same priority; \( \mathbb{P} \mathbb{C} \) is a set of priority classes with priorities given to the appliances.

- If there is a tie between priorities of appliances, for a given power budget, the greatest possible number of appliances should get scheduled.

Table 3.1 shows the notation used in the problem formulation. As we assume that more than one appliance can have same priority, we have priority
classes \( \mathbb{P}_C \) in Table 3.1. \( J \) denotes the set of appliances. As mentioned earlier, we consider the set of air conditioner, washing machine, refrigerator, dryer, toaster and dishwasher as indicative for our analyses. Our goal is to give power to appliances given higher priority by user, so that within the same priority class, we maximize the appliance count getting the available power.

We need to find a solution for the following LP optimization problem, which is the basis of our algorithm.

\[
\max \sum_{i=1}^{n} P_i \cdot x(i) \tag{3.1}
\]

subject to:

\[
\sum_{i=1}^{n} r_i \cdot x(i) \leq W_t \tag{3.2}
\]

Equation (3.1) shows the optimization problem as a maximization of the sum of the product of priority \( P_i \) given to an appliance and an indicator variable \( x \). This indicator variable is defined as mentioned in Table 3.1. So, in the results, the value of this indicator variable is set to 1 for appliances being scheduled with the required power at that instant of time. The next, equation (3.2) is the constraint which says that sum of the product of power required by scheduled appliances and corresponding indicator variable should be less than or equal to power budget available at that time instance. The given maximization problem is solved for each priority class, as is explained in detail in the next section. Equation (3.1) is for satisfying the goal of maximizing the scheduled appliances in case of a priority tie.

### Table 3.1: Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>Number of appliances</td>
</tr>
<tr>
<td>( J )</td>
<td>set of appliances 1, 2, \ldots, ( n )</td>
</tr>
<tr>
<td>( \mathbb{P}_C )</td>
<td>Set of priority classes 1, 2, \ldots, ( k ) and ( k \ll n )</td>
</tr>
<tr>
<td>( P_i )</td>
<td>Priority of appliance ( i ); ( 1 \ll P_j \ll k )</td>
</tr>
<tr>
<td>( T )</td>
<td>Total number of time instants</td>
</tr>
<tr>
<td>( t )</td>
<td>Particular time instance 1, 2, \ldots, ( T )</td>
</tr>
<tr>
<td>( W_t )</td>
<td>Power Budget available at time instant ( t )</td>
</tr>
<tr>
<td>( r_i )</td>
<td>Required power by appliance ( i )</td>
</tr>
<tr>
<td>( x )</td>
<td>Indicator variable; ( x(i) = 1 ) if appliance ( i ) is given required power ( r_i ), else ( x(i) = 0 )</td>
</tr>
</tbody>
</table>
Chapter 4

Algorithm

In this section, we briefly describe the algorithm for priority-based scheduling of electrical appliances. Table 4.1 shows the notation used to discuss the algorithm. In the notation, symbols $\alpha$, $\beta$ and $\gamma$ are matrices with $n \times m$ dimensions, i.e., each row in the matrix denotes that particular appliance’s starting, ending and required time instants over 1 to $T$ respectively. In order to take note of the changing priorities of appliances at every time instant, $\delta$ is a $n \times T$ matrix, where every row $i$ signifies the varying priorities of appliance $i$ over 1 to $T$. $\vec{A}$ is a $1 \times T$ vector consisting of the power budget values over 1 to $T$. $\vec{R}$, $\vec{Q}$, $\vec{F}$, $\vec{C}$, $\vec{O}$ are all $1 \times n$ dimensional vectors denoting the power required, constraint part needed for LP formulation, boolean vector, intermediate coefficients, and coefficients for all the appliances from 1 to $n$ respectively. $\vec{X}$ is the final output vector consisting if the appliances being scheduled for a particular time instant $t$. The appliances scheduled are represented by 1 and the rest (those not scheduled) by 0.

All the required input data is given to the algorithm using a file. So, $\text{Readfile}()$ is a function to read this input file and save the input data in the required format. The function $\text{Initialize}()$ initializes all the local parameters; $\text{SolveLP}()$ is used to solve the LP problem and get the scheduled appliances; $\text{Update}W_t()$ updates the value of $W_t$, as it is decreased if any appliance is scheduled in a particular priority class. The function $\text{Update}\gamma()$ is to change the required time matrix $\gamma$ if a particular appliance is getting scheduled.

4.1 Explanation

We use a file containing input data such as priorities, start and end times, and required power to work for all appliances. At a particular time instance $t$, all the appliances having their start times less than $t$ and end time greater that
### Table 4.1: Notation in Algorithms

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>Maximum number of times appliances are scheduled $1 &lt;&lt; m &lt;&lt; T$</td>
</tr>
<tr>
<td>$\text{count}$</td>
<td>variable to count appliances eligible for scheduling at particular time instant</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$n \times m$ matrix containing scheduled start time for all appliances</td>
</tr>
<tr>
<td>$\beta$</td>
<td>$n \times m$ matrix containing scheduled end time for all appliances</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>$n \times m$ matrix containing time required by appliance to complete the work</td>
</tr>
<tr>
<td>$\delta$</td>
<td>$n \times T$ matrix containing changing priorities of appliances according to time</td>
</tr>
<tr>
<td>$\vec{A}$</td>
<td>$1 \times T$ vector containing available power from the source at a particular time instance</td>
</tr>
<tr>
<td>$\vec{R}$</td>
<td>$1 \times n$ vector containing power required by appliance</td>
</tr>
<tr>
<td>$\vec{Q}$</td>
<td>$1 \times n$ vector containing power required by appliance eligible for scheduling at particular time instance</td>
</tr>
<tr>
<td>$\vec{F}$</td>
<td>$1 \times n$ boolean vector which defines whether a appliance is allocated the power or not</td>
</tr>
<tr>
<td>$\vec{X}$</td>
<td>$1 \times \text{count}$ vector containing the selected appliances for a particular time instant $t$ under a priority class $l$</td>
</tr>
<tr>
<td>$\vec{C}$</td>
<td>$1 \times n$ vector containing intermediate coefficients for objective function</td>
</tr>
<tr>
<td>$\vec{O}$</td>
<td>$1 \times n$ vector containing coefficients for final objective function</td>
</tr>
</tbody>
</table>

*Readfile*() Function for reading the file containing input data

*Initialize*() Function to initialize all the local parameters

*SolveLP*() Solves the LP using bintprog method from Matlab

*UpdateW*(*)() Update the value of $W_i$ after giving power to higher priority class appliances

*Updateγ*(*)() Decreases required time of appliance by 1 if it has been given power
are considered and possibly scheduled according to their given priorities. We find real data for required power of some home appliances from GE’s Data Visualization \[20\] website, and power generated using wind (renewable source) from NETA Generation By Fuel Type report \[21\].

Scheduling of appliances based on their priorities is done in the following manner, where the algorithm mentioned is divided into three sections, and any appliance being scheduled should satisfy two conditions. The three sections are: one \textit{for loop} starting at line number 6, second \textit{for loop} starting at line number 12 and then LP problem solving part at line number 20 in Algorithm 1. Among the two conditions required to be satisfied, first condition checks appliance’s start time, end time and required time and decides if the appliance is eligible to be scheduled or not. This is taken care by the first \textit{for loop} of the algorithm. In the second condition, priorities of only those appliances satisfying the first condition are considered eliminating others. This is taken care by the second and third sections of the algorithm. In the second section the appliances not satisfying the first condition are eliminated where in the third section an objective function is constructed consisting of appliances satisfying the first condition. Then further using power budget, power required by appliances and objective function data the linear program is solved. The output of the linear program consists of the appliances scheduled for that particular time instant $t$.

Equation (3.1) shows the LP problem formulation, and (3.2) gives the constraint. In the following we give a stepwise explanation of the algorithm.

1. Read input data from file and initialize all the required variables.
2. Update $W_t$ with $\vec{A}$ value at instant $t$.
3. For each appliance $i$, check if the $\alpha$ value at a particular instant $t$ is less than $t$, the $\beta$ value at a particular instant $t$ is greater than or equal to $t$, and the remaining time is greater than or equal to $\gamma$; if these conditions are satisfied then update the $\vec{F}$ value to 1 for that particular appliance indicating that the appliance can be allotted power.
4. For each priority class $l$, starting from the highest priority $k$, for each appliance $i$ at a particular instant $t$ check if the $\delta[i, t]$ value is equal to the priority class value $l$, if so then update $\vec{C}$ with $\delta \times \vec{F}$, the product of the priority value of a appliance at a particular time instant and its flag value.
5. In parallel, during the formation of the objective function, check if the value of $\vec{C}$ is not equal to zero; if so then update $\vec{O}$ with the $\vec{C}$ value thereby computing the final required objective function.
6. Update the $\vec{Q}$ using $\vec{R}$ and then increase the count; by doing so the matrix required for the constraint part in the LP is computed.

7. The LP is solved and the outputs are stored in $\vec{X}$. Computing $\text{SolveLP}()$ gives as a result the appliances which that be getting required power.

8. After computing the value of $\vec{X}$ for one time instant, the values of $W_t$ and $t$ are updated. This $\vec{X}$ is for current particular priority class $l$. This has to be calculated for each priority class $l$ at each time instance $t$.

---

**ALGORITHM 1:** Priority based appliance scheduling

```plaintext
Data: N
Data: PC
Result: $\vec{X}$

begin
  Algorithm
  2 Readfile(fileName)
  3 Initialize()
  4 while $t \leq T$ do
  5     $W_t \leftarrow A[t]$
  6     foreach appliance 'i' in n do
  7         $s \leftarrow 1$
  8         while $s \leq m$ do
  9             if $\gamma[i,s] \neq 0$ then
 10                 if $\alpha[i,s] \leq t$ and $\beta[i,s] \geq t$ and $\gamma[i,s] \leq T - t$ then
 11                     $F[i] \leftarrow 1$
 12             end
 13         end
 14     end
 15     foreach priority class 'l' starting from highest priority 'k' do
 16         foreach appliance 'i' in n do
 17             if $\delta[i,t] \equiv l$ then
 18                 $C[i] \leftarrow \delta[i,t] \times F[i]$
 19                 if $C[i] \neq 0$ and count $\leq n$ then
 20                     $O[count] \leftarrow C[i]$
 21                     $Q[count] \leftarrow R[i]$
 22                     count $\leftarrow$ count + 1
 23             end
 24         end
 25     end
 26     $\vec{X} \leftarrow \text{SolveLP}($$O,$$\vec{Q}, W_t)$
 27     Update$W_t()$
 28     Update$\gamma()$
 29     $t \leftarrow t + 1$
 30 end
end
```
4.2 Argument For Correctness

Here we discuss the correctness of the algorithm. As our aim is to give power to high-priority appliances, with maximum appliance count in case of ties between appliances of the same priority class, our algorithm should ensure this.

As written in Algorithm 1, the for loop at line 12 takes care of giving power to high-priority appliances, as we execute the loop from the highest priority class to the lowest. Now we have to prove that if there are more than one appliances with the same priority then out of them maximum appliances should get the power. This also mean that maximum number of appliances will get the power according to available power budget at that instant of time.

Let for given input data, \(m_1, m_2, m_3\), etc., be the counts of appliances in particular priority classes which get power according to the algorithm in case of tie between priority. So finally overall sum \(m = m_1 + m_2 + m_3\ldots\) is the count of appliances getting the power at a instant of time. We have to show that these \(m_1, m_2, m_3\ldots\) are the maximum counts of appliances.

To prove by contradiction, take the case of \(m_1\) appliances getting power for particular priority class having same priority \(P\) for more than one appliance. And let it be that the \((m_1+1)\)th appliance also can get power. So this means, there was power budget available and still our algorithm did not do the right thing and give power to the \((m_1+1)\)th appliance. But the decision of to which appliances power will be given is taken by solving the LP problem given by equation (3.1). In our algorithm this is solved at line number 20 which is in the for loop starting at line 12. This LP problem is solved for each priority class.

According to our assumption of the \((m_1+1)\)th appliance,

\[
P \times (m_1 + 1)
\]

is the maximum sum for LP formulation such that

\[
P \times (m_1 + 1) > P \times m_1
\]

So this is a contradiction, because the LP is formulated in such a way that it gives the maximum sum of priorities of allowed appliances. Hence the result.
Chapter 5

Experimentation and Results

We have simulated the algorithm using MATLAB. Real data for simulation were collected from the NETA: Generation By Fuel Type website [21]. These data are taken from renewable energy source, in this case being a wind power plant. Power generated by this wind power plant on 23rd October 2012 is used for the simulation. The data collected from this site are used as input power budget in our implementation.

5.1 Input Data

We consider that the power generated from renewable energy source is equally distributed among homes (or alternatively, that a smaller scaled-down source with the same pattern exists). On checking data from the NETA website [21], we compute that the wind energy generated from a plant can give power to approximately 100000 homes. Without loss of generality, we have considered that the varying power generated is being distributed equally among 100000 homes resulting in generation of our $W_t$. As a result, the power generated at wind power plant gets distributed across these homes. So now the available power budget for particular time instances is a corresponding fraction of the power generated by the source. Table 5.1 indicates the input $\vec{A}$ values (in Watts).

Table 5.2 shows sample generated input data in order to verify the performance of the algorithm. Here the Priority, Start time, End time, and Required time of appliances are defined. Table 5.3 shows the power required by some home appliances respectively. For sample results in the algorithm, we have considered four appliances: Air Conditioner, Toaster, Water Heater and Dish Washer, considering their published power ratings.
Table 5.1: Power Budget Input to the Algorithm

<table>
<thead>
<tr>
<th>Time Instance</th>
<th>Power (Watts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>6700</td>
</tr>
<tr>
<td>t2</td>
<td>5920</td>
</tr>
<tr>
<td>t3</td>
<td>4060</td>
</tr>
<tr>
<td>t4</td>
<td>5100</td>
</tr>
<tr>
<td>t5</td>
<td>4550</td>
</tr>
<tr>
<td>t6</td>
<td>4260</td>
</tr>
<tr>
<td>t7</td>
<td>3960</td>
</tr>
<tr>
<td>t8</td>
<td>4520</td>
</tr>
<tr>
<td>t9</td>
<td>4520</td>
</tr>
<tr>
<td>t10</td>
<td>5100</td>
</tr>
</tbody>
</table>

Table 5.2: Generated Sample Input Data

<table>
<thead>
<tr>
<th>Appliance</th>
<th>Time Instance</th>
<th>Priority</th>
<th>Start Time</th>
<th>End Time</th>
<th>Required Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air-conditioner</td>
<td>t1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Toaster</td>
<td>t1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Water-Heater</td>
<td>t2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Dish-Washer</td>
<td>t1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Air-conditioner</td>
<td>t3</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Water-Heater</td>
<td>t3</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Toaster</td>
<td>t3</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Dish-Washer</td>
<td>t3</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.3: Power Required By Some Appliances

<table>
<thead>
<tr>
<th>Appliance</th>
<th>Power Required (Watts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Air Conditioner</td>
<td>5000</td>
</tr>
<tr>
<td>Toaster Oven</td>
<td>1550</td>
</tr>
<tr>
<td>Water Heater</td>
<td>2475</td>
</tr>
<tr>
<td>Dishwasher</td>
<td>1800</td>
</tr>
<tr>
<td>Clothes Dryer</td>
<td>3400</td>
</tr>
<tr>
<td>Coffee Machine</td>
<td>1500</td>
</tr>
<tr>
<td>Iron</td>
<td>1100</td>
</tr>
<tr>
<td>Refrigerator-Freezer 20cu.ft</td>
<td>800</td>
</tr>
<tr>
<td>Washing Machine</td>
<td>512</td>
</tr>
<tr>
<td>Well Pump</td>
<td>2238</td>
</tr>
</tbody>
</table>
5.2 Scheduled Appliances According to Priority

For the given input values in Tables 5.1, 5.2, 5.3 the algorithm is applied and it is observed that the appliances are scheduled according to the priorities given. Figure 5.1 shows priorities of appliances at time instants t1, t2 and t3. At t1 and t2 time instance Air Conditioner and Toaster are given higher priority, where at t3, Dish Washer is given higher priority.

![Figure 5.1: Priorities of appliances at time instance t1, t2, t3](image)

Figure 5.2 is plotted to explain the result obtained from the algorithm for above input data condition. At time instance t1, Air Conditioner and Toaster have been scheduled as they have higher priorities as shown in Figure 5.1. At instance t2, Air Conditioner, Toaster and Water Heater have higher and the same priority, but only the Toaster and Water Heater are getting the power. This is because of the available power budget at time instance t2 is as given in Table 5.1.

When computed manually we could observe a match in the results. On comparing Figures 5.1 and 5.2, we observe that scheduling is done according to the priorities of appliances, and our priority-based appliance scheduling algorithm works well in this context.
5.3 Priority Tie Case

As discussed earlier and given the proof, Algorithm 1 ensures that maximum appliances get scheduled if there is priority tie. To show this practically, in the simulation, we consider inputs for appliances as shown in Table 5.4. At time instance t1, t2, t3, the given appliances have the same priorities, and the available power budget as given in Figure 5.3.

<table>
<thead>
<tr>
<th>Time Instance</th>
<th>Appliance With Same Priority</th>
<th>Available Power Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Air Conditioner, Toaster, Dish Washer</td>
<td>6700</td>
</tr>
<tr>
<td>t2</td>
<td>Air Conditioner, Toaster, Water Heater</td>
<td>5920</td>
</tr>
<tr>
<td>t3</td>
<td>Toaster, Water Heater, Dish Washer</td>
<td>4060</td>
</tr>
</tbody>
</table>
Now, in this situation at instant t1, only the Air Conditioner is not getting scheduled (as it requires 5000 Watts of power), but the Toaster and Dish Washer have been scheduled by the algorithm (as they require 1550 Watts and 1800 Watts respectively). This means the algorithm schedules the most possible appliances within the given power budget. Figure 5.4 shows the same result, and the same explanation can be given in case of t2 and t3 as according to Figure 5.4.

5.4 Resulting Total Power Consumption By Appliances

For a given power budget, how much power the algorithm is utilizing is given by Figure 5.5. From this, the efficacy can also be verified; although maximum power utilization is not our main goal we can see that the algorithm is power-efficient. Here we have plotted a bar chart comparing the power budget $W_t$ and the total power consumed by appliances at $t$. The total power consumed is computed from $\vec{X}$ and $\vec{Q}$. Steps of the algorithm are repeated for $t$ varying from 1 to $T$. 

Figure 5.3: Eligible Appliances With Same Priorities At Time Instances t1, t2, t3

Figure 5.4 Resulting Total Power Consumption By Appliances

Figure 5.5
Figure 5.4: Maximum possible appliances getting power at time instance $t_1$, $t_2$ and $t_3$
Figure 5.5: Power Budget and Power Consumption Vs Time
Chapter 6

Conclusion

Energy production by many sources, especially wind, PV, and other renewable sources, is generally variable, which affects the balance between demand and supply. It is not sufficient to merely assume supplementary sources whenever there is a shortfall, but rather, it is desirable that a scheduling approach for appliances be used that allows the time-varying nature of such sources.

To address this problem we are taking into consideration the changing power budget from such sources as well as the needs of the system consuming power. We give a priority-based scheduling algorithm for home appliances using a Linear Programing formulation. According to our problem formulation and algorithm, the appliances with highest priority and satisfying the constraint given (3.2) get required power, with other appliances scheduled later, with priority ties also being handled.

This approach can be a part of large smart home applications intended for use with variable sources, where users can set priorities for appliances and times when they are to be scheduled. Given this information, our algorithm finds the detailed schedule of appliances to meet task requirements while obeying supply constraints.

To extend this work, scheduled appliances can be further categorized as interruptible and non-interruptible. Also, different power levels can be considered for appliances to work. According to category and power level required by a particular appliance, the scheduling algorithm can be enhanced. Pricing can be considered for the power supply, with scheduling to take place accordingly to reduce overall cost incurred.
Bibliography


