



26/05/2018

SMART GRID ALGORITHMS



Sanjivan Naicker, 26425637 (Supervised by Professor Manos Varvarigos)

ELECTRICAL AND COMPUTER SYSTEMS ENGINEERING DEPARTMENT, MONASH UNIVERSITY
ECE4095 FINAL YEAR PROJECT B

Significant Contributions

The main contributions of this project are the design and implementation an algorithm for the smart grid that has not been done before by previous works in the same field of study. The use of a simple, deterministic algorithm (Tariff-Proportional Algorithm) is compared to a more complex, stochastic optimisation algorithm (Genetic Algorithm). The simple, deterministic algorithm is designed from scratch and it is a novel design not implemented by any of the works in this field of study. On the other hand, the stochastic optimisation algorithm which has been quite commonly used in similar contexts but it has not been used on the model that this project is based on. Simulated and evaluated the algorithms on MATLAB to minimise the average cost per hour in a day spent by the user and to lower the peak-to-average ratio (PAR) of power demand. The results from the investigations of this project suggests that using the stochastic optimisation algorithm is better at minimising the average costs per hour in a day spent by the user and lowering the PAR of power demand. The results also suggest that using the simple, deterministic algorithm uses very little computational time and has the best scores for users' satisfaction. The investigations from this project have given some interesting results that can be of great use to future works in the same field of study.

Poster



MONASH University
Engineering

Department of Electrical and
Computer Systems Engineering

ECE4095 Final Year Project 2017

Sanjivan Naicker

Smart Grid Algorithms

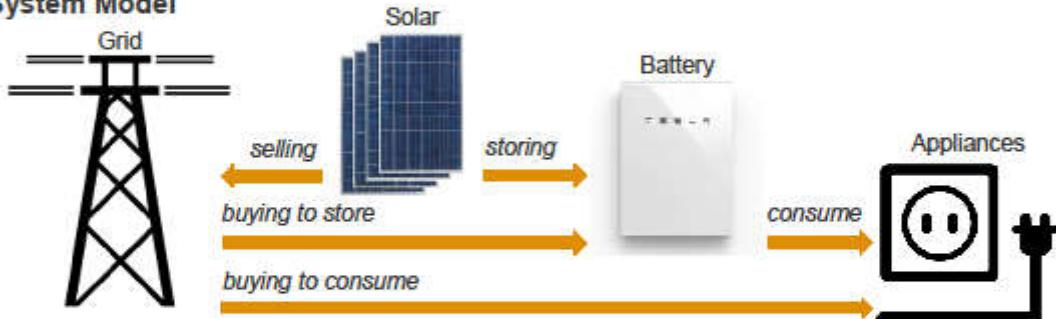
Supervisor: Professor Manos Varvarigos

Project Goals

Smart grids allow communication between the user and the supplier to ensure power demands are met. Smart grids are more necessary now that the use of solar power by users have become more prevalent. This project focuses on the user end of the smart grid.

The aim of this project is to design and compare an algorithm that takes advantage of the solar power harnessed, battery store and varying tariffs (for buying and selling) to minimise the expenditure of the user and to lower the peak-to-average ratio (PAR) of the user's power consumption. The algorithm designed is compared to an existing algorithm to compare their performance.

System Model



Algorithms

The objective function is to minimise the average cost over a day. The amount of solar power to store, power bought from grid, consumption from battery and delays for tasks are decisions that the algorithms are supposed to make.

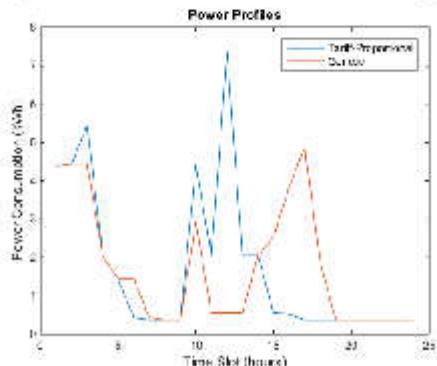
Tariff-Proportional Algorithm (No task scheduling performed, delays for tasks are zero)

The algorithm bases its decisions on the tariffs.

- 1 - If next tariff is greater than current tariff
- Power stored to battery (from solar & grid) proportional to change in tariff
- 2 - If next tariff less than current tariff
- Power consumed from battery proportional to change in tariff
- 3 - If tariff remains the same
- No change in decisions

Results

The following results show that Genetic Algorithm is superior Tariff-Proportional Algorithm. The use of task scheduling has a major effect on the PAR and cost. (Time slots are from 8am to 7am the next day)



Genetic Algorithm (Task scheduling performed)

Genetic algorithm is an optimisation algorithm based on evolution that optimises the decision variables to fit the objective function.

	Tariff-Proportional	Genetic
PAR	5.25	3.36
Average Cost	27.66¢, 29.58¢ (No selling)	20.76¢, 41.68¢ (No selling)
Cost		

Changes made in Results: (1) Column 1 title changed, (2) Column 1, Average Cost values changed, (3) Row 1, PAR values changed

Executive Summary

This report elaborates the design and implementation of algorithms in a smart grid. The algorithms are designed to minimise the average cost per hour in a day spent by the user. The algorithms that are being evaluated are Tariff-Proportional Algorithm and Genetic Algorithm. The resulting average cost per hour in a day and the peak-to-average ratios (PAR) of power demand of these two algorithms are compared.

Both algorithms takes advantage of the ability to harness solar power and the ability to store power in a battery storage. The decisions made by the algorithms are the proportion of harnessed solar power that is siphoned to the battery (remainder solar power is sold), the battery power discharged to the electrical appliances (tasks) (remainder of tasks uses power from grid), the power bought from the grid to charge the battery and the delays that each task should make for task scheduling. MATLAB is used to program these algorithms and display the results. A crucial difference between the Tariff-Proportional Algorithm and the Genetic Algorithm is that the former sets the delays for each task to zero. In other words, no task scheduling is performed by the Tariff-Proportional Algorithm. The effect of selling excess harnessed solar power on the resulting average cost per hour in a day spent by the user is investigated. The effect of using solar power to directly offset the power demand and the effect of using task scheduling alone on the PAR of power demand are investigated. The computational time and the user's dissatisfaction are additional aspects that are investigated.

The final results show that the Genetic Algorithm is far superior to the Tariff-Proportional Algorithm in minimising the cost and lowering the PAR of power demand but the Tariff-Proportional Algorithm is superior in user satisfaction and computational time. The inability of the Tariff-Proportional Algorithm to schedule tasks, resulted in much poorer PAR of power demand and average cost per hour spent in a day. The ability to sell excess harnessed solar power resulted in a huge decrease in the average cost per hour in a day. Taking away this ability to sell harnessed solar power, the Tariff-Proportional Algorithm performed better than the Genetic Algorithm.

In conclusion, the main take from this investigation is that a stochastic optimisation algorithm such as the Genetic Algorithm provides far better results compared to a deterministic algorithm such as the Tariff-Proportional Algorithm in minimising the average cost per hour in a day and lowering the PAR of power demand. However, the tariff-Proportional Algorithm is better in terms of computational time and user's dissatisfaction.

Table of Contents

Significant Contributions	i
Poster.....	ii
Executive Summary	iii
Table of Contents.....	iv
List of Figures	vi
List of Tables	vii
Abbreviations	viii
1. Introduction.....	1
2. Literature Review.....	4
3. Overview.....	6
3.1. System Model	6
3.1.1. Solar-Battery Model.....	6
3.1.2. Task Scheduling Model	7
3.2. Constraints	7
3.2.1. Equations.....	7
3.2.2. Inequalities	7
3.2.3. Bounds	8
4. Detailed Discussion	9
4.1. Input Data.....	9
4.1.1. Parameters.....	9
4.1.2. Input Variables.....	13
4.1.3. Independent Variables	22
4.2. Decision Variables	22
4.2.1. Proportion of Solar Power Harnessed.....	22
4.2.2. Battery Power Charged/Discharged.....	23
4.2.3. Task Delays.....	23
4.3. Algorithms	23
4.3.1. Implementation Platform	23
4.3.2. Tariff-Proportional Algorithm	24
4.3.3. Genetic Algorithm	27
5. Results & Discussion	31
5.1. Caveats.....	31

5.2. Procedure	32
5.2.1. Genetic Algorithm	32
5.2.2. Tariff-Proportional Algorithm	33
5.3. Decisions Made	33
5.3.1. Proportion of Solar Power Harnessed.....	33
5.3.2. Battery Power Charged/Discharged.....	35
5.3.3. Task Delays.....	37
5.4. Main Aspects.....	39
5.4.1. Average Cost per Hour in a Day.....	39
5.4.2. Peak-to-Average Ratio Power.....	40
5.5. Other Aspects	44
5.5.1. Users' Dissatisfaction	44
5.5.2. Computational Time	45
6. Conclusion & Recommendations	46
6.1. Results Conclusion.....	46
6.2. Limitations & Recommendations	47
6.3. Code Repository & Video Presentation Links.....	47
7. References.....	48
8. Appendix.....	51

List of Figures

Figure 1.1. Smart Grid Architecture	1
Figure 1.2. Duck- Curve [14].....	2
Figure 2.1. Solar-battery System Model	6
Figure 2.2. Task Scheduling System Model	7
Figure 3.1. Power Categories and Percentage of Consumption.....	12
Figure 3.2. Solar Power Harnessed.....	15
Figure 3.3. Comparing the Solar Power Harnessed to the Power Demand	15
Figure 3.4. Comparing Relative Rates for Buying from Grid to the Relative Power Demand	17
Figure 3.5. Tariffs for Buying and Selling.....	18
Figure 3.6. Comparing Relative Rates for Selling from Grid to the Relative Power Demand	19
Figure 3.7. Tariff-Proportional Algorithm Decision Making	26
Figure 3.8. Optimisation Process	28
Figure 4.1. Comparing Proportion of Solar Power Harnessed Siphoned to Battery	35
Figure 4.2. Comparing Battery Charged/Discharged	36
Figure 4.3. Delays of Tasks in Bar Chart.....	38
Figure 4.4. Effect of Task Scheduling Alone	41
Figure 4.5. Effect of Harnessed Solar Power.....	42
Figure 4.6. Effect of Battery Power	43
Figure 4.7. Utility Function	44

List of Tables

Table 1.1. Parameters.....	viii
Table 1.2. Input variables.....	viii
Table 1.3. Independent Variables	viii
Table 1.4. Decision Variables.....	viii
Table 1.5. Totals	viii
Table 2.1. List of Tasks and Parameters	9
Table 2.2. Harnessed Solar Power	14
Table 2.3. Tariff for Buying from Grid.....	17
Table 2.5. Battery Specifications [33], [34].....	20
Table 2.6. Set of Tasks.....	21
Table 2.7. Settings.....	29
Table 2.8. Stopping Criteria.....	30
Table 3.1. Proportion of Solar Power Harnessed.....	34
Table 3.2. Battery Power Charged/Discharged.....	36
Table 3.3. Task Delays.....	37
Table 3.5. Average Costs per Hour in a Day	39
Table 3.6. Peak-to-Average Ratio Power	40
Table 3.7. User Dissatisfaction	45
Table 3.8. Computational Time	45

Abbreviations

Table 1.1. Parameters

Task duration time	r_i
Task deadline time	d_i
Task arrival time	a_i
Task power demand	p_i

Table 1.2. Input variables

Power harnessed from solar panels	$S(t)$
Power level of battery	$B(t)$
Tariff for buying from the grid	$C(t)$
Tariff for selling to the grid	$G(t)$
Set of tasks	$n(t)$

Table 1.3. Independent Variables

Time slots	t
Tasks	i

Table 1.4. Decision Variables

Proportion of harnessed solar power stored in batteries	$k(t)$
Power to charge battery (from grid), power discharged from battery (to tasks)	$b(t)$
Task delay time	s_i

Table 1.5. Totals

Total number of time slots	T
Total number of tasks	N
Total number of tasks in time t	$N(t)$
Total power demand in time t	$W(t)$

1. Introduction

Smart grids are electric grids that incorporate two way communication between the power supplier or power distributor and the user [1]. A smart grid uses sensors and computers to automate power transmission [1]. The smart grid can be divided into three main aspects. These are from the aspect of the power supplier and power distributor and the ways that they could optimise the use of their power generators. The next aspect considered is the cooperation between the users and the power distributor (assuming that users can harness solar power and store power) to optimise the use of power (ref.). Lastly, the smart grid can be studied from the aspect of the individual user. In this project, the smart grid will be investigated from the third aspect which is from the individual user's aspect. Figure 1.1. shows these different aspects clearly.

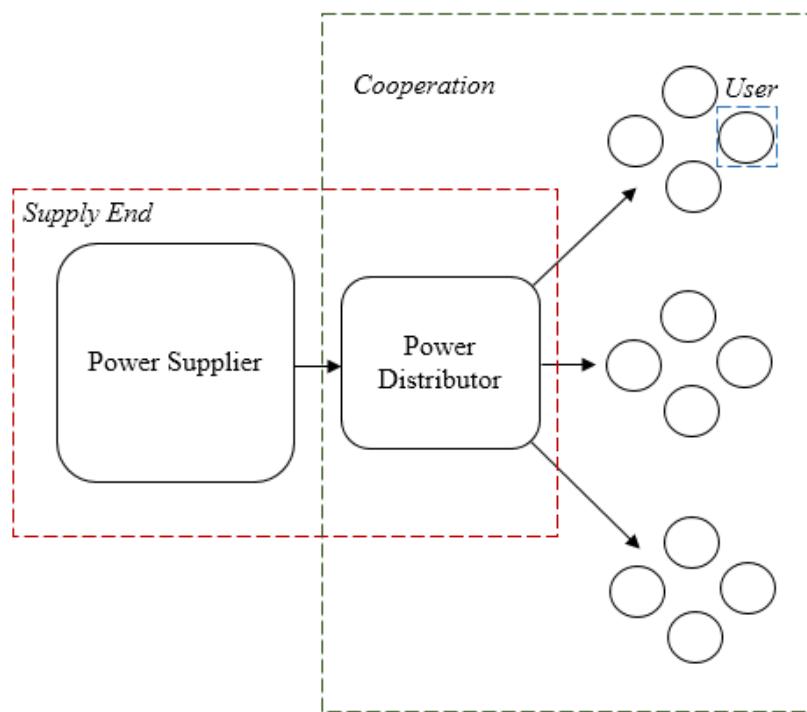


Figure 1.1. Smart Grid Architecture

Compared to a traditional grid, the smart grid monitors the power demands of the users which varies throughout the day. The monitoring is done using Smart Meters [1]. This is useful as the smart grid can try to predict the volume of power demand in the near future. Depending on the prediction, the smart grid can then decide on the most appropriate use of power generators. Turning on power generators is expensive and fluctuating power demands mean that power generators have to be turned on and off frequently to meet the power demands. The ability to predict future power demand would allow the power supplier to make an optimised decision on using the power generators. The prevalence of users harnessing solar power also has a major effect on the grid. Since solar power peaks during the day when power demands are not peaking, there exists a greater dip in demand just before the

peaking in power demand at night. This is known as the ‘duck-curve’ due to the shape of the curve as shown in Figure 1.2. [2]. This increases the peak-to-average ratio (PAR) of the power demand. It is advantageous for the power supplier to have to supply constant power throughout the day as this would bring down transmission costs [1]. To circumvent the ‘duck-curve’, most solar panel installers offer the installation of batteries as well. This reduces the PAR of power demand which reduces the burden on the power supplier and allows for more efficient power distribution which is essential to sustainability [2]. Hence, this project investigates the user’s ability to take advantage of installed solar systems and battery storage to reduce the PAR power that the user needs from the grid.

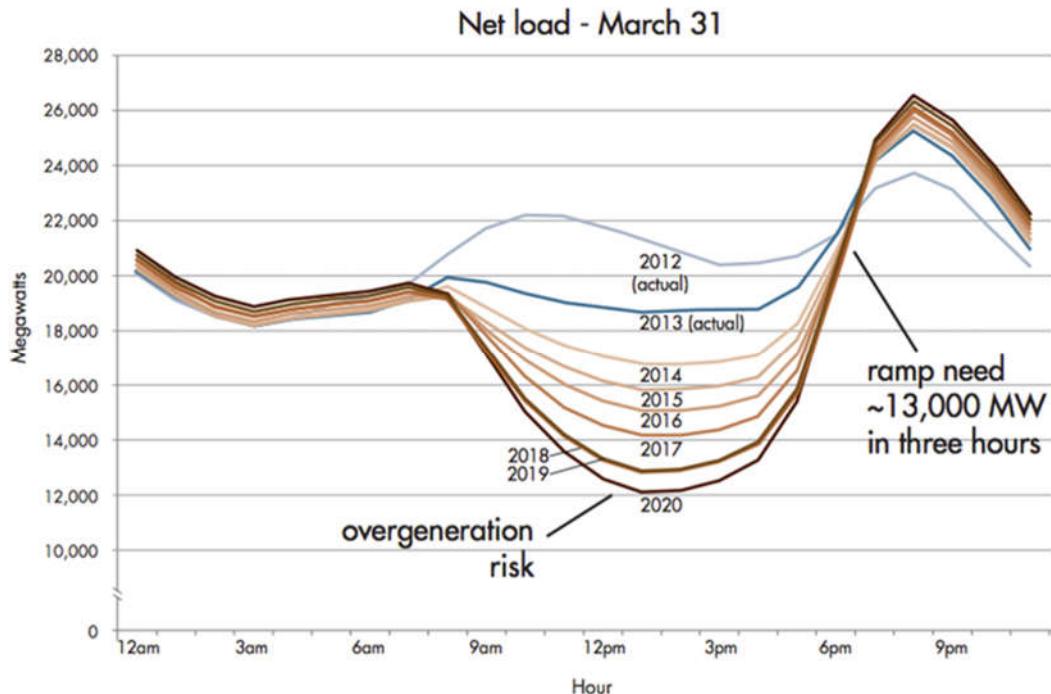


Figure 1.2. Duck- Curve [14]

From the user’s aspect, it is most beneficial to them that they the rates for buying power from the grid goes down. So if using harnessed solar power and battery storage would reduce the cost spent by the user, the user would be interested to invest in these products. The power supplier would also have a lower operating cost as the use of the power generators will be more efficient. This would benefit both the power supplier and the user.

Since smart grids should be able to operate automatically based on decisions made by computers, the use of algorithms to optimise the decision making needs to be implemented in all aspects of the smart grid. In this project, the use of algorithms on the user-end of the smart grid is being investigated. The algorithms implemented should minimise the average cost per hour in a day spent by the user and indirectly lower the PAR of power demand by the user. The project focuses more on minimising the average cost per hour in a day as it is assumed that by minimising the cost, the user’s PAR power would be minimised as well. This is because the consumption of power at peak demand times incur a higher cost to the user as the rates are set to be higher during peak demand times. To minimise the cost, the algorithm designed has to shift the loads to non-peak demand times as the rates are lower which also reduces the PAR of power consumed by the user.

The varying rates set by the power distributor is a form of Demand Side Management (DSM), which is the management of the customer's power demands [3]. Varying rates are meant to encourage users to shift their power demands [1]. In a smart grid, the two-way communication links between the power supplier and the user will inform the Smart Meter installed by the user on the varying rates [1].

The algorithms investigated in this project are designed to take advantage of the harnessed solar power, the battery storage and the varying rates set by the power supplier. The algorithms then make the decision on the proportion of solar power harnessed that is siphoned to the battery storage, the amount of power to store in the battery from the grid or the amount of power to consume from the battery and the delays that the tasks (electrical appliances) should have. The main objective of these algorithms are **to minimise the average cost per hour spent by the user and to lower the PAR of power consumed**. The algorithms are designed with the objective of minimising the average cost per hour in a day. The PAR of power demand is simply compared and is not part of the objective function.

2. Literature Review

A number of literatures on the subject of smart grid algorithms are reviewed to investigate the works done prior to this project. The literatures investigated the use of algorithms on the different aspects of a smart grid. They investigate the use of algorithms in the aspect of the interaction between the power supplier or power distributor and the user, in the aspect of cooperation of users in a smart grid and in the aspect of the individual user. All the literatures reviewed has aims to either minimise the peak-to-average ratio (PAR) of power demand or to minimise the cost or both. The literatures make use of task scheduling, solar power and battery storage, cooperation between users or a combination of these things.

Firstly, there are works done on the methods power can be supplied more efficiently by interaction between the power supplier and the user. Doulamis [4] suggests employing spectral clustering to assign tasks to different processors whilst maintaining a certain level of user satisfaction. Mohsenian-Rad [5] focuses more on reducing the aggregate PAR power by clumping a group of users together and basing the power distributor's decision on the aggregate power of the group. This method is used because simply using Real-Time Pricing (RTP) on the individual user may cause a shift of the peak instead of actually reducing the PAR power. Game theory is the approach employed by Mohsenian-Rad [5] for task scheduling by the users. Samadi [6] investigates the interaction between the power supplier and the user to optimise the consumption of the users to fit the supply of power. Samadi [6] also encourages cooperation between users and aims to maintain a certain level of user satisfaction. These literatures address the issue of using power efficiently from the supply side through task scheduling. These literatures claim that communication between the power supplier or power distributor and the user is beneficial to both parties in minimising cost and PAR power.

Some of the literatures investigated the benefits of cooperation between users in the smart grid. Guo [7] and Tsaousoglou [8] investigate the benefits of cooperation between users with access to harnessed solar power and battery storage. Basically, these users are clumped together in Micro Grids [8] where the battery storage is shared and the harnessed solar power can be shared as well. The literatures show that cooperation in smart grid is beneficial to the users in minimising cost spent. In both literatures by Guo [7] and Tsaousoglou [8], the decisions are made by the power supplier or the power distributor. Guo [7] employs a Lyapunov optimisation method to make the decisions whereas in Tsaousoglou [8], a list of simple deterministic algorithms are compared. In the literature by Guo [7], task scheduling is used but the literature does not investigate the ability to sell power back to the grid whereas Tsaousoglou [8] does. These literatures claim that cost spent by the users can be minimised with cooperation between the users and with the power supplier or distributor.

The literatures that have models that are more similar to the model employed in this project are the literatures that focus solely on the individual user end of the smart grid. Logenthiran [9] and Burcea [10] focus on task scheduling using different algorithms but with different objectives. Logenthiran [9] aims to minimise the PAR of power demand whereas Burcea [10] aims to minimise the cost spent by the user. Albalas [11] investigates the use of Genetic Algorithm in task scheduling but not in the decisions involving the battery. Albalas [11] uses very simple decision making for the use of the battery storage. Chen [12] and Mary [13] investigate the use of harnessed solar power and battery storage for the individual user to

minimise their cost spent. In the literature by Chen [12] however, task scheduling is also a part of the model whereas in the literature by Mary [13] task scheduling is not part of the model. The literature by Chen [12] compares the results of a model with the ability to sell back to the grid with a model that does not. One literature [12] employs the Lyapunov optimisation algorithm whereas the other [13] employs Genetic Algorithm to solve its optimisation problem. These literatures claim that employing the use of task scheduling, using harnessed solar power and battery storage or both would minimise the cost spent by the user or minimise the PAR of power demand.

Most of these works either focus solely on task scheduling or using harnessed solar power and battery storage. In the literature by Albalas [11], the model implemented only employs task scheduling and battery storage but does not consider using harnessed solar power. In the literature by Chen [12] however, the model implemented employs task scheduling, harnessed solar power and battery storage. This [12] is the only literature that has all of these three components. Most of these literatures do not include the ability to sell back to the grid except for a few [8], [12]. The ability to sell back to the grid may significantly affect the resulting costs. The literature by Mary [13] does not implement this but instead, the excess harnessed solar power is simply wasted. This project's model is based mainly on the literature by Chen [12] as it has all the components that this project uses in its model. Most of these works reviewed also uses stochastic optimisation algorithms that are based on probabilities. Only one literature [8] looks at using simple and deterministic stepwise algorithms to make the decisions. Not many of these works consider the utility function to determine the user's satisfaction. Only a few literatures [4], [6], [12] consider using the utility function to gauge user's satisfaction. This especially important from the power supplier's or power distributor's end as they will want to keep customers satisfied. Hence, the utility function is not really part of the objective function in this project but it is still investigated from the results. A lot of work has been done on smart grids and the works reviewed here are only a fraction of the sum of all related works that have been published.

The design choices of this project are made based on these reviewed works. The use of a simple and deterministic stepwise algorithm such as the Tariff-Proportional Algorithm has not been done in any of the works reviewed except for one [8]. The choice of using Genetic Algorithm in a model similar to the model designed in the literature by Chen [12] has not been done in any of the works reviewed. The ability to sell to the grid has only been investigated by so few works [8], [12] but may be a big contributor to the results. So most of the design choices in this project are chosen because they have not been done in such a manner before. All of these literatures are available in the Appendix.

3. Overview

The list of notations used for the parameters and variables can be seen in Table 1.1, Table 1.2., Table 1.3. The parameters in Table 1.1., are used to set the properties of the tasks hence, they differ from task to task. The parameters have the subscript i to show that they are a parameter of the task i . The input variables in Table 1.2. are time-dependent hence, they change depending on the time slot that they are in. The total number of time slots is 24 hours and is denoted by T (shown in Table 1.5). These variables are denoted with a t to show that they are a function of time. The time slot t starts from 8 a.m. and ends at 7 a.m. the following day. The variables t and i shown in Table 1.3. are considered independent variables as they do not depend on any other variable.

3.1. System Model

There are two parts to the system model. These are the solar-battery part and the task scheduling part. They are the base and structure for deciding the values of the decision variables. The system model also indicates the components that are considered for the design of the algorithms.

3.1.1. Solar-Battery Model

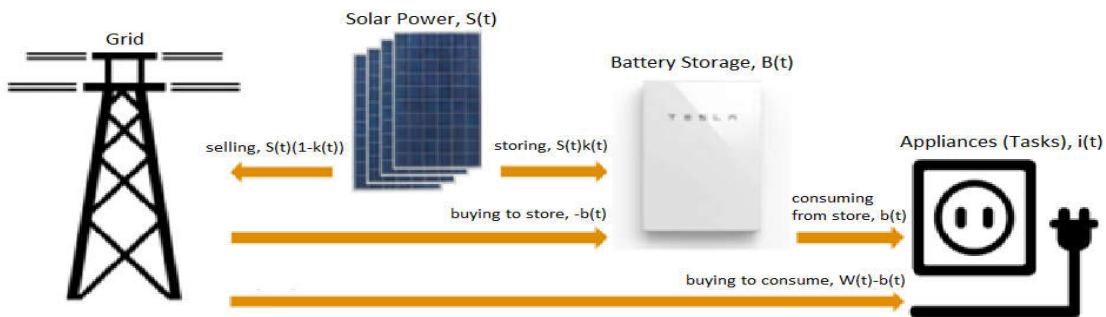


Figure 2.1. Solar-battery System Model

The solar-battery system model is based on the model used in the literatures by Chen [12] and Mary [13]. This model is a representation of the ability of the user to take advantage of harnessed solar power and battery storage. As can be seen from Figure 2.1., the solar-battery system model has four main components which are the grid power, harnessed solar power, battery storage and the electrical appliances (tasks) demanding the power. The decisions made are directed as shown with the arrows in Figure 2.1. Some decisions are mutually exclusive hence, only one variable is needed to identify the decision that affects two components. The first line of arrows are decisions based on the value of decision variable $k(t)$ which is the proportion of harnessed solar power to be stored in the battery. The decision to sell harnessed solar power is based on $k(t)$ as the proportion of excess solar power is identified as $1-k(t)$. On the second and third lines of arrows the decisions are based on $b(t)$ which is the battery power consumed by the tasks. If the battery power is not being consumed, then power is being bought from the grid to store in the battery as $-b(t)$. The remaining power demand is satisfied by buying power from the grid as $W(t)-b(t)$ where $W(t)$ (shown in Table 1.5.) is the total power demand for that time slot. This model only has two

decision variables that are $k(t)$ and $b(t)$ and does not include the third decision variable for task scheduling.

3.1.2. Task Scheduling Model

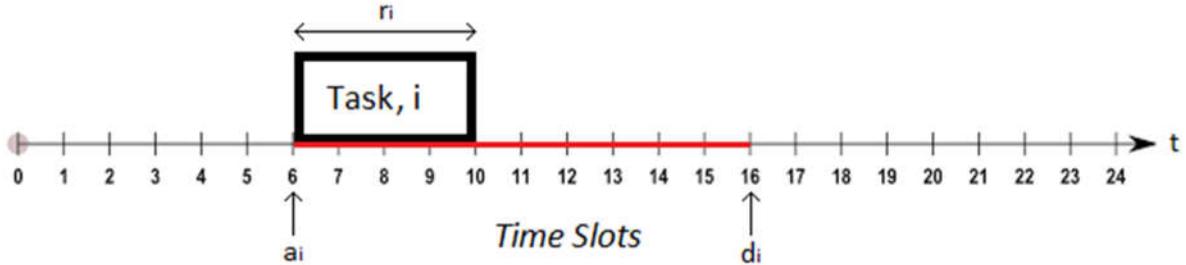


Figure 2.2. Task Scheduling System Model

The task scheduling model is based on the one used in the literature by Chen [12]. The model represents the ability to shift tasks to different time slots. As can be seen in Figure 2.2., the three parameters needed for this model are the r_i , a_i and d_i which are the duration of the task, the arrival time of the task and the deadline for when the task should be completed. The set of tasks $n(t)$ arrive in time slot t where i is a member of $n(t)$. The total number of tasks in $n(t)$ is denoted by $N(t)$ (shown in Figure 1.5). In the Figure 2.2., the set of tasks is $n(6)$ where task i has the parameters a_i as 6, r_i as 4 and d_i as 10. The entire task must be scheduled to a time slot on the red line. More than one task may arrive at one time slot and it is up to the algorithm to redistribute these tasks within their constraints.

3.2. Constraints

The constraints are the upper and lower bounds of the decision variables and the inequalities and equations that govern them. The constraints are important to prevent the decision variables from diverging to infinity or producing illogical values.

3.2.1. Equations

$$B(1) = B_{max}/2 \quad (i)$$

$$B(t + 1) = B(t) + k(t)S(t) - b(t) \quad (ii)$$

The equations (i) and (ii) are used to determine the battery levels at the beginning of each time slot. The battery power level is assumed to be half of the maximum battery capacity at the beginning of the day as shown in equation (i). The battery power level thereafter is based on equation (ii). The second term in equation (ii) is the harnessed solar power siphoned into the battery storage and the third term is the battery charged with power from the grid or battery power discharged to be consumed by the tasks. The term $b(t)$ is negative when power bought from the grid is charging the battery. The decision variables are needed to feed the equation (ii).

3.2.2. Inequalities

$$B(t) \geq 0, B(t + 1) \geq 0 \quad (iii)$$

$$b(t) - k(t)S(t) \leq B(t), k(t)S(t) - b(t) \leq B(t + 1) \quad (iv)$$

The inequalities (iii) are used to maintain the battery power levels above zero as it is not possible for the battery power levels to be negative. The inequalities (iv) are derived from equation (ii). The decision variable need to abide to these inequalities as they make up the terms of the inequalities.

3.2.3. Bounds

$$0 \leq k(t) \leq 1 \quad (\text{v})$$

$$|b(t)| \leq b_{max} \quad (\text{vi})$$

$$0 \leq s_i \leq d_i - r_i \quad (\text{vii})$$

These bounds are used to ensure that the values of the decision variables are kept within a reasonable range.

The bounds (v) set the range for $k(t)$ which is the proportion of harnessed solar power to be siphoned to the battery storage. $k(t)$ must be within zero to one because it is a percentage. $k(t)$ cannot be a negative value as that would mean that the solar panels are absorbing electric power and $k(t)$ cannot be greater than one as that would mean that more power is being siphoned to the battery storage than the solar panels actually harness.

The bounds (vi) set the range for $b(t)$ which is the battery power discharged to the tasks or if $b(t)$ is negative, the battery power charged from the grid. b_{max} signifies the maximum that the battery can charge or discharge in an hour. Hence, the maximum $b(t)$ that can be discharged to the tasks in an hour is b_{max} and the maximum $-b(t)$ to charge the battery is $-b_{max}$.

The bounds (vii) set the range for s_i which is the delay for the task i . s_i cannot be less a negative value as a negative delay is not applicable for this model. The task can only be delayed but not brought forward. s_i cannot be greater than the deadline d_i and since it takes a duration r_i for the task to run, the latest time that the task can start is $d_i - r_i$ as can be seen in Figure 2.2.

4. Detailed Discussion

This section explains the details of the algorithms. A detailed explanation of the parameters used, input variables used and decision variables expected from the algorithms is also available in this section.

4.1. Input Data

The input data represent the data that the algorithm needs to produce results. The input data are split into three categories which are the parameters, the input variables and the independent variables. A more detailed explanation on their differences will be given in their respective sections.

4.1.1. Parameters

The parameters are used to define the properties of the electrical appliances (tasks). The parameters vary from task to task. They are kept constant throughout the project and do not vary with time. The parameters for the tasks can be seen in Table 2.1. The values given for the duration time (r_i), deadline time (d_i) and arrival time (a_i) for the tasks are based on reasonable estimates whereas the values for the power rating of the tasks (p_i) are taken from “Energy Use Calculator” [15] which gives an average power rating for the task.

Table 2.1. List of Tasks and Parameters

i	Task	r_i	d_i	a_i	p_i (kW)	Total Power Consumed (kW)	Percentage of Power Consumed (%)
1	Dryer	1	6	19	3.000	3.000	7.245%
2	Washing machine	1	6	19	0.500	0.500	1.207%
3	Oven	1	3	17	2.400	2.400	5.796%
4	Dish washer	1	5	19	1.800	1.800	4.347%
5	Microwave	1	4	11	0.600	0.600	1.449%
6	Space heater	5	9	17	1.500	7.500	18.112%
7	Air-conditioner	3	6	10	1.000	3.000	7.245%
8	LCD TV (22”)	5	7	18	0.030	0.150	0.362%
9	Laptop (15”)	6	7	8	0.060	0.360	0.869%
10	Water heater	3	4	8	4.000	12.000	28.979%
11	Fridge	24	24	8	0.180	4.320	10.432%
12	Freezer	24	24	8	0.200	4.800	11.591%
13	CFL Lights	7	7	17	0.140	0.980	2.367%

4.1.1.1. Tasks

The tasks chosen in Table 2.1. are based on the Department of Industry, Innovation of Science Australia's booklet "Energy Use in the Australian Residential Sector" [16]. These are some of the more common appliances that households have. The tasks listed in Table 2.1. may not all be turned on in the same day. This set of tasks represent a worst case scenario where all these tasks are switched on in the same day. The list of tasks also does not differentiate with the seasons and a day is assumed to run all of these tasks for all its duration. The season-dependent tasks such as the air-conditioner and the space heater are switched on in the same day assuming warm weather during the day and colder weather during the evenings. This may lead to some inaccuracies in the results.

4.1.1.2. Duration Time

The duration times that these tasks take are measured in hours hence, an hour is the shortest possible time that a task can be running for. This is for the sake simplicity in designing the algorithms. Tasks that do not actually take an hour or more will have their power ratings reduced to compensate for the total power that said task demands. The microwave only takes a duration of half an hour in the worst case scenario, so the power rating for the microwave is actually halved [17]. The microwave is the only task that has its power rating halved. Other tasks such as the dryer is assumed to run for an hour even though the "Energy Use Calculator" states that a dryer only runs for 15 minutes a day [18]. The duration given by the "Energy Use Calculator" represents the use of a dryer for 52.5 minutes a day, twice a week [18] but for the worst case scenario, the dryer is assumed to run for one hour each day. The same goes for the washing machine [19]. These two laundry appliances are usually used only once or twice in a week but for the sake of simplicity and worst case scenario they are assumed to run each day. The duration of lights turned on is extended to 7 hours instead of the 5 hour duration suggested by "Energy Use Calculator" [20]. This to account for the worst case scenario where the user may very likely switch on their lights for more than just 5 hours. All other tasks have durations suggested by "Energy Use Calculator" [15]. The durations are set for the worst case scenario but this does not reflect real-world scenarios which may result in some inaccuracies in the results obtained.

4.1.1.3. Deadline Time

The deadline for all the tasks is the number of time slots from the arrival time that the task must be complete by. This is to ensure that the tasks are not unreasonably delayed. If there are no deadlines in place, the tasks will all be delayed to the time of day with the lowest rates for buying from the grid. Delaying the tasks with no deadline is also unrealistic as the user will definitely not agree to delay the tasks for potentially long periods of time.

The tasks consist of a mixture of elastic and inelastic tasks. Elastic tasks are simply tasks that have flexible deadlines (can be shifted) whereas inelastic tasks are tasks that have hard deadlines (cannot be shifted). From Table 2.1., these inelastic tasks are tasks 10, 11 and 12 as they have hard deadlines (the duration time of the task is the same as the deadline time). These tasks cannot be shifted as the users will need these tasks switched on right away and cannot afford to delay it. The fridge and freezer cannot be turned off at any time of the day as these are needed for the preservation of food. The lights have to be turned on immediately as the users will definitely not agree to delay such an important task.

The deadlines are decided based on the type of task. Tasks that are laundry-related such as the dryer and washing machine that arrive at night can be delayed to the time before the user goes to bed. Tasks that are kitchen-related such as the microwave or oven will have shorter deadlines because the user needs to have their meal within the meal times. Although the dishwasher is a kitchen-related task but the dishwasher has an extended deadline since the user will only need to wash the dishes before the day ends. The heating and cooling tasks may have long deadlines but this is due to the tasks long durations as well. These heating and cooling tasks can only be extended for the period of time that the user may agree to. This depends very much on the user's tolerance to indoor climate changes. For this project, the delays for these climate control tasks are chosen based on the author's lifestyle.

Entertainment-related tasks such as the TV needs to be met as soon as possible as the user will definitely want to use entertainment devices as soon as the feeling to do so comes to the user. The laptop which may be used for work, is a task that needs to be met as soon as possible, so short deadline of is given. The water heater is a task that must be switched on almost immediately as this is needed for the user to attend to their daily routines such having a shower. The deadlines set are very subjective and based largely on the author's experiences and lifestyle.

4.1.1.4. Arrival Time

The arrival times are the time slots that the task is needed to be used. The arrival times for the tasks are based on a user's daily routine. For the values in Table 2.1., the arrival times are based on the author's lifestyle. Based on the author's lifestyle, the user does not go to work and is assumed to be in the household for the entire day. The refrigerator and the freezer are tasks that arrive at the beginning of the day as these tasks run for the entire day. The user begins the day with a shower and using the laptop. Hence, these two tasks arrive at the beginning of the day. The next task to arrive is the air-conditioner which the user will want to use during that time of day, if it is a hot day. Then, in the afternoon the task that arrives is the microwave since lunch is around that time of day. The user could be using either a microwave or an oven during mealtimes so for the sake of simplicity, the user needs to use the microwave during lunch time and the oven during dinner time. The entertainment-related tasks arrive during the evening when the user will probably want to wind down and relax. The user will want to use the heater in the evening as the temperature drops. Almost all the other tasks arrive sometime in the evening causing the peak in power demand that is common in most households.

4.1.1.5. Power Demand

The power ratings for all the tasks are obtained from the “Energy Use Calculator” [15]. As mentioned earlier. Most modern washing machines are front-loading as this is a more efficient system [19]. The type of washing machine indicated in Table 2.1. is a modern front-loading one. The power rating of the microwave is altered because the microwave only runs for 30 minutes in the worst case scenario but since the duration of tasks uses one hour as its smallest denominator, the power rating is halved so that the total power consumed from the microwave remains the same [17]. Power ratings of air-conditioners usually depend on the size of the room. The size of the room the air-conditioner and space heater are in is assumed to be the size of a single room [21], [22]. The power rating of a laptop varies from 20 W to 100 W depending on screen size so, the power rating for the average 15-inch screen size is used [23]. The refrigerator and the freezer are assumed to be two separate devices although most modern refrigerators come with a built-in freezer [24]. The power ratings for both these appliances are based on large modern refrigerators [24] and modern freezers which are a lot more efficient than older models [25]. The power rating for the lights is actually the power rating for a combination of ten CFC light bulbs which are a lot more efficient than their incandescent predecessors. The number of light bulbs is taken as the average between the sources “Energy Use Calculator” [20] and Origin Energy’s “Understanding Your Energy Usage” [26].

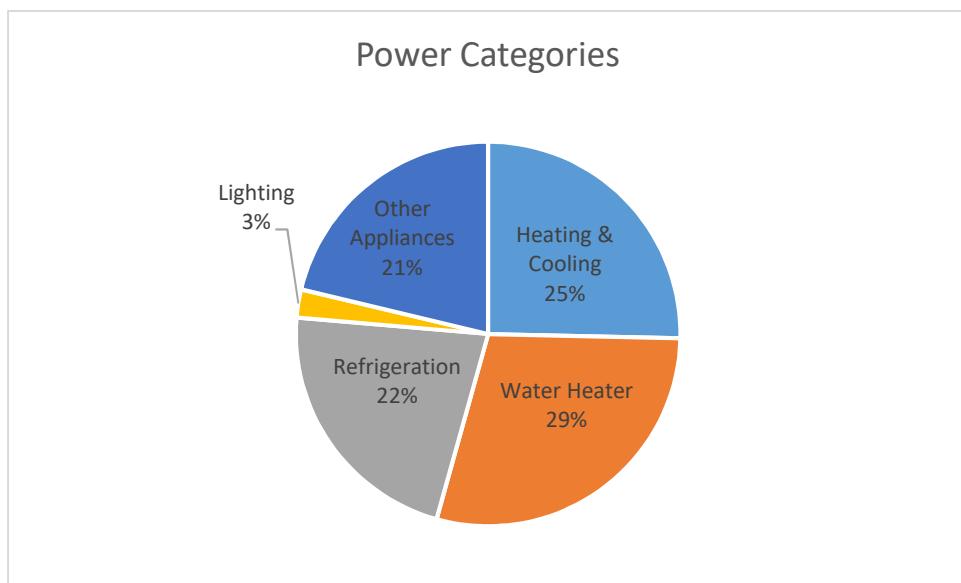


Figure 3.1. Power Categories and Percentage of Consumption

The total power demand for each task and the percentage of that power demand is shown in Table 2.1. The task with the greatest demand is the water heater followed by the space heater and the refrigeration appliances. There are newer water heaters that run at half the time but the water heater indicated in Table 2.1. is of an older and less efficient make [27]. Figure 3.1. shows a clearer depiction of which type of household appliance consumes the most power. Figure 3.1. is a sensible depiction as the household power demand is presented in a similar figure by the Australian government’s guide to a sustainable home [28]. The total power demand in a day is **41.410 kW**.

4.1.2. Input Variables

The input variables are simply variables that vary with time and are used as inputs to the algorithms. The input variables are independent of each other and the values for each variable is decided beforehand except for the battery power level which depends on the equation (ii). The input variables include the power generated from the solar panels, the battery power level at the beginning each time slot, the tariff for buying from the grid, the tariff for selling to the grid and the sets of tasks arriving in each time slot.

4.1.2.1. Solar Power Harnessed

As households with installed solar systems become more prevalent, the need to include harnessed solar power in this project's model is apparent. Not only does the use of solar systems in households reduce the cost that the user spends on electric power but it also pushes for a greener, more sustainable future where reliance on the grid is lessened and the use of renewables is widespread. However, the prevalence of solar systems in a traditional grid is disrupting the transmission of power [14]. This may incur losses on the supplying end which will affect the users as the power suppliers will increase the rates they charge [14]. Nevertheless, the government is providing incentives for households to install solar systems by allowing the users to sell back to the grid at a feed-in tariff [29]. The ability to sell the harnessed solar power and the ability to use the harnessed solar power for direct consumption will lead to reducing the cost spent by the users which is the main reason harnessed solar power is one of the input variables.

The solar system used is a **1.5 kW** rating solar system as this is one of the most common installation types in Australia [30]. This means that the maximum power output from the solar panels is 1.5 kW in an hour. The percentage output from the solar system varies during the day and it also varies with different days. The percentage is a fraction of its maximum rating which in this project is 1.5 kW. In a country like Australia where there are four seasons in a year, the intensity of sunlight varies by quite a margin from a summer day to a winter day. Hence, the percentage output of the most intense summer day (6th January 2018) and the output of one the less sunny days in winter (1st June 2017) is averaged and used as the input data for solar power harnessed [30]. The location of this data is set to be in Victoria [30]. The Table 2.2. shows the expected solar power harnessed at the different times of day for the two days mentioned above. The average harnessed solar power is plotted against the time slots in the day as shown in Figure 3.1.

As can it be seen in Table 2.2. and Figure 3.2., the maximum solar power is harnessed at some time between 11 a.m. to 3 p.m. where the average power is above 0.5 kW. This is expected since the sun is at its zenith so the solar power received should be the highest. The mean solar power harnessed in a day is shown in Figure 3.2. to be about 0.2 kW which is a bit less than half of the peak value. This is probably due to the long hours of no solar power during the night and early in the morning. The difference in solar power harnessed between the two days is quite a large. It is 0.8025 kW on the peak summer day and only 0.1485 kW on a winter day. This is a decrease in solar power harnessed of 82%. This major difference is not considered in this project for simplicity and only the average power is taken into consideration. Figure 3.3. shows that the magnitude of solar power harnessed is dwarfed by the power demand of the user. Hence, having a meagre 1.5 kW solar system may not be sufficient to make any major difference in power demand from the grid.

Table 2.2. Harnessed Solar Power

Time Slot, t	Harnessed Solar Power, $S(t)$					Average Power (kW)	
	Summer Day (6 th January 2018)		Winter Day (1 st June 2017)				
	Percentage of maximum rating	Power Harnessed (W)	Percentage of maximum rating	Power Harnessed (W)			
8	13	195	1	15	0.1050		
9	30	450	5	75	0.2625		
10	45	675	10	150	0.4125		
11	57	855	15	225	0.5400		
12	64	960	18	270	0.6150		
13	68	1020	17	255	0.6375		
14	67	1005	16	240	0.6225		
15	63	945	11	165	0.5550		
16	54	810	6	90	0.4500		
17	40	600	0	0	0.3000		
18	22	330	0	0	0.1650		
19	10	150	0	0	0.0750		
20	2	30	0	0	0.0150		
21	0	0	0	0	0.0000		
22	0	0	0	0	0.0000		
23	0	0	0	0	0.0000		
0	0	0	0	0	0.0000		
1	0	0	0	0	0.0000		
2	0	0	0	0	0.0000		
3	0	0	0	0	0.0000		
4	0	0	0	0	0.0000		
5	0	0	0	0	0.0000		
6	0	0	0	0	0.0000		
7	2	30	0	0	0.0150		
Total		8025			1485	0.4755	

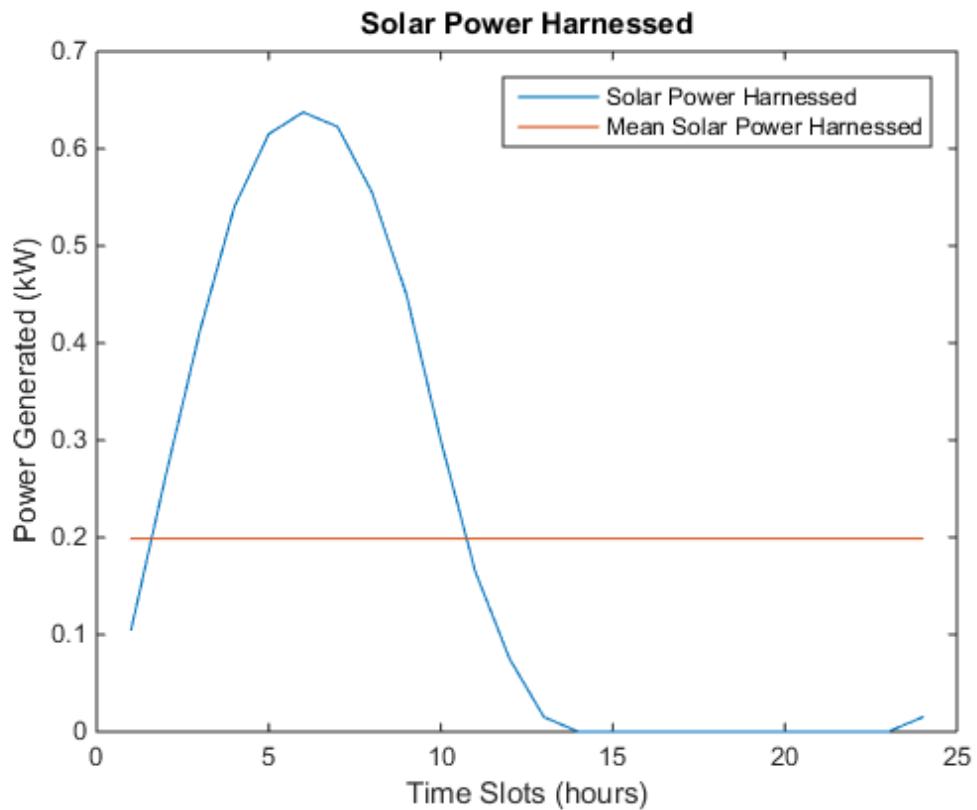


Figure 3.2. Solar Power Harnessed

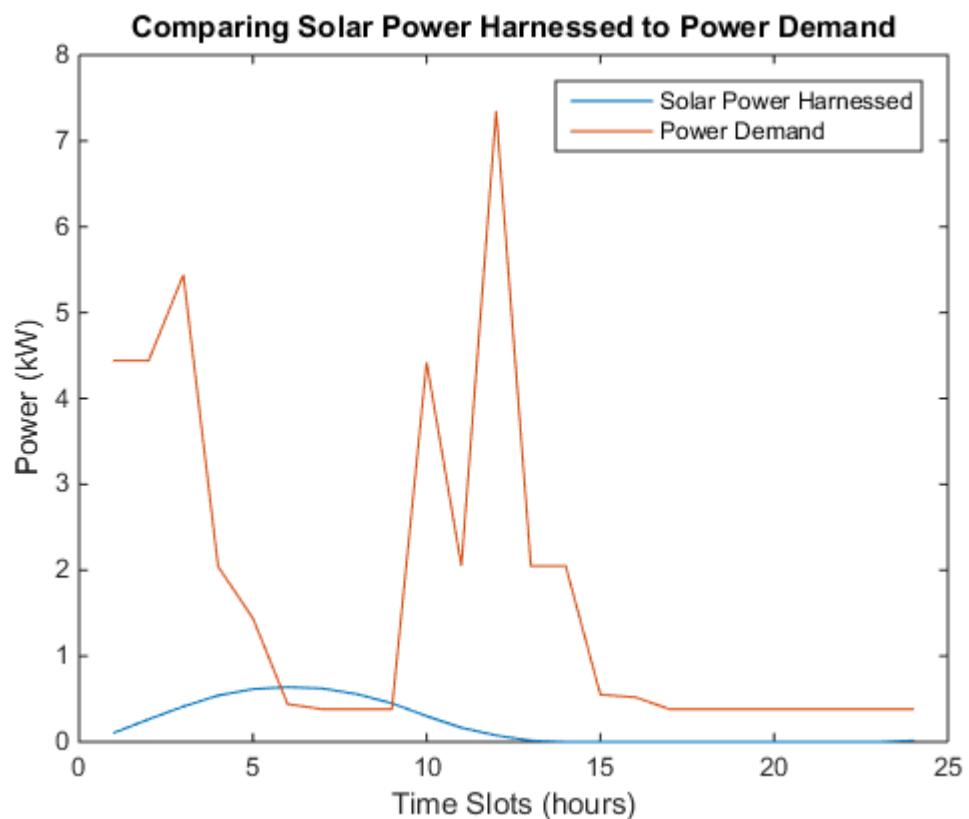


Figure 3.3. Comparing the Solar Power Harnessed to the Power Demand

4.1.2.2. Tariff for Buying from Grid

As a form of Demand-Side Management (DSM), the rates for buying power from the grid varies depending on the time of day to encourage users to schedule their tasks from periods of peak power demand to other times of the day [3]. The tariff is structured such that the rates during the periods of peak power demand are higher than the off-peak periods, encouraging users to shift their loads. The power supplier or power distributor employs DSM to reduce the peak-to-average ratio (PAR) of the power demand. The less the demand of power fluctuates, the lower the operational costs are for the power supplier or power distributor. Hence, it is beneficial to the power supplier that the user shifts their off peak power demand hours to other times. The user can take advantage of this varying rates to minimise their cost spent and at the same time reduce the PAR of power demand.

The tariff for buying power from the grid is provided by Alinta Energy's Victorian electricity pricing [32]. The rates are used in the Jemena Distribution Area and are applicable for both summer and winter days [32]. The rates include GST as well [32]. The rates vary for three different time periods of the day. In ascending order, these are the off-peak usage, shoulder usage and peak usage as shown in Table 2.3. The periods of these varying rates are different during the weekends [32] but those different periods are not considered in this project. The Figure 3.4. depicts a clearer picture of the rates and the power demand at that period.

The relative values of the rates and the power demand in Figure 3.4., are obtained by normalising the actual values with their maximum values. Figure 3.4. shows that the power demand with no task scheduling is quite aligned with the tariff for buying from the grid. This is expected as the higher rates should be during the peak power demand periods and the lower rates during lower power demand periods.

Table 2.3. Tariff for Buying from Grid

Time Slot, t	Period	Rates (cents/kWh)
8	Shoulder Usage	33.462
9	Shoulder Usage	33.462
10	Shoulder Usage	33.462
11	Shoulder Usage	33.462
12	Shoulder Usage	33.462
13	Shoulder Usage	33.462
14	Shoulder Usage	33.462
15	Peak Usage	48.136
16	Peak Usage	48.136
17	Peak Usage	48.136
18	Peak Usage	48.136
19	Peak Usage	48.136
20	Peak Usage	48.136
21	Shoulder Usage	33.462
22	Off-Peak Usage	22.132
23	Off-Peak Usage	22.132
0	Off-Peak Usage	22.132
1	Off-Peak Usage	22.132
2	Off-Peak Usage	22.132
3	Off-Peak Usage	22.132
4	Off-Peak Usage	22.132
5	Off-Peak Usage	22.132
6	Off-Peak Usage	22.132
7	Shoulder Usage	33.462

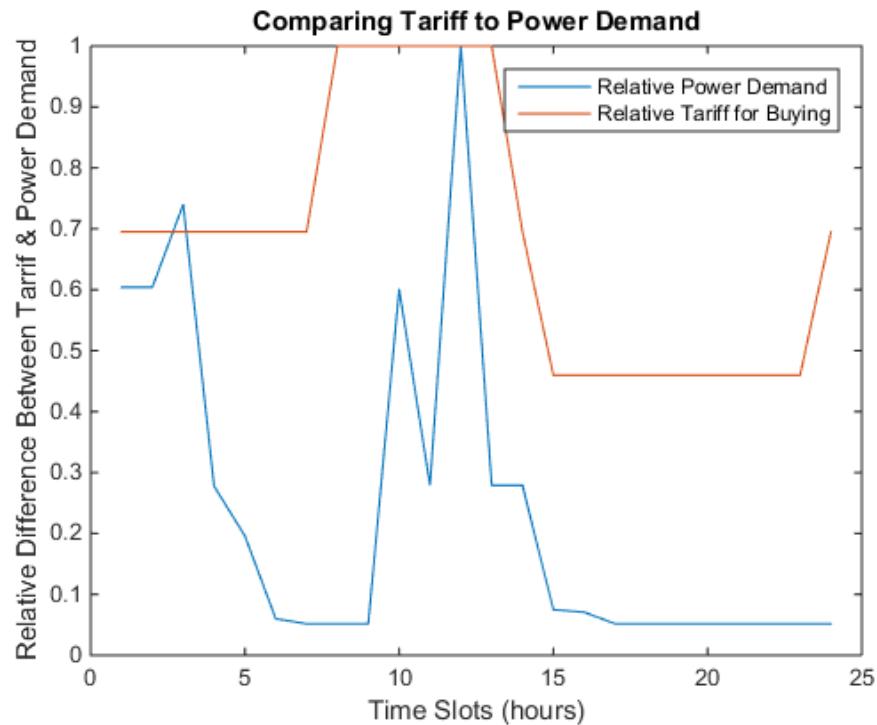


Figure 3.4. Comparing Relative Rates for Buying from Grid to the Relative Power Demand

4.1.2.3. Tariff for Selling to Grid

Many developed nations with widespread use of solar systems in residential households have introduced feed-in tariffs to encourage more people to install solar systems or any form of renewable power source (e.g. wind turbines) (ref.). The feed-in tariffs are the rates set by the grid for users to sell their harnessed solar power back to the grid. There are two types of feed-in tariffs, gross feed-in tariffs and net feed-in tariffs. Gross feed-in tariffs rebates the users for any amount of power that is generated whereas net feed-in tariffs rebate the user based on the excess power generated. In this project, the net feed-in tariff is assumed to fit the system model.

The incentives vary from state to state in Australia as there is no nationwide program for feed-in tariffs [29]. In Victoria, Australia at the time of this project, a fixed rate of 11.3 cents/kW (used to be 5.0 cents/kW before 1st July 2017) is rebated for users of the grid who feeds their harnessed solar power back to the grid [29]. Nevertheless, the feed-in tariff in Victoria is set to change as of 1st July 2018 [29]. This is the tariff that has been chosen for this project. Regardless of the provider that a user subscribes to (this depends on the area a user resides in), the tariff will remain the same. The tariff used for selling back to the grid is as shown in Table 2.4.

As shown in Table 2.4 and Figure 3.5., the feed-in tariff follows a similar trend to the tariff for buying from the grid. The rates peak during peak power demand hours and lowers during non-peak power demand hours as shown in Figure 3.6. This encourages users to sell their harnessed solar power during the peak power demand periods to offset the power demand which will lower the PAR of power demand that the power supplier or power distributor needs to meet. Figure 3.5. also shows that the feed-in tariff is about one third the tariff for buying from the grid which is expected as the feed-in tariff cannot exceed the tariff for buying from the grid.

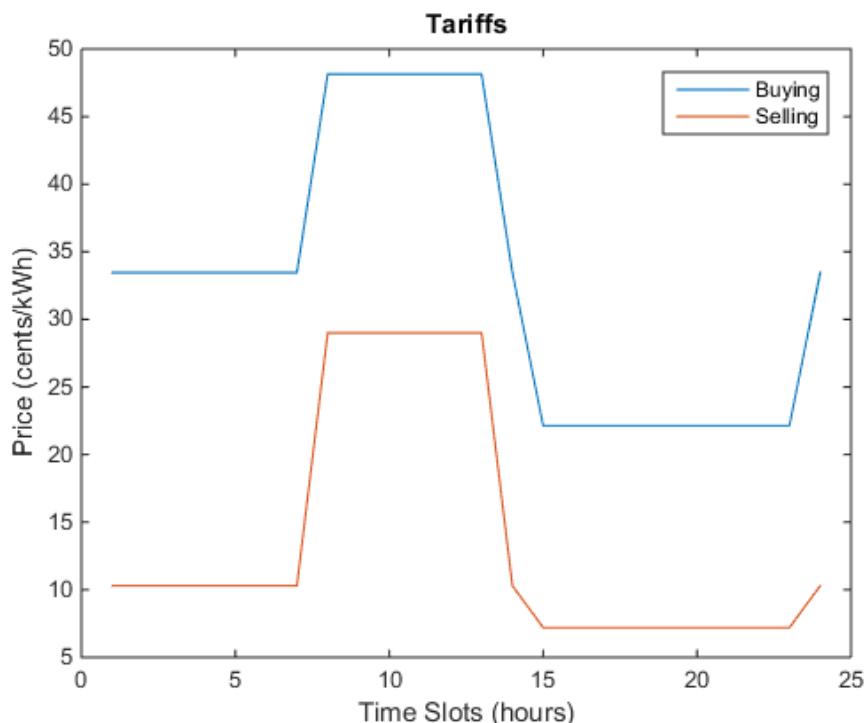


Figure 3.5. Tariffs for Buying and Selling

Table 2.4. Tariff for Selling to the Grid

Time Slot, t	Period	Rates (cents/kWh)
8	Shoulder Usage	10.300
9	Shoulder Usage	10.300
10	Shoulder Usage	10.300
11	Shoulder Usage	10.300
12	Shoulder Usage	10.300
13	Shoulder Usage	10.300
14	Shoulder Usage	10.300
15	Peak Usage	29.000
16	Peak Usage	29.000
17	Peak Usage	29.000
18	Peak Usage	29.000
19	Peak Usage	29.000
20	Peak Usage	29.000
21	Shoulder Usage	10.300
22	Off-Peak Usage	7.200
23	Off-Peak Usage	7.200
0	Off-Peak Usage	7.200
1	Off-Peak Usage	7.200
2	Off-Peak Usage	7.200
3	Off-Peak Usage	7.200
4	Off-Peak Usage	7.200
5	Off-Peak Usage	7.200
6	Off-Peak Usage	7.200
7	Shoulder Usage	10.300

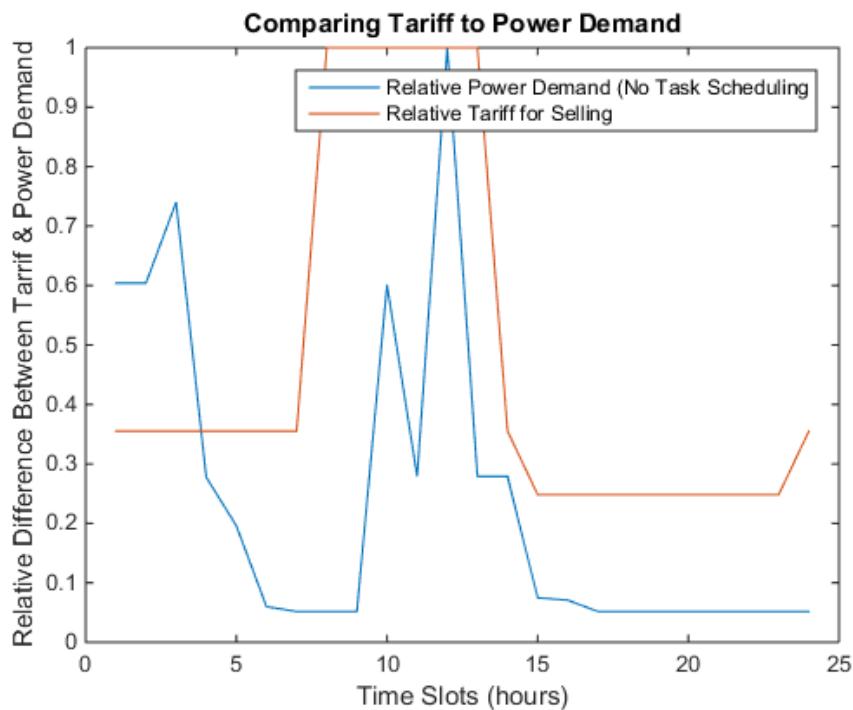


Figure 3.6. Comparing Relative Rates for Selling from Grid to the Relative Power Demand

4.1.2.4. Battery Specifications

The need for a battery storage to be installed together with a solar system is becoming common practice. Since solar power can only be harnessed during periods with intense sunlight (daytime), the battery storage will be a useful addition as the power harnessed during these peak solar output hours (11 a.m. to 3 p.m. as shown in Figure 3.2.) can be stored and used during the later parts of the day when the power demand peaks and the peak solar output dips. The addition of a battery will also lower the peak-to-average ratio (PAR) of power consumed from the grid. Using a battery may circumvent the ‘duck-curve’ [2]. The ‘duck-curve’ shown in Figure 1.2. is a plot of the power demand that the grid has to meet when the use of harnessed solar power is prevalent [2]. Solar power harnessed peaks during mid-day when power demands are low, dropping the power demand needed to be met even lower. However, in the evening when power demand peaks, the solar power harnessed dips causing the curve to have a steep ramping up right after a major dip [2]. This causes problems for the power supplier as there is a sudden tremendous peak in power demand that needs to be met [14]. A battery solves this by storing the harnessed solar power during those peak solar output hours and using that power during peak power demand hours.

The battery that has been chosen for this project is based on specifications provided by Redback Technologies [33] and Tesla [34]. The batteries take in DC input from the solar panels and AC input from the grid and redistributes the power to household appliances and back to the grid [35]. The specifications of these batteries are shown in Table 2.5. and are available in the Appendix. The battery capacities are about the same but considering that the battery should not be totally drained as this may shorten the battery’s cycle life, the maximum capacity set for the sake of this project is **12 kWh** [36]. The maximum discharge and charge rate for one hour is taken to be 5kW as shown in Table 2.5. The nominal voltage and the weight of the batteries are similar hence, these do not really play a major role in choosing the battery. There are other minor factors that will affect the battery’s performance but are not taken into considerations for the sake of simplicity.

Table 2.5. Battery Specifications [33], [34]

Specification	Battery	Redback (BE13200)	Tesla (Powerwall)
Capacity (kWh)	13.2	13.5	
Charge/Discharge Rate (kW)	N/A	5	
Nominal Voltage (V)	48	50	
Weight (kg)	130	125	

4.1.2.5. Sets of Tasks

The sets of tasks that arrive in time slot t is shown in Table 2.6. This input variable only shows the tasks arriving in that time slot and not the tasks that are still running. Sets of tasks are used to refer the algorithm to the tasks in a time slot as the algorithm runs through all the time slots.

Table 2.6. Set of Tasks

Time Slot, t	Tasks, i
8	9,10,11,12
9	0
10	7
11	5
12	0
13	0
14	0
15	0
16	0
17	3,6,13
18	8
19	1,2,4
20	0
21	0
22	0
23	0
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0

4.1.3. Independent Variables

The independent variables are kept constant all throughout. The variables and parameters depend on these independent variables. The two independent variables are the time slots, t and the tasks, i . All the input variables and two out of the three decision variables depend on the time slots whereas all the input parameters and one of the decision variables are based on the task i .

4.1.3.1. Time Slots

The time slots represent a full 24 hour day hence, the total number of time slots is 24 which is represented by T . The time slots start at 8 a.m. and end at 7 a.m. the following day. These times are chosen because they represent the cycle of a full day for a user (from waking up to going to bed). Another reason that these times are chosen is so that the deadlines for the tasks are kept within the same day and do not crossover into the following day. This may occur if the time slots are set to be from 12 a.m. to 11 p.m. Although, the time slots used are from 8 a.m. to 7 a.m. the following day, all the plots used in this project refer to the time slots from 0 to 24 and this does not indicate the actual time of day.

4.1.3.2. Tasks

The tasks shown in Table 2.1. are independent of all the other variables. These tasks are elements of the set $n(t)$ which is used to refer to tasks that arrive in a certain time slot as shown in Table 2.6. There are a total of 13 tasks which is denoted by an N (shown in Table 1.5.).

4.2. Decision Variables

The decision variables are set by the algorithms to fit the objective functions. Basically, the decision variables are the output of the algorithms. The decision variables that need to be set are the proportion of solar power harnessed that is siphoned to the battery storage, $k(t)$, the power discharged from the battery storage, $b(t)$ or the power bought from the grid to charge the battery, $-b(t)$ and the delays that the tasks should take, s_i . These decision variables are subject to the constraints mentioned earlier in 3.2. Constraints.

4.2.1. Proportion of Solar Power Harnessed

The proportion of solar power harnessed is a percentage of the total solar power harnessed for that time slot. This decision variable is a time-dependent variable and the value is set for each time slot. Harnessed solar power that is siphoned to the battery storage may contribute to reducing the cost spent by the user. This is because solar power is only harnessed during daylight hours and by siphoning some of this power to the battery storage, the power harnessed can be used later when there is zero solar power harnessed as shown in Figure 3.2.

4.2.2. Battery Power Charged/Discharged

The battery power charged or discharged must be within the maximum charge and discharge rate which is stated in Table 2.5. The variable is negative when the battery power is charged from the grid and the variable is positive when battery power is discharged to be used by the tasks. This is a time-dependent variable, so the value for this variable needs to be set for each time slot. The decision to buy power from the grid to store in the battery storage is useful as this power can be used in time slots that have higher buying-in rates. The decision to discharge the battery power to the tasks is useful in the occasion that the current time slots' buying-in rates are high. This may contribute to reducing the cost spent by the user. This could also be used to distribute the harnessed solar power to the time slots with less sunlight. Hence, lowering the peak-to-average ratio (PAR) of power demand.

4.2.3. Task Delays

The delay for each task is the number of time slots that the task can be delayed without surpassing the deadline set for these tasks. This is the task scheduling decision that the algorithm has to make. This delay is useful as it allows the user to schedule tasks during periods with peak power demand and peak rates to periods with lower power demand and cheaper rates. This may contribute to reducing the cost spent by the user and to reducing the PAR of power demand.

4.3. Algorithms

The input data is used by the algorithms to set the decision variables with the aim of fitting the objective function. The Tariff-Proportional Algorithm is the algorithm designed for this project to investigate how it fares in comparison to other algorithms that have already been used for a similar purpose. Genetic Algorithm is the algorithm that is being compared with Tariff-Proportional Algorithm. Although Genetic Algorithm is one of the more popular optimisation algorithms that is implemented for smart grids, the use of Genetic Algorithm in a similar model has not been done before as the section 2. Literature Review discusses. The model used in this project is similar to the model used by Chen [12]. However, the model used in by Chen [12] is a Lyapunov optimisation algorithm which is a lot more complex than Genetic Algorithm. Hence, the algorithms being compared in this project have not been implemented in a similar system model by the works reviewed.

4.3.1. Implementation Platform

These algorithm are modelled and executed using MATLAB. MATLAB is primarily used for programming mathematical problems which makes it a suitable platform to implement these algorithms. The programming language used in MATLAB is very high-level and easy to implement as well, making MATLAB the ideal platform to program these algorithms.

4.3.2. Tariff-Proportional Algorithm

Tariff-Proportional Algorithm is a simple and deterministic stepwise algorithm that sets the decision variables based on the rates that are shown in Table 2.3. One of the reasons that this type of simple and deterministic algorithm is used, is that it uses very little computational power to run. Hence, the aim is to investigate whether such a simple algorithm can outperform a more complex, stochastic optimisation algorithm such as Genetic Algorithm. A major difference between the Tariff-Proportional Algorithm from the Genetic Algorithm implemented is that there are no decision making processes for task scheduling. Therefore, all the task delays for the Tariff-Proportional Algorithm are set to zero. To implement a stepwise, “if-else” type of algorithm for task scheduling has not been done by any of the previous works and implementing such an algorithm may be too difficult and complicated. Thus, it is not used in this project.

4.3.2.1. Objective Function

$$\text{Average Cost} = \min_{k(t), b(t)} \frac{1}{T} \sum_{t=1}^T W(t)C(t) - b(t)C(t) - (1 - k(t))S(t)G(t) \quad (\text{viii})$$

The objective function is the aim that the algorithm attempts to achieve. This objective function’s main aim is to minimise the average cost per hour in a day. The function is subject to the decision variables noted at the min in (viii). As the function (viii) shows, the sum of the entire equation over all the time slots is divided by the total number of time slots to give the average cost per hour in a day. The first term of the function $W(t)C(t)$ represents the total cost of the total power demand in time slot t . Then the second term $-b(t)C(t)$ represents the reduction in cost or the increase in cost if the battery power is discharged to meet power demands or charged from the grid. The second term will become positive when the battery power is being charged from the grid. The final term $-(1-k(t))S(t)G(t)$ represents the selling of the excess harnessed solar power back to the grid. This final term is dropped when investigating the effect of not selling back to the grid. As can be seen, the objective function (viii) does not include the delays needed for task scheduling. To sum it up, the objective function takes the total expected cost and subtract the savings that the battery and excess harnessed solar power may contribute.

4.3.2.2. Equations

$$k(t+1) = \frac{c(t)}{c(t+1)} k(t) \quad (\text{ix})$$

The equation (ix) is used to determine the proportion of harnessed solar power that should be siphoned into the battery storage. The equation takes the buying-in rate from the current time slot and divides it with the rate of the next time slot, so that if the rate in the next time slot is greater than the rate in the current time slot, the fraction will be less than one. If the rate in the next time slot is less than the rate in the current time slot, the fraction will be greater than one. This will increase or decrease $k(t)$ inversely proportional to the increase or decrease in the rates. For an increase in the rate of the next time slot, the algorithm will want to sell more power to the grid in the time slot with the increased rate (the tariff to sell to the grid increases together with the tariff to buy from the grid as shown in Figure 3.5.). Hence, the value of $k(t)$ decreases so that more harnessed solar power can be sold. The reverse is true for a decrease in the rate of the next time slot. The variable $k(t)$ is initialised in the beginning of the first time slot as $k(1) = 0.5$. This value is chosen as it is the midpoint between the bounds of $k(t)$.

shown in (v). Selling half of the harnessed solar power is also typically done [30]. This also ensures that the values of $k(t)$ following this remain with the bounds (v).

$$b(t+1) = \frac{c(t+1)}{c(t)} b_{max}/2 \text{ OR } -\frac{c(t)}{c(t+1)} b_{max}/2 \quad (\text{x})$$

The equations (x) are used to determine the battery power to charge from the grid or discharge to the tasks. The equation takes the buying-in rate from the next time slot and divides it with the rate from the current time slot if the rate from the next time slot is greater than the current time slot. This gives the fraction a value greater than one. The equation takes the buying-in rate from the current time slot and divides it with the rate from the next time slot if the rate from the next time slot is less than the current time slot. This gives the fraction a value greater than one as well. Hence, the fraction in both equations in (x) will be greater than one which only increases the term $b_{max}/2$. For an increase in the rate of the next time slot, the algorithm will set $b(t+1)$ to be positive as the algorithm will want to discharge the battery power for the tasks since it is more expensive to buy from the grid in the next time slot. For a decrease in the rate in the next time slot, the algorithm will set $b(t+1)$ to be negative as the algorithm will want to stock up on power and buy more from the grid proportional to the change in the rate. The term $b_{max}/2$ is used to ensure that the increase and decrease in battery power charged or discharged are within the bounds stated in (vi).

4.3.2.3. Decision Making

The flow chart for the algorithm's decision making is shown in Figure 3.7. In the beginning, the decision variables are initialised as shown. Then the algorithm runs through three conditional statements to determine whether the rate for buying power from the grid in the next time slot is greater than, less than or the same as the current rate. At each of these conditions, if the statement is true, the decision made is shown on the right side of Figure 3.7. The entire algorithm is run through for all time slots t as shown in Figure 3.7. by the *for* loop that the flow chart is placed in. Finally, these decision variables are used in the objective function (viii) to obtain the final average cost. The code in MATLAB is written in a similar manner to Figure 3.7.

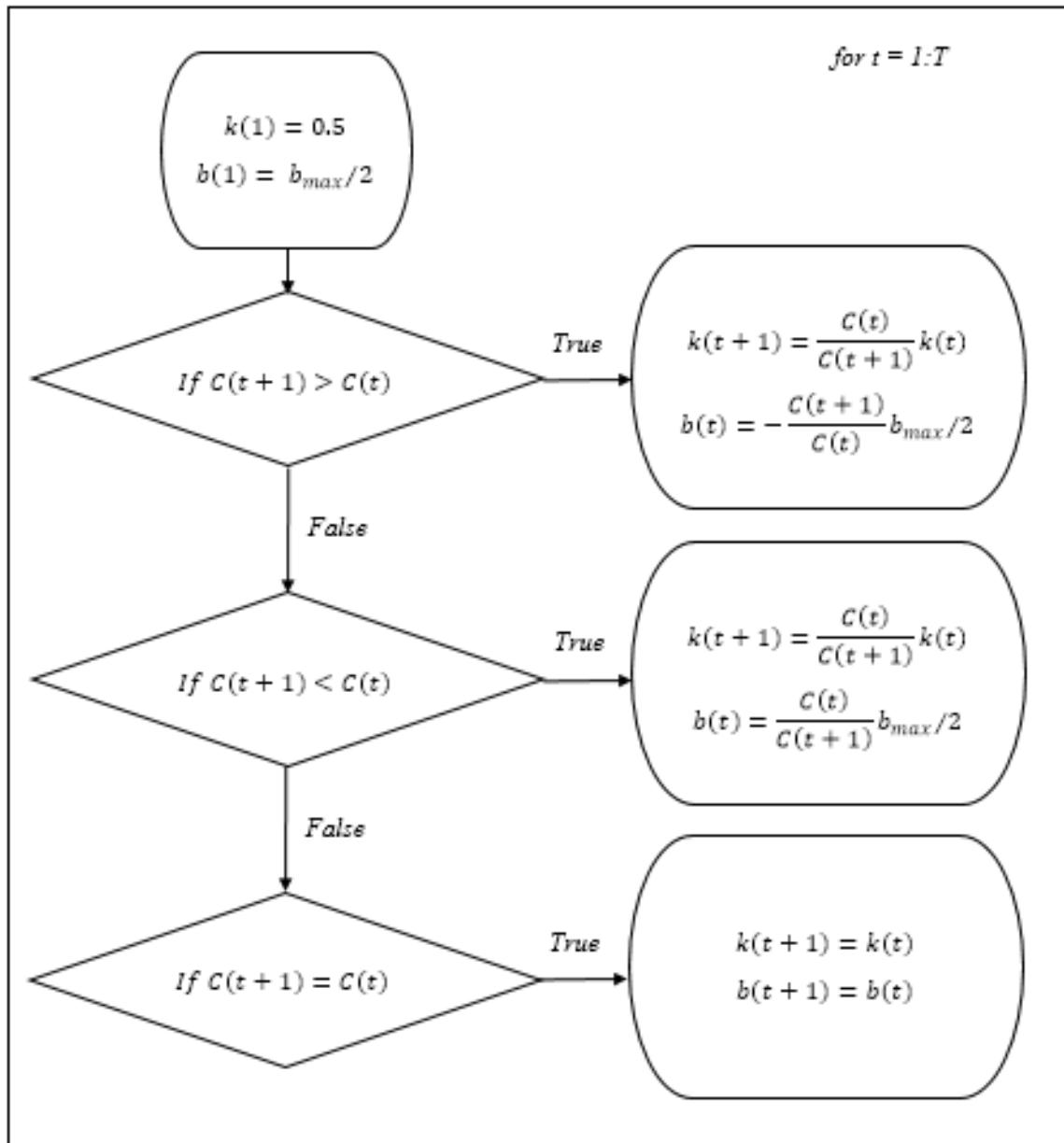


Figure 3.7. Tariff-Proportional Algorithm Decision Making

4.3.3. Genetic Algorithm

Genetic Algorithm is a stochastic optimisation algorithm that is designed to solve for constrained and unconstrained problems [37]. It is a stochastic optimisation algorithm as opposed to a deterministic optimisation algorithm as the iteration processes are probabilistic. This means that the results of the algorithm may change each time it is run. Although, the results should not differ by much. Genetic Algorithm borrows the idea of natural selection from biological evolution which involves the survival of the fittest [37]. The fittest points being the points that are able to achieve the best fitness values (lowest values for finding the minimum of an objective function). Genetic Algorithm is especially useful in finding the global minimum of a problem and not converging to a local minimum instead if the algorithm has a large enough disparity in its randomly generated population. The computation time that Genetic Algorithm takes depends on the number of iterations that is needed to achieve a reasonably precise fitness value. Hence, if the algorithm does not converge the iterations will continue forever unless a time limit is set. Implementing Genetic Algorithm with the model described for this project has not been investigated in the works reviewed in the section 2.

Literature Review. Most of the works with Genetic Algorithm only involve using the harnessed solar power and battery storage [13] or only task scheduling [11] but not both models together. The Genetic Algorithm used in this project covers both these models.

4.3.3.1. Objective Function

$$Avg. Cost = \min_{k(t), b(t), s_i} \frac{1}{T} \sum_{t=1}^T \left[\sum_{i=1}^{N(t)} \sum_{j=0}^{r_i-1} p_i C(t+j+s_i) - b(t)C(t) - (1-k(t))S(t)G(t) \right] \quad (xi)$$

The objective function is the fitness function for Genetic Algorithm. The aim of this objective function (xi) is to minimise the average cost per hour in a day. The function is subject to the decision variables noted at the min in (xi). As the function (xi) shows, the sum of the entire equation in the square brackets over all the time slots is divided by the total number of time slots to give the average cost per hour in a day. Within the brackets, the sum of the cost of power demand for all the tasks in a specific time slot is taken into account. The first term $p_i C(t+j+s_i)$ represents the cost of power demand for each task where j represents the time slots that a task may cover (if the task has a duration over one hour). So the third summing notation is for those tasks with a duration of more than one hour. The t in the cost term C is actually the arrival time for the task. Then the second term $-b(t)C(t)$ represents the reduction in cost or the increase in cost if the battery power is discharged to the tasks or charged from the grid (term will become positive). The final term $-(1-k(t))S(t)G(t)$ represents the selling of the excess harnessed solar power back to the grid. This final term is dropped when investigating the effect of not selling back to the grid.

4.3.3.2. Optimisation Process

Figure 3.8 describes the optimisation process used by Genetic Algorithm that is inspired by biological evolution [37]. Firstly, a population of randomly generated points that are within the constraints (if any) are used to initialise the algorithm [38]. These points are in the form of vectors if the solution has multiple variables. The points are scored (the fitness value evaluated) and the points with better scores are selected for propagation of the next generation [38]. The fitness value is taken as the reciprocal of the fitness function. So, the larger the fitness value, the closer the point is to the minimum. Elite points (points with the best fitness values) from the ‘parent’ generation are passed on to the ‘children’ generation without undergoing any changes [38]. Crossover occurs to a portion of remaining points in the ‘parent’ generation [38]. Random mutations occur to the remaining single points in the ‘parent’ generation to diversify the ‘children’ population [38]. It is important that the population maintains a certain level of disparity to prevent converging into a local minimum [39]. The ‘children’ generation replaces the ‘parent’ generation and undergoes the same process [38]. The iteration continues until a stopping condition is met. The stopping conditions are shown in Table 2.8.

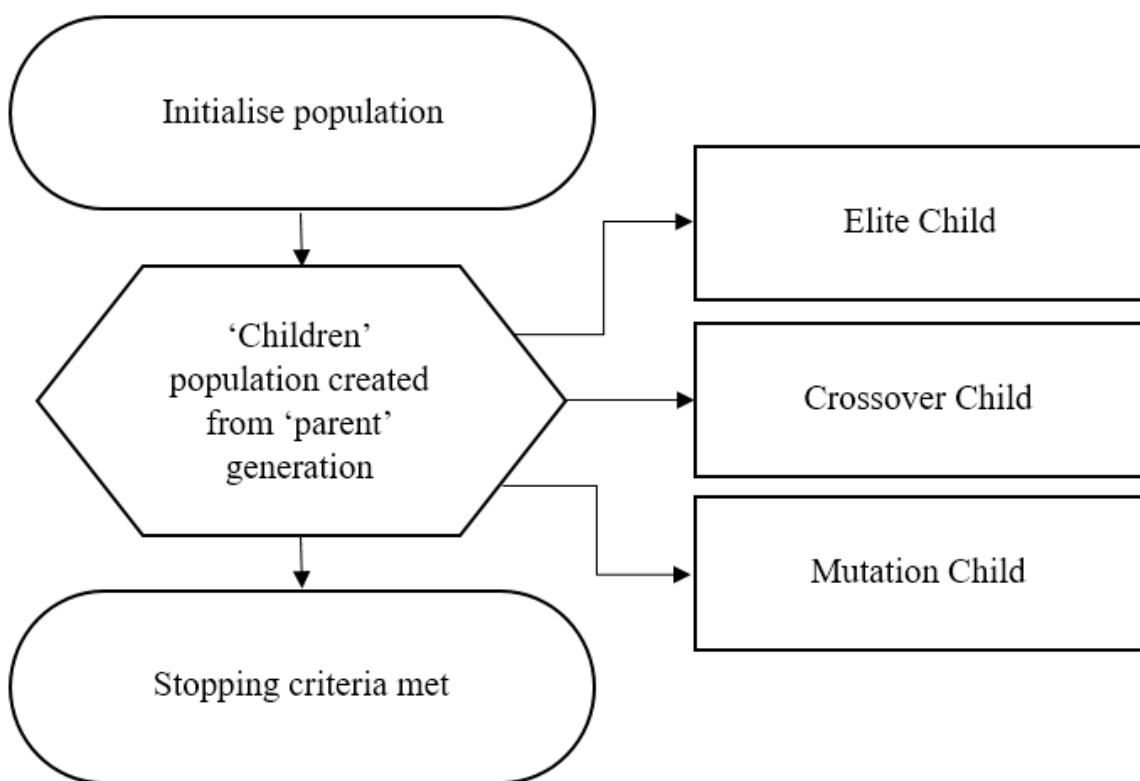


Figure 3.8. Optimisation Process

4.3.3.3. Settings

The settings used for the Genetic Algorithm are all left to the default values provided by MATLAB for its *ga()* function [40]. Table 2.7. shows the significant settings that MATLAB has set for the Genetic Algorithm. These values are used for the optimisation process detailed in Figure 3.8. Table 2.8. shows the stopping criteria used to end the iteration process shown in Figure 3.8. These settings can be altered using the *options* input for the *ga()* function [40]. In this project, the *options* input is set to *gaoptimset* which creates a Genetic Algorithm options structure [40].

Table 2.7. Settings

Number of Variables (<i>nvars</i>)	61 ($2T+N - 24 k(t), 24 b(t), 13 s_i$)
Population Size (<i>psize</i>)	200 (If $nvar > 5$)
Selection Function	Stochastic Uniform
Elite Count	10 ($0.05 \times psize$)
Crossover Function	Scattered
Crossover Fraction	0.8
Mutation Function	Gaussian

The setting of the number of variables (*nvars*) is actually an input to the *ga()* function [40]. The *nvars* is 61 because there are 24 time slots for the first two decision variables ($k(t)$, $b(t)$) and 13 tasks for the third decision variable (s_i) which totals up to 61. The population size is the number of points iterated in each round for each variable. The selection function is the method of choosing the ‘parents’ for the next generation based on scaled fitness values (scores) [38]. Stochastic Uniform function lays the scored ‘parents’ in a line and arranges them according to their scores [41]. The length of the section of line each ‘parent’ is on is proportional to the score of the ‘parent’ [41]. The algorithm then goes through the line in equal periods hence, the ‘parents’ with better scores (longer line segments) will be chosen more than the ‘parents’ with weaker scores [41]. The elite count, crossover values and the mutation function all determine the three types of ‘children’ that the ‘parents’ will propagate (shown in Figure 3.8.). The elite count determines the number of scored ‘parents’ that will be carried over to the next generation [38]. The elite children have the highest fitness values, so an elite count of 10 means the top 10 scores are chosen to be carried over. The crossover function determines the method of crossing over [38]. A Scattered crossover takes a random binary vector and chooses the ‘genes’ from ‘parent’ A and ‘parent’ B by assigning one ‘parent’ to the 1s in the binary vector and the other ‘parent’ to the 0s in the binary vector [38]. Hence, there will be a random mix of the two ‘parents’ genes. The crossover fraction determines the fraction of the ‘parent’ generation (excluding the elite ‘parents’) that will undergo the crossover decided by the crossover function [40]. The mutation function is applied to the remainder of the ‘parent’ generation. A Gaussian function mutation function adds random numbers from a Gaussian distribution to the remainder of the ‘parent’ generation [41].

Table 2.8. Stopping Criteria

Constraint Tolerance	1e-3
Maximum Stall Generations	50
Function Tolerance	1e-6
Maximum Generation	6100 (100 x <i>nvars</i>)

The stopping criteria determines the stopping of the iterative process. The constraint tolerance is not exactly a stopping criteria but it is checked to ensure that the results are within the constraints [40]. The maximum stall generations is used by the function tolerance to check for any changes in the results within that many generations [40]. So based on Table 2.8., in 50 iterations, the changes in the results must be no more than 1e-6 for this stopping criteria to be satisfied. The maximum generation is the maximum number of iterations before the algorithm stops [40]. The iterations will continue until any one of these stopping criteria are met. There are other stopping criteria such as the maximum time which has a default setting of infinity [40]. Hence, the algorithm will go on forever if there is no convergence in the results. The maximum time may be useful if the algorithm takes too long to come to a stopping criteria.

5. Results & Discussion

The Tariff-Proportional Algorithm and the Genetic Algorithm are compared in several different aspects. The main aspect that is being compared is the **average cost per hour in a day**. This aspect is generated by the objective functions (viii) and (xi) based on the decision variables set by the algorithms. The second major aspect that is being compared is the **peak-to-average ratio (PAR) of power demand**. These are the aims of this project hence, the results of these aspects will be investigated most thoroughly.

Other aspects that are investigated include the effects of selling harnessed solar power. The algorithms' objective functions are altered for this purpose (by dropping the selling term). The effect of using harnessed solar power directly (use of battery not considered) to offset the power profiles and the effect that the discharged battery power has on the power profile are also investigated. User dissatisfactions, determined by a utility function for both algorithms are compared. Finally, the computation times that each algorithm takes are also compared.

5.1. Caveats

There are some caveats to this experiment. All the results obtained are based on interpreted data sourced from a variety of sources. Any discrepancies between these sources are not taken into consideration. These data may not reflect actual values that one may obtain from real world measurements. The data used are all averaged hence, the experiment does not take into consideration the anomalies and extreme data values that may occur in the real world. In real-world scenarios the products used (solar system and battery storage) will not have the exact specifications given in the datasheet as there will be minor faults that occur in the manufacturing of these products. The use of the solar panels and the batteries are assumed to function ideally. These products also deteriorate with time and will not function as specified in their datasheets [33], [34] (especially applicable to the battery storage which deteriorates the fastest).

The choice of tasks shown in Table 2.1. are very subjective and differs vastly in different households. The number of people living in a household and the size of the households are not taken into consideration as well. As for the objective function which calculates the average cost per hour for a day, the function does not take other costs into consideration. Such additional costs may include installation costs, supply charges [32] and maintenance cost. Besides this, the experiment is entirely simulated on MATLAB which excludes any real-world, practical issues that may be faced. The simulation may also have errors present in the algorithms that the author has overlooked which will lead to skewed results.

5.2. Procedure

All the results were simulate on MATLAB. The procedure or the pseudocode for obtaining the results are discussed in detail in this section. The Genetic Algorithm codes are implemented before the Tariff-Proportional Algorithm codes. The input data for these algorithms are set as global variables so that they can be accessed across multiple workspaces. The codes for these algorithms can be found in a link to the code repository found in the 6. Conclusion & Recommendations section.

5.2.1. Genetic Algorithm

Firstly, the function for obtaining the average cost per hour in a day is constructed based on (xi) with the inputs specified. The function accepts one variable which is denoted as x . The single variable x is an array of all the decision variable vectors. Then within the function, the variable x is split into the decision variable vectors to be used by the rest of the function. Another function is used when investigating the effect of not selling on the average cost per hour in a day. The pseudocode for the function is as shown below:

1. Initialise variables and parameters
2. Run loop to go through all time slots
 - Loop through *all time slots*
 - If *task exists in the time slot*
 - Loop through *all tasks in the time slot*
 - ❖ If *task has a duration that is more than one time slot*
 - Loop through *all time slots that task covers*
Calculate the costs for that task in each time slot
 - End of loop
Sum the costs of the task in each time slot
 - ❖ Else, *calculate the cost of task in current time slot*
 - End of loop
 - Else, *move on to next line of code*
Calculate the battery power level at the end of the time slot
 - End of loop
 - 3. Calculate the average cost per hour in a day

This function is called upon using the $ga()$ function in the workspace to obtain the decision variables. In the workspace, the input data are called in from an Excel spreadsheet and set as global variables. The battery power level at the beginning of the first time slot is set to half the maximum capacity. Then, the inequalities, bounds and equations that govern the Genetic Algorithm are set to be the input of the $ga()$ function. The $ga()$ function takes in the function to be evaluated, the number of variables, the inequalities, the equalities, and the bounds for the decision variables. The inequalities set are as shown in (iii) and (iv) whereas the bounds for each decision variable are set as shown in (v), (vii) and (viii). Another $ga()$ function is called to get the decision variables used when investigating the effect of not selling on the average cost per hour in a day. The pseudocode to implement the Genetic Algorithm is as shown below:

1. Initialise input variables and parameters
2. Set initial battery power level
3. Set inputs for $ga()$ function
4. Run $ga()$ function
5. Break apart x (array of decision variable vectors) into the respective decision variables

5.2.2. Tariff-Proportional Algorithm

Implementing the Tariff-Proportional Algorithm is much simpler than implementing the Genetic Algorithm. The input data that has already been set in the Genetic Algorithm workspace as a global variable is called into another workspace for this algorithm. Since no task scheduling is involved in the Tariff-Proportional Algorithm, a set with the tasks running for all time slots is initialised. This is used to calculate the total power consumed in one time slot and also used to plot the power demand plot that is seen in Figure 3.3., 3.4. and 3.6. The equation (viii) to obtain the average cost per hour in a day is altered to investigate the effect of not selling. The pseudocode to implement the Tariff-Proportional Algorithm is as below:

1. Initialise the input variables and parameters
2. Set initial battery power level, initial battery power charged/discharged and proportion of solar power harnessed to be siphoned into battery storage.
3. Power demand for each time slot is obtained
4. Run loop to go through all time slots (shown in Figure 3.7.)
 - Loop through *all time slots*
 - If *rate in next time slot is greater than current rate*
Decide values of decision variables
 - Else, if *rate in next time slot is less than current rate*
Decide values of decision variables
 - Else, *Maintain decision variable values*
 - End of loop
5. Total cost over all time slots is obtained
6. Calculate the average cost per hour in a day

5.3. Decisions Made

The values set for the decision variables are compared between the two algorithms. Table 3.1., 3.2. and 3.3. show the values for the decision variables. The Figure 4.1., 4.2. and 4.3. compares these values graphically. The decision variables being compared are the proportion of harnessed solar power that is siphoned to the battery storage, the power charged to the battery from the grid or discharged from the battery to meet the power demands of tasks and the delays for each task (only applicable for Genetic Algorithm). These decision variables are the results of the algorithms trying to achieve their respective objective functions.

5.3.1. Proportion of Solar Power Harnessed

The proportion of harnessed solar power that is siphoned to the battery storage is denoted by $k(t)$ as shown in Table 1.4. and it is a percentage of the solar power harnessed that is stored in a battery storage. The Table 3.1. and Figure 4.1. shows the proportions that have been decided by both algorithms.

As can be seen in Figure 4.1., the $k(t)$ for Genetic Algorithm fluctuates for every time slot as the algorithm tries to find an ideal value. The $k(t)$ seems to fluctuate about an average value of 0.5. The $k(t)$ for Tariff-Proportional Algorithm is more stable as it is inversely proportional to the tariff for buying from the grid as shown in Figure 3.5. As the rates go up the $k(t)$ goes down. The opposite seems to be true for the Genetic Algorithm as it shows in Figure 4.1 where the $k(t)$ dips at about time slot 15 whereas the $k(t)$ for the Tariff-Proportional Algorithm increases at about the same time slot. The values of $k(t)$ fall within the bounds (v) that have been specified. Hence, the constraints have been met by the algorithms.

Table 3.1. Proportion of Solar Power Harnessed

Time Slots, t	Proportioned of Solar Power Harnessed, $k(t)$	
	Tariff-Proportional Algorithm	Genetic Algorithm
8	0.50000	0.39193
9	0.50000	0.30149
10	0.50000	0.11975
11	0.50000	0.73051
12	0.50000	0.45353
13	0.50000	0.50734
14	0.50000	0.32080
15	0.34758	0.45965
16	0.34758	0.68300
17	0.34758	0.52582
18	0.34758	0.37966
19	0.34758	0.81742
20	0.34758	0.57199
21	0.50000	0.41810
22	0.75596	0.80240
23	0.75596	0.28501
0	0.75596	0.21832
1	0.75596	0.50249
2	0.75596	0.18885
3	0.75596	0.54272
4	0.75596	0.59680
5	0.75596	0.43923
6	0.75596	0.57494
7	0.50000	0.41864

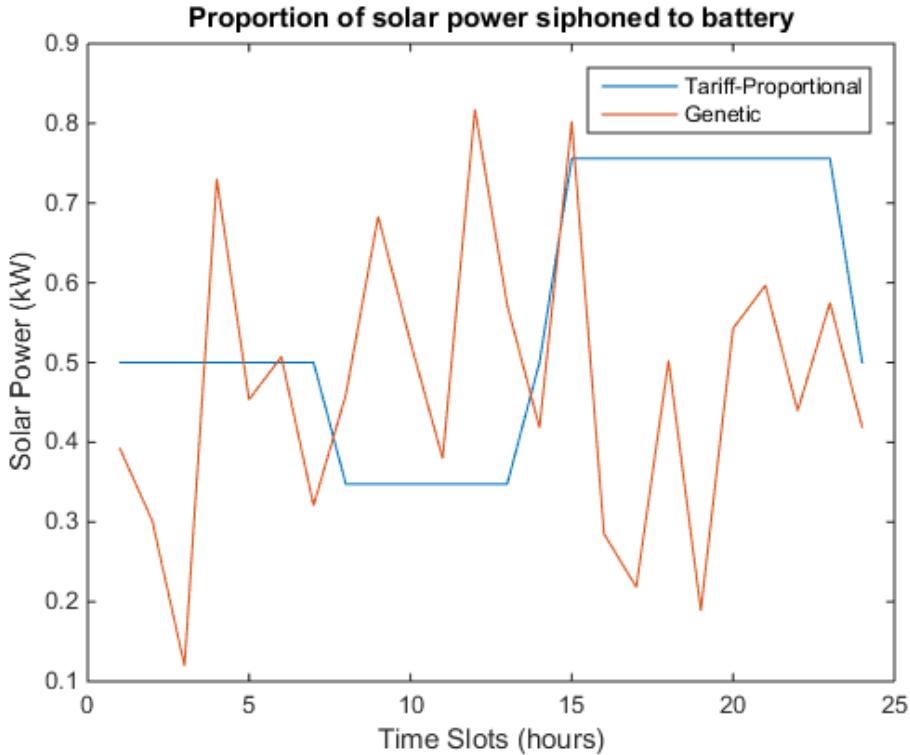


Figure 4.1. Comparing Proportion of Solar Power Harnessed Siphoned to Battery

5.3.2. Battery Power Charged/Discharged

The battery power charged from the grid or discharged to the tasks are denoted as $b(t)$ as shown in Table 1.4. The Table 3.2. and Figure 4.2. show the values that have been decided by both algorithms.

As shown in Figure 4.2. the $b(t)$ value tends to be at zero most of the time. This seems to show that the battery does not really have much of a function. The $b(t)$ for the Genetic Algorithm starts of at around 5 kW as this is the maximum charge/discharge rate that has been specified for the battery then the $b(t)$ falls to zero and remains there for the rest of the time slots. The $b(t)$ for the Tariff-Proportional Algorithm starts of at half the maximum charge/discharge rate and discharges more to the tasks when the rates are higher. The $b(t)$ then goes negative as power is bought from the grid to charge the battery. The values of $b(t)$ fall within the bounds (vi) that have been specified. Hence, the constraints have been met by the algorithms.

Table 3.2. Battery Power Charged/Discharged

Time Slots, t	Battery Power Charged/Discharged, $b(t)$	
	Tariff-Proportional Algorithm	Genetic Algorithm
8	2.50000	4.99997
9	2.50000	0.07953
10	2.50000	0.04931
11	2.50000	0.39527
12	2.50000	0.27923
13	2.50000	0.32436
14	2.50000	0.20061
15	3.59632	0.25607
16	3.59632	0.30826
17	3.59632	0.15874
18	3.59632	0.06358
19	3.59632	0.06228
20	3.59632	0.00954
21	-3.59632	0.00068
22	-3.77982	0.00090
23	-3.77982	0.00072
0	-3.77982	0.00096
1	-3.77982	0.00044
2	-3.77982	-0.00025
3	-3.77982	0.00091
4	-3.77982	0.00092
5	-3.77982	0.00070
6	-3.77982	0.00091
7	3.77982	0.00708

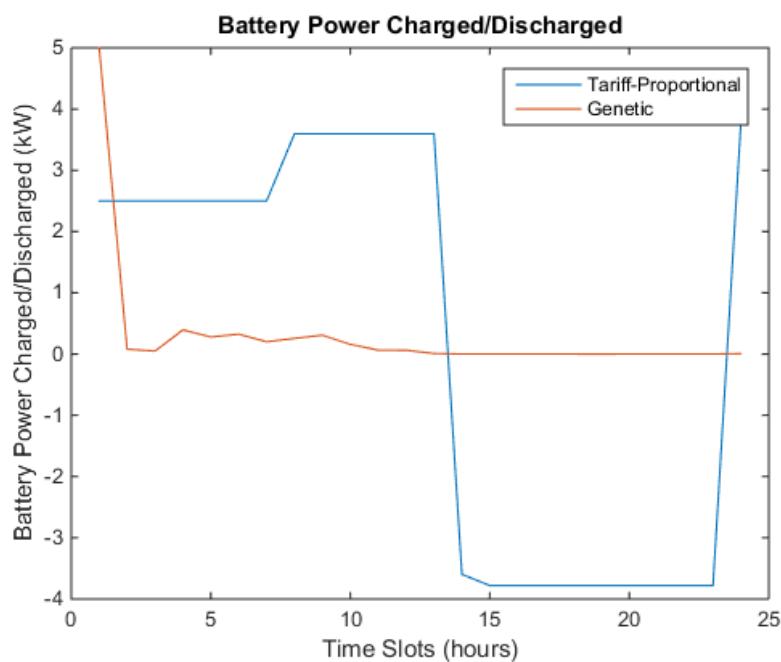


Figure 4.2. Comparing Battery Charged/Discharged

5.3.3. Task Delays

The delays set for the tasks in are denoted as s_i as shown in Table 1.4. The task delays are needed for task scheduling performed by the Genetic Algorithm. The Table 3.3. and Figure 4.3. show the delays set for the tasks listed out. The task delays for the Tariff-Proportional Algorithm are all set to zero as no task scheduling decisions are performed in the Tariff-Proportional Algorithm.

The first two tasks are laundry-related tasks and have been given the highest delay times. This is probably due to their large deadline periods. The oven and microwave do not undergo any delay which has an added advantage as the user will not have to delay a meal and lose satisfaction points (less delay, better user satisfaction). The dishwasher and the space heater undergo a delay of four hours which may be a little inconvenient if the weather is cold but it does not really matter for the dishwasher as it is a task that does not need immediate attention. The air-conditioner and the laptop undergo a delay of one hour which will not really affect the users' satisfaction. As for the Tariff-Proportional Algorithm, the users' satisfaction will be maximised as there is zero delay for all the tasks. Most of the tasks (1, 4, 6 and 9) have been delayed to their limit as shown in Table 3.3. The constraints for all the tasks are met as the delays do not exceed their bounds (vi) as shown in Table 3.3.

Table 3.3. Task Delays

i	Task	Bound, $d_i - r_i$	Task Delays, s_i	
			Tariff-Proportional Algorithm	Genetic Algorithm
1	Dryer	5	0	5
2	Washing machine	5	0	3
3	Oven	2	0	0
4	Dish washer	4	0	4
5	Microwave	3	0	0
6	Space heater	4	0	4
7	Air-conditioner	3	0	1
8	LCD TV (22")	2	0	0
9	Laptop (15")	1	0	1
10	Water heater	1	0	0
11	Fridge	0	0	0
12	Freezer	0	0	0
13	CFL Lights	0	0	0

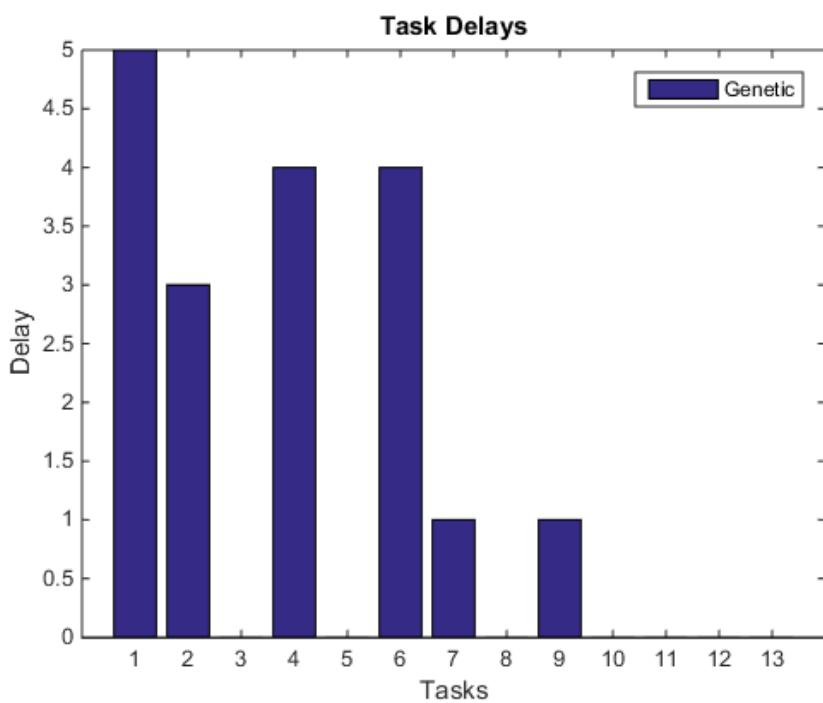


Figure 4.3. Delays of Tasks in Bar Chart

5.4. Main Aspects

This section presents the results and the discussions of the different aspects of the investigation that are being looked into and the way that they differ between the two algorithms. The main aspects to be compared are the aims of this project. These aspects are the average cost per hour in a day and the peak-to-average ratio (PAR) of power demand from the tasks.

5.4.1. Average Cost per Hour in a Day

This aspect is supposed to be minimised by both algorithms to result in the lowest possible cost that a user should have to spend. The objective functions (viii) and (xi) both describe this aspect. The Table 3.4. shows the results of the two algorithms and compares them to a model that does not use any algorithm or harnessed solar power or batteries, a control model. The effect of the ability to sell to the grid has also been investigated and the results shown in Table 3.4.

Both models work as the results of both models shown in Table 3.5. indicate a major decrease in the cost compared to the control model. The Genetic Algorithm model fares better against the Tariff-Proportional Algorithm if the ability to sell to the grid is taken into consideration but fares poorly when the ability to sell is removed.

Table 3.5. Average Costs per Hour in a Day

Selling Ability	Average Cost per Hour in a Day (cents)		
	Control	Tariff-Proportional Algorithm	Genetic Algorithm
Yes	66.14	27.66	20.76
No	66.14	29.58	41.68

As expected, both algorithms fare a lot better compared to the control model. The control model represents a traditional household feeding off the grid as it does not have any solar system or batteries installed. It is not surprising that Genetic Algorithm fares better than the Tariff-Proportional Algorithm as Genetic Algorithm uses an optimisation process that searches for the most minimal point whereas the Tariff-Proportional Algorithm does no such optimisation and bases its decisions on the buying-in tariff alone. Surprisingly, the Tariff-Proportional Algorithm fares better when the ability to sell is removed. This is most likely because of its better usage of the battery storage as shown in Figure 4.2. Besides, the solar power harnessed is almost negligible when compared to the power demands from the tasks as shown in Figure 3.3.

Therefore, in a grid that has feed-in tariffs (ability to sell to the grid), Genetic Algorithm is the better option but in a grid that has not introduced feed-in tariffs, the Tariff-Proportional Algorithm will be the better option.

5.4.2. Peak-to-Average Ratio Power

$$PAR\ Power = \frac{Peak\ Power}{Average\ Power} \quad (xii)$$

This aspect is compared between the two algorithms using the equation (xii) but it is not included in the objective function. The results are shown in Table 3.6. Shifting the tasks to minimise the cost may lead to a decrease in the peak-to-average ratio (PAR) of power demand. Shifting tasks minimises the cost because the rates for buying-in power vary according to the power demand received by the power supplier. The varying rates is a form of Demand Side Management (DSM) used to lower the PAR of power demand. So, the tasks will need to be shifted to time slots with lower rates in order to minimise the cost. Hence, task scheduling is an essential part of reducing PAR of power demand. First, the effect of task scheduling alone is investigated as shown in Figure 4.4. The effect of using harnessed solar power directly to offset the power demand is investigated in Figure 4.5. The effect of using the battery power that has been decided by the algorithms to offset the power demand of the tasks is also investigated as shown in Figure 4.6. As for the effect of the battery power discharged, any decision to charge the battery (negative values) is brought to zero as it is not possible to draw power from the tasks. The PAR of power demand resulting from the algorithms is represented by the PAR of power demand offset by the battery power discharged to the tasks.

Genetic Algorithm fares better than Tariff-Proportional Algorithm in reducing the PAR of power demand when only task scheduling (no battery power) is considered. Genetic Algorithm also fares better than Tariff-Proportional Algorithm when harnessed solar power is used directly to offset the power demand. Finally, Genetic Algorithm fares better compared to Tariff-Proportional Algorithm when the decided battery power is used. Hence, this shows that the Genetic Algorithm fares better than the Tariff-Proportional Algorithm in all of the effects investigated.

Table 3.6. Peak-to-Average Ratio Power

Effect of	Peak-to-Average Ratio Power	
	Tariff-Proportional Algorithm	Genetic Algorithm
Task Scheduling	4.2598	2.8283
Harnessed Solar Power	4.6778	3.1596
Battery Power	5.2446	3.3621

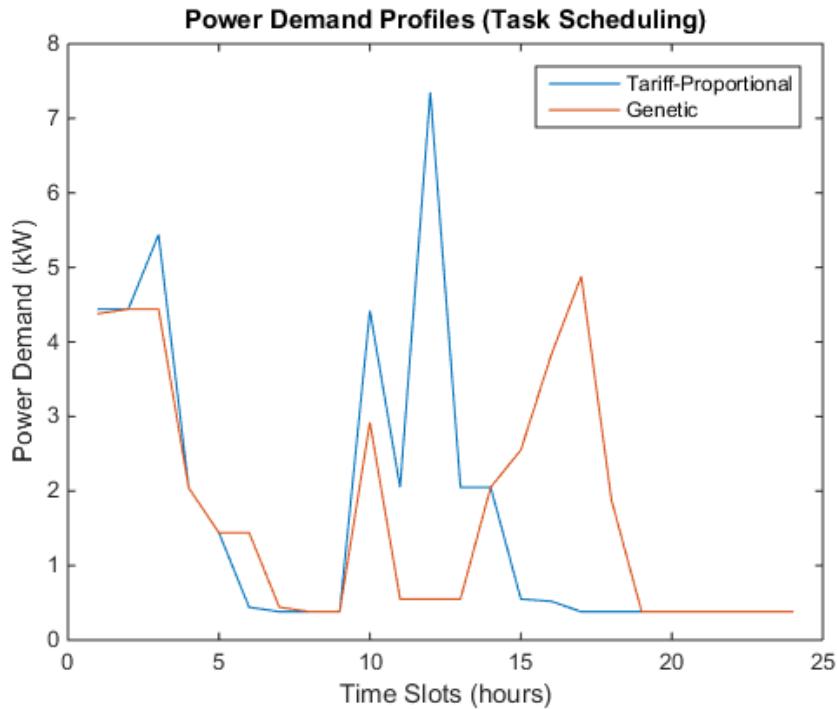


Figure 4.4. Effect of Task Scheduling Alone

The difference between the two algorithms shown in Figure 4.4. shows that task scheduling used in the Genetic Algorithm model has a major effect on the power demand profile. As expected, the use of task scheduling reduces the PAR of power demand for Genetic Algorithm compared to the Tariff-Proportional Algorithm. This can be seen in Table 3.6 where Genetic Algorithm is superior all the time. It definitely shows that task scheduling has a significant role in reducing PAR of power demand.

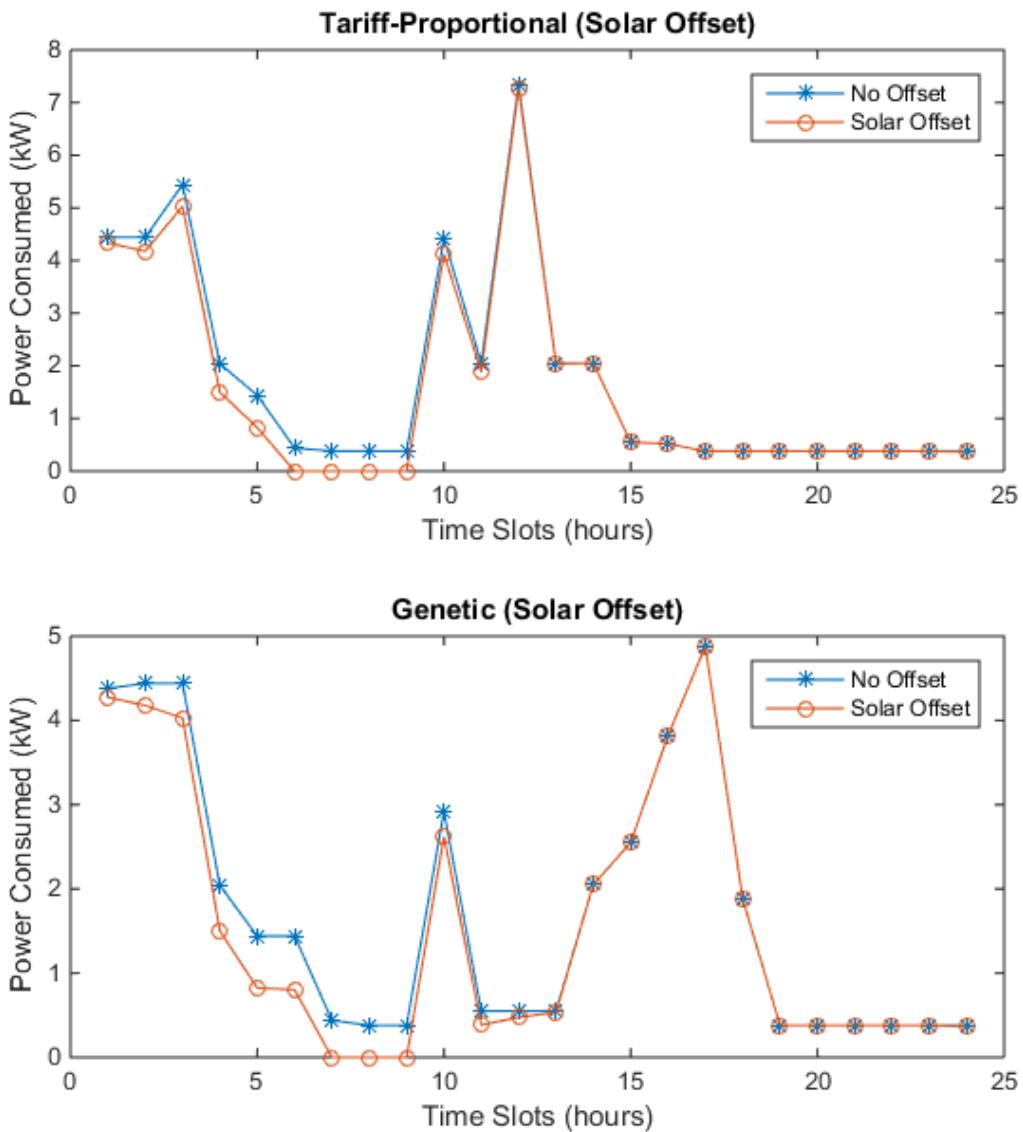


Figure 4.5. Effect of Harnessed Solar Power

As can be seen from Figure 4.5. and Table 3.6., the direct use of harnessed solar power does not have a major effect on the power demand profiles resulting from both algorithms. This is expected as it shows in Figure 3.3. that the solar power harnessed is almost insignificant when compared to the power demand profile. On some occasions, the solar power harnessed exceeds the power demand resulting in some time slots with zero power demand. The PAR of power demand has deteriorated slightly compared to the PAR of power demand without the effect of harnessed solar power. This is as predicted by the 'duck-curve' shown in Figure 1.2. Perhaps with the use of a higher rating solar system, the power demand profile may have a larger offset that may bring about a larger PAR of power demand.

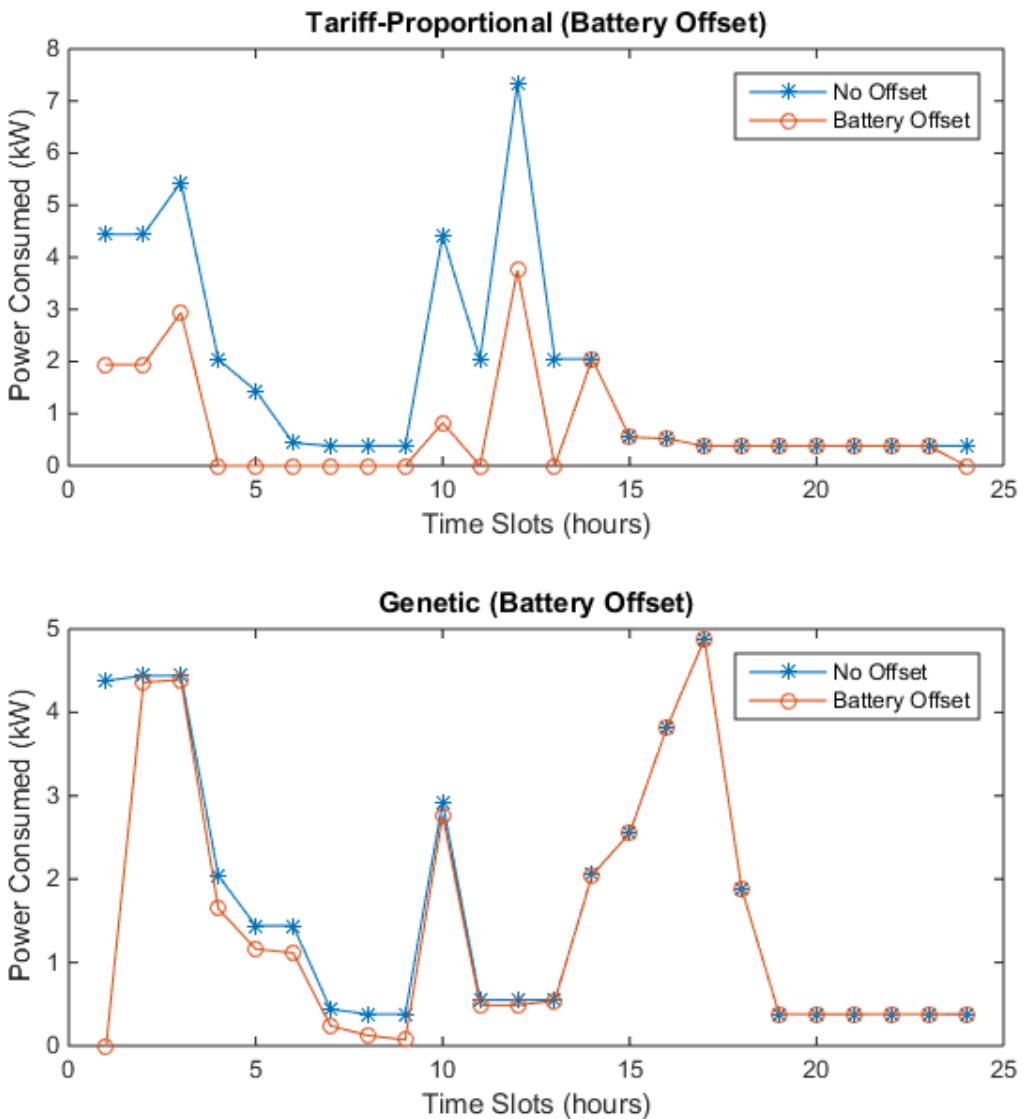


Figure 4.6. Effect of Battery Power

The effect of the battery power discharged to the tasks shown in Figure 4.6. corresponds to the decisions made by the algorithms shown in Figure 4.2. The battery power has a major effect on the power demand profile of the Tariff-Proportional Algorithm but barely any effect on the Genetic Algorithm power demand profile. The Genetic Algorithm decisions for battery power used (shown in Figure 4.2.) is zero for most of the time slots and only has a major discharge in the first time slot which Figure 4.6. shows. Surprisingly, the use of battery power has deteriorated the PAR of power demand for both algorithms. This may be because of the decrease in the average power demand due to the many time slots with zero power demands as shown in Figure 4.6. This result is in contrary to the expected results of using a battery storage as the battery's purpose is to even out the power demand profile.

With all these effects considered, the Genetic Algorithm still fares better in reducing the PAR of power demand.

5.5. Other Aspects

The other aspects that have been considered in this investigation are the users' dissatisfaction based on a utility function and the computational time. These two aspects will be compared for both the algorithms.

5.5.1. Users' Dissatisfaction

$$U = s_i^2 \quad (\text{xiii})$$

The dissatisfaction of a user is measured using a convex utility function [12]. The utility function measures the dissatisfaction of a user based on the time delays that the user has to wait to carry out a task. The Figure 4.7. shows the utility function with respect to the delay times. The equation (xiii) is used as the convex utility function. The choice of this equation is based loosely on some utility functions in a literature [6]. As the delay increases, the user's dissatisfaction increases quadratically. For the task delays shown in Table 3.7, the dissatisfaction of the user is calculated. The total score of dissatisfaction is then compared between the two algorithms.

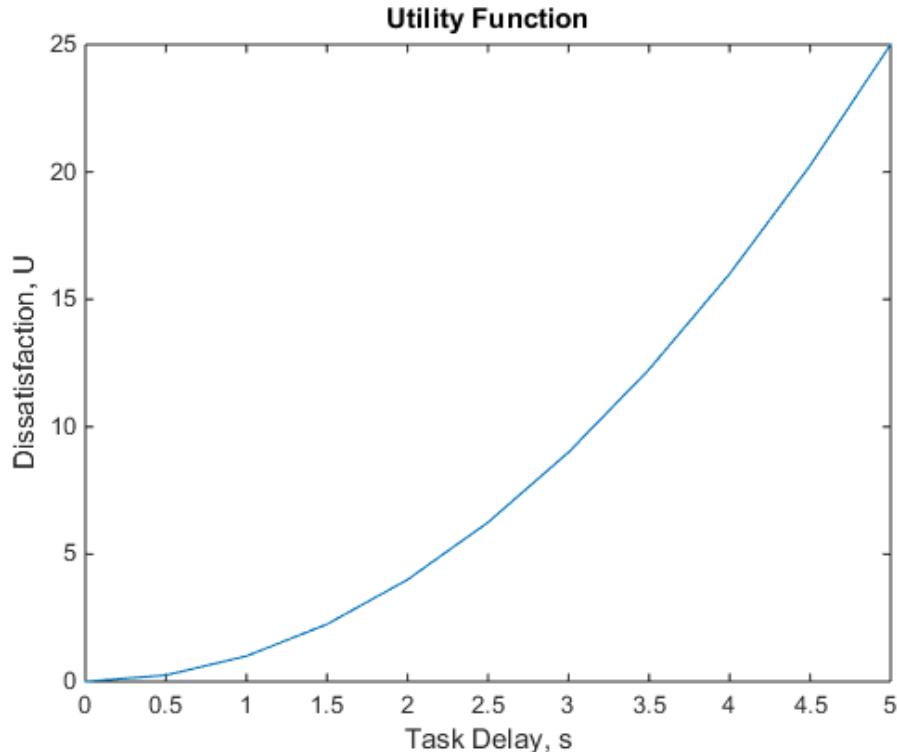


Figure 4.7. Utility Function

Table 3.7. User Dissatisfaction

i	Task	Tariff-Proportional Algorithm		Genetic Algorithm	
		Delay, s_i	Dissatisfaction, U	Delay, s_i	Dissatisfaction, U
1	Dryer	0	0	5	25
2	Washing machine	0	0	3	9
3	Oven	0	0	0	0
4	Dish washer	0	0	4	16
5	Microwave	0	0	0	0
6	Space heater	0	0	4	16
7	Air-conditioner	0	0	1	1
8	LCD TV (22")	0	0	0	0
9	Laptop (15")	0	0	1	1
10	Water heater	0	0	0	0
11	Fridge	0	0	0	0
12	Freezer	0	0	0	0
13	CFL Lights	0	0	0	0
Total			0		68

It is obvious that the user satisfaction for Tariff-Proportional Algorithm is much better as there is no task scheduling performed in this algorithm. Hence, the Tariff-Proportional Algorithm is better than the Genetic Algorithm in this aspect.

5.5.2. Computational Time

The computational time measured is the actual, physical time that has elapsed from the beginning of the algorithm to the end. The elapsed time is measured using the functions *tic* [42] and *toc* [43] that starts and ends an internal stopwatch. The time is given in seconds and shown in Table 3.8. This is not the CPU time which is considered an inaccurate measure of performance [44] as it does not consider the time for input or output operations, fetching data and storing data [45]. As a measure of performance, the actual elapsed time is usually measured [44].

Table 3.8. Computational Time

Algorithm	Computational Time (s)
Tariff-Proportional	0.0057
Genetic	121.8656

As Table 3.8. shows, the Tariff-Proportional Algorithm takes a lot less time than the Genetic Algorithm to run. This is expected as the Tariff-Proportional Algorithm is a much simpler, deterministic algorithm as compared to the more complex, stochastic and iterative Genetic Algorithm. Hence, the Tariff-Proportional Algorithm is far superior compared to the Genetic Algorithm in terms of computational time.

6. Conclusion & Recommendations

6.1. Results Conclusion

This project focusses on designing algorithms for the user end of the smart grid. The objectives of these algorithms are to **minimise the average cost per hour in a day** that the user has to spend and to indirectly **lower the peak-to-average ratio (PAR) of power demand**. These objectives have been achieved. These objectives are achieved through the use of two algorithms which are the Tariff-Proportional Algorithm and the Genetic Algorithm. The Tariff-Proportional Algorithm is a novel design and has not been used by any of the works reviewed. The use of Genetic Algorithm in a model identical to the one used in this project has not been done by previous works reviewed as well. The performances of these two algorithms are then compared and evaluated in several different aspects.

Based on these aspects, the Genetic Algorithm is more suited for the objectives of this project. The Genetic Algorithm is better than the Tariff-Proportional Algorithm in minimising the cost that the user has to spend. However, the Tariff-Proportional Algorithm is better at minimising the cost the user has to spend when the model is altered to remove the ability to sell back to the grid. The results of the cost minimisation objective are shown in Table 3.5. Since this project's system model allows the user to sell back to the grid, the results from this model show that the Genetic Algorithm is significantly better at minimising the average cost per hour in a day spent by the user. This is probably because of the Genetic Algorithm's better capabilities at optimising to find the global minimum of the function.

In terms of achieving the second objective of lowering the PAR of power demand, the Genetic Algorithm is the better option again. Surprisingly, investigations on the effect of not using the battery (task scheduling only) has shown better PAR of power demand than the results of the project's system model. Using harnessed solar power to directly offset the power demand has also resulted in a poorer PAR of power demand. Although the maximum power demand and the mean power demand has significantly dropped from the use of the battery, the difference between the maximum power demand and the mean power demand has widened causing the increase in PAR of power demand. The results of this objective of lowering the PAR of power demand is shown in Table 3.6. The results definitely show that the Genetic Algorithm is superior in lowering PAR of power demand compared to the Tariff-Proportional Algorithm. This is largely due to the task scheduling ability of the Genetic Algorithm.

Other aspects such as computational time taken for the algorithms to run and the users' dissatisfaction are investigated as well. The results of these other aspects have shown that the Tariff-Proportional Algorithm is superior to the Genetic Algorithm. The Tariff-Proportional Algorithm takes much less computation time compared to Genetic Algorithm which is an iterative optimisation algorithm. Tariff-Proportional Algorithm also scored zero for user's dissatisfaction as it does not perform any task scheduling. The results of these are shown in Table 3.7. and 3.8. Hence, the Tariff-Proportional Algorithm is far superior in these minor aspects but definitely inferior in the major aspects (the objectives of the project). In conclusion, the Genetic Algorithm is the better algorithm to be used in achieving the objectives of this project.

6.2. Limitations & Recommendations

This project's model is not without limitations. As for minimising the cost a user has to spend, the possible installation cost, maintenance cost and the supply charge to bring in power from the grid [32] are not taken into consideration. Although these costs would offset the cost minimisation results of both algorithms by the same amount but the total resulting cost will be higher. The deterioration of the solar panels are not taken into consideration as the errors resulting from these deteriorations are not significant. However, the deterioration of the battery storage may cause significant errors in the results if the battery is into the late stages of its lifespan. Another limitation that has not been considered is the type of household that is being investigated. This may vary the results by quite a bit depending on the type of household considered. The use of MATLAB to simulate the model will also provide results that differ to a real-world model. These are some of the limitations that have been overlooked for the sake of simplicity and the time constraint on this project.

Based on the limitations discussed, the additional costs mentioned should be taken into consideration. The current state of deterioration that the battery is in should also be considered as this may limit the discharge/charge rate and the maximum capacity of the battery. To obtain more realistic results, the use of data from a few common households should be used or the model should consider different sets of tasks for a few common types of households. The algorithms should also be tested out in an actual household to observe the performance and compare it to the simulated model.

Based on the results of this project, some recommendations for future works similar to this include the following. Both algorithms should include task scheduling to better gauge the performances of the algorithms. Besides this, a higher rating should be used for the solar system installed. The current solar system has a power rating of 1.5 kW which does not offset the power demand by much. The difference that the harnessed solar power makes is insignificant compared to the power demand, so a solar system with a higher power rating may have a greater effect. The objective functions should also include minimising the PAR of power demand to get better results on lowering the PAR of power demand. The decisions for battery power discharged to meet power demands should also take into consideration the size of the demand. This is to prevent the battery from discharging excessively which may damage electrical appliances if more than the required power is running through them. The decisions for battery power discharged could also consider the solar power present at the time. This is more for the purpose of reducing the PAR of power demand by correcting the duck-curve [2].

There are plenty of improvements that could be made for future works similar to this. This project has many limitations and faults partly due to the limited time that has been given to carry out this project.

6.3. Code Repository & Video Presentation Links

Code Repository: <https://github.com/snaicker95/Smart-Grid-Algorithms>

Video Presentation Link: <https://youtu.be/qoFacYmfc7c>

7. References

- [1] Smart Grid. (undated). *What is the Smart Grid?* [Online]. Viewed 2018 May 16. Available: https://www.smartgrid.gov/the_smart_grid/smart_grid.html
- [2] RE New Economy. (2018 April). *Will battery storage kill “duck curve” market for gas generators?* [Online]. Viewed 2018 May 16. Available: <https://reneweconomy.com.au/will-battery-storage-kill-duck-curve-market-for-gas-generators-68301/>
- [3] Ausgrid. (undated). *What is Demand Management?* [Online]. Viewed 2018 May 16. Available: <https://www.ausgrid.com.au/Common/Industry/Demand-management/What-is-demand-management.aspx>
- [4] N.Doulamis, P. Kokkinos, E. Varvarigos, “Spectral Clustering Scheduling Techniques for Tasks with Strict QoS Requirements” in *European Conference on Parallel Processing*, Santiago de Compostela, Spain, 2017, pp. 478-488.
- [5] A. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, A. Leon-Garvie, “Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid”, *IEEE Transactions on Smart Grid*, vol. 1, iss. 3, pp. 320-331, Dec 2010.
- [6] P. Samadi, A. Mohsenian-Rad, R. Schober, V. W. S. Wong, J. Jatskevich, “Optimal Real-time Pricing Algorithm Based on Utility Maximization for Smart Grid” in *Smart Grid Communications*, Gaithersburg, MD, USA, 2010.
- [7] Y. Guo, M. Pan, Y. Fang, P. P. Khargonekar, “Decentralized Coordination of Energy Utilization for Residential Households in the Smart Grid”, *IEEE Transactions on Smart Grid*, vol. 4, iss. 3, pp. 1341- 1350, Sep 2013.
- [8] G. Tsousoglou, P. Makris, E. Varvarigos, “Micro grid scheduling policies, forecasting errors, and cooperation based on production correlation” in *International Green Building and Smart Grid*, Prague, Czech Republic, 2016.
- [9] T. Logenthiran, D. Srinivasan, Z. S. Tan, “Demand Side Management in Smart Grid Using Heuristic Optimization”, *IEE Transactions on Smart Grid*, vol. 3, iss. 3, pp. 1244-1252, Sep 2012.
- [10] M. Burcea, W. Hon, H. Liu, P. W. H. Wong, D. K. Y. Yau, “Scheduling for Electricity Cost in Smart Grid” in *Combinatorial Optimizations an Applications*, Chengdu, China, 2013, pp. 306-317.
- [11] F. Albalas, W. Mardini, D. Bani-Salameh, “Optimized Job Scheduling Approach based on Genetic Algorithms in Smart Grid Environment”, *International Journal of Communication Networks and Information Security*, vol. 9, iss. 2, pp. 172-176, Aug 2017.
- [12] S. Chen, P. Sinha, N. B. Schroff, “Heterogeneous Delay Tolerant Task Scheduling and Energy Management in the Smart Grid with Renewable Energy”, *IEEE Journal in Selected Areas of Communication*, vol. 31, iss. 7, pp. 1258-1267, July 2013.
- [13] G. A. Mary, R. Rajarajeswari, “Smart Grid Cost Optimization Using Genetic Algorithm”, *International Journal of Research in Engineering and Technology*, vol. 3, iss. 7, pp. 282-287, May 2014.

- [14] The Conversation. (2015 March). *Why rooftop solar is disruptive to utilities?* [Online]. Viewed 2018 May 16. Available: <http://theconversation.com/why-rooftop-solar-is-disruptive-to-utilities-and-the-grid-39032>
- [15] Energy Use Calculator. (undated). [Online]. Viewed 2018 May 13. Available: <http://energyusecalculator.com/>
- [16] Energy Efficient Strategies, *Energy Use in the Australian Residential Sector 1986-2020 Part 1*, Canberra: Department of the Environment, Water, Heritage and the Arts, 2008, pp. 43-68.
- [17] Energy Use Calculator. (undated). *Electricity Usage of a Microwave*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_microwave.htm
- [18] Energy Use Calculator. (undated). *Electricity Usage of a Clothes Dryer*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_clothesdryer.htm
- [19] Energy Use Calculator. (undated). *Electricity Usage of a Clothes Washer*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_clotheswasher.htm
- [20] Energy Use Calculator. (undated). *Electricity Usage of a CFL Light Bulb*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_cfllightbulb.htm
- [21] Energy Use Calculator. (undated). *Electricity Usage of an Air Conditioner*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_airconditioner.htm
- [22] Energy Use Calculator. (undated). *Electricity Usage of a Space Heater*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_spaceheater.htm
- [23] Energy Use Calculator. (undated). *Electricity Usage of a Laptop, Notebook or Netbook*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_laptop.htm
- [24] Energy Use Calculator. (undated). *Electricity Usage of a Refrigerator*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_refrigerator.htm
- [25] Energy Use Calculator. (undated). *Electricity Usage of a Freezer*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_freezer.htm
- [26] *Understanding Your Energy Usage*, Origin Energy Ltd., S. A., 2015.
- [27] Energy Use Calculator. (undated). *Electricity Usage of a Water Heater*. [Online]. Viewed 2018 May 13. Available: http://energyusecalculator.com/electricity_waterheater.htm
- [28] Your Home. (2013). *Hot Water Service*. [Online]. Viewed 2018 May 14. Available: <http://www.yourhome.gov.au/energy/hot-water-service>
- [29] Energy Matters. (undated). *Information on Australian Solar Feed-In Tariffs*. Viewed 2018 May 14. Available: <https://www.energymatters.com.au/rebates-incentives/feedintariff/#victoria>
- [30] Sustainability Victoria. (undated). *Solar Power*. [Online]. Viewed 2018 May 14. Available: <http://www.sustainability.vic.gov.au/You-and-Your-Home/Save-energy/Solar-power>

- [31] Australian PV Institute. (undated). *Live Solar PV*. [Online]. Viewed 2018 May 14. Available: <http://pv-map.apvi.org.au/live#2018-01-06>
- [32] Alinta Energy. (undated). *Victoria Electricity Pricing*. [Online]. Viewed 2018 May 14. Available: <https://www.alintaenergy.com.au/energy-products/customer-information/victoria-pricing>
- [33] Redback Technologies, “Smart Hybrid Battery Enclosure”, BE13200 Datasheet, 2018 March 1.
- [34] Tesla, “Powerwall”, Tesla Powerwall 2 Datasheet, 2017 April 4.
- [35] Jim’s Energy. (undated). *Residential Energy Storage*. [Online]. Viewed 2018 May 15. Available: <http://www.jimsenergy.com.au/residential-energy-storage/>
- [36] Battery University. (2018 May). *Summary of Do’s and Don’ts*. [Online]. Viewed 2018 May 16. Available: http://batteryuniversity.com/learn/article/do_and_dont_battery_table
- [37] MathWorks. (undated). *Genetic Algorithm*. [Online]. Viewed 2018 May 17. Available: <https://au.mathworks.com/discovery/genetic-algorithm.html>
- [38] MathWorks. (undated). *How the Genetic Algorithm Works?* [Online]. Viewed 2018 May 19. Available: <https://au.mathworks.com/help/gads/how-the-genetic-algorithm-works.html#f6243>
- [39] MathWorks. (undated). *Global vs Local Minima Using ga*. [Online]. Viewed 2018 May 19. Available: <https://au.mathworks.com/help/gads/example-global-vs-local-minima-with-ga.html>
- [40] MathWorks. (undated). *ga*. [Online]. Viewed 2018 May 20. Available: https://au.mathworks.com/help/gads/ga.html#mw_4a8bfdb9-7c4c-4302-8f47-d260b7a43e26
- [41] MathWorks. (undated). *Genetic Algorithm Options*. [Online]. Viewed 2018 May 20. Available: <https://au.mathworks.com/help/gads/genetic-algorithm-options.html#f6633>
- [42] MathWorks. (undated). *Tic*. [Online]. Viewed 2018 May 23. Available: <https://au.mathworks.com/help/matlab/ref/tic.html>
- [43] MathWorks. (undated). *Toc*. [Online]. Viewed 2018 May 23. Available: <https://au.mathworks.com/help/matlab/ref/toc.html>
- [44] MathWorks. (undated). *Measure Performance of Your Program*. [Online]. Viewed 2018 May 23. Available: https://au.mathworks.com/help/matlab/matlab_prog/measure-performance-of-your-program.html
- [45] Techopedia. (undated). *CPU Time*. [Online]. Viewed 2018 May 23. Available: <https://www.techopedia.com/definition/2858/cpu-time>

8. Appendix

The appendix contains all the journal articles, conference papers and datasheets used for this project. The order of the documents are based as ordered in the reference list.

Spectral Clustering Scheduling Techniques for Tasks with Strict QoS Requirements

Nikos Doulamis, Panagiotis Kokkinos, and Emmanouel Varvarigos

Department of Computer Engineering and Informatics, University of Patras and
Research Academic Computer Technology Institute, Patras, Greece
ndoulam@cs.ntua.gr

Abstract. Efficient task scheduling is fundamental for the success of the Grids, since it directly affects the Quality of Service (QoS) offered to the users. Efficient scheduling policies should be evaluated based not only on performance metrics that are of interest to the infrastructure side, such as the Grid resources utilization efficiency, but also on user satisfaction metrics, such as the percentage of tasks served by the Grid without violating their QoS requirements. In this paper, we propose a scheduling algorithm for tasks with strict timing requirements, given in the form of a desired start and finish time. Our algorithm aims at minimizing the violations of the time constraints, while at the same time minimizing the number of processors used. The proposed scheduling method exploits concepts derived from spectral clustering, and groups together for assignment to a computing resource the tasks so to a) minimize the time overlapping of the tasks assigned to a given processor and b) maximize the degree of time overlapping among tasks assigned to different processors. Experimental results show that our proposed strategy outperforms greedy scheduling algorithms for different values of the task load submitted.

1 Introduction

Task scheduling is fundamental for the success of the Grids, especially with regards to its ability to support real-life commercial applications, by directly affecting a) the efficiency with which Grid resources are used and b) the Quality of Service (QoS) offered to the users. Evaluating a task scheduling algorithm should be based not only on resource utilization metrics, but also on user satisfaction factors, such as the percentage of tasks that are served by the Grid without violating their QoS requirements, e.g., their start and finish times [1]. Currently, several open source schedulers have been developed for clusters of servers such as Maui [2] and portable batch system (PBS) [3]. However, the primary objective of most existing approaches is to improve overall system performance (e.g., resource utilization), while the QoS experienced by Grid users is, at best, a secondary consideration [4], [5].

Without QoS guarantees (e.g., given in the form of task deadlines that should not be violated), users may be reluctant to pay for Grid services or contribute resources to the Grid, reducing its economic impact. On the other hand, designing a scheduling algorithm that satisfies only the end-to-end users' QoS, without taking into account Grid utilization efficiency, would result in a wasteful Grid architecture, that uses

many more processors than necessary in order to satisfy users' QoS requirements. Equivalently, processor utilization would be relatively small, meaning that only a small percentage of the available resources would be exploited. Therefore, we need scheduling and resource allocation schemes that are able to simultaneously meet these two sometimes contradictory requirements: *optimize Grid utilization efficiency while simultaneously guaranteeing the tasks' strict QoS requirements* (e.g., deadlines).

Several computing toolkits and systems have been developed to meet the task QoS requirements in a Grid computing architecture. Globus is probably the most well known [6]. Additionally Condor-G is an enhanced version of Condor that uses the Globus toolkit to manage Grid jobs [7]. The Nimrod-G [8] is a Grid aware version of the Nimrod, which provides a simple declarative parametric modeling language for expressing a parametric experiment. A dynamic Grid resource allocation method is adopted in [9] using market economy notions (the G-commerce architecture). Finally, a new scheduling algorithm developed in the framework of the GrADS (Grid Application Development Software) tool has been proposed in [10]. A survey of state of the art methods for Grid scheduling is presented in [11].

In general, scheduling parallel and distributed applications is a known NP-complete problem. For this reason, several heuristic algorithms have been proposed for task scheduling. Some approaches use genetic algorithms to maximize the overall system performance [12], [13], while others use Directed Acyclic Graphs (DAG) for scheduling on heterogeneous or homogeneous computing environments [14], [15]. Performance evaluation results for these algorithms are presented in [16]. However, all the aforementioned approaches try to maximize overall system performance, (that is, Grid resource utilization), without respecting task deadlines (that is, user's QoS). Advance reservation of resources, which is the ability of the scheduler to guarantee the availability of resources at a particular time in the future, is one mechanism Grid providers may employ in order to offer specific QoS guarantees to the users [4]. However, these algorithms lack scalability, as they are unable to efficiently perform task scheduling in short time for large numbers of Grid resources. Using concepts from computational geometry, [1] solves the scalability problem for task scheduling under a user's satisfaction framework. The scalability problem is also addressed in [17]. Furthermore, fair scheduling algorithms and reservation schemes have been discussed in [5].

The main drawback of the above mentioned approaches is that scheduling is performed either in the direction of maximizing overall system performance (resource utilization efficiency) or minimizing the degradation of user's QoS requirements satisfaction. As mentioned before, a successful Grid scheduling algorithm should actually take into account both directions. This problem is addressed in this paper, by proposing a novel task scheduling algorithm that assigns tasks to processors so that a) *the time overlapping between tasks assigned to the same processor are minimized* (users QoS requirements are met to the degree possible), while simultaneously b) *maximizing overall Grid utilization efficiency*.

As we show in this paper, the two above mentioned objectives can be described by a matrix representation and then the proposed optimal scheduling strategy can be obtained by introducing the notions of generalized eigenvalues through the use of the Ky-Fan theorem [19]. The Ky-Fan theorem states that an optimal schedule that satisfies both aforementioned criteria can be derived as a solution of the largest

eigenvectors of the two matrices that represent the two conditions. Therefore, we have a scheduling algorithm of polynomial order with respect to the number of tasks, that simultaneously satisfies the users' QoS and the system's performance conditions.

The paper is organized as follows. Section 2 discusses the proposed scheduling algorithm for jointly optimizing resource utilization efficiency and tasks' QoS requirements. The solution of the joint optimization problem is given in Section 3. In Section 4, we discuss a lower bound on the number of processors required to achieve no task overlapping (no QoS violations) and propose objective criteria for evaluating scheduling efficiency. Experimental results and comparisons with other approaches are given in Section 5, while Section 6 concludes the paper.

2 Joint Optimization of Resource Performance and QoS Requirements

Let us denote by T_i , $i=1,2,\dots,N$, the tasks that request service in a Grid infrastructure consisting of M processors. Let us also denote by ST_i the desired *Start Time* for Task T_i and by FT_i its desired *Finish Time*. In this paper, we assume that the tasks are scheduled in a *non-preemptable, non-interruptible way*. Under this assumption, if a task has been assigned for execution on a processor and another task requests service on an overlapping time interval, then, the second task should either be assigned to another processor (which is not reserved at the requested time interval) or undergo violation of its QoS, i.e., its start or finish time or both of them.

We denote by σ_{ij} the non-overlapping measure between tasks T_i and T_j . Assuming that the task i Start and Finish Time, ST_i and FT_i , are hard constraints that should not be violated, a proper selection for the non-overlapping measure σ_{ij} is to take zero values when tasks T_i and T_j overlap in time and positive non-zero values when they do not overlap.

$$\sigma_{ij} = \begin{cases} \alpha, & \text{if } T_i, T_j \text{ are non-overlapping in time} \\ 0, & \text{if } T_i, T_j \text{ overlap in time} \end{cases} \quad (1)$$

where $\alpha > 0$.

Let us assume, without loss of generality that the *Start time* ST_i and *Finish time* FT_i for all tasks that are to be scheduled are within a *time horizon* T , which can be considered as the time interval within which one instance of the scheduling algorithm is executed. Let us denote by C_r the set of tasks assigned for execution on processor r .

As stated in Section 1, an efficient scheduling scheme for a commercially successful Grid should assign all the N pending tasks to the M processors so as to a) *minimize the tasks' QoS violations*, while simultaneously b) maximizing the *overall utilization* of the M processors, so that the Grid resources do not stay *idle* most of the time. The first requirement, in terms of the scheduling algorithm, means that the tasks assigned to a *given* processor should present *minimal overlapping*. The second requirement

indicates that the task overlapping among *different* processors should be *maximized*, that is, the utilization of all processors in Grid should be as high as possible. These two requirements can be written as

$$Q_r = \frac{\sum_{i \in C_r, j \in C_r} \sigma_{ij}}{\sum_{i \in C_r, j \in V} \sigma_{ij}}, \quad P_r = \frac{\sum_{i \in C_r, j \notin C_r} \sigma_{ij}}{\sum_{i \in C_r, j \in V} \sigma_{ij}}, \quad (2)$$

where $V = \{T_i\}_{i=1, \dots, N}$ the set of tasks that request service in a Grid infrastructure.

The denominator of equations (2) is used for normalization purposes. Otherwise, optimizing would favor the trivial solution of one task per processor. Parameter Q_r expresses a measure of the overall QoS violation for the tasks assigned to the r^{th} processor. Instead, parameter P_r expresses the Grid utilization. Taking into account all the M processors of the Grid, we can define a measure Q for the total tasks' QoS violation and a measure P for the overall processor utilization as

$$Q = \sum_{r=1}^M Q_r, \quad P = \sum_{r=1}^M P_r. \quad (3)$$

An efficient scheduler that tries to meet user QoS requirements should maximize Q and simultaneously minimize P . However, it is clear that

$$P + Q = M. \quad (4)$$

Equation (4) shows that the *maximization* of Q simultaneously yields a *minimization* of P and vice versa. Hence, in our problem, the two aforementioned optimization objectives require in fact the use of identical means and they can be met simultaneously. This is intuitively satisfying, since scheduling a set of tasks in a way that makes efficient use of resources is also expected to help meet the QoS requirements of the set of tasks that are scheduled. Therefore, it is enough to optimize (maximize or minimize) only one of the two criteria. In our case, we select to minimize variable P . Thus,

$$\hat{C}_r : \min P = \min \sum_{r=1}^M \frac{\sum_{i \in C_r, j \notin C_r} \sigma_{ij}}{\sum_{i \in C_r, j \in V} \sigma_{ij}}, \text{ for all } r=1, \dots, M, \quad (5)$$

where \hat{C}_r is the set of tasks assigned for execution on processor r .

3 The Proposed Task Scheduling Policy

3.1 Matrix Representation

Optimizing equation (5) is still a NP-complete problem, even for the special case of $M=2$ processors. To overcome this difficulty, we transform the problem of (5) into a matrix based representation. Let us denote by $\Sigma = [\sigma_{ij}]$ a matrix which contains the values of the non-overlapping measures σ_{ij} for all $N \times N$ combinations of tasks T_i and

T_j . Let us now denote as $\mathbf{e}_r = [\dots e_r^u \dots]^T$ an $N \times 1$ *indicator vector* whose u -th entry is given by

$$e_r^u = \begin{cases} 1, & \text{if Task } T_u \text{ is assigned to processor } r \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

The indicator vector \mathbf{e}_r points out which of the N tasks are executed on processor r . That is, indices of tasks executed on processor r are marked with one, while the remaining indices take zero values. Since the Grid infrastructure consists of M processors, M different indicator vectors \mathbf{e}_r , $r = 1, 2, \dots, M$ are defined, each indicating the tasks assigned for execution on each processor. This way, we can express the left hand of (5) with respect to vectors \mathbf{e}_r , $r = 1, 2, \dots, M$. However, we also need to express the right hand of (5) as a function of \mathbf{e}_r . For this reason, we denote by $\mathbf{L} = \text{diag}(\dots l_i \dots)$ the diagonal matrix, whose elements l_i express the cumulative non-overlapping degree of task T_i with the remaining tasks. That is,

$$l_i = \sum_j \sigma_{ij}. \quad (7)$$

Using matrices \mathbf{L} and Σ , we can express equation (5) as,

$$\hat{\mathbf{e}}_r, \forall r : \min P = \min \sum_{r=1}^M \frac{\mathbf{e}_r^T (\mathbf{L} - \Sigma) \mathbf{e}_r}{\mathbf{e}_r^T \mathbf{L} \mathbf{e}_r}. \quad (8)$$

3.2 Optimization in the Continuous Domain

Let us form the indicator matrix $\mathbf{E} = [e_1 \dots e_M]$, the columns of which correspond to the M processors in the Grid, while the rows to the N tasks, then the rows of \mathbf{E} have only one unit entry and the remaining entries are zero. Optimization of (8) under the discrete representation of matrix \mathbf{E} is still a NP hard problem. However, if we relax the indicator matrix \mathbf{E} to take values in the continuous domain, we can solve the problem in polynomial time. We denote by \mathbf{E}_R the *relaxed version of the indicator matrix \mathbf{E}* , i.e. a matrix whose rows take real values instead of binary values as is the case of the indicator matrix \mathbf{E} . Then, it can be proven that (8) can be rewritten as

$$P = M - \text{trace}(\mathbf{Y}^T \mathbf{L}^{-1/2} \Sigma \mathbf{L}^{-1/2} \mathbf{Y}), \quad (9a)$$

$$\text{subject to } \mathbf{Y}^T \mathbf{Y} = \mathbf{I}. \quad (9b)$$

\mathbf{Y} is a matrix that is related to the matrix \mathbf{E}_R through

$$\mathbf{L}^{-1/2} \mathbf{Y} = \mathbf{E}_R \boldsymbol{\Lambda}, \quad (10)$$

where $\boldsymbol{\Lambda}$ is any $M \times M$ matrix. In this paper, we select $\boldsymbol{\Lambda}$ to be equal to the identity matrix, $\boldsymbol{\Lambda} = \mathbf{I}$. Then, the relaxed indicator matrix \mathbf{E}_R is given as

$$\mathbf{E}_R = \mathbf{L}^{-1/2} \mathbf{Y}. \quad (11)$$

Minimization of (11) is obtained through the Ky-Fan theorem [19]. The Ky-Fan theorem states that the maximum value of the $\text{trace}(\mathbf{Y}^T \mathbf{L}^{-1/2} \boldsymbol{\Sigma} \mathbf{L}^{-1/2} \mathbf{Y})$ subject to the constraint of $\mathbf{Y}^T \mathbf{Y} = \mathbf{I}$ is equal to the sum of the M ($M < N$) largest eigenvalues of matrix $\mathbf{L}^{-1/2} \boldsymbol{\Sigma} \mathbf{L}^{-1/2}$. This maximum value is provided for the matrix

$$\mathbf{Y} = \mathbf{U} \cdot \mathbf{R}, \quad (12)$$

where \mathbf{U} is a $N \times M$ matrix whose columns are the eigenvectors corresponding to the M largest eigenvalues of matrix $\mathbf{L}^{-1/2} \boldsymbol{\Sigma} \mathbf{L}^{-1/2}$ and \mathbf{R} is an arbitrarily rotation matrix (i.e., orthogonal with determinant of one). Again, a simple approach is to select matrix \mathbf{R} as the identity matrix, $\mathbf{R} = \mathbf{I}$, that is $\mathbf{Y} = \mathbf{U}$. Therefore, we have that the optimal relaxed indicator matrix $\hat{\mathbf{E}}_R$ in the continuous domain is given as

$$\hat{\mathbf{E}}_R = \mathbf{L}^{-1/2} \mathbf{U}. \quad (13)$$

3.3 Discrete Approximation

The optimal matrix $\hat{\mathbf{E}}_R$ given by equation (13) does not have the form of the indicator matrix \mathbf{E} since the values of $\hat{\mathbf{E}}_R$ are continuous, while \mathbf{E} 's entries are binary. Recall that a unit entry indicates the processor a task is assigned to for execution, under the non-interruptible, non-preemptable assumption. Consequently, the problem is how to round the continuous values of $\hat{\mathbf{E}}_R$ in a discrete form that approximates matrix \mathbf{E} .

One simple approach, regarding the rounding process, is to set the maximum value of each row of matrix $\hat{\mathbf{E}}_R$ to be equal to 1 and let the remaining values be equal to 0. However, such an approach yields unsatisfactory performance when there is no dominant maximum value for each row of $\hat{\mathbf{E}}_R$. Furthermore, it handles the rounding process as N (that is the number of tasks) independent problems. An alternative approach, which is adopted in this paper, is to treat the N rows of matrix $\hat{\mathbf{E}}_R$ as M -dimensional feature vectors. Each one of these feature vectors indicates the association degree of each task and the respective M processor of the Grid. Then, we apply the k-means to form the indicator matrix \mathbf{E} .

4 Lower Bound - Scheduling Efficiency

An important aspect, which determines the scheduling efficiency is the task granularity g , and the task arrival rate λ defined as

$$\lambda = \frac{N}{T}, \quad g = \frac{D}{T}, \quad (14)$$

where N is the number of tasks requesting service over the corresponding time interval T and D the average task delay.

Given a granularity g and a rate λ , the *lower bound* of Grid resources required for achieving no task overlapping is given by the following equation

$$B = \frac{ND}{T} = N \cdot g \leq M_{opt}, \quad (15)$$

where M_{opt} refers to minimum number of processors required for achieving no task overlapping by an optimal (exhaustive search) scheduling algorithm. The lower bound of (15) is achieved in the extreme case that the tasks request execution intervals of a constant duration D that appear one right after the other, completely filling the gaps within the time horizon T . Thus, this lower bound is usually smaller than the M_{opt} .

Given the lower bound B on the number of processors required for no overlapping, the scheduling efficiency is defined as

$$e(A) = \frac{B}{M(A)} \text{ or } \varepsilon(A) = \frac{\lceil B \rceil}{M(A)}, \quad (16)$$

where A refers to the algorithm used to approximate the exhaustive search policy and $M(A)$ is the number of processors required for achieving no task overlapping when algorithm A is used. $e(A)$ is the scheduling efficiency, while $\varepsilon(A)$ is the rounded efficiency for algorithm A . The $\lceil \cdot \rceil$ indicates the ceil operator.

5 Experimental Results

Two different algorithms were implemented in this paper and compared with respect to their scheduling efficiency. The first algorithm is the *proposed scheme*, presented in Section 2. The second scheme is a *greedy approach*, which, for each task, a locally optimum choice is selected. In particular, the algorithm assigns each task to a processor, so that no task overlapping is encountered, by taking into account the current local load of the processors.

Our proposed algorithm is recursively applied assuming different number of processors in Grid. Then, we select the minimum number of processors that provide *no task overlapping* that is no violation of the tasks' QoS. This number $M(\text{Proposed Algorithm})$ is used for evaluating the scheduling efficiency. In the greedy algorithm, each time a newly considered task overlaps with the already assigned tasks, then a new resource is activated and this task is assigned to this resource. The number of resources that have been activated after scheduling all tasks, without overlaps, is denoted by $M(\text{Greedy})$. We assume that the tasks' *Start and Finish Times* ST_i and FT_i are uniformly distributed within the time horizon T and that the average tasks' duration is constant and equals D . Experiments where the task duration varies significantly from task to task, have also been performed, but are not included here due to space limitations.

Fig. 1(a) presents the efficiency e [see equation (16)] versus the task granularity g for different values of lower bound B . As is observed, the efficiency increases as the granularity decreases for low values of g . However, the ratio of improvement decreases, meaning that the efficiency converges as g increases. We also observe from

Fig. 1 that for values of granularity greater than $g \geq 0.2$ the efficiency also increases as g increases. This is due to singularity issue, since in this case the minimum number of processors required for achieving no task overlapping equals the number of tasks N . In Fig. 1(b), we compare the continuous and rounded efficiency e and ε for the lower bound $B=1$. As expected, the rounded efficiency is a discontinuous function and several peaks are encountered due to the ceiling operator $\lceil \cdot \rceil$ involved in (16).

However, in general terms, the overall behavior resembles that of the continuous case.

In Fig. 2(a), we depict the effect of the number of tasks N (equivalently, of the task arrival rate λ , for a given time window T) on the efficiency ε for different values of the granularity g . As we observe, the rounded efficiency presents a periodically discontinuous behavior that depends on the granularity value. This periodicity is due to the ceiling operator involved in the rounded efficiency ε [see equation (16)]. Next, we examined the effect of the number of iterations of the k -means algorithm used for estimating the indicator matrix \mathbf{E} –that is tasks' partitioning– from the relaxed matrix

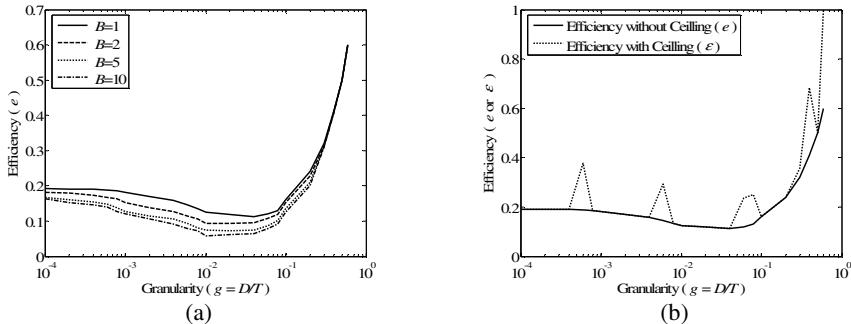


Fig. 1. (a) Efficiency (e) versus granularity (g) for different values of lower bound B in case that the proposed scheduling policy is used. (b) Comparison of the continuous and rounded efficiency (e and ε) for lower bound $B=1$.

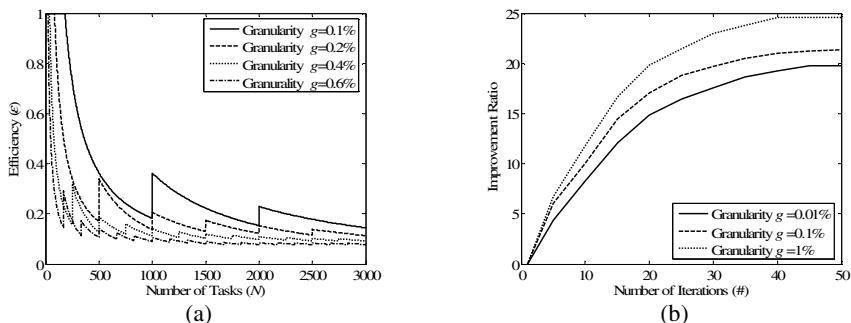


Fig. 2. (a) Efficiency ε versus the number of tasks for different values of granularity in case of $B=1$. (b) Improvement ratio of the efficiency versus the number of iterations of the proposed algorithm.

E_R computing by the Ky-Fan Theorem (see Section 3). In particular, Fig. 2(b) presents the improvement ratio versus the number of iterations for different granularity values, assuming $B=1$. We observe that as the number of iterations increases the scheduling efficiency increases for all granularity values. However, convergent is achieved for large number of iterations.

Fig. 3 presents the comparison results between the proposed algorithm for iterations of 1 and 50 and the greedy scheduling scheme. As we observe, the proposed algorithm exhibits better efficiency at any value of granularity. At low task load (low values of B) the improvement is more evident than for high values of B . Additionally, for low granularity values the improvement is smaller. This is because in this case, task durations are very small compared to the time window and thus both algorithms can schedule more effectively the tasks.

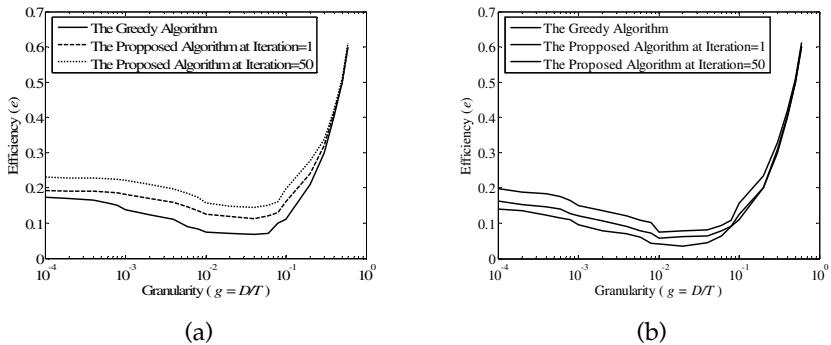


Fig. 3. Comparison of the proposed method for different iterations with the greedy algorithm. (a) $B=1$. (b) $B=10$.

6 Conclusions

We proposed an efficient scheduling strategy that maximizes Grid utilization efficiency, while resulting in a minimal degradation of the QoS offered to the submitted tasks. These two objectives are transformed into a matrix representation and then the scheduling problem is solved by introducing the notions of generalized eigenvalues through the use of the Ky-Fan theorem. Optimization using eigenvectors has the advantage that scheduling is performed in polynomial complexity.

Experimental results and comparisons with a greedy scheduling policy are presented to indicate the efficiency of the proposed scheme. In particular, we investigate the number of processors required for achieving no task overlapping (no QoS violations) under the two scheduling policies. We also define a lower bound on the minimum number of processors required and we estimate the scheduling efficiency of an algorithm as the ratio of the lower bound over the number of processors achieved by the algorithm. Comparison with the greedy scheduling policy demonstrates the efficiency of the proposed scheme for all granularities and different assumptions on the number and durations of the tasks. In addition, as the number of iterations of the proposed algorithm increases better scheduling efficiency is achieved. Algorithm

convergence is achieved even for a small number of iterations, e.g., 30. We find that task granularity affects more significantly the scheduling efficiency rather than the task arrival rate. Finally, efficiency is better at low values of granularity, however, convergence is noticed for very low granularities.

Acknowledgment

This work has been supported by the European Commission through the IP Phosphorus project.

References

- [1] Castillo, C., Rouskas, G., Harfoush, K.: On the Design of Online Scheduling Algorithms for Advance Reservations and QoS in Grids. In: Int'l Parallel and Distributed Processing Symp., pp. 1–10 (2007)
- [2] Jackson, D., Snell, Q., Clement, M.: Core Algorithms of the Maui Scheduler. In: Feitelson, D.G., Rudolph, L. (eds.) JSSPP 2001. LNCS, vol. 2221, pp. 87–102. Springer, Heidelberg (2001)
- [3] Bode, B., et al.: The Portable Batch Scheduler and the Maui Scheduler on Linux Clusters. Usenix Conf. (2000)
- [4] Al-Ali, R.J., et al.: Analysis and provision of QoS for distributed grid applications. Journal of Grid Computing 2(2), 163–182 (2004)
- [5] Doulamis, N., Doulamis, A., Varvarigos, E., Varvarigou, T.: Fair Scheduling Algorithms in Grids. IEEE Trans. on PDS 18(11), 1630–1648 (2007)
- [6] Foster, I., Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications 11(2), 115–128 (1997)
- [7] Thain, D., Tannenbaum, T., Livny, M.: Condor and the Grid. In: Berman, F., Hey, A.J.G., Fox, G. (eds.) Grid Computing: Making the Global Infrastructure a Reality. John Wiley & Sons, Chichester (2003)
- [8] Abramson, D., Giddy, J., Kotler, L.: High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid. In: Int'l Parallel and Distributed Processing Symp. (2000)
- [9] Wolski, R., Plank, J.S., Brevik, J., Bryan, T.: G-commerce: Market Formulations Controlling Resource Allocation on the Computational Grid. In: Int'l Parallel and Distributed Processing Symp. (2001)
- [10] K. Cooper et al., “New Grid Scheduling and Rescheduling Methods in the GrADS Project,” Int'l Parallel and Distributed Processing Symp., pp. 199–207, 2004.
- [11] Maheswaran, M., Krauter, K., Buyya, R.: A taxonomy and survey of grid resource management systems for distributed computing. Software: Practice and Experience 32(2), 135–164 (2002)
- [12] Shu, W., et al.: A Grid Computing Task Scheduling Method Based on Target Genetic Algorithm. The Sixth World Congress on Intelligent Control and Automation 1, 3528–3532 (2006)
- [13] Ye, G., Rao, R., Li, M.: A Multiobjective Resources Scheduling Approach Based on Genetic Algorithms in Grid Environment. In: Fifth International Conference on Grid and Cooperative Computing Workshops, pp. 504–509 (2006)

- [14] Topcuoglu, H., Hariri, S., Wu, M.-Y.: Performance Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. *Transactions on Parallel and Distributed Systems* 2(13), 260–274 (2002)
- [15] Mandal, A., et al.: Scheduling Strategies for Mapping Application Workflows onto the Grid. In: Symp. on High Performance Distributed Computing, pp. 125–134 (2005)
- [16] Zhang, Y., Koelbel, C., Kennedy, K.: Relative Performance of Scheduling Algorithms in Grid Environments. In: Int'l Conf. on Cluster Computing and the Grid, pp. 521–528 (2007)
- [17] Zhang, Y., et al.: Scalable Grid Application Scheduling via Decoupled Resource Selection and Scheduling. In: Int'l Conf. on Cluster Computing and the Grid, pp. 568–575 (2006)
- [18] Varvarigos, E., Doulamis, N., Doulamis, A., Varvarigou, T.: Timed/Advance Reservation Schemes and Scheduling Algorithms for QoS Resource Management in Grids. In: Di Martino, B., Dongarra, J., Hoisie, A., Yang, L.T., Zima, H. (eds.) *Engineering the Grid*, pp. 355–378. American Scientific Publishers (2006)
- [19] Nakic, I., Veselic, K.: Wielandt and Ky-Fan Theorem for Matrix Pairs. *Linear Algebra and its Applications* 369(17), 73–77 (2003)

Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid

Amir-Hamed Mohsenian-Rad, *Member, IEEE*, Vincent W. S. Wong, *Senior Member, IEEE*, Juri Jatskevich, *Senior Member, IEEE*, Robert Schober, *Fellow, IEEE*, and Alberto Leon-Garcia, *Fellow, IEEE*

Abstract—Most of the existing demand-side management programs focus primarily on the interactions between a utility company and its customers/users. In this paper, we present an autonomous and distributed demand-side energy management system among users that takes advantage of a two-way digital communication infrastructure which is envisioned in the future smart grid. We use game theory and formulate an energy consumption scheduling game, where the players are the users and their strategies are the daily schedules of their household appliances and loads. It is assumed that the utility company can adopt adequate pricing tariffs that differentiate the energy usage in time and level. We show that for a common scenario, with a single utility company serving multiple customers, the global optimal performance in terms of minimizing the energy costs is achieved at the Nash equilibrium of the formulated energy consumption scheduling game. The proposed distributed demand-side energy management strategy requires each user to simply apply its best response strategy to the current total load and tariffs in the power distribution system. The users can maintain privacy and do not need to reveal the details on their energy consumption schedules to other users. We also show that users will have the incentives to participate in the energy consumption scheduling game and subscribing to such services. Simulation results confirm that the proposed approach can reduce the peak-to-average ratio of the total energy demand, the total energy costs, as well as each user's individual daily electricity charges.

Index Terms—Demand-side management, distributed algorithms, energy consumption scheduling, energy pricing, game theory, market incentives, smart grid, smart meter.

I. INTRODUCTION

D EMAND-SIDE management (DSM) commonly refers to programs implemented by utility companies to control the energy consumption at the customer side of the meter

Manuscript received April 07, 2010; revised July 05, 2010; accepted August 15, 2010. Date of current version November 19, 2010. Paper no. TSG-00045-2010.

A. H. Mohsenian-Rad was with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, V6T 1Z4 and with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, M5S 2E4. He is now with the Department of Electrical and Computer Engineering, Texas Tech University, Lubbock, TX 79409, USA (e-mail: hamed.mohsenian-rad@ttu.edu).

V. W. S. Wong, J. Jatskevich, and R. Schober are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, V6T 1Z4 (e-mail: vincentw@ece.ubc.ca; juri@ece.ubc.ca; rschober@ece.ubc.ca).

A. Leon-Garcia is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, M5S 2E4 (e-mail: alberto.leongarcia@utoronto.ca).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSG.2010.2089069

[1]. These programs are employed to use the available energy more efficiently without installing new generation and transmission infrastructure. DSM programs include conservation and energy efficiency programs, fuel substitution programs, demand response programs, and residential or commercial load management programs [2]–[4]. Residential load management programs usually aim at one or both of the following design objectives: *reducing consumption* and *shifting consumption* [5]. The former can be achieved among users by encouraging energy-aware consumption patterns and by constructing more energy efficient buildings. However, there is also a need for practical solutions to shift the high-power household appliances to off-peak hours to reduce the peak-to-average ratio (PAR) in load demand. Appropriate load-shifting is foreseen to become even more crucial as plug-in hybrid electric vehicles (PHEVs) become popular. Most PHEVs need 0.2–0.3 KWh of charging power for one mile of driving [6]. This will represent a significant new load on the existing distribution system. In particular, during the charging time, the PHEVs can almost double the average household load and drastically exacerbate the already high PAR. Moreover, unbalanced conditions resulting from an increasing number of PHEVs may lead to further degradation of the power quality, voltage problems, and even potential damage to utility and consumer equipment if the system is not properly reinforced [6].

One approach in residential load management is direct load control (DLC) [7]–[10]. In DLC programs, based on an agreement between the utility company and the customers, the utility or an aggregator, which is managed by the utility, can remotely control the operations and energy consumption of certain appliances in a household. For example, it may control lighting, thermal comfort equipment (i.e., heating, ventilating, and air conditioning), refrigerators, and pumps. However, when it comes to residential load control and home automation, users' privacy can be a major concern and even a barrier in implementing DLC programs [11].

An alternative for DLC is smart pricing, where users are encouraged to individually and voluntarily manage their loads, e.g., by reducing their consumption at peak hours [12]–[14]. In this regard, critical-peak pricing (CPP), time-of-use pricing (ToUP), and real-time pricing (RTP) are among the popular options. For example, in RTP tariffs, the price of electricity varies at different hours of the day. The prices are usually higher during the afternoon, on hot days in the summer, and on cold days in the winter [15]. RTP programs have been adopted in some places in

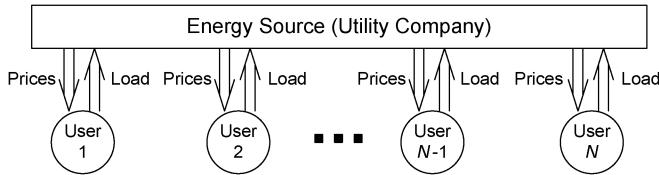


Fig. 1. A demand-side management (DSM) strategy focused on individual interactions between the utility and each user.

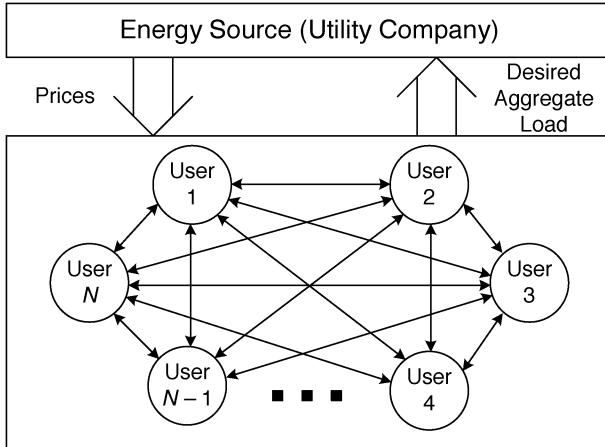


Fig. 2. A demand-side management strategy for the smart grid with enabled interactions among the users/customers and the utility company.

North America, e.g., by the Illinois Power Company in Chicago [15]. While it is usually difficult and confusing for the users to manually respond to prices that are changing every hour [16], [17], another problem that RTP may face is load synchronization, where a large portion of load is shifted from a typical peak hour to a typical nonpeak hour, without significantly reducing the PAR [18].

In most of the DSM programs that have been deployed over the past three decades (e.g., in [7]–[14]), the key focus has been on interactions between the utility company and each end user. For example, in RTP programs, each user is expected to individually respond to the time-differentiated prices by shifting its own load from the high price hours to the low price hours. Under this paradigm, each customer communicates with the energy source individually as depicted in Fig. 1. However, we argue in this paper that such an approach to the residential load control may not always achieve the best solution to the energy consumption problem. Instead, rather than focusing only on how each user behaves individually, a good DSM program should have the objective that the aggregate load satisfies some desired properties. For example, only the total load at each hour is important when it comes to solving the economic dispatching problem [19]. Also the PAR depends only on the total load demand. Therefore, while it is useful to employ aggregator units for load shaping [20], it is also important to design more efficient residential load management strategies that work by enabling interactions among users via message exchange as depicted in Fig. 2. If the users are provided with sufficient incentives, they can coordinate their usage to reduce the PAR or minimize the energy cost. Due to the recent advancements in smart grid technologies [21]–[24], the interactions between users do

not have to be manual, but can be automatic through two-way digital communication.

In this paper, we propose an incentive-based energy consumption scheduling scheme for the future smart grid. We consider a scenario where a source of energy (e.g., a generator or a step-down substation transformer which is connected to the grid) is shared by several customers, each one of which is equipped with an automatic energy consumption scheduler (ECS). The ECS functionality is deployed inside the smart meters that are connected to not only the power line, but also to a communication network. The smart meters with ECS functions interact automatically by running a distributed algorithm to find the optimal energy consumption schedule for each user. The optimization objective is to minimize the energy cost in the system. As can be shown with a game-theoretic analysis [25], a simple pricing mechanism can provide the users with the incentives to cooperate. The overall system performance is improved. Each user also pays less. In other words, through an appropriate pricing scheme, the Nash equilibrium of the energy consumption game among the participating users who share the same energy source is the optimal solution of a system-wide optimization problem.

The discussions and analysis in this paper extend the preliminary results in our earlier conference paper in [26] in various directions. First, here we consider not only the energy cost minimization problem but also the problem of minimizing the PAR in the total load. In this regard, we also explain the relationship between the two problems. Second, our game-theoretic analysis in this paper is more elaborate and includes new discussions on strategy-proof properties of the proposed algorithm, i.e., the ability to prevent users from cheating and misleading during their interactions with each other. Finally, the simulation results in this paper are more extensive and further include a detailed assessment of the convergence and optimality properties of our proposed algorithm, enable a better understanding of the relationship between PAR in the total load demand and the PAR in each user's individual load, and highlight the impact of changes in the number of appliances to energy consumption scheduling.

The rest of this paper is organized as follows. We introduce the system model in Section II. The PAR and energy cost minimization problems are formulated in Section III. The energy consumption games are introduced in Section IV. A distributed DSM algorithm is presented in Section V. Simulation results are given in Section VI. The paper is concluded in Section VII. All analytical proofs are summarized in Appendixes A–D.

II. SYSTEM MODEL

In this section, we provide analytical descriptions for the representation of the power system, the energy cost, and residential load control. Based on these definitions, we will formulate two design optimization problems in Section III.

A. Power System

Consider a smart power system with multiple load customers and one energy source, e.g., a generator or a step-down substation transformer connected to the electric grid. The block diagram of such a power distribution system is shown in Fig. 3.

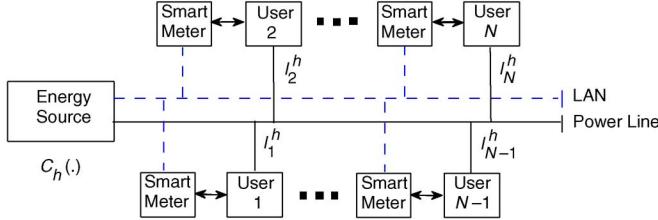


Fig. 3. Block diagram of smart grid system composed of an energy source, users, a distribution power line, and a local area communication network.

We assume that each customer is equipped with a smart meter that has an ECS capability for scheduling the household energy consumption. The smart meters are all connected to the power line coming from the energy source. They are also connected to each other and to the energy source through a local area network (LAN). All communications between the utility and the customers' smart meters and all message exchanges among the smart meters are done through the LAN by using appropriate communication protocols.

Throughout the paper, let \mathcal{N} denote the set of users, where the number of users is $N \triangleq |\mathcal{N}|$. For each customer $n \in \mathcal{N}$, let I_n^h denote the total load at hour $h \in \mathcal{H} \triangleq \{1, \dots, H\}$, where $H = 24$. Without loss of generality, we assume that time granularity is one hour. The daily load for user n is denoted by $\mathbf{I}_n \triangleq [I_n^1, \dots, I_n^H]$. Based on these definitions, the total load across all users at each hour of the day $h \in \mathcal{H}$ can be calculated as

$$L_h \triangleq \sum_{n \in \mathcal{N}} I_n^h. \quad (1)$$

The daily peak and average load levels are calculated as

$$L_{\text{peak}} = \max_{h \in \mathcal{H}} L_h \quad (2)$$

and

$$L_{\text{avg}} = \frac{1}{H} \sum_{h \in \mathcal{H}} L_h \quad (3)$$

respectively. Therefore, the PAR in load demand is

$$\text{PAR} = \frac{L_{\text{peak}}}{L_{\text{avg}}} = \frac{H \max_{h \in \mathcal{H}} L_h}{\sum_{h \in \mathcal{H}} L_h}. \quad (4)$$

B. Energy Cost Model

We define a *cost function* $C_h(L_h)$ indicating the cost of generating or distributing electricity by the energy source at each hour $h \in \mathcal{H}$. In general, the cost of the same load can be different at different times of the day. In particular, the cost can be less at night compared to the day time. In addition, we make the following assumptions throughout this paper.

Assumption 1: The cost functions are increasing. That is, for each $h \in \mathcal{H}$, the following inequality holds:

$$C_h(\hat{L}_h) < C_h(\tilde{L}_h), \quad \forall \hat{L}_h < \tilde{L}_h. \quad (5)$$

From (5), energy cost increases if the total load increases.

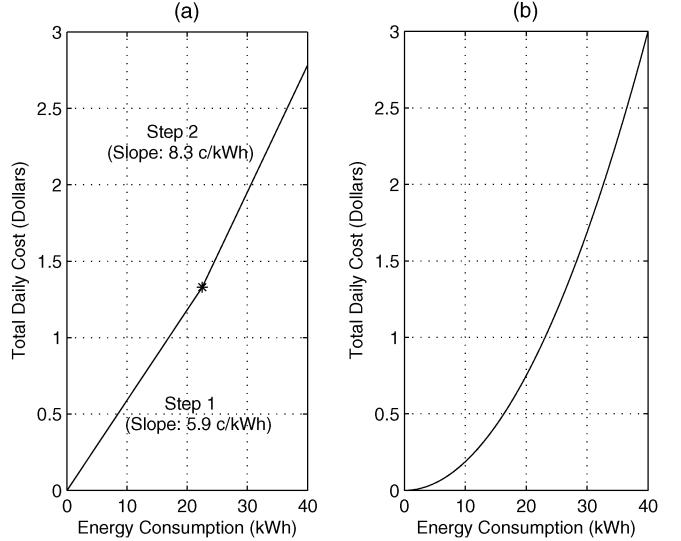


Fig. 4. Two sample convex and increasing cost functions: (a) Two-step conservation rate model used by BC Hydro [28]; (b) A quadratic cost function.

Assumption 2: The cost functions are *strictly convex*. That is, for each $h \in \mathcal{H}$, any real number $\hat{L}_h, \tilde{L}_h \geq 0$, and any real number $0 < \theta < 1$, we have [27]

$$C_h(\theta \hat{L}_h + (1 - \theta) \tilde{L}_h) < \theta C_h(\hat{L}_h) + (1 - \theta) C_h(\tilde{L}_h). \quad (6)$$

An interesting example for a class of actual energy cost functions that satisfy Assumptions 1 and 2 is the quadratic cost function for thermal generators with [19]

$$C_h(L_h) = a_h L_h^2 + b_h L_h + c_h \quad (7)$$

where $a_h > 0$ and $b_h, c_h \geq 0$ at each hour $h \in \mathcal{H}$.

Note that the cost functions that we consider in this paper can represent either the *actual* energy cost as for thermal generators or simply *artificial* cost tariffs which are used by the utility to impose a proper load control. For example, British Columbia (BC) Hydro in Canada adopts a convex price model in form of a two-step piecewise linear function to encourage energy conservation as shown in Fig. 4(a) [28]. A smoother quadratic cost function is also shown in Fig. 4(b), which is more tractable for the purpose of optimization.

C. Residential Load Control

For each user $n \in \mathcal{N}$, let \mathcal{A}_n denote the set of household appliances such as washer and dryer, refrigerator, dishwasher, air conditioner, PHEV, etc. For each appliance $a \in \mathcal{A}_n$, we define an energy consumption scheduling vector

$$\mathbf{x}_{n,a} \triangleq [x_{n,a}^1, \dots, x_{n,a}^H] \quad (8)$$

where scalar $x_{n,a}^h$ denotes the corresponding one-hour energy consumption that is scheduled for appliance a by user n at hour h . Clearly, the total load of the n th user is obtained as

$$l_n^h = \sum_{a \in \mathcal{A}_n} x_{n,a}^h, \quad h \in \mathcal{H}. \quad (9)$$

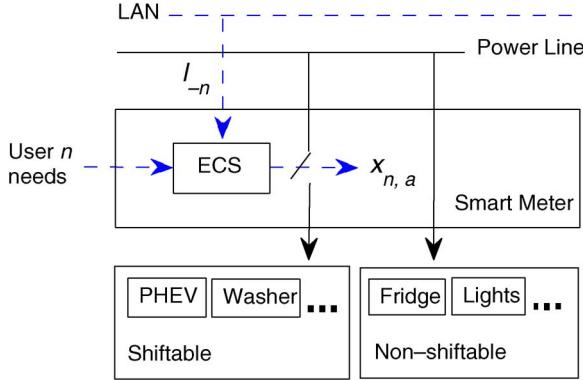


Fig. 5. The operation of the smart meter with ECS capability in our design.

In our design, as illustrated in Fig. 5, the task of the ECS function in user n 's smart meter is to determine the optimal choice of the energy consumption vector $\mathbf{x}_{n,a}$ for each appliance a . This will shape user n 's daily load profile due to (9). Next, we identify the feasible choices of the energy consumption scheduling vectors based on users' energy needs.

For each user $n \in \mathcal{N}$ and each appliance $a \in \mathcal{A}_n$, we denote the predetermined total daily energy consumption as $E_{n,a}$. For example, $E_{n,a} = 16$ kWh for a PHEV for a 40-mile daily driving range [6]. In this paper, our designed energy consumption scheduler does not aim to change the amount of energy consumption, but instead to systematically manage and shift it, e.g., in order to reduce the PAR or minimize the energy cost. In this case, the user needs to select the beginning $\alpha_{n,a} \in \mathcal{H}$ and the end $\beta_{n,a} \in \mathcal{H}$ of a time interval that appliance a can be scheduled. Clearly, $\alpha_{n,a} < \beta_{n,a}$. For example, a user may select $\alpha_{n,a} = 11$ PM and $\beta_{n,a} = 8$ AM for its PHEV to have it ready before going to work. This imposes certain constraints on vector $\mathbf{x}_{n,a}$. In fact, the time interval for which appliance a can be scheduled equals its required predetermined daily consumption, that is

$$\sum_{h=\alpha_{n,a}}^{\beta_{n,a}} x_{n,a}^h = E_{n,a} \quad (10)$$

and

$$x_{n,a}^h = 0, \quad \forall h \in \mathcal{H} \setminus \mathcal{H}_{n,a} \quad (11)$$

where $\mathcal{H}_{n,a} \triangleq \{\alpha_{n,a}, \dots, \beta_{n,a}\}$. For each appliance, the time interval provided by the user needs to be larger than or equal to the time interval needed to finish the operation. For example, for a PHEV the normal charging time is 3 h [6]; therefore, it is required that $\beta_{n,a} - \alpha_{n,a} \geq 3$. We also note that from (10) and (11), the total energy consumed by all appliances in the system over 24 h is equal to the sum of the daily energy consumption of all loads/appliances. That is, we always have the following energy balance relationship:

$$\sum_{h \in \mathcal{H}} L_h = \sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}_n} E_{n,a}. \quad (12)$$

In general, the operation of some appliances may not be time shiftable and they may have strict energy consumption sched-

uling constraints. For example, a refrigerator may have to be on all the time. In that case, $\alpha_{n,a} = 1$ and $\beta_{n,a} = 24$. As shown in Fig. 5, the ECS function in the smart meter essentially has no impact on the energy consumption scheduling for nonshiftable household appliances.

We define the minimum standby power level $\gamma_{n,a}^{\min}$ and the maximum power level $\gamma_{n,a}^{\max}$ for each appliance $a \in \mathcal{A}_n$ for each user $n \in \mathcal{N}$. Standby power refers to the electric power consumed by each appliance while it is switched off or it is in a standby mode. We assume that

$$\gamma_{n,a}^{\min} \leq x_{n,a}^h \leq \gamma_{n,a}^{\max}, \quad \forall h \in \mathcal{H}_{n,a}. \quad (13)$$

For notational simplicity, for each user n , we introduce vector \mathbf{x}_n , which is formed by stacking up energy consumption scheduling vectors $\mathbf{x}_{n,a}$ for all appliances $a \in \mathcal{A}_n$. In this regard, we can define a *feasible* energy consumption scheduling set corresponding to user n as follows:

$$\mathcal{X}_n = \left\{ \mathbf{x}_n \mid \sum_{h=\alpha_{n,a}}^{\beta_{n,a}} x_{n,a}^h = E_{n,a}, \quad x_{n,a}^h = 0, \quad \forall h \in \mathcal{H} \setminus \mathcal{H}_{n,a}, \right. \\ \left. \gamma_{n,a}^{\min} \leq x_{n,a}^h \leq \gamma_{n,a}^{\max}, \quad \forall h \in \mathcal{H}_{n,a} \right\}. \quad (14)$$

An energy consumption schedule calculated by the ECS unit in user n 's smart meter is valid only if we have $\mathbf{x}_n \in \mathcal{X}_n$. We are now ready to formulate various energy consumption scheduling optimization problems in a smart grid.

III. PROBLEM FORMULATION

In this section, we formulate two optimization problems based on two common design objectives in a power distribution system: *PAR minimization* and *energy cost minimization*. We show that these two problems can be related to each other depending on the choice of the energy cost function.

A. Peak-to-Average Ratio Minimization

By using the expressions in (1), (9), (10), and (11) in (4), we can rewrite the PAR in terms of energy consumption scheduling vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$ as

$$\frac{H \max_{h \in \mathcal{H}} \left(\sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}_n} x_{n,a}^h \right)}{\sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}_n} E_{n,a}}. \quad (15)$$

In general, a low PAR is preferred [19]. Therefore, given complete knowledge about the users' needs and the smart grid depicted in Fig. 3, an efficient energy consumption scheduling can be characterized as the solution to the following problem:

$$\underset{\mathbf{x}_n \in \mathcal{X}_n, \forall n \in \mathcal{N}}{\text{minimize}} \frac{H \max_{h \in \mathcal{H}} \left(\sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}_n} x_{n,a}^h \right)}{\sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}_n} E_{n,a}}. \quad (16)$$

Next, we note that since H and $\sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}_n} E_{n,a}$ are fixed as far as the optimization variables $\mathbf{x}_1, \dots, \mathbf{x}_N$ are concerned,

they can be removed from the objective function and we can rewrite problem (16) as the following equivalent problem:

$$\underset{\mathbf{x}_n \in \mathcal{X}_n, \forall n \in \mathcal{N}}{\text{minimize}} \max_{h \in \mathcal{H}} \left(\sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}_n} x_{n,a}^h \right). \quad (17)$$

However, problem (17) is still difficult to solve in its current form due to the max term in the objective function. This can be resolved by introducing a new auxiliary variable Γ and rewriting (17) in equivalent form as

$$\begin{aligned} & \underset{\Gamma, \mathbf{x}_n \in \mathcal{X}_n, \forall n \in \mathcal{N}}{\text{minimize}} \quad \Gamma \\ & \text{subject to} \quad \Gamma \geq \sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}_n} x_{n,a}^h, \quad \forall h \in \mathcal{H}. \end{aligned} \quad (18)$$

Problem (18) is a linear program. It can be solved in a centralized fashion by using either the *simplex method* or the *interior point method* (IPM) [27], [28]. We also note that (18) may have more than one optimal solution. That is, the same minimum PAR in the total load demand can be achieved through different energy consumption schedules.

B. Energy Cost Minimization

An efficient energy consumption scheduling can also be formulated in terms of minimizing the energy costs to all users, which can be expressed as the following optimization problem:

$$\underset{\mathbf{x}_n \in \mathcal{X}_n, \forall n \in \mathcal{N}}{\text{minimize}} \sum_{h=1}^H C_h \left(\sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}_n} x_{n,a}^h \right). \quad (19)$$

Optimization problem (19) is convex and can be solved in a centralized fashion using convex programming techniques such as IPM [27], [29]. Since the cost functions are assumed to be strictly convex, minimization problem (19) always has a unique solution, given the choices of the cost functions [27]. This is one of the differences between the energy cost minimization problem and the PAR minimization problem. Recall that the latter could have multiple optimal solutions.

IV. ENERGY CONSUMPTION GAME

Although the optimization problems defined in the previous section can be solved in a centralized fashion to obtain an optimal solution for a given configuration of the users and their energy schedules, it is more advantageous to define a solution approach that can be implemented and updated autonomously to accommodate the changes within the system. For these reasons, we are interested in solving (19) in a distributed way at the level of the smart meter using its ECS functionality, and with a minimum amount of information exchanges among the smart meters and the energy source. In particular, the goal is to let each smart meter with ECS functionality schedule the energy consumption of the household according to the individual needs of the users. It is also important to make sure that the users have an incentive to actually use the ECS features and to follow the schedules they determine. We focus on the energy cost minimization problem. Nevertheless, we will see in Section VI that by achieving the minimum cost, we also achieve a low PAR in the total load demand.

A. Pricing and Billing Tariffs

For each user $n \in \mathcal{N}$, let b_n denote the daily billing amount in dollars to be charged to user n by the utility at the end of each day. The prices should reflect the users' total daily energy consumption and relate it to the total energy cost of the system. It is reasonable to assume that

$$\sum_{n \in \mathcal{N}} b_n \geq \sum_{h=1}^H C_h \left(\sum_{n \in \mathcal{N}} l_n^h \right) \quad (20)$$

where the left-hand side in (20) denotes the total daily charge to the users and the right hand side denotes the total daily cost. For notational simplicity we define

$$\kappa \triangleq \frac{\sum_{n \in \mathcal{N}} b_n}{\sum_{h=1}^H C_h (\sum_{n \in \mathcal{N}} l_n^h)} \geq 1. \quad (21)$$

If $\kappa = 1$, then the billing system is *budget-balanced* and the utility company charges the users only with the same amount that generating/providing energy costs for the utility. On the other hand, if $\kappa > 1$, then the difference between the total charges to the users and the total energy cost would indicate the profit made by the utility company. We further assume that

$$\frac{b_n}{b_m} = \frac{\sum_{h=1}^H l_n^h}{\sum_{h=1}^H l_m^h}, \quad \forall n, m \in \mathcal{N}. \quad (22)$$

That is, users are charged proportional to their total daily energy consumption. For example, if user n consumes twice as much energy as user m , then he/she will be charged twice as much as user m . The exact amount of the charge depends on the cost of energy at each hour of the day, which itself results from a strictly convex function $C(\cdot)$ such as the one shown in Fig. 4. While the assumption in (22) helps in keeping our analysis tractable by directly relating every user's total payment to the total energy cost in the system, it is also consistent with the existing residential metering models. We are now ready to introduce an efficient energy pricing model which satisfies both assumptions (20) and (22).

After summing up the two sides of the equality in (22) across all users $m \in \mathcal{N}$, for each $n \in \mathcal{N}$, we have

$$\begin{aligned} \sum_{m \in \mathcal{N}} b_m &= \sum_{m \in \mathcal{N}} \left(b_n \frac{\sum_{h=1}^H l_m^h}{\sum_{h=1}^H l_n^h} \right) \\ &= b_n \frac{\sum_{m \in \mathcal{N}} \sum_{h=1}^H l_m^h}{\sum_{h=1}^H l_n^h}. \end{aligned} \quad (23)$$

Together from (9), (10), (21), and (23) and after reordering the terms we can show that for each user $n \in \mathcal{N}$ we have

$$\begin{aligned} b_n &= \frac{\sum_{h=1}^H l_n^h}{\sum_{m \in \mathcal{N}} \sum_{h=1}^H l_m^h} \left(\sum_{m \in \mathcal{N}} b_m \right) \\ &= \frac{\kappa \sum_{h=1}^H l_n^h}{\sum_{m \in \mathcal{N}} \sum_{h=1}^H l_m^h} \left(\sum_{h=1}^H C_h \left(\sum_{m \in \mathcal{N}} l_m^h \right) \right) \\ &= \Omega_n \sum_{h=1}^H C_h \left(\sum_{m \in \mathcal{N}} \sum_{a \in \mathcal{A}_m} x_{m,a}^h \right) \end{aligned} \quad (24)$$

where

$$\Omega_n \triangleq \frac{\kappa \sum_{a \in \mathcal{A}_n} E_{n,a}}{\sum_{m \in \mathcal{N}} \sum_{a \in \mathcal{A}_m} E_{m,a}}. \quad (25)$$

Next, we study the behavior of the users when they are charged according to (24) by using techniques from game theory.

B. Game Model

From (24), the charge on each user depends on how he and all other users schedule their consumptions. This naturally leads to the following game among users:

Energy Consumption Game Among Users:

- *Players*: Registered users in set \mathcal{N} .
- *Strategies*: Each user $n \in \mathcal{N}$ selects its energy consumption scheduling vector \mathbf{x}_n to maximize its payoff.
- *Payoffs*: $P_n(\mathbf{x}_n; \mathbf{x}_{-n})$ for each user $n \in \mathcal{N}$, where

$$\begin{aligned} P_n(\mathbf{x}_n; \mathbf{x}_{-n}) &= -b_n \\ &= -\Omega_n \times \left(\sum_{h=1}^H C_h \left(\sum_{m \in \mathcal{N}} \sum_{a \in \mathcal{A}_m} x_{m,a}^h \right) \right). \end{aligned}$$

Here, $\mathbf{x}_{-n} \triangleq [\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, \mathbf{x}_{n+1}, \dots, \mathbf{x}_N]$ denotes the vector containing the energy consumption schedules of all users *other than* end user n .

Based on the definitions of the payoffs and strategies in Game 1, the users try to select their energy consumption schedule to minimize their payments to the utility company.

Theorem 1: Suppose Assumptions 1 and 2 hold. The Nash equilibrium of Game 1 always exists and is unique.

The proof of Theorem 1 is given in Appendix A. Note that Nash equilibrium is a solution concept in game theory that characterizes how the players play a game [25]. The energy consumption scheduling variables $(\mathbf{x}_n^*, \forall n \in \mathcal{N})$ form a Nash equilibrium for Game 1 if and only if we have

$$P_n(\mathbf{x}_n^*; \mathbf{x}_{-n}^*) \geq P_n(\mathbf{x}_n; \mathbf{x}_{-n}^*), \quad \forall n \in \mathcal{N}, \quad \mathbf{x}_n \geq 0. \quad (26)$$

If the energy consumption game is at its unique Nash equilibrium, then no user would benefit by deviating from schedule $(\mathbf{x}_n^*, \forall n \in \mathcal{N})$. Next, we show the following result regarding the performance at Nash equilibrium of Game 1.

Theorem 2: The unique Nash equilibrium of Game 1 is the optimal solution of energy cost minimization problem (19).

The proof of Theorem 2 is given in Appendix B. From Theorems 1 and 2, as long as the cost functions $C_h(\cdot)$ are increasing and strictly convex for each $h \in \mathcal{H}$ and also the price model satisfies the requirements (20) and (22), the users have an incentive to cooperate with each other to reduce their own payments by solving problem (19).

V. DISTRIBUTED ALGORITHM

From the results in Section IV, the users would be willing to cooperate and allow their ECS units to schedule their household energy consumption to pay less. In particular, we showed that the unique Nash equilibrium of the energy consumption game among the users is the same as the global optimal solution of the energy consumption scheduling problem (19). In this section, we provide an algorithm to be implemented in each ECS

unit to reach the Nash equilibrium of Game 1 and achieve the optimal system performance. We prove the convergence and optimality properties of the proposed algorithm. We also show that it is strategy-proof and users will not benefit from misleading each other by providing inaccurate information about their usage during their interactions.

A. Principle of the Algorithm

Consider any user $n \in \mathcal{N}$. Given \mathbf{x}_{-n} and assuming that all other users fix their energy consumption schedule according to \mathbf{x}_{-n} , user n 's *best response* can be determined by solving the following local optimization problem:

$$\underset{\mathbf{x}_n \in \mathcal{X}_n}{\text{maximize}} \quad P_n(\mathbf{x}_n; \mathbf{x}_{-n}). \quad (27)$$

Notice that we refer to optimization problem (27) as a local problem for user n because the only optimization variable is user n 's energy consumption scheduling vector \mathbf{x}_n . Since Ω_n is fixed and does not depend on the choice of \mathbf{x}_n , the maximization in (27) can be replaced by

$$\underset{\mathbf{x}_n \in \mathcal{X}_n}{\text{minimize}} \quad \sum_{h=1}^H C_h \left(\sum_{m \in \mathcal{N}} \sum_{a \in \mathcal{A}_m} x_{m,a}^h \right). \quad (28)$$

After reordering the terms, we can also rewrite (28) as

$$\underset{\mathbf{x}_n \in \mathcal{X}_n}{\text{minimize}} \quad \sum_{h=1}^H C_h \left(\sum_{a \in \mathcal{A}_n} x_{n,a}^h + \sum_{m \in \mathcal{N} \setminus \{n\}} l_m^h \right). \quad (29)$$

We notice that problems (29) and (19) have the same objective functions. However, problem (29) has only local variables for user n . Moreover, problem (29) is convex and can be solved by IPM [27]. User n can solve problem (29) as long as it knows the cost functions C_h for all $h \in \mathcal{H}$ as well as $\mathbf{l}_{-n} \triangleq [\mathbf{l}_1, \dots, \mathbf{l}_{n-1}, \mathbf{l}_{n+1}, \dots, \mathbf{l}_N]$, i.e., the vector containing the scheduled daily energy consumption for all other users. These observations motivate us to propose Algorithm 1 to solve problem (19) in a distributed fashion.

Algorithm 1: Executed by each user $n \in \mathcal{N}$.

- 1: Randomly initialize \mathbf{l}_n and \mathbf{l}_{-n} .
 - 2: **Repeat**
 - 3: **At** random time instances **Do**
 - 4: Solve local problem (29) using IPM [27].
 - 5: **If** \mathbf{x}_n changes compared to current schedule **Then**
 - 6: Update \mathbf{x}_n according to the new solution.
 - 7: Broadcast a control message to announce \mathbf{l}_n to the other ECS units across the system.
 - 8: **End**
 - 9: **End**
 - 10: **If** a control message is received **Then**
 - 11: Update \mathbf{l}_{-n} accordingly.
 - 12: **End**
 - 13: **Until** no ECS unit announces any new schedule.
-

Next, we explain how the proposed algorithm works. In Line 1, each user $n \in \mathcal{N}$ starts with some random initial conditions. That is, each user n assumes a random vector \mathbf{l}_m for any $m \in \mathcal{N} \setminus \{n\}$. This assumption is required since at the beginning, user n has no prior information about other users. Then, the loop in Lines 2 to 13 is executed until the algorithm converges.¹ Within this loop, each ECS individually solves its own version of local optimization problem (29) using IPM in Line 4. That is, each user simply plays its best response as discussed in Section V-A. The new schedule is announced to the other users through broadcasting a control message. In fact, the message exchange between the users in the general framework in Fig. 2 are implemented in Algorithm 1 in form of each user n broadcasting \mathbf{l}_n over the LAN. Note that users do *not* reveal the details about the energy consumption of their own appliances due to privacy concerns. They only announce their total hourly usage, which is collected at the energy source for billing purposes anyways. In this setting, each user also updates its local memory whenever it receives a control message from other users in Line 11.

The proposed DSM strategy in this section has key differences with the existing DSM programs in the literature. First, unlike DLC, here each user has control over the operation of its own appliances. Therefore, user privacy is not a concern. Moreover, the users simply follow what is best for them in order to decide on their own consumption schedules while they select the best-response in the energy consumption game. Different from DLC and RTP which are based on the framework in Fig. 1, our design is based on the new setting in Fig. 2 and incorporates the interactions among users.

B. Convergence and Optimality

In this section, we prove the convergence and optimality properties of the proposed distributed algorithm. The cornerstone of our assessment is the following theorem.

Theorem 3: If the updates of the individual energy consumption scheduling vectors are asynchronous among the users, i.e., no two users $n, m \in \mathcal{N}$ update their energy consumption scheduling vectors \mathbf{x}_n and \mathbf{x}_m at the same time, then starting from any randomly selected initial conditions, Algorithm 1 converges to its fixed point, i.e., to the Nash equilibrium of the energy consumption game.

The proof of Theorem 3 is given in Appendix C. Theorem 3 provides a sufficient condition to guarantee convergence. This condition only requires the users to update their energy consumption scheduling vectors sequentially. For example, this can be achieved if the energy source can determine the timing when each user should update its energy consumption. In that case, instead of Line 3 in Algorithm 1, user n would execute Lines 4 to 7 only if it receives a command from the energy source telling him/her that it is user n 's turn to update its energy consumption scheduling vectors. Other turn-taking scenarios can also be used to coordinate the energy consumption scheduling updates among the users.

Together, from Theorems 2 and 3, starting from any initial point, Algorithm 1 automatically converges to the global optimal solution of energy cost minimization problem (19). We

notice that if all users' energy consumption needs remain unchanged within the next 24 h, then Algorithm 1 becomes a day-ahead energy consumption scheduling design. However, if the energy consumption needs for the users change frequently, then Algorithm 1 will converge to the new optimal energy consumption schedules in a more real-time fashion (cf. [18]).

C. Strategy-Proof Property

In this section, we would like to answer this question: *Is it beneficial for a user or a group of users to cheat and announce an incorrect energy consumption schedule to the other users?* That is, does it help user n , in terms of increasing its daily payoff at a fixed point of Algorithm 1, to be *untruthful* and set $\mathbf{l}_n \neq (\sum_{a \in \mathcal{A}_n} x_{n,a}^h)$? Of course, it is possible to have the energy source supervise and monitor all message exchanges among users such that the users' truthfulness can be enforced. But it is still interesting to see if Algorithm 1 itself enforces truthfulness with no extra supervisory effort from the utility.

Theorem 4: When the users are running Algorithm 1, no user or group of users would benefit from being untruthful. That is, each user $n \in \mathcal{N}$ will end up having a higher electricity payment on its daily bill if he announces its daily energy consumption schedule \mathbf{l}_n incorrectly.

The proof of Theorem 4 is given in Appendix D. From Theorem 4 we can assure that all users release their daily energy consumption schedule accurately. This includes the case when a user believes that all other users in the system are being truthful as well. This implies that an energy consumption game is *not* tempting for cheating and is fundamentally different from some well-known games such as the *prisoner's dilemma* game [25, p. 110] where the users *do* cheat if they think that other players are truthful. It is worth clarifying that the underlying cause for Algorithm 1 to be automatically strategy-proof is that in our billing model, we have directly related every user's payoff to the total energy cost in the system. In fact, the global and individual cost minimizations are closely related in our model. Therefore, any behavior by a user or a group of users which results in deviating from the optimal system performance will also harm the cheating user or the group of cheating users in terms of individual payments, deterring users from any malicious behavior.

VI. SIMULATION RESULTS

In this section, we present simulation results and assess the performance of our proposed algorithm. In our considered benchmark smart grid system there are $N = 10$ customer/users that subscribe to the ECS services. For the purpose of study, each user is selected to have between 10 to 20 appliances with *nonshiftable* operation, i.e., with strict energy consumption scheduling constraints. Such appliances may include refrigerator-freezer (daily usage: 1.32 kWh), electric stove (daily usage: 1.89 kWh for self-cleaning and 2.01 kWh for regular), lighting (daily usage for 10 standard bulbs: 1.00 kWh), heating (daily usage: 7.1 kWh) [30]. Moreover, each user is selected to also have between 10 to 20 appliances with *shiftable* operation, i.e., with soft energy consumption scheduling constraints. Recall that the smart meter with ECS capability may schedule only the appliances with soft energy consumption scheduling constraints. Such appliances may include dishwasher (daily

¹We will discuss the convergence property of Algorithm 1 in Theorem 3.

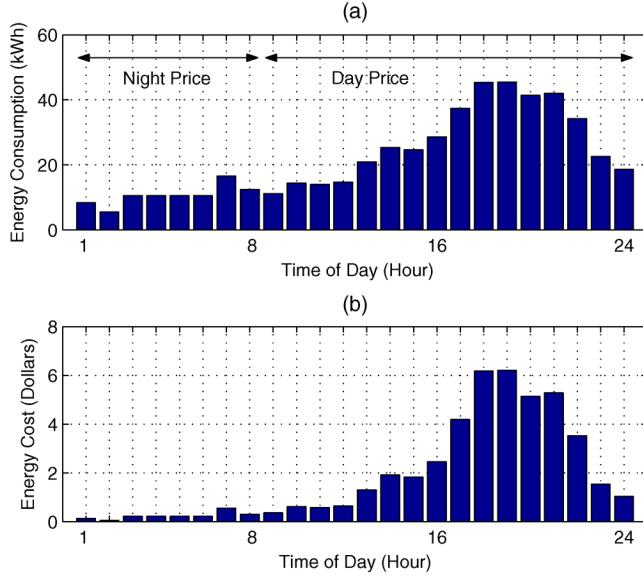


Fig. 6. Scheduled energy consumption and corresponding cost when ECS units are not used. In this case, PAR is 2.1 and the total daily cost is \$44.77.

usage: 1.44 kWh), washing machine (daily usage: 1.49 kWh for energy-star, 1.94 kWh for regular), clothes dryer (daily usage: 2.50 kWh), and PHEV (daily usage: 9.9 kWh) [6], [30]. In our simulation model, we assume that each user has a randomly selected combination of the considered shiftable and nonshiftable loads to be used at different times of the day by taking into account that the load demand is higher in the evening and lower during the night. For example, we assumed that when PHEVs become popular as widely predicted, it is reasonable to assume that most users (four out of five users in our setting) have electric cars to be charged some time between the afternoon hours on each day and the early morning hours on the next day. The energy cost function is assumed to be quadratic as in (7). For simplicity we assume that $b_h = c_h = 0$ for all $h \in \mathcal{H}$. We also have $a_h = 0.3$ cents at daytime hours, i.e., from 8:00 in the morning to 12:00 at night and $a_h = 0.2$ cents during the night, i.e., from 12:00 at night to 8:00 AM the day after. The power system is assumed budget-balanced, i.e., $\kappa = 1$ (see (21)). The timing Algorithm 1 works based on a round-robin scenario which is coordinated by the energy source. In this scenario, at each user's turn it will start its local computation to update its own energy consumption schedule according to Line 4 in Algorithm 1. Then it will inform the energy source who will allocate turn to another randomly selected user and this procedure continues until the algorithm converges. In this setting, the energy source makes sure that every user takes a turn once in a while.

A. Performance Comparison

The simulation results on total scheduled energy consumptions and the energy cost for a single scenario are shown in Figs. 6 and 7, *without* and *with* the deployment of the ECS function in the smart meters, respectively. For the case without ECS deployment, each appliance $a \in \mathcal{A}_n$ for each user $n \in \mathcal{N}$ is assumed to start operation right at the beginning of the time interval $[\alpha_{n,a}, \beta_{n,a}]$ and at its typical power level. For the case with ECS deployment, the timing and the power level for the

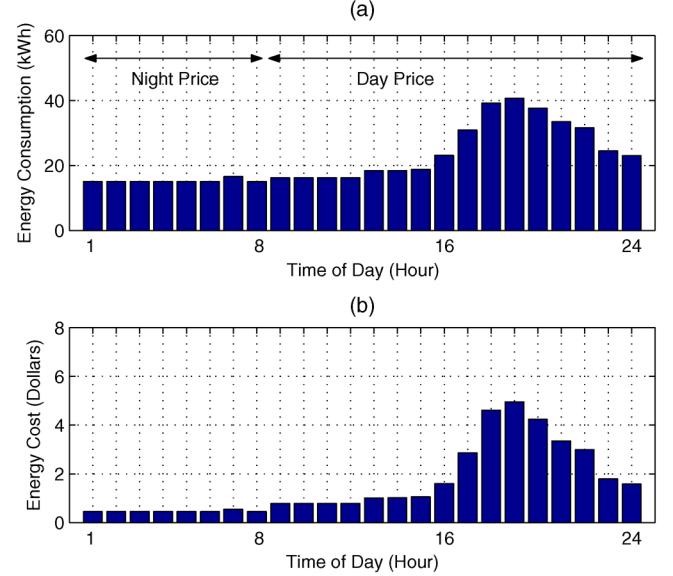


Fig. 7. Scheduled energy consumption and corresponding cost when ECS units are deployed. In this case, PAR is 1.8 and the total daily cost is \$37.90.

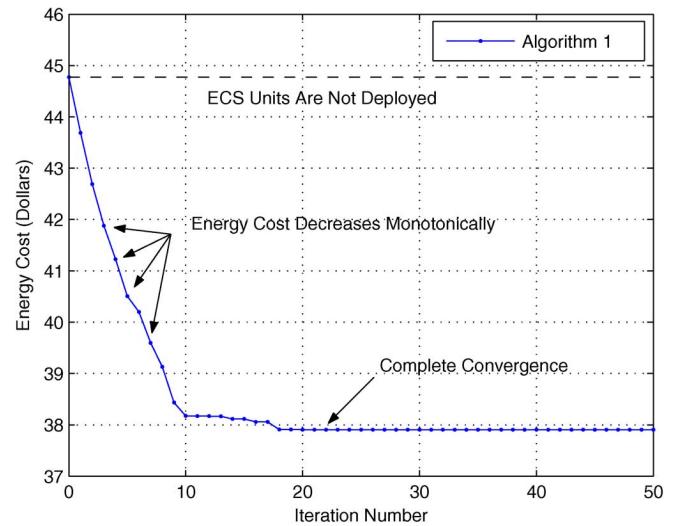


Fig. 8. Trend of resulting energy cost along the iterations of Algorithm 1. We can see that the proposed distributed algorithm converges quickly. Steady state is reached after only 22 iterations when the energy cost is minimized.

operation of each household appliance is determined by Algorithm 1. By comparing the results in Figs. 6 and 7, we can see that when the ECS functions are *not* used/implemented, the PAR is 2.1 and the energy cost is \$44.77. At the same time, when the ECS feature is enabled, the PAR reduces to 1.8 (i.e., 17% less) and the energy cost reduces to \$37.90 (i.e., 18% less). In fact, in the latter case, there is a more evenly distributed load across different hours of the day. Note that each user consumes the same amount of energy in the two cases, but it simply schedules its consumption more efficiently in the case that the ECS units are used. On the other hand, the trends of the resulting total energy cost while Algorithm 1 proceeds along its distributed iterations are shown in Fig. 8. We can see that as the users run Algorithm 1, the energy cost monotonically decreases until the algorithm converges after 22 iterations only, i.e., around 2 iterations per user on average.

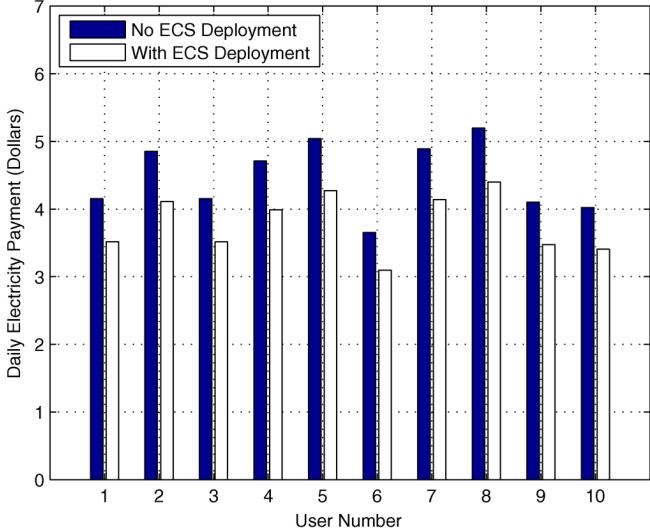


Fig. 9. Daily charges for each subscriber without and with ECS deployment.

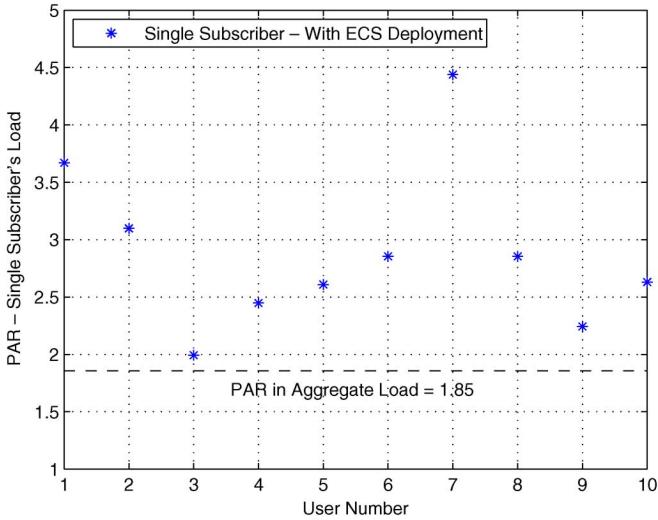


Fig. 10. PAR in each end user's individual daily load and comparison with the PAR in the aggregate load across all end users.

B. User Payment

While the proposed distributed DSM strategy leads to less total energy cost and lower PAR in the aggregate load demand, it is also beneficial for each individual end user. To see this, the daily payments for all users are shown in Fig. 9. Here, the simulation setting is the same as the one in Section VI-A. We can see that all users would pay significantly less to the utility company when the ECS is enabled in the smart meter. Therefore, the users would be willing to participate in the proposed automatic demand-side management system.

Another interesting aspect is shown in Fig. 10. In this figure, we have plotted the PAR in each user's load and compared it with the PAR in the aggregate load across all users. Here, for each user $n \in \mathcal{N}$, the individual PAR is calculated as

$$\text{PAR}_n \triangleq \frac{H \max_h l_n^h}{\sum_{h=1}^H l_n^h}. \quad (30)$$

We can see in Fig. 10 that the PAR in the aggregate load is significantly less than the PAR in each user's individual load. In fact, for some users, such as user 7 in our example scenario, the load is quite unbalanced and the PAR is around 4.5. This confirms our discussions in Section I that the utility company may not necessarily need all users to individually balance their load, as opposed to the design objective in real-time pricing tariffs which expect each individual end user to shift its consumption from peak hours to off-peak hours.

C. Optimal PAR Reduction

Recall from Section III that in order to achieve the minimum PAR in the total load demand, we can schedule energy consumption according to the optimal solution of problem (18). However, our proposed distributed algorithm in this paper only aims at minimizing the energy cost, i.e., solving problem (19). In this section, we show that by solving problem (19), Algorithm 1 results in PAR values which are very close to the PAR values obtained by solving problem (18). In addition, Algorithm 1 significantly reduces the energy cost. Corresponding simulation results are shown in Fig. 11. Here, we simulated 50 different scenarios to have a more accurate comparison. From the results in Fig. 11(a), we can see that the PARs achieved by solving the PAR minimization problem (18) and the energy cost minimization problem (19) are almost the same. In fact, in 32 out of 50 scenarios, the two PAR values are identical. In the remaining 18 scenarios, the solution of the PAR minimization problem results in strictly lower PAR, e.g., as in scenario number 17. However, even in these cases, the improvement is minor. On average, the PAR reduces from 1.8325 to 1.8315 (i.e., only 0.05% improvement). On the other hand, from Fig. 11(b), we can see that Algorithm 1 significantly outperforms the optimal solution of the PAR minimization problem in terms of reducing the energy cost. We notice that while the average energy cost when no ECS unit is deployed is as high as \$51.83, the average energy cost at the optimal solution of the PAR minimization problem reduces to \$48.04. However, the average energy cost at the optimal solution of the energy cost minimization problem obtained by running Algorithm 1 is only \$41.65. Therefore, we can conclude that Algorithm 1, which is designed for energy cost minimization, also efficiently reduces the PAR in the aggregate load. Moreover, it can significantly reduce the energy cost.

D. Impact of Amount of Shiftable Load

For the simulation scenarios so far, we have assumed that around half of the residential load is shiftable while the other half is not shiftable. Clearly, the ECS units are expected to have a more significant impact if more appliances have shiftable operation. To better see this, we have plotted the PAR in the aggregate load demand when the percentage of shiftable load varies from 10% to 90% in Fig. 12. In this regard, we can see that if 9 out of 10 appliances have shiftable operation, then on average, the PAR in the aggregate load can be reduced down to only 1.35, indicating almost a flat load.

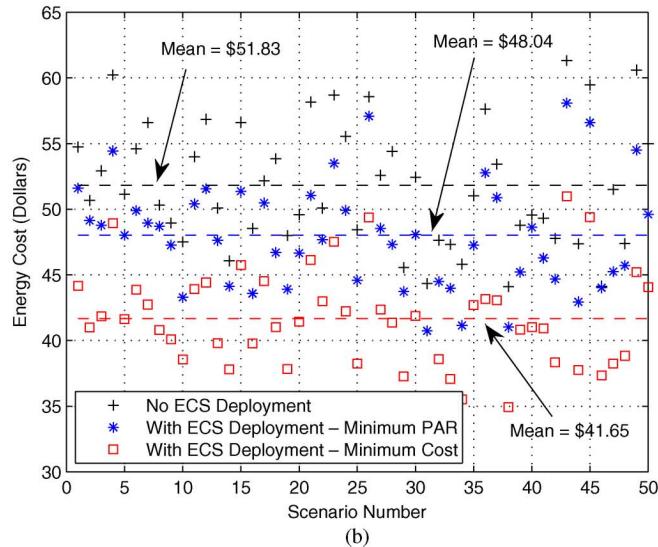
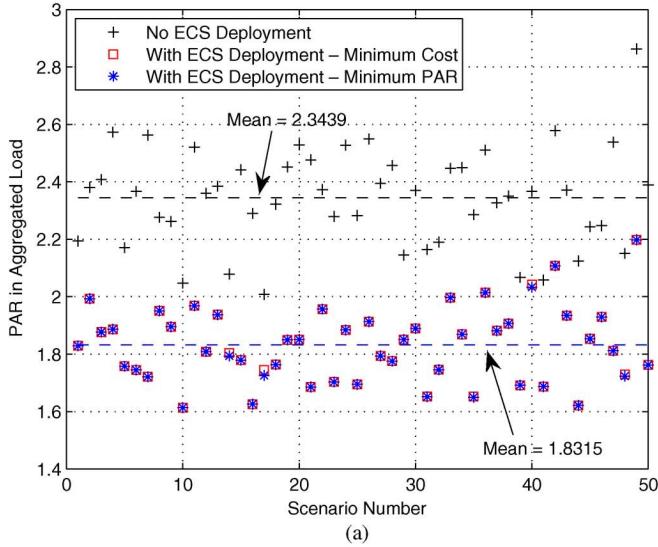


Fig. 11. Comparison between the optimal solution of the PAR minimization problem (18), the optimal solution of the energy cost minimization problem (19), and the case with no ECS function deployment in smart meters. (a) Minimizing the PAR in the aggregate load. (b) Minimizing the energy cost.

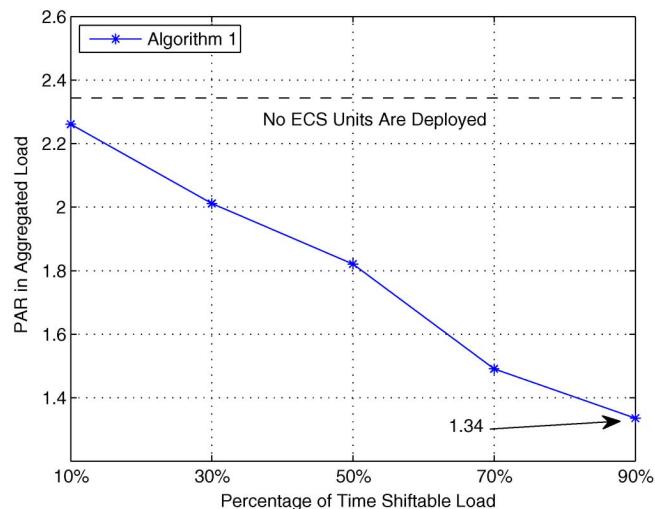


Fig. 12. The PAR in the aggregate load when the percentage of shiftable load varies from 10% to 90%. In a scenario when most of the residential load is shiftable, the aggregate load can be flat, resulting in a very low PAR.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an optimal, autonomous, and distributed incentive-based energy consumption scheduling algorithm in order to minimize the cost of energy and also to balance the total residential load when multiple users share a common energy source. Unlike most of the previous DSM strategies that focus solely on the interactions between the utility company and each user, the basis of our design are the interactions among the users. Our proposed distributed algorithm requires only some limited message exchanges between users when each of them tries to maximize its own benefits in a game-theoretic setting. In order to encourage users to behave in a desired way (i.e., to minimize the energy cost) we proposed a smart pricing tariff such that the interactions among the users automatically lead to an optimal aggregate load profile at the equilibrium of an energy consumption scheduling game. Simulation results confirm that the proposed distributed demand-side management strategy can reduce the PAR, the energy cost, and each user's daily electricity charges.

The results in this paper can be extended in several directions. First, the proposed distributed DSM strategy can be modified to address the case when there are multiple energy sources in the system. In that case, the users need to determine not only the total amount of their energy consumption at each hour of the day, but also the portion of the total energy that they need to obtain from each available energy source. Second, it is interesting to extend our design to address both shifting and reducing energy consumption. This can be done by introducing new energy cost functions which depend on not only the energy consumption at each hour, but also the total daily energy consumption. In this regard, the linear billing model in (22) in Section IV-A can be extended to more general nonlinear models. Third, one may relax the convexity assumption on the choices of the energy cost functions to cover a wider range of energy cost models. Of course, this will introduce optimization problems which are more difficult to solve. Fourth, our system model can be extended to a scenario where users can store energy at certain hours, e.g., in their PHEV batteries during the night. They can then sell the energy back to the grid at peak hours. This can be done by allowing our energy consumption scheduling variables to take negative values for appliances that have energy storage capability, where a negative value for these variables indicates providing rather than consuming energy. Finally, while our analysis focused only on residential load control, similar techniques can be used to better shape the aggregate profile of commercial load in an industrial region.

APPENDIX

A. Proof of Theorem 1

We first notice that since $C_h(\cdot)$ is strictly convex for each $h \in \mathcal{H}$, the payoff function $P_n(\mathbf{x}_n; \mathbf{x}_{-n})$ is strictly concave with respect to \mathbf{x}_n . Therefore, Game 1 is a strictly concave N -person game. In this case, the existence of a Nash equilibrium directly results from [31, Th. 1]. Moreover, the Nash equilibrium is unique due to [31, Th. 3]. ■

B. Proof of Theorem 2

We first show that the global optimal solution of problem (19) forms a Nash equilibrium for Game 1. For notational simplicity, let $\mathbf{x}_1^*, \dots, \mathbf{x}_N^*$ denote the optimal solution of problem (19). We also define

$$C^* \triangleq \sum_{h=1}^H C_h \left(\sum_{m \in \mathcal{N}} \sum_{a \in \mathcal{A}_m} x_{m,a}^{h*} \right). \quad (31)$$

By definition of optimality, for each subscriber $n \in \mathcal{N}$ and for any arbitrary $\mathbf{x}_n \geq 0$, we have

$$C^* \leq \sum_{h=1}^H C_h \left(\sum_{m \in \mathcal{N} \setminus \{n\}} \sum_{a \in \mathcal{A}_m} x_{m,a}^{h*} + \sum_{a \in \mathcal{A}_n} x_{n,a}^h \right). \quad (32)$$

After multiplying both sides in (32) by $-\Omega_n$ it becomes

$$P_n(\mathbf{x}_n^*; \mathbf{x}_{-n}^*) \geq P_n(\mathbf{x}_n; \mathbf{x}_{-n}^*), \quad \forall \mathbf{x}_n \geq 0. \quad (33)$$

Comparing (33) and (26), we can conclude that the optimal solution $\mathbf{x}_1^*, \dots, \mathbf{x}_N^*$ forms a Nash equilibrium for Game 1. However, from Theorem 1, Game 1 has a unique Nash equilibrium. Thus, the optimal solution of problem (19) is equivalent to the Nash equilibrium of Game 1. ■

C. Proof of Theorem 3

Recall that playing the best response for each user $n \in \mathcal{N}$ would be equivalent to solving optimization problem (29). Therefore, if users play the best responses sequentially through running Algorithm 1 in an asynchronous fashion, the energy cost in the system either decreases or remains unchanged every time a user updates its energy consumption schedule. Since the energy cost is bounded below (e.g., the energy cost is always nonnegative), the convergence to some fixed point is evident. On the other hand, at the fixed point of Algorithm 1, no user can improve its payoff by deviating from the fixed point when playing its best response. This directly indicates that the fixed point is the Nash equilibrium of Game 1 among the users. ■

D. Proof of Theorem 4

Let $\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_N$ denote the Nash equilibrium of Game 1 when a nonempty set of users $\mathcal{M} \subseteq \mathcal{N}$ are untruthful, while all other users $\mathcal{N} \setminus \mathcal{M}$ are truthful. Also let $\mathbf{x}_1^*, \dots, \mathbf{x}_N^*$ denote the optimal solution of problem (19). Recall from Theorem 2 that $\mathbf{x}_1^*, \dots, \mathbf{x}_N^*$ is also the Nash equilibrium of Game 1 when all users are truthful. For each user $n \in \mathcal{M}$ to benefit from being untruthful, it is required that we have

$$P_n(\bar{\mathbf{x}}_n; \bar{\mathbf{x}}_{-n}) \geq P_n(\mathbf{x}_n^*; \mathbf{x}_{-n}^*). \quad (34)$$

By dividing both sides in (34) by $-\Omega_n$, we should have

$$\sum_{h=1}^H C_h \left(\sum_{m \in \mathcal{N}} \sum_{a \in \mathcal{A}_m} \bar{x}_{m,a}^h \right) \leq \sum_{h=1}^H C_h \left(\sum_{m \in \mathcal{N}} \sum_{a \in \mathcal{A}_m} x_{m,a}^{h*} \right). \quad (35)$$

However, this contradicts the fact that $\mathbf{x}_1^*, \dots, \mathbf{x}_N^*$ is the optimal solution of problem (19). Therefore, user n does not benefit from announcing inaccurate information with respect to its daily energy consumption schedule in Line 7 of Algorithm 1. In fact, since every user's individual payoff is nothing but the total energy cost times a negative constant $-\Omega_n$, for each user or a group of users, the only way to increase the payoff at Nash equilibrium is to reduce the total energy cost. Therefore, any behavior, such as being untruthful, which leads to increasing the energy cost from its optimal/minimum value would harm the cheating users or the group of cheating users as well as every other user in the system. ■

ACKNOWLEDGMENT

The authors would like to thank the support from the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Canada Research Chair program.

REFERENCES

- [1] G. M. Masters, *Renewable and Efficient Electric Power Systems*. Hoboken, NJ: Wiley, 2004.
- [2] B. Ramanathan and V. Vital, "A framework for evaluation of advanced direct load control with minimum disruption," *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1681–1688, Nov. 2008.
- [3] C. W. Gellings and J. H. Chamberlin, *Demand Side Management: Concepts and Methods*, 2nd ed. Tulsa, OK: PennWell Books, 1993.
- [4] M. A. A. Pedrasa, T. D. Spooner, and I. F. MacGill, "Scheduling of demand side resources using binary particle swarm optimization," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1173–1181, Aug. 2009.
- [5] Reducing electricity consumption in houses, Ontario Home Builders' Assoc., May 2006, Energy Conservation Committee Report and Recommendations.
- [6] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power Energy Mag.*, vol. 7, no. 2, pp. 52–62, Mar.–Apr. 2009.
- [7] N. Ruiz, I. Cobelo, and J. Oyarzabal, "A direct load control model for virtual power plant management," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 959–966, May 2009.
- [8] A. Gomes, C. H. Antunes, and A. G. Martins, "A multiple objective approach to direct load control using an interactive evolutionary algorithm," *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1004–1011, Aug. 2007.
- [9] D. D. Weers and M. A. Shamsedin, "Testing a new direct load control power line communication system," *IEEE Trans. Power Del.*, vol. 2, no. 3, pp. 657–660, Jul. 1987.
- [10] C. M. Chu, T. L. Jong, and Y. W. Huang, "A direct load control of air-conditioning loads with thermal comfort control," in *Proc. IEEE PES Gen. Meet.*, San Francisco, CA, Jun. 2005.
- [11] OpenHAN Task Force of the UtilityAMI Working Group, Home Area Network System Requirements Specification Aug. 2008.
- [12] K. Herter, "Residential implementation of critical-peak pricing of electricity," *Energy Policy*, vol. 35, pp. 2121–2130, Apr. 2007.
- [13] C. Triki and A. Violi, "Dynamic pricing of electricity in retail markets," *Q. J. Oper. Res.*, vol. 7, no. 1, pp. 21–36, Mar. 2009.
- [14] P. Centolella, "The integration of price responsive demand into regional transmission organization (RTO) wholesale power markets and system operations (article in press)," *Energy*, vol. 35, no. 4, pp. 1568–1574, Apr. 2010.
- [15] H. Allcott, Real time pricing and electricity markets, Working Paper, Harvard Univ.. Cambridge, MA, Feb. 2009.
- [16] Quantum Consulting Inc. and Summit Blue Consulting, LLC Working Group 2 Measurement and Evaluation Committee and Southern California Edison Company, Demand response program evaluation—Final report Apr. 2005.
- [17] M. Ann-Piette, G. Ghatikar, S. Kilicotte, D. Watson, E. Koch, and D. Hennage, "Design and operation of an open, interoperable automated demand response infrastructure for commercial buildings," *J. Comput. Inf. Sci. Eng.*, vol. 9, pp. 1–9, Jun. 2009.

- [18] A. H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 120–133, Sep. 2010.
- [19] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. New York: Wiley-Interscience, 1996.
- [20] Converge Demand Response Aggregatos [Online]. Available: <http://www.converge.com/demandmanager/> 2010
- [21] U.S. Dept. Energy, *The Smart Grid: An Introduction*, 2009.
- [22] A. Vojdani, "Smart integration," *IEEE Power Energy Mag.*, vol. 6, no. 6, pp. 72–79, Nov. 2008.
- [23] L. H. Tsoukalas and R. Gao, "From smart grids to an energy internet: Assumptions, architectures, and requirements," presented at the 3rd Int. Conf. Elect. Utility Deregulation Restructuring Power Technol., Nanjing, China, Apr. 2008.
- [24] M. Amin and B. F. Wollenberg, "Toward a smart grid: Power delivery for the 21 st century," *IEEE Power Energy Mag.*, vol. 4, no. 6, pp. 34–41, Nov. 2006.
- [25] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA: MIT Press, 1991.
- [26] A. H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, and R. Schober, "Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid," presented at the IEEE PES Conf. Innovative Smart Grid Technol., Gaithersburg, MD, Jan. 2010.
- [27] S. Boyd and L. Vandenberghe, *Convex Optimization*. : Cambridge University Press, 2004.
- [28] British Columbia Hydro Conservation Electricity Rates [Online]. Available: https://www.bchydro.com/youraccount/content/residential_inclining_block.jsp 2009
- [29] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. : Athena Scientific, 1997.
- [30] Office of Energy Efficiency, Natural Resources Canada, Energy Consumption of Household Appliances Shipped in Canada Dec. 2005.
- [31] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, pp. 347–351, 1965.



Amir-Hamed Mohsenian-Rad (S'04–M'09) received the M.S. degree in electrical engineering from Sharif University of Technology in 2004 and the Ph.D. degree in electrical and computer engineering from the University of British Columbia (UBC), Canada, in 2008.

Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering, Texas Tech University, Lubbock. His research interests include design, optimization, and game-theoretic analysis of communication networks and smart power systems.

Dr. Mohsenian-Rad is the recipient of the UBC Graduate Fellowship, Pacific Century Graduate Scholarship, and the Natural Sciences and Engineering Research Council of Canada (NSERC) postdoctoral fellowship. He is an Associate Editor of the *International Journal of Electronics and Communication* and serves as a Technical Program Committee member in various conferences, including IEEE Globecom, ICC, and CCNC.



Vincent W. S. Wong (SM'07) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000.

From 2000 to 2001, he was a Systems Engineer at PMC-Sierra Inc. He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently an Associate Professor. His research interests are in resource and mobility management for wireless mesh networks, wireless sensor networks, and heterogeneous wireless networks.

Dr. Wong is a Member of the ACM. He is an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and an Editor of KICS/IEEE Journal of Communications and Networks. He serves as TPC member in various conferences, including IEEE Infocom, ICC, and Globecom.



Juri Jatskevich (M'99–SM'07) received the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette IN, in 1997 and 1999, respectively.

Since 2002, he has been a faculty member at the University of British Columbia, Vancouver, Canada, where he is now an Associate Professor of Electrical and Computer Engineering. His research interests include smart energy grids, power electronic systems, electrical machines and drives, and modeling and simulation of electromagnetic transients.

Dr. Jatskevich is currently a Chair of the IEEE CAS Power Systems and Power Electronic Circuits Technical Committee, Editor of IEEE TRANSACTIONS ON ENERGY CONVERSION, Editor of IEEE POWER ELECTRONICS LETTERS, and Associate Editor of IEEE TRANSACTIONS ON POWER ELECTRONICS. He is also chairing the IEEE Task Force on Dynamic Average Modeling, under the Working Group on Modeling and Analysis of System Transients Using Digital Programs.



Robert Schober (M'01–SM'08–F'10) was born in Neuendettelsau, Germany, in 1971. He received the Diplom (Univ.) and Ph.D. degrees in electrical engineering from the University of Erlangen-Nuernberg in 1997 and 2000, respectively.

From May 2001 to April 2002 he was a Postdoctoral Fellow at the University of Toronto, Canada, sponsored by the German Academic Exchange Service (DAAD). Since May 2002 he has been with the University of British Columbia (UBC), Vancouver, Canada, where he is now a Full Professor and Canada

Research Chair (Tier II) in Wireless Communications. His research interests fall into the broad areas of communication theory, wireless communications, and statistical signal processing.

Dr. Schober received the 2002 Heinz MaierLeibnitz Award of the German Science Foundation (DFG), the 2004 Innovations Award of the Vodafone Foundation for Research in Mobile Communications, the 2006 UBC Killam Research Prize, the 2007 Wilhelm Friedrich Bessel Research Award of the Alexander von Humboldt Foundation, and the 2008 Charles McDowell Award for Excellence in Research from UBC. In addition, he received best paper awards from the German Information Technology Society (ITG), the European Association for Signal, Speech and Image Processing (EURASIP), IEEE ICUWB 2006, the International Zurich Seminar on Broadband Communications, and European Wireless 2000. He is also the Area Editor for Modulation and Signal Design for the IEEE TRANSACTIONS ON COMMUNICATIONS.



Alberto Leon-Garcia (S'74–M'77–SM'97–F'99) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1973, 1974, and 1976, respectively.

He was founder and CTO of AcceLight Networks in Ottawa, from 1999 to 2002, which developed an all-optical fabric multiterabit, multiservice core switch. He is currently a Professor in Electrical and Computer Engineering at the University of Toronto, ON, Canada. He holds a Canada Research Chair in Autonomic Service Architecture. He holds several patents and has published extensively in the areas of switch architecture and traffic management. His research team is currently developing a network testbed that will enable at-scale experimentation in new network protocols and distributed applications. He is recognized as an innovator in networking education. In 1986, he led the development of the University of Toronto—Northern Telecom Network Engineering Program. He has also led in 1997 the development of the Master of Engineering in Telecommunications program, and the communications and networking options in the undergraduate computer engineering program. He is the author of the leading textbooks *Probability and Random Processes for Electrical Engineering and Communication Networks: Fundamental Concepts and Key Architecture*. His current research interests include application-oriented networking and autonomic resources management with a focus on enabling pervasive smart infrastructure.

Prof. Leon-Garcia is a Fellow of the Engineering Institute of Canada. He received the 2006 Thomas Eadie Medal from the Royal Society of Canada and the 2010 IEEE Canada A. G. L. McNaughton Gold Medal for his contributions to the area of communications.

Optimal Real-time Pricing Algorithm Based on Utility Maximization for Smart Grid

Pedram Samadi, Amir-Hamed Mohsenian-Rad, Robert Schober, Vincent W.S. Wong, and Juri Jatskevich

Department of Electrical and Computer Engineering

The University of British Columbia, Vancouver, Canada

E-mail:{psamadi, hamed, rschober, vincentw, jurij}@ece.ubc.ca

Abstract—In this paper, we consider a smart power infrastructure, where several subscribers share a common energy source. Each subscriber is equipped with an *energy consumption controller* (ECC) unit as part of its smart meter. Each smart meter is connected to not only the power grid but also a communication infrastructure such as a local area network. This allows two-way communication among smart meters. Considering the importance of energy pricing as an essential tool to develop efficient demand side management strategies, we propose a novel *real-time pricing* algorithm for the future smart grid. We focus on the interactions between the smart meters and the energy provider through the exchange of control messages which contain subscribers' energy consumption and the real-time price information. First, we analytically model the subscribers' preferences and their energy consumption patterns in form of carefully selected *utility functions* based on concepts from microeconomics. Second, we propose a distributed algorithm which automatically manages the interactions among the ECC units at the smart meters and the energy provider. The algorithm finds the optimal energy consumption levels for each subscriber to maximize the aggregate utility of all subscribers in the system in a fair and efficient fashion. Finally, we show that the energy provider can encourage some desirable consumption patterns among the subscribers by means of the proposed real-time pricing interactions. Simulation results confirm that the proposed distributed algorithm can potentially benefit both subscribers and the energy provider.

I. INTRODUCTION

Electricity is currently provided through an infrastructure consisting of utility companies, power plants, and transmission lines which serve millions of customers. For example, the electric power grid in the United States includes more than 3,100 electric utilities operating more than 10,000 power plants, and there are about 157,000 miles of high voltage electric transmission lines which bring energy to more than 131 million customers [1]. The dependency of almost all parts of industry and different aspects of our life on electrical energy makes this massive infrastructure a strategic entity.

Given the increased expectations of customers, both in quality and quantity [1], the limited energy resources, and the lengthy and expensive process of exploiting new resources, the reliability of the grid has been put in danger and there is a need to develop new methods to increase the grid efficiency. Currently, the electricity consumption is not efficient in most buildings (e.g., due to poor thermal isolation). This results in the waste of a large amount of natural resources, since most of the electricity consumption occurs in buildings [2]. In addition, the arising of new types of demand such as plug-in hybrid

electric vehicles (PHEVs), which can potentially double the average household load, have further increased the need to develop new methods for *demand side management* (DSM).

There is a wide range of DSM techniques such as voluntary load management programs [3]–[5] and direct load control [6]. However, *smart pricing* is known as one of the most common tools that can encourage users to consume wisely and more efficiently. Given the recent increases in the price of energy, the users are more willing to improve the insulation conditions of their buildings or try to shift the energy consumption schedule of their high-load household appliances to off-peak hours. DSM has been considered since the early 1980s [7]–[11]. DSM can be used as a tool for load shaping, where the electricity demand is being re-distributed over a certain period of time (e.g., time-of-day, day-of-week). Broad categories of load shaping objectives include peak clipping, load shifting, valley filling, strategic conservation, and flexible load shaping [7]. For example, peak clipping includes direct load control of the utilities on customers' appliances to reduce the peak load.

Several pricing schemes have already been proposed in the smart grid literature. In general, flat pricing, peak load pricing, and adaptive pricing are among the most popular approaches to pricing which have been practiced extensively [12]–[15]. Flat pricing refers to those methods where the utility company announces a fixed price for all periods. In peak load pricing, the intended cycle is divided into several periods and a distinct price value for each period is announced at the beginning of the operation [14]. On the other hand, in adaptive pricing, instead of announcing a pre-determined price for each period of operation at the beginning of the day, the exact price value for each period is calculated in real-time and is announced only at the beginning of each operation period. Clearly, in this method, the realization of random events and the reaction of users with respect to the previous prices will influence the price in the upcoming operation periods [12].

Based on a report of the U.S. Department of Energy [16], *smart grid* is an electricity delivery system enhanced with communication facilities and information technologies to enable more efficient and reliable grid operation with an improved customer service and a cleaner environment. By exploiting the *two-way* communication capabilities of smart meters it becomes possible to replace the current power system with a more intelligent infrastructure [17]. From this and given the importance of demand side management, in this paper,

we focus on the *real-time interactions* among subscribers and the energy provider and introduce a novel real-time pricing algorithm for the future smart grid. The contributions of this paper can be summarized as follows:

- We propose a real-time pricing algorithm for DSM programs to encourage desired energy consumption behaviors among users and to keep the total consumption level below the power generation capacity.
- In our system model, the subscribers and the energy provider automatically interact with each other through a limited number of message exchanges and by running a distributed algorithm to find the optimal energy consumption level for each subscriber, the optimal price values to be advertised by the energy provider, and also the optimal generating capacity for the energy provider.
- We model the subscriber's preferences and their energy consumption patterns in form of carefully selected *utility functions* based on concepts from microeconomics.
- We formulate the real-time pricing as an optimization problem to maximize the aggregate utility of all subscribers in the system while minimizing the imposed energy cost to the energy provider. Moreover, we include constraints to limit the total energy consumption level of all users to the total electricity generation capacity of the system offered by the energy provider.
- We prove the existence and the uniqueness of the optimal solution for the formulated optimization problem.
- Simulation results confirm that both subscribers and the energy provider will benefit from the proposed algorithm.

This paper is organized as follows. The system model is presented in Section II. In Section III, we formulate our design as a convex optimization problem and propose a distributed pricing algorithm. Simulation results are given in Section IV, and conclusions are drawn in Section V.

II. SYSTEM MODEL

Consider a smart power system consisting of a single energy provider, several load subscribers or users, and a regulatory authority. For each user, we assume that there is an *energy consumption controller* (ECC) unit which is embedded in the user's smart meter. The role of the ECC is to control the user's power consumption, and to coordinate each user with other users and also with the energy provider. All ECC units are connected to each other and to the energy provider through a communication infrastructure such as a local area network.

The intended time cycle for the operation of the users is divided into K time slots, where $K \triangleq |\mathcal{K}|$, and \mathcal{K} is the set of all time slots. This division can be based on the behavior of the users and their power demand pattern: peak load time slots, valley load time slots, and normal load time slots. Also, let \mathcal{N} denote the set of all users, where $N \triangleq |\mathcal{N}|$. For each user $i \in \mathcal{N}$, let x_i^k denote the amount of power consumed by user i in time slot k . For each subscriber $i \in \mathcal{N}$ and each time slot $k \in \mathcal{K}$, we define the power consumption interval I_i^k as

$$I_i^k \triangleq [m_i^k, M_i^k] \quad (1)$$

and the consumed power x_i^k has to satisfy $m_i^k \leq x_i^k \leq M_i^k$. Here, m_i^k and M_i^k denote the minimum and the maximum power consumption of user i , respectively. The minimum power consumption level may represent the load from appliances such as refrigerator which *always* need to be *on* during the day. The maximum power consumption level may also represent the *total* power consumption level of household appliances assuming that *all* appliances are *on*.

The regulatory authority ensures that the energy provider will provide the minimum capacity to cover the minimum power requirements of all users L_k^{min} in each time slot.

$$L_k^{min} \triangleq \sum_{i \in \mathcal{N}} m_i^k, \quad \forall k \in \mathcal{K}. \quad (2)$$

The *generation capacity* in each time slot $k \in \mathcal{K}$ is denoted by L_k , which may differ among time slots. We also define L_k^{max} as the maximum generating capacity in each time slot $k \in \mathcal{K}$.

A. User Preference and Utility Function

Each individual subscriber in a power system is an entity which can behave *independently*. The energy demand of each subscriber may vary based on different parameters. For example, we can take into account the time of day, climate conditions, and also the price of electricity. The energy demand also depends on the type of the users. For example, household users may have different *responses* to the same price than industrial users. The different response of different users to various price scenarios can be modeled analytically by adopting the concept of *utility function* from microeconomics [18]. In fact, we can model the behavior of different users through their different choices of utility functions [4]. For all users, we represent the corresponding utility function as $U(x, \omega)$, where x is the power consumption level of the user and ω is a parameter which may vary among users and also at different times of the day. More formally, for each user, the utility function represents the *level of satisfaction* obtained by the user as a function of its power consumption. We assume that the utility functions fulfill the following properties:

1) *Property I*: Utility functions are *non-decreasing*. That is, users are always interested to consume more power if possible until they reach their maximum consumption level. Mathematically, this implies that we have

$$\frac{\partial U(x, \omega)}{\partial x} \geq 0. \quad (3)$$

For notational convenience we define

$$V(x, \omega) \triangleq \frac{\partial U(x, \omega)}{\partial x}, \quad (4)$$

as the *marginal benefit* [3], [4].

2) *Property II*: The marginal benefit of customers is a non-increasing function and we have

$$\frac{\partial V(x, \omega)}{\partial x} \leq 0. \quad (5)$$

In other words, the utility functions are *concave* and the level of satisfaction for users can gradually get *saturated*. While the class of utility functions that fulfill (3) and (5) is very large, it is convenient to have a linear marginal benefit [3], [4].

3) *Property III*: We have to be able to rank the customers based on their utilities. In our formulation, we assume, for a fixed consumption level x , a larger ω implies a larger $U(x, \omega)$, which can be expressed as

$$\frac{\partial U(x, \omega)}{\partial \omega} > 0. \quad (6)$$

4) *Property IV*: We assume the general expectation that no power consumption brings no benefit, so we have

$$U(0, \omega) = 0, \quad \forall \omega > 0. \quad (7)$$

Various choices of utility functions are widely used in the communications and networking literature [19]. However, recent reports indicate that the behavior of power users can also be accurately modeled by certain utility functions [3]. In this paper, we consider *quadratic utility* functions corresponding to *linear decreasing marginal benefit* [5]:

$$U(x, \omega) = \begin{cases} \omega x - \frac{\alpha}{2} x^2 & \text{if } 0 \leq x \leq \frac{\omega}{\alpha}, \\ \frac{\omega}{\alpha} & \text{if } x \geq \frac{\omega}{\alpha}, \end{cases} \quad (8)$$

where α is a pre-determined parameter. Sample utility functions from this class are shown in Fig. 1.

A subscriber that consumes x kW electricity during a designated number of hours at a rate of P dollars per kWh is charged Px dollars per hour. Hence, the *welfare* of each user can simply be represented as

$$W(x, \omega) = U(x, \omega) - Px, \quad (9)$$

where $W(x, \omega)$ is the user's *welfare function*, $U(x, \omega)$ is the utility function of the user, Px is the cost imposed by the energy provider to the user, and x is the user's power consumption. For each announced price value P , each user tries to adjust its power consumption x to maximize its *own* welfare, and this can be achieved by setting the derivative of (9) equal zero which means that at the optimal consumption level, the marginal benefit of the user would be equal to the announced price. For example, different power consumption responses of a user with a decreasing linear marginal benefit to two different announced prices are depicted in Fig. 2.

B. Energy Cost Model

We consider a *cost function* $C_k(L_k)$ indicating the cost of providing L_k units of energy offered by the energy provider in each time slot $k \in \mathcal{K}$. We make the following assumptions:

Assumption 1: The cost functions are *increasing* in the offered energy capacity. That is, for each $k \in \mathcal{K}$, we have

$$C_k(\hat{L}_k) \leq C_k(\tilde{L}_k), \quad \forall \hat{L}_k \leq \tilde{L}_k. \quad (10)$$

Assumption 2: The cost functions are *strictly convex*. For each $k \in \mathcal{K}$, any $0 \leq \theta \leq 1$, and $\hat{L}_k, \tilde{L}_k \geq 0$, we have [20]

$$C_k(\theta \hat{L}_k + (1 - \theta) \tilde{L}_k) \leq \theta C_k(\hat{L}_k) + (1 - \theta) C_k(\tilde{L}_k). \quad (11)$$

Piece-wise linear functions and *quadratic functions* are two example cost functions that satisfy Assumption 1 and Assumption 2. In this paper, we consider quadratic cost functions [10]:

$$C_k(L_k) = a_k L_k^2 + b_k L_k + c_k, \quad (12)$$

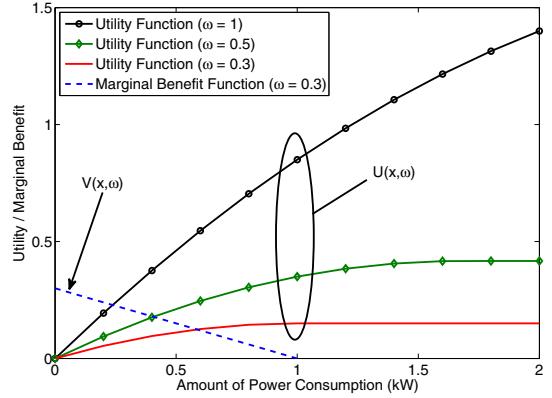


Fig. 1. Sample utility functions for power subscribers ($\alpha = 0.3$).

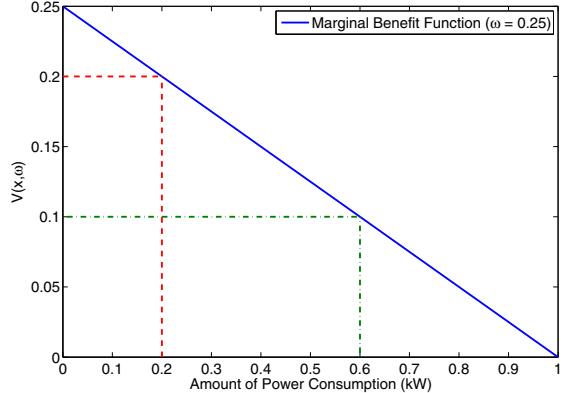


Fig. 2. Different power consumption reactions of a subscriber to two different announced prices ($P_1 = 0.2, P_2 = 0.1$ and $\alpha = 0.25$).

where $a_k > 0$ and $b_k, c_k \geq 0$ are pre-determined parameters.

III. REAL-TIME PRICING FORMULATION

In this section, we formulate the interactions between the users and the energy provider as an optimization problem and analyze the existence and uniqueness of the solution. In our model, the energy provider announces the price of electricity in *real-time* based on the total load demand.

A. Optimization Problem Formulation

From a social fairness point of view, it is desirable to utilize the available capacity provided by the energy provider in such a way that the sum of the utility functions of *all* subscribers is maximized and the cost imposed to the energy provider is minimized. However, each subscriber will choose its consumption level to maximize its *own* welfare function introduced in (9). These individually optimal consumption levels may *not* be socially optimal for a general price announced by the energy provider. To align these individual optimal consumption levels with the social optimal case, we need to adopt the sum of all utility functions *minus* the cost imposed to the energy provider as the objective function while the consumption levels of all users are coupled via the *limited available generation capacity*. Having a centralized control over all subscribers, and also being provided with complete information about the

subscribers' needs, an efficient energy consumption schedule can be characterized as the solution of the following problem:

$$\underset{\substack{x_i^k \in I_i^k, L_k^{min} \leq L_k \leq L_k^{max}, \\ i \in \mathcal{N}, k \in \mathcal{K}}}{\text{maximize}} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} U(x_i^k, \omega_i^k) - C_k(L_k)$$

$$\text{subject to} \quad \sum_{i \in \mathcal{N}} x_i^k \leq L_k, \quad \forall k \in \mathcal{K}, \quad (13)$$

where $U(x_i^k, \omega_i^k)$ is defined in (8), $C_k(L_k)$ is defined in (12), and ω_i^k is the ω parameter of user i in time slot k .

The problem formulated in (13) is a concave maximization problem and can be solved using *convex programming* techniques such as the interior point method (IPM) [20] in a central fashion. However, the problem arising in solving (13) in a central manner is that we need to know the exact utility function of users. Since it is assumed that the utility parameter ω_i^k for each user $i \in \mathcal{N}$ is private, the energy provider may not have sufficient information to solve problem (13).

B. Dual Decomposition Approach

We notice that (13) can be solved independently for each time slot $k \in \mathcal{K}$. In other words, we have the following optimization problem for each fixed time slot $k \in \mathcal{K}$:

$$\underset{\substack{x_i^k \in I_i^k, i \in \mathcal{N}, L_k^{min} \leq L_k \leq L_k^{max}}}{\text{maximize}} \sum_{i \in \mathcal{N}} U(x_i^k, \omega_i^k) - C_k(L_k)$$

$$\text{subject to} \quad \sum_{i \in \mathcal{N}} x_i^k \leq L_k. \quad (14)$$

Problem (14) is again convex and can be solved easily in a centralized manner. In practice, this problem has to be solved in a distributed fashion. Although the objective function in (14) is further separable in x_i^k and L_k , the variables x_i^k and L_k are coupled by the imposed constraint that the total consumed power cannot exceed the available capacity in (14).

For primal problem (14), the Lagrangian is defined as [20]:

$$\begin{aligned} \mathcal{L}(\mathbf{x}, L_k, \lambda^k) &= \sum_{i \in \mathcal{N}} U(x_i^k, \omega_i^k) - C_k(L_k) \\ &\quad - \lambda^k (\sum_{i \in \mathcal{N}} x_i^k - L_k), \\ &= \sum_{i \in \mathcal{N}} (U(x_i^k, \omega_i^k) - \lambda^k x_i^k) \\ &\quad + \lambda^k L_k - C_k(L_k), \end{aligned} \quad (15)$$

where λ^k is the Lagrange multiplier and $\mathbf{x} = (x_i^k, i \in \mathcal{N})$ for a fixed $k \in \mathcal{K}$. Due to the separability of the first term in the Lagrangian, we can write the objective function of the *dual optimization problem* as [20]:

$$\begin{aligned} \mathcal{D}(\lambda^k) &= \underset{\substack{x_i^k \in I_i^k, i \in \mathcal{N}, L_k^{min} \leq L_k \leq L_k^{max}}}{\text{maximize}} \quad \mathcal{L}(\mathbf{x}, L_k, \lambda^k) \\ &= \sum_{i \in \mathcal{N}} B_i^k(\lambda^k) + S_k(\lambda^k), \end{aligned} \quad (16)$$

where

$$B_i^k(\lambda^k) = \underset{x_i^k \in I_i^k}{\text{maximize}} \quad U(x_i^k, \omega_i^k) - \lambda^k x_i^k, \quad (17)$$

and

$$S_k(\lambda^k) = \underset{L_k^{min} \leq L_k \leq L_k^{max}}{\text{maximize}} \quad \lambda^k L_k - C_k(L_k). \quad (18)$$

The dual problem is

$$\underset{\lambda^k > 0}{\text{minimize}} \quad \mathcal{D}(\lambda^k). \quad (19)$$

The first term in $\mathcal{D}(\lambda^k)$ in (16) can be decomposed into N separable subproblems in form of (17), which can be solved by the users, and another subproblem in form of (18), which can be solved by the energy provider.

We can show that *strong duality* holds, and we can solve the dual problem (19) instead of the primal problem (14). In this case, we can obtain the solution of the dual problem λ^{k*} , and each individual subscriber and also the energy provider can simply solve their own local optimization problem determined by (17) and (18) to obtain x_i^{k*} and L_k^* , respectively.

The key idea which motivates us to propose a real-time pricing algorithm can be understood if we compare the local problem (17) that has to be solved by each individual user with (9), introducing each user's *welfare*. In fact, if the energy provider would be able to charge the users at a rate $P = \lambda^{k*}$, and each individual user tries to maximize its own welfare function, it will be guaranteed by strong duality that the total power consumption will not exceed the provided capacity.

C. Distributed Algorithm

We explained in the previous section that by charging the users with the solution of the dual problem λ^{k*} , we can achieve the solution of primal problem (14). Interestingly, it is possible to solve the dual problem in an iterative manner using the gradient projection method, and in this case we have

$$\begin{aligned} \lambda_{t+1}^k &= [\lambda_t^k - \gamma \frac{\partial \mathcal{D}(\lambda_t^k)}{\partial \lambda^k}]^+ \\ &= [\lambda_t^k + \gamma (\sum_{i \in \mathcal{N}} x_i^{k*}(\lambda_t^k) - L_k^*(\lambda_t^k))]^+, \end{aligned} \quad (20)$$

where $t \in \mathcal{T}$, and \mathcal{T} is the set of time instances at which the energy provider updates λ^k . Here, $x_i^{k*}(\lambda_t^k)$ is the local optimizer of (17), and $L_k^*(\lambda_t^k)$ is the local optimizer of (18) for a given λ_t^k , respectively. Also, λ_t^k is the value of λ^k in instance $t \in \mathcal{T}$, and γ is the step size. The interaction between the energy provider and the subscribers is depicted in Fig. 3.

The distributed algorithms of each subscriber and the energy provider are summarized in Algorithms 1 and 2, respectively. Consider Algorithm 1. In Line 1, each subscriber starts with its initial condition, which is assumed to be random. Then, the loop in Lines 2 to 6 describes the responses of each subscriber to the newly announced price λ^k . Within this loop, each subscriber receives the new value of λ^k in Line 3 and solves local problem (17) to get the optimal consumption $x_i^{k*}(\lambda^k)$ corresponding to the new value of λ^k in Line 4. In Line 5, the user communicates the new value of $x_i^{k*}(\lambda^k)$ to the energy provider. We note that in each time slot $k \in \mathcal{K}$, users apply their new loads only after the algorithm has converged.

In Algorithm 2, the energy provider starts with random initial conditions in Line 1. The loop in Lines 2 to 11 continues during the operational cycle of the system. Within this loop, the energy provider updates λ^k in each instance $t \in \mathcal{T}$ in Lines 4 and 5. It further calculates the new value of $L_k(\lambda^k)$ which

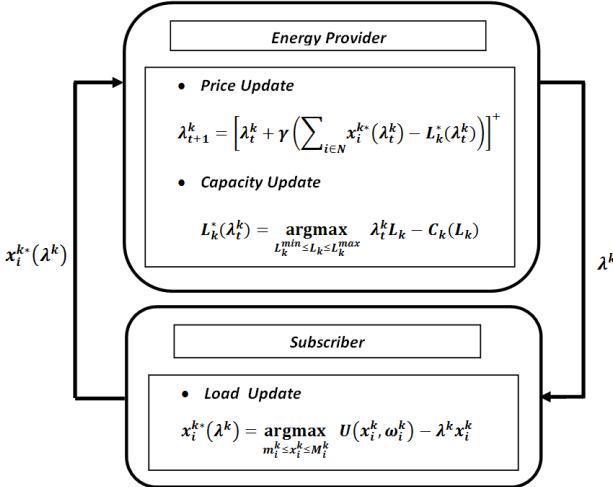


Fig. 3. Illustration of the operation of the proposed algorithm and the interactions between the energy provider and subscribers in the system.

Algorithm 1 : Executed by each subscriber $i \in \mathcal{N}$.

- 1: Initialization.
- 2: **for each** $t \in \mathcal{T}$
- 3: Receive the new value of λ^k from energy provider.
- 4: Update the consumption value $x_i^{k*}(\lambda^k)$ by solving (17).
- 5: Communicate the updated $x_i^{k*}(\lambda^k)$ to energy provider.
- 6: **end for**

maximizes its welfare and updates its information about the total consumption level of the system in Lines 7 to 9.

We note that network utility maximization has already been applied successfully in computer networking. The problem formulation in this section is similar to the congestion control problem in the Internet (e.g., [19]). However, the pricing algorithm in this paper differs from the rate allocation problem for the Internet in two aspects: (a) The capacity can be adjusted by the energy provider and may change periodically while the capacity constraint in [19] is fixed; (b) We consider the energy cost imposed to the energy provider and formulate the problem as utility maximization together with cost minimization.

IV. PERFORMANCE EVALUATION

In this section, we present simulation results and assess the performance of our proposed distributed algorithm. In our simulation model, we assume there are $N = 10$ subscribers. The entire time cycle is divided into 24 time slots representing the 24 hours of the day. The minimum and the maximum power requirements of all users vary in each time slot, and the minimum generating capacity to meet the minimum power requirements is guaranteed. However, we also assume the maximum generating capacity L_k^{max} is equal to the maximum total power requirements of all the users, so we have $L_k^{max} = \sum_{i \in \mathcal{N}} M_i^k$, for all $k \in \mathcal{K}$.

We also assume the ω parameter of each user is selected randomly from the interval $[1, 4]$ and remains fixed within the entire cycle. Parameter α of the utility function introduced in

Algorithm 2 : Executed by the energy provider.

- 1: Initialization.
 - 2: **repeat**
 - 3: **if** time $t \in \mathcal{T}$
 - 4: Compute the new value of λ^k using (20).
 - 5: Broadcast the new value of λ^k to all the subscribers.
 - 6: **else**
 - 7: Update the capacity value $L_k(\lambda^k)$ by solving (18).
 - 8: Receive $x_i^{k*}(\lambda^k)$ from all the subscribers $i \in \mathcal{N}$.
 - 9: Update the total load $\sum_{i \in \mathcal{N}} x_i^{k*}(\lambda^k)$ accordingly.
 - 10: **end**
 - 11: **until** end of intended period.
-

(8) is chosen to be 0.5, and we set the parameters of the cost function introduced in (12) to $a_k = 0.01$, $b_k = 0$, and $c_k = 0$.

Simulation results for the total consumed power for the proposed algorithm are shown in Fig. 4. As illustrated in Fig. 4, due to real-time interaction of the subscribers and the energy provider, the two curves corresponding to the total power consumption of the users and the desired generating capacity of the energy provider coincide. The high utilization of the available resources while keeping the total power consumption below the desired threshold is one of the advantages of the proposed algorithm. As expected, the generating capacity and also the total power consumption are bounded within the minimum and the maximum total power requirements of all the users in each time slot.

To have a baseline scheme for comparison with the proposed real-time pricing strategy, we also consider a fixed pricing scenario with a hard constraint to keep the total consumption below the generating capacity without *interaction* with the users. In the fixed pricing algorithm, the energy provider announces a price for each time slot $k \in \mathcal{K}$ at the beginning of the time slot which guarantees for any type of users with different choices of the ω parameter that the total consumption level will not exceed the generating capacity. Therefore, in the fixed pricing algorithm, the worst case situation where the ω parameter of all the users assumes the maximum value $\omega_{max} = 4$ is being considered. Hence, the price in each time slot $k \in \mathcal{K}$ can be calculated as

$$P_{fixed}^k = \omega_{max} - \frac{L^k \alpha}{N}. \quad (21)$$

Simulation results for the aggregate utility of all users for the two different methods are shown in Fig. 5. We can see that the aggregate utility is much higher for our proposed real-time pricing algorithm than for the fixed pricing algorithm.

Last but not least, our proposed distributed real-time pricing algorithm can also benefit the users. Let us consider 24 time slots with different power requirements for different users in each time slot. Simulation results for the time averaged welfare of each individual subscriber for our proposed real-time pricing algorithm as well as the fixed pricing algorithm are shown in Fig. 6. We can see that the average welfare of each individual subscriber is much higher for our proposed

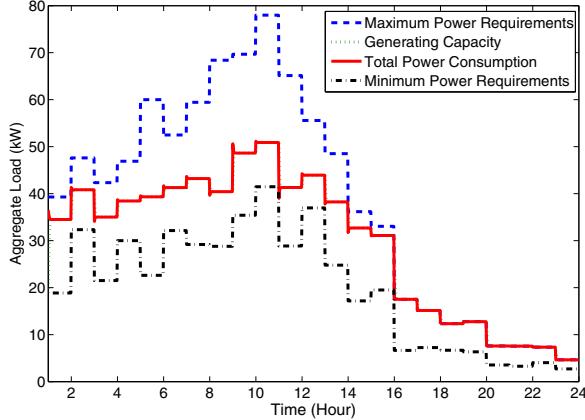


Fig. 4. Total consumed power when the proposed pricing algorithm is used.

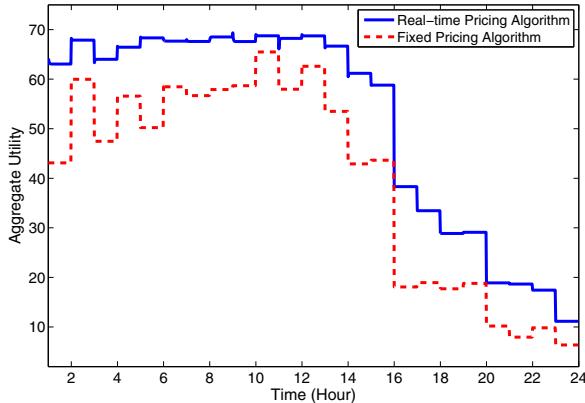


Fig. 5. Obtained aggregate utility of all users when our proposed real-time pricing algorithm as well as a fixed pricing algorithm are used.

algorithm than for the fixed pricing algorithm.

V. CONCLUSIONS

In this paper, we proposed an optimal real-time pricing algorithm for DSM in the future smart grid. The proposed algorithm is based on utility maximization. It can be implemented in a distributed manner to maximize the aggregate utility of all users and minimize the cost imposed to the energy provider while keeping the total power consumption below the generating capacity. Simulation results confirmed that by using our proposed optimization-based real-time pricing model, not only the energy provider, but also the users will benefit. The ideas developed in this paper can be extended in several directions. A system with multiple energy providers can be considered. The effect of malicious users can also be explored.

ACKNOWLEDGMENT

This research is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

REFERENCES

- [1] L. H. Tsoukalas and R. Gao, "From smart grids to an energy internet: Assumptions, architectures, and requirements," in *Proc. of Third Int'l. Conf. on Electric Utility Deregulation and Restructuring and Power Technologies*, Nanjing, China, Apr. 2008.
- [2] U.S. Department of Energy, *The 2008 Buildings Energy Data Book*. Energy Efficiency and Renewable Energy, Mar. 2009.
- [3] M. Fahrioglu, M. Fern, and F. Alvarado, "Designing cost effective demand management contracts using game theory," in *Proc. of IEEE Power Eng. Soc. 1999 Winter Meeting*, New York, NY, Jan. 1999.
- [4] M. Fahrioglu and F. Alvarado, "Using utility information to calibrate customer demand management behavior models," *IEEE Trans. on Power Systems*, vol. 16, no. 2, pp. 317–322, May 2001.
- [5] R. Faranda, A. Pievatolo, and E. Tironi, "Load shedding: A new proposal," *IEEE Trans. on Power Systems*, vol. 22, no. 4, pp. 2086–2093, Nov. 2007.
- [6] N. Ruiz, I. Cobelo, and J. Oyarzabal, "A direct load control model for virtual power plant management," *IEEE Trans. on Power Systems*, vol. 24, no. 2, pp. 959–966, May 2009.
- [7] C. Gellings, "The concept of demand-side management for electric utilities," *Proceedings of the IEEE*, vol. 73, no. 10, pp. 1468–1470, Oct. 1985.
- [8] M. Fahrioglu and F. Alvarado, "Designing incentive compatible contracts for effective demand management," *IEEE Trans. on Power Systems*, vol. 15, no. 4, pp. 1255–1260, Nov. 2000.
- [9] B. Ramanathan and V. Vittal, "A framework for evaluation of advanced direct load control with minimum disruption," *IEEE Trans. on Power Systems*, vol. 23, no. 4, pp. 1681–1688, Oct. 2008.
- [10] A. H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, and R. Schober, "Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid," in *Proc. of IEEE PES Conf. on Innovative Smart Grid Technologies*, Gaithersburg, MD, Jan. 2010.
- [11] A. H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Trans. on Smart Grid*, vol. 1, no. 2, Sept. 2010.
- [12] P. Luh, Y. Ho, and R. Muralidharan, "Load adaptive pricing: An emerging tool for electric utilities," *IEEE Trans. on Automatic Control*, vol. 27, no. 2, pp. 320–329, Apr. 1982.
- [13] Y. Tang, H. Song, F. Hu, and Y. Zou, "Investigation on TOU pricing principles," in *Proc. of IEEE PES Transmission and Distribution Conf. Exhibition: Asia and Pacific*, Dalian, China, Aug. 2005.
- [14] M. Crew, C. Fernando, and P. Kleindorfer, "The theory of peak-load pricing: A survey," *Journal of Regulatory Economics*, vol. 8, no. 3, pp. 215–248, Nov. 1995.
- [15] S. Zeng, J. Li, and Y. Ren, "Research of time-of-use electricity pricing models in China: A survey," in *Proc. of IEEE Int'l. Conf. on Industrial Engineering and Engineering Management*, Singapore, Dec. 2008.
- [16] U.S. Department of Energy, *The Smart Grid: An Introduction*, 2009.
- [17] A. Vojdani, "Smart integration," *IEEE Power and Energy Magazine*, vol. 6, no. 6, pp. 72–79, Nov. 2008.
- [18] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*, 1st ed. USA: Oxford University Press, 1995.
- [19] S. Low and D. Lapsley, "Optimization flow control I: Basic algorithm and convergence," *IEEE/ACM Trans. on Networking*, vol. 7, no. 6, pp. 861–874, Dec. 1999.
- [20] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

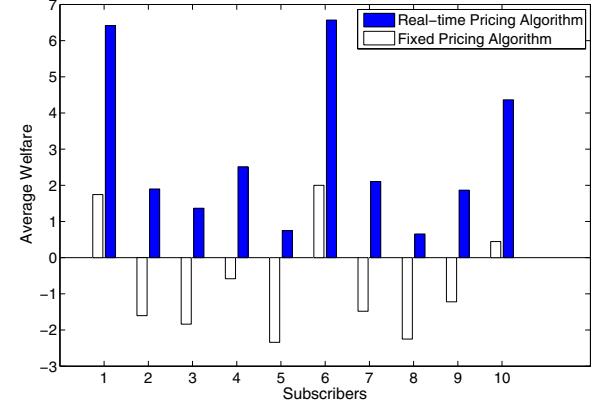


Fig. 6. Average welfare of each subscriber when our proposed real-time pricing algorithm as well as a fixed pricing algorithm are used.

Decentralized Coordination of Energy Utilization for Residential Households in the Smart Grid

Yuanxiong Guo, *Student Member, IEEE*, Miao Pan, *Student Member, IEEE*, Yuguang Fang, *Fellow, IEEE*, and Pramod P. Khargonekar, *Fellow, IEEE*

Abstract—In this paper, we investigate the minimization of the total energy cost of multiple residential households in a smart grid neighborhood sharing a load serving entity. Specifically, each household may have renewable generation, energy storage as well as inelastic and elastic energy loads, and the load serving entity attempts to coordinate the energy consumption of these households in order to minimize the total energy cost within this neighborhood. The renewable generation, the energy demand arrival, and the energy cost function are all stochastic processes and evolve according to some, possibly unknown, probabilistic laws. We develop an online control algorithm, called Lyapunov-based cost minimization algorithm (LCMA), which jointly considers the energy management and demand management decisions. LCMA only needs to keep track of the current values of the underlying stochastic processes without requiring any knowledge of their statistics. Moreover, a distributed algorithm to implement LCMA is also developed, which can preserve the privacy of individual household owners. Numerical results based on real-world trace data show that our control algorithm can effectively reduce the total energy cost in the neighborhood.

I. INTRODUCTION

The growing demands of electricity and concerns over global climate change and carbon emission have motivated the grid modernization, which transforms the current power grids to the future “smart grid”. As stated in [1], the smart grid will enable deep penetration of renewable generation, customer driven demand response, widespread adoption of electric vehicles, and electric energy storage. Sensing, communication, computation, and control technologies in conjunction with advances in renewable generation, energy storage, power electronics, etc. are critical to realizing the vision and promise of the smart grid.

Within the smart grid, demand side management (DSM) is a key component, which can help reduce peak load, increase grid reliability, and lower generation cost [2]. There are mainly two types of demand side management techniques: direct load control (DLC) and demand response based on time-varying pricing [3]. In DLC, the load serving entity, usually a utility company, enters into a contract with the consumers beforehand, so that certain amount of energy load can be curtailed during the peak hours in order to release the congestion on the power grid or to avoid the operation of high

Y. Guo, M. Pan, Y. Fang, and P. P. Khargonekar are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, 32611, USA (email: {guoyuanxiong@, miaopan@, fang@ece., ppk@}ufl.edu).

This work was partially supported by the U.S. National Science Foundation under Grants CNS-0916391, CNS-1147813, ECCS-1129062, ECCS-1129061, and Eckis Professor endowment at the University of Florida.

cost peak generators. Currently, it is mainly employed by large industrial and commercial customers. On the other hand, the demand response based on time-varying pricing encourages the customers to adjust their normal energy consumption, either reducing or shifting consumption, based on the pricing signal issued by the load serving entity (LSE) in return for some benefits, such as reduced electricity bill. Several popular schemes already exist in this regard, such as critical-peak pricing (CPP), time-of-use (TOU) pricing, and real-time pricing (RTP). With the introduction of the advanced metering infrastructure (AMI), which can provide two-way communication between utility companies and smart meters, it is expected that there will be a widespread deployment of such demand response programs for residential and business customers in the smart grid [4].

Meanwhile, nearly 7% of electricity is lost during transmission and distribution (T&D) from remote power plants to distant homes [5]. Distributed generation (DG) from many small on-site energy sources deployed at individual homes and businesses can be used to decrease both T&D losses and carbon emissions. Typical examples of these small on-site energy sources include rooftop solar panels, fuel cells, microturbines, and micro-wind generators. Distributed energy storage devices are usually used in combination with these renewable sources to better utilize them. We envision residential households in the smart grid which use on-site renewable generation, modest energy storage, and the electric grid to meet their energy demands, within which some are elastic and can be served in a flexible manner. How to simultaneously manage these components for households within a neighborhood in order to reduce the total energy cost as well as the impact on the distribution network of the power grids is a challenging problem, especially considering the random dynamics in the system.

There have been many previous studies on energy consumption scheduling in households, renewable energy integration, and demand response schemes. On the residential energy consumption scheduling side, Mohsenian-Rad et al. [6] formulate the optimal control of multiple flexible appliances as a linear program to achieve a desired trade-off between the electricity payment and the waiting time for the operation of each appliance in a household, where customers are subject to a real-time pricing tariff combined with inclining block rates. A game theory based approach is proposed in [7] to handle the case of multiple households. Kim et al. [8] use dynamic programming to solve the problem of causally scheduling power consumption of a single appliance in real-time pricing

environment in order to minimize the expected cost. On the side of renewable energy integration, the problem of supplying renewable energy to demand-flexible customers is investigated in [9], [10]. Specifically, Papavasiliou et al. [10] address the optimal allocation for renewable sources to demand-flexible customers in a real-time pricing environment using dynamic programming, while in [9], the authors develop a Lyapunov optimization based method. Bitar et al. [11] consider optimal selling strategies for uncertain and variable wind production into the current electricity market. On the demand response side, Li et al. [12] consider the problem of optimal demand response as a convex optimization problem and study the role of dynamic pricing. Jiang et al. [13] propose a model that integrates two-period electricity markets, uncertainty in renewable generation, and real-time dynamic demand response, and derive the optimal control decisions to optimize the social welfare. However, most of the previous studies either only consider optimization for one household, or assume perfect future information, or do not consider on-site distributed generation and energy storage.

This paper extends the single household case in our previous work [14] to that of multiple households within a smart grid neighborhood. In our work, not only does the energy cost of an individual household matter, but also the total energy cost in a neighborhood is of equal importance. We show that, through our collaborative and distributed energy consumption scheduling algorithm in multiple households, the impact of the household energy consumption on the power system and the total energy cost can be greatly reduced. Due to the distributed and online properties of our proposed algorithm, it can be easily implemented in the smart grid. In summary, our paper makes the following contributions:

- In the setting of multiple households within a neighborhood, we propose a new system architecture to incorporate the following essential components in the smart grid: distributed renewable generation, energy storage, demand response, and smart appliances.
- We develop a distributed online algorithm, called Lyapunov-based cost minimizing algorithm (LCMA), to approximately minimize time-average total energy cost for households within a neighborhood without the knowledge of the statistics of related stochastic models.
- Through theoretical analysis, we show that our algorithm can obtain an explicit trade-off between cost saving and energy storage capacity. Moreover, through extensive simulations based on real-world data sets, we demonstrate the effectiveness of our proposed algorithms.

The paper is organized as follows. In Section II, we describe our system model and formulate the problem as a stochastic programming problem. We describe the design principle behind our algorithm and present an online algorithm in Section III. We then analyze our algorithm in Section IV. We present numerical results based on real-world data in Section V. Finally, some concluding remarks are presented in Section VI.

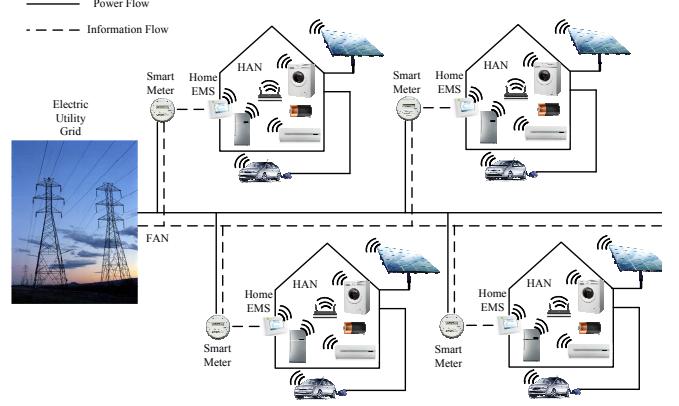


Fig. 1. Schematic of Household Energy Management in a Smart Grid Neighborhood

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we provide mathematical descriptions for load serving entity, energy load, energy storage, and distributed renewable generation in residential households. Based on these definitions, we will formulate our control problem as a stochastic program.

Consider a set of N households/customers that are served by a single load serving entity (LSE) in a smart grid neighborhood setting as depicted in Fig. 1. The LSE may be a utility company and the smart grid neighborhood may cover all households connected to a step-down transformer in the distribution network of power grids. The LSE participates into wholesale electricity markets (day-ahead, hour-ahead, real-time balancing, ancillary service) to purchase electricity from power generators and then sell it to the N customers in the retail market. Currently, the electricity price in the retail market is usually flat because of the simplicity and predictability. However, it does not encourage efficient usage of electricity, causing high peak demand and low load factor. We consider a time-slotted model with an infinite horizon. Each slot represents a suitable period for control decisions (e.g., 1 hour or 15 min) and is indexed by $t = \{0, 1, \dots\}$.¹

A. Load Serving Entity

The LSE serves as an agent that is responsible for purchasing enough electricity from wholesale electricity markets to serve the energy demand of the households in its service area. The retail price is set in order to at least recover the running cost of the LSE. In the future smart grid, field area network (FAN) would be deployed, which can provide convenient communications between utility companies and smart meters of residential households. For simplicity, we make the assumption that the cost of the LSE can be represented by a time-varying cost function $C_t(D)$ that specifies the cost of providing amount D of electricity to the N customers at time slot t . We assume that the cost function $C_t(D)$ is increasing, continuously differentiable, and convex in D for any t with

¹In this paper, all power quantities such as $r_i(t)$, $s_i(t)$, $y_i(t)$, $d_{i,1}(t)$, $d_{i,2}(t)$ are in the unit of energy per slot, so the energy produced/consumed in time period t are $r_i(t)$, $s_i(t)$, $y_i(t)$, $d_{i,1}(t)$, $d_{i,2}(t)$, respectively.

a bounded first derivative. We use α^{\min} and α^{\max} to denote the minimum and the maximum first derivatives of $C_t(D)$, respectively.

B. Energy Load

In general, the energy loads in a household can be roughly divided into two categories: inelastic and elastic loads. Examples of inelastic energy loads include lights, TVs, microwaves, and computers. For this type of energy loads, the energy requests must be met exactly at the time t when needed. In contrast, there are some energy loads in households that are elastic in the sense that they can be controlled (using smart appliances, for example,) to adjust the times of their operations and the amount of their energy usage without impacting the satisfaction of customers. Examples include refrigerators, dehumidifiers, air conditioners, and electric vehicles. Actually, the vast majority of household loads are inelastic. However, as observed in [15], while the elastic energy loads comprise less than 7.5% of the total loads in a household, they account for 59% of the average energy consumption. Therefore, there is great hidden potential in exploiting the inherent flexibility of such elastic loads for various important individual and system level objectives.

Inside a household, electric loads can communicate with the smart meter via the home area network (HAN), which may be Wi-Fi or ZigBee. For each household $i \in N$, denote by $d_{i,1}(t)$ the inelastic energy loads (in unit of kWh) and by $d_{i,2}(t)$ the elastic energy loads (in unit of kWh) at time t . As in [9], we assume that the elastic energy loads are “buffered” (i.e., the energy requests are held or delayed) first in a queue $Q_i(t)$ before being served. Denote by $y_i(t)$ the amount of energy that is used for serving the queued energy loads at time t . Then the dynamics of $Q_i(t)$ is as follows:

$$Q_i(t+1) = \max\{Q_i(t) - y_i(t), 0\} + d_{i,2}(t), \quad \forall i. \quad (1)$$

For each i , we assume that

$$0 \leq y_i(t) \leq y_i^{\max}, \quad (2)$$

where $y_i^{\max} \geq d_{i,2}^{\max}$ so that the queue Q_i can always be stabilized. For any feasible control decision, we need to ensure that the average delay of the elastic loads in the queue is finite. In other words, we cannot delay arbitrarily long time for the service of elastic energy loads. This can be stated as follows:

$$\overline{Q_i} \doteq \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{Q_i(t)\} < \infty. \quad (3)$$

C. Energy Storage

In addition to energy loads, each household may have some kind of energy storage device, possibly in the form of the battery in PHEV. For each household i , we denote by E_i^{\max} the battery capacity, by $E_i(t)$ the energy level of the battery at time t , and by $r_i(t)$ the power charged to (when $r_i(t) > 0$) or discharged from (when $r_i(t) < 0$) the battery during slot t . Assume that the battery energy leakage is negligible and

batteries at households operate independently of each other. Then we model the dynamics of the battery energy level by

$$E_i(t+1) = E_i(t) + r_i(t). \quad (4)$$

For each household i , the battery usually has an upper bound on the charge rate, denoted by r_i^{\max} , and an upper bound on the discharge rate, denoted by $-r_i^{\min}$, where r_i^{\max} and $-r_i^{\min}$ are positive constants depending on the physical properties of the battery. Therefore, we have the following constraint on $r_i(t)$:

$$r_i^{\min} \leq r_i(t) \leq r_i^{\max}. \quad (5)$$

The battery energy level should be always nonnegative and cannot exceed the battery capacity. So in each time slot t , we need to ensure that for each household i ,

$$0 \leq E_i(t) \leq E_i^{\max}. \quad (6)$$

From constraints (4), (5), and (6), we obtain the following equivalent constraints in each slot t at household i :

$$r_i(t) \geq \max\{r_i^{\min}, -E_i(t)\}, \quad (7)$$

$$r_i(t) \leq \min\{r_i^{\max}, E_i^{\max} - E_i(t)\}. \quad (8)$$

However, the cost of battery use cannot be ignored. In practice, there are limited times of charging/discharging cycles for each battery. Besides, conversion loss occurs both in charging and discharging processes. Stored energy is also subject to leakage with time. All these factors depend on how fast/much/often it is charged and discharged. Instead of modeling these factors exactly, we use an amortized time-invariant cost function $F_i(r_i)$ (in unit of dollars) to model the impact of charging or discharging operation r_i on the battery during one slot for household i . Each battery cost function $F_i(r_i)$ is assumed to be increasing, continuously differentiable, and convex in r_i with a bounded first derivative and $F_i(0) = 0$. We use β_i^{\min} and β_i^{\max} to denote the minimum and the maximum first derivatives of $F_i(r_i)$ for each household i , respectively.

D. Renewable Distributed Generation (DG)

Each household i may possess a distributed renewable generator installed on its site, such as rooftop PV panel or small wind turbine. Denote by $s_i(t)$ the renewable energy generated in slot t by the renewable DG, which is usually intermittent, uncertain, and uncontrollable. In this paper, we assume it is i.i.d. across different slots and has the maximum value given by its rated power s_i^{\max} . Therefore, we have

$$0 \leq s_i(t) \leq s_i^{\max} \quad \forall i, t. \quad (9)$$

Note that the energy generation density from renewable generator is usually lower than the normal energy consumption density of households. Households need to connect to the utility electric grid for backup power and, therefore, are mostly grid-tied systems. In this paper, we assume that the renewable energy is free and should be utilized as much as possible.

E. Problem Formulation

With the above models for the battery and the distributed renewable generator, at each time t , the total power demand of household i needed from the utility electric grid is

$$g_i(t) \doteq \max\{d_{i,1}(t) + y_i(t) + r_i(t) - s_i(t), 0\}. \quad (10)$$

Note that in the formula above, we have assumed that power cannot be fed from the household into the utility electric grid through, for example, net metering. We plan to incorporate the option of two-way energy flow in our future investigation.

In this paper, we are interested in minimizing the time-average total energy cost of the whole smart grid neighborhood, or the negative of social welfare, which equals to the generation cost plus the battery cost. Therefore, the control problem can be stated as follows: for the dynamic system defined by equations (1) and (4), design a control strategy which, given the past and present random renewable supplies, the battery energy levels, the energy demands, and the energy cost function, chooses the battery charge/discharge vector \mathbf{r} and the elastic load serving rate vector \mathbf{y} such that the time-average total energy cost of the whole smart grid neighborhood is minimized. It can be formulated as the following stochastic programming problem, called **P1**:

$$\min_{\mathbf{y}, \mathbf{r}} : \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{C_t(\sum_{i=1}^N g_i(t)) + \sum_{i=1}^N F_i(r_i(t))\}, \quad (11)$$

s.t.

$$r_i(t) \geq \max\{r_i^{min}, -E_i(t)\}, \quad \forall i, t, \quad (12)$$

$$r_i(t) \leq \min\{r_i^{max}, E_i^{max} - E_i(t)\}, \quad \forall i, t, \quad (13)$$

$$0 \leq y_i(t) \leq y_i^{max}, \quad \forall i, t, \quad (14)$$

$$\overline{Q_i} < \infty, \quad \forall i. \quad (15)$$

Here the expectation in the objective is w.r.t. the random renewable generation $s_i(t)$, the random inelastic energy loads $d_{i,1}(t)$, the random elastic energy loads $d_{i,2}(t)$ for each household, and the random cost function C_t . Define $\mathbf{P1}^*$ as the infimum time average cost associated with **P1**, considering all feasible control actions subject to the queue stability and the finite battery energy level. We will design a control algorithm, parameterized by a constant $V > 0$, that satisfies the constraints above and achieves the average cost within $O(1/V)$ of the optimal value $\mathbf{P1}^*$. Moreover, it can guarantee that the worst-case delay is within $O(1/V)$.

III. ONLINE DISTRIBUTED ALGORITHM

In this section, we design algorithms to solve **P1**. One challenge of solving the stochastic optimization problem above is the uncertainty of future renewable generation, time-varying cost function, inelastic or elastic energy loads. Moreover, the constraints on $E_i(t)$ bring the ‘‘time-coupling’’ property to the stochastic optimization problem above. That is to say, the current control action may impact the future control actions, making it more challenging to solve. Our solution is based on the technique of Lyapunov optimization [16] and requires minimum information on the random dynamics in the system.

To carry out our design, we first consider a related problem and obtain a lower bound on the optimal operation cost of **P1**, which is useful for our later analysis. After that, we will describe our proposed online algorithms to approximately solve it.

A. Relaxed Problem

Before presenting the solution to **P1**, we first consider a relaxed problem. Define the time-average expected value of battery charge/discharge rate for household i under any feasible control policy of **P1** as follows:

$$\overline{r_i} = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{r_i(t)\}. \quad (16)$$

Since the dynamics of battery energy level is (4), summing over all $t \in \{0, 1, \dots, T-1\}$, and taking expectation of both sides, we obtain

$$\mathbb{E}\{E_i(T)\} - E_i(0) = \sum_{t=0}^{T-1} \mathbb{E}\{r_i(t)\}. \quad (17)$$

As $0 \leq E_i(t) \leq E_i^{max}$ for all slots t , dividing both sides by T , and taking $T \rightarrow \infty$ yields $\overline{r_i} = 0$. Hence we have the following relaxed problem, called **P2**:

$$\min_{\mathbf{y}, \mathbf{r}} : \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{C_t(\sum_{i=1}^N g_i(t)) + \sum_{i=1}^N F_i(r_i(t))\}, \quad (18)$$

s.t.

$$r_i^{min} \leq r_i(t) \leq r_i^{max}, \quad \forall i, t, \quad (19)$$

$$\overline{r_i} = 0, \quad \forall i, \quad (20)$$

$$0 \leq y_i(t) \leq y_i^{max}, \quad \forall i, t, \quad (21)$$

$$\overline{Q_i} < \infty, \quad \forall i. \quad (22)$$

Denote by $\mathbf{P1}^*$ the optimal objective value of **P1** and by $\mathbf{P2}^*$ the optimal objective value of **P2**, respectively. Since any feasible solution to **P1** is also a feasible solution to **P2**, $\mathbf{P2}^* \leq \mathbf{P1}^*$. Note that **P2** is a decoupled control problem since no correlation exists in any constraint. From the framework of Lyapunov optimization [16], we have the following theorem for the solution to **P2**:

Theorem 1: If $s_i(t)$, $d_{i,1}(t)$, $d_{i,2}(t)$, and $C_t(\cdot)$ are i.i.d. over slots, then there exists a stationary, randomized policy that takes control decisions \mathbf{y}^{stat} and \mathbf{r}^{stat} every slot purely as a function (possibly randomized) of the current system status while satisfying the constraints of **P2** and providing the following guarantees:

$$\mathbb{E}\{r_i^{stat}(t)\} = 0, \quad \forall i, \quad (23)$$

$$\mathbb{E}\{y_i^{stat}(t)\} \geq \mathbb{E}\{d_{i,2}(t)\}, \quad \forall i, \quad (24)$$

$$\mathbb{E}\{C_t(\sum_{i=1}^N g_i^{stat}(t)) + \sum_{i=1}^N F_i(r_i^{stat}(t))\} = \mathbf{P2}^*, \quad (25)$$

where the expectation in the objective is w.r.t. the random renewable generation $s_i(t)$, the random inelastic energy loads

$d_{i,1}(t)$, the random elastic energy loads $d_{i,2}(t)$ for each household, and the random cost function C_t .

Proof: It can be proven using Caratheodory's theorem in [16], [17] and is similar to that in [18]. It is omitted here for brevity. ■

In order to derive such a stationary policy, we need to know the statistical distributions of all combinations of $s_i(t)$, $d_{i,1}(t)$, $d_{i,2}(t)$, and $C_t(\cdot)$, which usually has the problem of “curse of dimensionality” [19] if solved by dynamic programming. Moreover, this control policy may not be a feasible solution to **P1**. Instead, we use the existence of such a policy to help us design our control policy that meets all constraints of **P1** and derive the analytical performance of our algorithm as illustrated in the proof of our algorithm performance properties later.

B. Delay-Aware Virtual Queue

Since the constraint $\overline{Q}_i < \infty$ only ensures finite average delay for the elastic energy loads in household i , worst-case delay guarantee is usually desired in practice. For this purpose, we leverage the technique of “virtual queue” in the Lyapunov optimization framework. Specifically, the following virtual queues $Z_i(t)$, $i = 1, 2, \dots, N$ are defined to provide the worst-case delay guarantee on any buffered elastic energy loads in $Q_i(t)$:

$$Z_i(t+1) = \max\{Z_i(t) - y_i(t) + \epsilon_i 1_{\{Q_i(t)>0\}}, 0\}, \quad (26)$$

where $1_{\{Q_i(t)>0\}}$ is an indicator function that is 1 if $Q_i(t) > 0$ or 0 otherwise; ϵ_i is a fixed positive parameter to be specified later. The intuition behind this virtual queue is that since $Z_i(t)$ has the same service process as $Q_i(t)$, but has an arrival process that adds ϵ_i whenever the actual backlog is nonempty, this ensures that $Z_i(t)$ grows if there are energy loads in the queue $Q_i(t)$ that have not been serviced for a long time. The following lemma shows that if we can control the system to ensure that the queues $Q_i(t)$ and $Z_i(t)$ have finite upper bounds, then any buffered energy load is served within a worst-case delay as follows:

Lemma 1: Suppose we can control the system to ensure that $Z_i(t) \leq Z_i^{\max}$ and $Q_i(t) \leq Q_i^{\max}$ for all slots t , where Z_i^{\max} and Q_i^{\max} are some positive constants. Then, the worst-case delay for all buffered energy loads in household i is upper bounded by δ_i^{\max} slots where

$$\delta_i^{\max} \triangleq \lceil \frac{(Q_i^{\max} + Z_i^{\max})}{\epsilon_i} \rceil. \quad (27)$$

Proof: The proof follows directly from the framework of Lyapunov optimization [16] and is given in Appendix A for completeness. ■

We will show that there indeed exist such constants Z_i^{\max} and Q_i^{\max} for all households i later.

C. The Lyapunov-based Approach

The idea of our algorithm is to construct a Lyapunov-based scheduling algorithm with perturbed weights for determining the optimal power usage. By carefully perturbing the weights, we can ensure that whenever we charge or discharge the

battery, the energy level in the battery always lies in the feasible region.

First, we choose a perturbation vector $\theta = (\theta_i, \forall i)$ (to be specified later). We define a perturbed Lyapunov function as follows:

$$L(t) \doteq \frac{1}{2} \sum_{i=1}^N [(E_i(t) - \theta_i)^2 + Q_i^2(t) + Z_i^2(t)]. \quad (28)$$

Now define $\mathbf{K}(t) = (\mathbf{Q}(t), \mathbf{Z}(t), \mathbf{E}(t))$, and define a one-slot conditional Lyapunov drift as follows:

$$\Delta(t) = \mathbb{E}\{L(t+1) - L(t) \mid \mathbf{K}(t)\}. \quad (29)$$

Here the expectation is taken over the randomness of load arrivals, cost function, and renewable generation, as well as the randomness in choosing the control actions. Then, following the Lyapunov optimization framework, we add a function of the expected cost over one slot (i.e., the penalty function) to (29) to obtain the following *drift-plus-penalty* term:

$$\Delta_V(t) \doteq \Delta(t) + V \mathbb{E}\{C_t(\sum_{i=1}^N g_i(t)) + \sum_{i=1}^N F_i(r_i(t)) \mid \mathbf{K}(t)\}, \quad (30)$$

where V is a positive control parameter to be specified later. Then, we have the following lemma regarding the *drift-plus-penalty* term:

Lemma 2: For any feasible action under constraints (12), (13), (14), (15) that can be implemented at slot t , we have

$$\begin{aligned} \Delta_V(t) &\leq B + \sum_{i=1}^N \mathbb{E}\{(E_i(t) - \theta_i)r_i(t) \mid \mathbf{K}(t)\} \\ &\quad + \sum_{i=1}^N \mathbb{E}\{Q_i(t)(d_{i,2}(t) - y_i(t)) \mid \mathbf{K}(t)\} \\ &\quad + \sum_{i=1}^N \mathbb{E}\{Z_i(t)(\epsilon_i - y_i(t)) \mid \mathbf{K}(t)\} \\ &\quad + V \mathbb{E}\{C_t(\sum_{i=1}^N g_i(t)) + \sum_{i=1}^N F_i(r_i(t)) \mid \mathbf{K}(t)\}, \end{aligned} \quad (31)$$

where B is a constant given by

$$\begin{aligned} B &\doteq \sum_{i=1}^N \left\{ \frac{\max\{(r_i^{\min})^2, (r_i^{\max})^2\}}{2} + \frac{\max\{(y_i^{\max})^2, \epsilon_i^2\}}{2} \right. \\ &\quad \left. + \frac{(y_i^{\max})^2 + (d_{i,2}^{\max})^2}{2} \right\}. \end{aligned} \quad (32)$$

Proof: From (4), subtracting both sides by θ_i , and squaring both sides, we have for each household i ,

$$\frac{(E_i(t+1) - \theta_i)^2 - (E_i(t) - \theta_i)^2}{2} = \frac{r_i^2(t)}{2} + r_i(t)(E_i(t) - \theta_i).$$

Moreover, we have the following inequality:

$$\frac{r_i^2(t)}{2} \leq \frac{\max\{(r_i^{\max})^2, (r_i^{\min})^2\}}{2}.$$

Taking expectations of both sides of (4) given $\mathbf{K}(t)$, and summing over all households i , we can get the following upper bound for the Lyapunov drift for $E_i(t) - \theta_i$:

$$\frac{(E_i(t+1) - \theta_i)^2 - (E_i(t) - \theta_i)^2}{2} \leq \frac{\max\{(r_i^{max})^2, (r_i^{min})^2\}}{2} + r_i(t)(E_i(t) - \theta_i).$$

Also, from (1), squaring both sides, and using the following inequality:

$$(\max\{Q_i(t) - y_i(t), 0\} + d_{i,2}(t))^2 \leq d_{i,2}^2(t) + Q_i^2(t) + y_i^2(t) + 2Q_i(t)(d_{i,2}(t) - y_i(t)),$$

we obtain

$$\frac{Q_i^2(t+1) - Q_i^2(t)}{2} \leq \frac{(y_i^{max})^2 + (d_{i,2}^{max})^2}{2} + Q_i(t)(d_{i,2}(t) - y_i(t)).$$

Similarly, from (26), we have

$$Z_i^2(t+1) \leq (Z_i(t) - y_i(t) + \epsilon_i)^2.$$

Then, we obtain the following inequality:

$$\begin{aligned} & \frac{Z_i^2(t+1) - Z_i^2(t)}{2} \\ & \leq \frac{(\epsilon_i - y_i(t))^2}{2} + Z_i(t)(\epsilon_i - y_i(t)) \\ & \leq \frac{\max\{(y_i^{max})^2, \epsilon_i^2\}}{2} + Z_i(t)(\epsilon_i - y_i(t)). \end{aligned}$$

Combining these three bounds together, taking the expectation w.r.t. $\mathbf{K}(t)$ on both sides, and adding penalty term $V\mathbb{E}\{C_t(\sum_{i=1}^N g_i(t)) + \sum_{i=1}^N F_i(r_i(t)) \mid \mathbf{K}(t)\}$ to both sides of the above inequality, we arrive at the conclusion in the Lemma. ■

We now present the LCMA algorithm. The main design principle of our algorithm is to choose control actions that approximately minimize the R.H.S. of (31).

Lyapunov-based Cost Minimization Algorithm (LCMA):

Initialize (θ_i, ϵ_i) , $\forall i$ and V . At each slot t , observe $(d_{i,1}(t), d_{i,2}(t), s_i(t))$, $\forall i$, C_t , $\mathbf{K}(t)$, and do:

- Choose control decisions \mathbf{y}^* and \mathbf{r}^* as the optimal solution to the following optimization, called **P3**:

$$\begin{aligned} \min : & \sum_{i=1}^N \left\{ (E_i(t) - \theta_i)r_i(t) + VF_i(r_i(t)) \right. \\ & \quad \left. - (Q_i(t) + Z_i(t))y_i(t) \right\} + VC_t\left(\sum_{i=1}^N g_i(t)\right), \\ \text{s.t.} & r_i^{min} \leq r_i(t) \leq r_i^{max}, \quad \forall i, \\ & 0 \leq y_i(t) \leq y_i^{max}, \quad \forall i. \end{aligned}$$

- Update $\mathbf{K}(t)$ according to the dynamics (1), (4), and (26), respectively.

The intuition behind our algorithm is trying to store excess renewable energy for later use, recharge the battery during the period of low electricity price while discharging it during the period of high electricity price, and delay elastic energy

loads to later slots with lower electricity price. Note that we do not need to consider the time-coupling constraints (6) of the battery energy level in the algorithm, since they can be automatically satisfied during our operation of the queues, as proven in Theorem 2 below. Moreover, the algorithm only requires the knowledge of the instantaneous values of system dynamics and does not require any knowledge of the statistics of these stochastic processes. However, the algorithm above should be able to run in a distributed manner in order to be implemented in practice. In the ensuing subsection, we design a distributed algorithm to solve the optimization problem **P3**.

D. Distributed Algorithm to **P3**

First, we introduce the following ancillary variables: $h_i(t)$, $\forall i$ to upper bound individual grid power demand and $D(t)$ to upper bound the total grid power demand. Then, we can transform **P3** into the following formulation, called **P4**:

$$\begin{aligned} \min_{\mathbf{y}, \mathbf{r}, \mathbf{h}} : & \sum_{i=1}^N \left\{ (E_i(t) - \theta_i)r_i(t) + VF_i(r_i(t)) \right. \\ & \quad \left. - (Q_i(t) + Z_i(t))y_i(t) \right\} + VC_t(D(t)), \\ \text{s.t.} & r_i^{min} \leq r_i(t) \leq r_i^{max}, \quad \forall i, \\ & 0 \leq y_i(t) \leq y_i^{max}, \quad \forall i, \\ & h_i(t) \geq d_{i,1}(t) + y_i(t) + r_i(t) - s_i(t), \quad \forall i, \\ & 0 \leq h_i(t) \leq h_i^{max}, \quad \forall i, \\ & \sum_{i=1}^N h_i(t) \leq D(t) \leq D^{max}, \end{aligned}$$

where the maximum grid power consumption h_i^{max} is imposed because of security and reliability considerations for household i . Since $C_t(\cdot)$ is a strictly increasing function, we can easily prove by contradiction that the formulation above and **P3** are equivalent and have exactly the same optimal solutions in terms of \mathbf{r}^* and \mathbf{y}^* . Since **P4** is a convex optimization problem and has decomposability structures, it motivates us to design the following distributed subgradient-based algorithm to iteratively solve it.

In each time slot t , the algorithm implements the steps as indicated in Algorithm 1. When the constant step-size α is small enough, the algorithm above converges to the optimal solution [20]. Note that other types of step-size can also be used with different convergence properties [21]. The algorithm is actually a standard dual decomposition algorithm, which has been extensively investigated in network utility maximization problem in communication networks [22]. The proof of the correctness of this algorithm can be found in standard textbooks [21] and is omitted here for brevity. A desirable feature of our distributed algorithm is that the LSE does not need to know the detailed information about the energy usage in each individual household and only requires the total grid energy usage for all N households. By operating in this manner, our algorithm can help preserve the privacy of homeowners, who are shown to be concerned with some privacy issues associated with the smart grid [23].

Algorithm 1: Distributed Algorithm to **P3**

1 Initialization: Set $\lambda_i^{(0)}(t), \forall i, \nu^{(0)}(t)$ equal to some nonnegative value, $k = 0$

2 foreach Iteration k do

3 while Not satisfying convergence criterion do

4 Each household i updates $r_i^{(k)}(t), y_i^{(k)}(t)$, and $h_i^{(k)}(t)$ after receiving the Lagrangian multiplier $\nu^{(k)}(t)$ according to the solution to the following optimization problem:

$$\begin{aligned} \min : & (E_i(t) - \theta_i)r_i(t) + VF_i(r_i(t)) + \nu^{(k)}(t)h_i(t) \\ & - (Q_i(t) + Z_i(t))y_i(t), \\ \text{s.t.} : & h_i(t) - y_i(t) - r_i(t) \geq d_{i,1}(t) - s_i(t), \\ & 0 \leq h_i(t) \leq h_i^{\max}, \\ & r_i^{\min} \leq r_i(t) \leq r_i^{\max}, \\ & 0 \leq y_i(t) \leq y_i^{\max}. \end{aligned}$$

5 The LSE collects the predication of total utility power demands $\sum_{i=1}^N h_i^k(t)$ from all households i over the FAN. Then, it solves the optimal generation power and update the Lagrange multiplier as follows:

$$D^{(k)}(t) = \underset{0 \leq D(t) \leq D^{\max}}{\operatorname{argmin}} VC_t(D(t)) - \nu^{(k)}(t)D(t), \quad (33)$$

$$\nu^{(k+1)}(t) = [\nu^{(k)}(t) - \alpha(D^{(k)}(t) - \sum_{i=1}^N h_i^{(k)}(t))]^+, \quad (34)$$

where $\alpha > 0$ is a constant step-size, and then, broadcasts $\nu^{(k+1)}(t)$ to all households over the FAN

6 Set $k \leftarrow k + 1$

IV. PERFORMANCE ANALYSIS

In this section, we analyze the performance of LCMA under the case that when the cost function $C_t(\cdot)$, renewable energy generation $s_i(t), \forall i$, energy load arrival processes $d_{i,1}(t), \forall i$ and $d_{i,2}(t), \forall i$ are all i.i.d.. Note that our results can also be extended to the more general setting where $C_t(\cdot), s_i(t), \forall i, d_{i,1}(t), \forall i$, and $d_{i,2}(t), \forall i$ all evolve according to some finite state irreducible and aperiodic Markov chains according to the Lyapunov optimization framework [16]. It is omitted here for brevity.

Theorem 2: If $Q_i(0) = Z_i(0) = 0$ and $\theta_i = V(\alpha^{\max} + \beta_i^{\max}) - r_i^{\min}$ for all households i , then under the LCMA algorithm for any fixed parameters $0 \leq \epsilon_i \leq \mathbb{E}\{d_{i,2}(t)\}$, and $0 < V \leq V^{\max}$, where

$$V^{\max} \doteq \min_i \frac{E_i^{\max} - r_i^{\max} + r_i^{\min}}{\alpha^{\max} + \beta_i^{\max} - \alpha^{\min} - \beta_i^{\min}}, \quad (35)$$

we have the following properties:

- 1) The queues $Q_i(t)$ and $Z_i(t)$ are deterministically upper bounded by Q_i^{\max} and Z_i^{\max} at every slot, where

$$Q_i^{\max} \doteq V\alpha^{\max} + d_{i,2}^{\max}, \quad (36)$$

$$Z_i^{\max} \doteq V\alpha^{\max} + \epsilon_i. \quad (37)$$

Further, $Q_i(t) + Z_i(t)$ are upper bounded by Θ_i^{\max} where

$$\Theta_i^{\max} \doteq V\alpha^{\max} + d_{i,2}^{\max} + \epsilon_i. \quad (38)$$

- 2) The worst-case delay of any buffered elastic energy load is given by:

$$d_i^{\max} = \lceil \frac{2V\alpha^{\max} + d_{i,2}^{\max} + \epsilon_i}{\epsilon_i} \rceil. \quad (39)$$

- 3) The energy queue $E_i(t)$ satisfies the following for all time slots t :

$$0 \leq E_i(t) \leq E_i^{\max}. \quad (40)$$

- 4) All control decisions are feasible.

- 5) If $C_t(\cdot), s_i(t), \forall i, d_{i,1}(t), \forall i$, and $d_{i,2}(t), \forall i$ are i.i.d. over slots, then the time-average expected operating cost under our algorithm is within bound B/V of the optimal value, i.e.,

$$\begin{aligned} \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{C_t(\sum_{i=1}^N g_i(t)) + \sum_{i=1}^N F_i(r_i(t))\} \\ \leq \mathbf{P1}^* + B/V, \end{aligned} \quad (41)$$

where B is the constant specified in (32).

Proof: For each household i and time slot t ,

- 1) First, we prove $Q_i(t) \leq Q_i^{\max}$ for every time slot t . Once again, we will use induction method. Obviously, $Q_i(0) \leq Q_i^{\max}$. Suppose it holds at time slot t , we need to show that it also holds at time slot $t + 1$. As $Q_i(t+1) = \max\{Q_i(t) - y_i(t), 0\} + d_{i,2}(t)$, if $Q_i(t) \leq V\alpha^{\max}$, and the maximum amount of inelastic energy load arrival is $d_{i,2}^{\max}$, we have $Q_i(t+1) \leq V\alpha^{\max} + d_{i,2}^{\max}$. If $V\alpha^{\max} < Q_i(t) \leq V\alpha^{\max} + d_{i,2}^{\max}$, LCMA will choose the maximum possible value for $y_i(t)$ since the partial derivative of the objective function in **P3** w.r.t. $y_i(t)$ is negative. If $Q_i(t) - y_i^*(t) > 0$, then, in time slot t the amount of energy demand being served is at least y_i^{\max} , which is larger than the maximum amount of arrival during time slot t . Hence, the queue cannot increase, i.e., $Q_i(t+1) \leq Q_i(t) \leq V\alpha^{\max} + d_{i,2}^{\max}$. If $Q_i(t) - y_i^*(t) \leq 0$, then $Q_i(t+1) \leq d_{i,2}^{\max} \leq V\alpha^{\max} + d_{i,2}^{\max}$. Therefore, we have proved $Q_i(t) \leq Q_i^{\max}$.
- Next, we prove $Z_i(t) \leq Z_i^{\max}$ for every time slot t . Obviously, $Z_i(0) \leq Z_i^{\max}$. Assuming that it holds for time slot t , we need to show that it also holds in time slot $t+1$. As $Z_i(t+1) = \max\{Z_i(t) - y_i(t) + \epsilon_i 1_{Q_i(t)>0}, 0\}$, if $Z_i(t) \leq V\alpha^{\max}$, then the maximum amount of queuing increase is ϵ_i . We have $Z_i(t+1) \leq V\alpha^{\max} + \epsilon_i$; if $V\alpha^{\max} < Z_i(t) \leq V\alpha^{\max} + \epsilon_i$, then LCMA will choose the maximum possible value for $y_i(t)$ since the partial derivative of the objective function in **P3** w.r.t. $y_i(t)$ is negative. If $Z_i(t) - y_i^*(t) > 0$, then, in time

slot t the amount of energy demand being served is at least y_i^{max} , which is larger than the maximum amount of arrival ϵ_i during time slot t . Hence, the queue cannot increase, i.e., $Z_i(t+1) \leq Z_i(t) \leq V\alpha^{max} + \epsilon_i$. If $Z_i(t) - y_i^*(t) \leq 0$, then $Z_i(t+1) \leq \epsilon_i \leq V\alpha^{max} + \epsilon_i$. Therefore, we have proved that $Z_i(t) \leq Z_i^{max}$.

Finally, we prove $Q_i(t) + Z_i(t) \leq \Theta_i^{max}$ for every time slot t . Obviously, $Q_i(0) + Z_i(0) \leq \Theta_i^{max}$. Suppose $Q_i(t) + Z_i(t) \leq \Theta_i^{max}$ holds for time slot t . If $Q_i(t) + Z_i(t) \leq V\alpha^{max}$, then, according to the dynamics of $Q_i(t)$ and $Z_i(t)$, the maximum increase during one slot is $d_{i,2}^{max} + \epsilon_i$. If $V\alpha^{max} < Q_i(t) + Z_i(t) \leq V\alpha^{max} + d_{i,2}^{max} + \epsilon_i$, then LCMA will choose the maximum possible value for $y_i(t)$ since the partial derivative of the objective function in **P3** w.r.t. $y_i(t)$ is negative. Using the proof above, $Q_i(t+1)$ and $Z_i(t+1)$ cannot increase. Hence, $Q_i(t+1) + Z_i(t+1) \leq Q_i(t) + Z_i(t) \leq \Theta_i^{max}$. This completes the proof.

- 2) This follows directly from Lemma 1.
- 3) Once again, we prove the result by induction. When $t = 0$, $E_i(0) = 0 \leq E_i^{max}$. Now suppose that the bound above holds for time slot t . We need to show that it also holds for time slot $t+1$. First, assuming that $0 \leq E_i(t) < \theta_i - V(\alpha^{max} + \beta_i^{max})$, then LCMA will choose the maximum value for $r_i(t)$ because the partial derivative of the objective function in **P3** w.r.t. $r_i(t)$ is always negative. Therefore, the battery would charge as much as possible, i.e., $0 \leq E_i(t) \leq E_i(t+1) < \theta_i - V(\alpha^{max} + \beta_i^{max}) + r_i^{max} \leq E_i^{max}$. Second, assuming that $\theta_i - V(\alpha^{max} + \beta_i^{max}) \leq E_i(t) \leq \theta_i - V(\alpha^{min} + \beta_i^{min})$, then the maximum charge and discharge rates for the battery are r_i^{max} and $-r_i^{min}$, respectively. Hence, $0 = \theta_i - V(\alpha^{max} + \beta_i^{max}) + r_i^{min} \leq E_i(t+1) \leq \theta_i - V(\alpha^{min} + \beta_i^{min}) + r_i^{max} \leq E_i^{max}$, where we have used the upper bound V_{max} of V . Third, suppose $\theta_i - V(\alpha^{min} + \beta_i^{min}) \leq E_i(t) \leq E_i^{max}$, then LCMA will choose the minimum value for $r_i(t)$ because the partial derivative of the objective function in **P3** w.r.t. $r_i(t)$ is always positive. Therefore, the battery would discharge as much as possible, i.e., $0 \leq \theta_i - V(\alpha^{min} + \beta_i^{min}) + r_i^{min} \leq E_i(t+1) \leq E_i(t) \leq E_i^{max}$. This completes the proof.
- 4) Since we choose our decisions to satisfy all constraints in **P3**, combining it with the results above together, all constraints of **P1** are satisfied. Therefore, our control decisions are feasible to **P1**.
- 5) As we have mentioned before, the LCMA is always trying to greedily minimize the R.H.S. of the upper bound (31) of the *drift-plus-penalty* term at every slot t over all possible feasible control policies including the optimal and stationary policy given in Theorem 1. Therefore, by plugging this policy into the R.H.S. of the inequality (31), we obtain the following:

$$\begin{aligned} \Delta_V(t) &\leq \\ B + \sum_{i=1}^N \mathbb{E}\{(E_i(t) - \theta_i)r_i^{stat}(t) \mid \mathbf{K}(t)\} & \end{aligned}$$

$$\begin{aligned} & + \sum_{i=1}^N \mathbb{E}\{Q_i(t)(d_{i,2}(t) - y_i^{stat}(t)) \mid \mathbf{K}(t)\} \\ & + \sum_{i=1}^N \mathbb{E}\{Z_i(t)(\epsilon_i - y_i^{stat}(t)) \mid \mathbf{K}(t)\} \\ & + V\mathbb{E}\{C_t(\sum_{i=1}^N g_i^{stat}(t)) + \sum_{i=1}^N F_i(r_i^{stat}(t)) \mid \mathbf{K}(t)\} \\ & \leq B + V \cdot \mathbf{P2}^* \leq B + V \cdot \mathbf{P1}^*, \end{aligned}$$

where the following facts haven been used:

$$\begin{aligned} \mathbb{E}\{r_i^{stat}(t) \mid \mathbf{K}(t)\} &= 0, \\ \mathbb{E}\{d_{i,2}(t) - y_i^{stat}(t) \mid \mathbf{K}(t)\} &\leq 0 \\ \mathbb{E}\{C_t(\sum_{i=1}^N g_i^{stat}(t)) + \sum_{i=1}^N F_i(r_i^{stat}(t))\} &= \mathbf{P2}^*, \\ \mathbb{E}\{\epsilon_i - y_i^{stat}(t) \mid \mathbf{K}(t)\} &\leq 0. \end{aligned}$$

The first three facts follow from Lemma 1 and the last one follow from the second fact above with together $\epsilon_i \leq \mathbb{E}\{d_{i,2}(t)\}$. Taking the expectation of both sides, using the law of iterative expectation, and summing over $t \in \{0, 1, 2, \dots, T-1\}$, we have

$$\begin{aligned} V\mathbb{E}\{C_t(\sum_{i=1}^N g_i(t)) + \sum_{i=1}^N F_i(r_i(t))\} \\ \leq BT + VT \cdot \mathbf{P1}^* - \mathbb{E}\{L(T)\} + \mathbb{E}\{L(0)\}. \end{aligned}$$

Dividing both sides by T , letting $T \rightarrow \infty$, and using the facts that $\mathbb{E}\{L(0)\}$ are finite and $\mathbb{E}\{L(t)\}$ are nonnegative, we arrive at the following performance guarantee:

$$\begin{aligned} \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^N \mathbb{E}\{C_t(\sum_{i=1}^N g_i(t)) + \sum_{i=1}^N F_i(r_i(t))\} \\ \leq \mathbf{P1}^* + B/V, \end{aligned}$$

where **P1*** is the optimal objective value, B is a constant, and V is a control parameter which has the maximum value given by (35). ■

V. NUMERICAL EXPERIMENTS

In this section, we provide numerical results based on real-world data sets to complement the analysis in the previous sections.

A. Experiment Setup

We consider a simple power system consisting of eight households in one neighborhood that share the same load serving entity and have on-site renewable generation, energy storage, and elastic and inelastic energy loads. The households are divided into two categories. For the first type of households (indexed by $i = 1, 2, 3, 4$), both the elastic and inelastic energy load arrivals during one slot are i.i.d. and take value from $[1, 5]$ kWh uniformly at random. For the second type of households (indexed by $i = 5, 6, 7, 8$), both the elastic and inelastic energy load arrivals during one slot are also i.i.d. and take value from $[1.5, 7.5]$ kWh uniformly at random. For the renewable generation, we use the hourly average solar irradiance data for Los Angels area from the Measurement and Instrumentation Data center (MIDC) [24] at National Renewable Energy Laboratory. The period we consider in this paper is half year from January 1, 2011 to June 30, 2011. In total, this duration includes 181 days or 4344 1-hour slots. The control interval is chosen to be 1-hour. For different households, we use different scaling factors to characterize the heterogeneity of households. Specifically, we choose the scaling factors such that the average solar energy production during one slot is about 3 kWh for the first type of households and 4.5 kWh for the second type of households. We fix the maximum charge and discharge rates of batteries in households as follows: for $i \in \{1, 2, 3, 4\}$, $r_i^{max} = 1$ kWh, $r_i^{min} = -1$ kWh, and for $i \in \{5, 6, 7, 8\}$, $r_i^{max} = 1.5$ kWh, $r_i^{min} = -1.5$ kWh. Also, we choose $y_i^{max} = d_{i,2}^{max}$ for all i . The battery cost is assumed to be a simple quadratic function as follows:

$$F_i(r_i) = b_1 r_i^2,$$

where b_1 is a constant coefficient. For the purpose of simple illustration, we choose the same battery cost function for all households i in the evaluations.

For the LSE, we assume that the energy cost function is a smooth quadratic function as follows:

$$C_t(D) = c_1(t)D^2 + c_2D + c_3.$$

where $c_1(t)$ is a time-varying coefficient used to model different electricity marginal costs across time slots while c_2 and c_3 are constant coefficients. In this evaluation, $c_1(t)$ takes value from $[0.1, 0.2]$ kWh uniformly at random, $c_2 = 0.1$, and $c_3 = 0.2$.

B. Results and Analysis

In order to analyze the performance improvement due to our LCMA, we compare it with the following two approaches.

- No storage, no demand response (B1): In this approach, households have no energy storage and do not differentiate between elastic and inelastic energy loads. The household tries to use the renewable energy as much as possible. When the renewable energy is not sufficient, the household draws energy from the utility grid. Unused renewable energy is wasted due to the lack of energy storage.

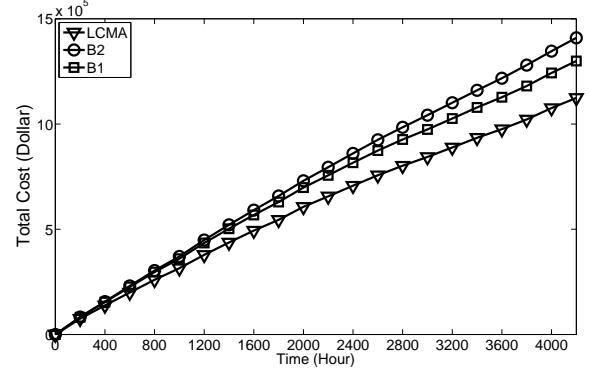


Fig. 2. Comparison of the total energy cost in three approaches

- Storage, no demand response (B2): In this approach, households have energy storage but do not consider demand response. The household uses renewable energy only as a supplement to the grid by consuming it whenever it is available. The household stores any extra renewable energy in its battery, but never charge the battery from the grid. The stored energy would be used to serve the future demands.

Note that LCMA differs from the approaches above in the sense that LCMA would actively charge the battery when the grid power is cheap while discharging it when the grid power is expensive. Moreover, LCMA differentiates between inelastic and elastic energy loads and delays the elastic energy loads to later time when the grid power cost is low.

In the first evaluation, we compare our algorithm with the two approaches above using the real-world solar power generation. Note that the performance of LCMA depends on the battery capacity, the battery cost, and the control parameters V and ϵ_i . We choose $b_1 = 0.5$, $E_i^{max} = 20$ kWh, $i \in \{1, 2, 3, 4\}$, and $E_i^{max} = 30$ kWh, $i \in \{5, 6, 7, 8\}$. The initial battery energy level at each household is chosen to be zero. Let $V = V^{max}$ and $\epsilon_i = \mathbb{E}\{d_{i,2}\}, \forall i$. As can be seen in Figure 2, our proposed LCMA can reduce the total energy cost by approximately 20% compared with B1 and 13% compared with B2 in the six-month period. Also, the slopes of the lines are different, meaning that the savings are unbounded as the time increases. Meanwhile, the LCMA has on average a much smaller delay than the worst-case guarantee (27), as shown in Fig. 3.

In the following, we consider the impact of varying control parameters on the performance of LCMA.

- **Impact of Battery Capacity:** In this evaluation, we vary the battery capacities of households with other parameters fixed. We set $E_i^{max} = \{20, 30, 40\}$ kWh for $i \in \{1, 2, 3, 4\}$, $E_i^{max} = \{30, 40, 50\}$ kWh for $i \in \{5, 6, 7, 8\}$, and $V = V^{max}$. The result is illustrated in Figure 4. From the figure, it is clear that the larger the capacity is, the more cost saving LCMA can obtain, which coincides with the algorithmic performance results of our algorithm in Theorem 2. As we have mentioned before, the saving comes from the fact that our algorithm charges the battery when the marginal energy cost is low,

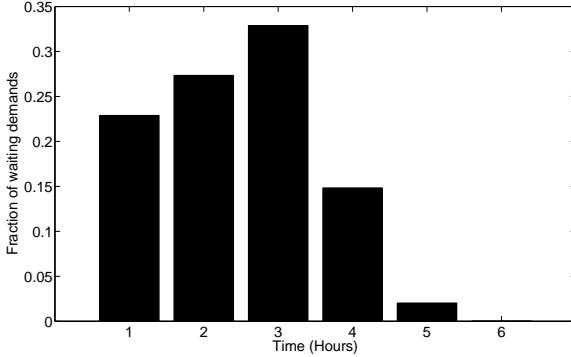


Fig. 3. Histogram of delay for the elastic demands in the service queue in household 1

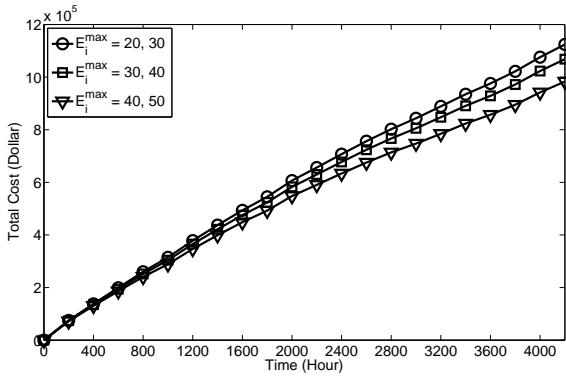


Fig. 4. The impact of battery capacity E_i^{max} on the cost saving

while discharging it when the marginal energy cost is high.

- **Impact of Battery Cost:** Currently, the battery is still expensive. The charging or discharging operation would reduce the lifetime of the battery. However, it is expected that the cost of battery would decrease greatly in the next decade. In this evaluation, we estimate the impact of battery cost on the cost saving of our algorithm. We set $b_1 = \{0, 1, 2, 5, 20, 200\}$ and keep $E_i^{max} = 20, i \in \{1, 2, 3, 4\}$, and $E_i^{max} = 30, i \in \{5, 6, 7, 8\}$ fixed. The result is shown in Figure 5. Note that when the battery cost per usage during one-slot b_1 is very large (e.g., 200 \$), our algorithm would not charge or discharge the battery at all, so it is the same as the approach B1. As the battery cost increases, the total cost saving of LCMA compared with B1 would decrease until they are the same since the opportunity to utilize the temporal variation of electricity prices is smaller.

- **Impact of Worst-case Delay Requirement:** In this setting, we adjust the parameters ϵ_i while fixing other parameters to see the impact of the worst-case delay guarantee for elastic energy loads on the performance of LCMA. We choose $\epsilon_i = \{0, 1, 2, 3\}, \forall i$, respectively. As observed in Figure 6, the increase of ϵ_i (i.e., the worst-case delay) gives more opportunity to optimize the energy cost, since the elastic energy loads are more likely to be

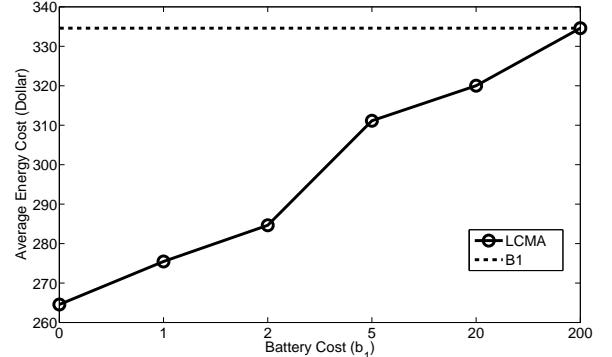


Fig. 5. The impact of battery cost b_1 on the cost saving

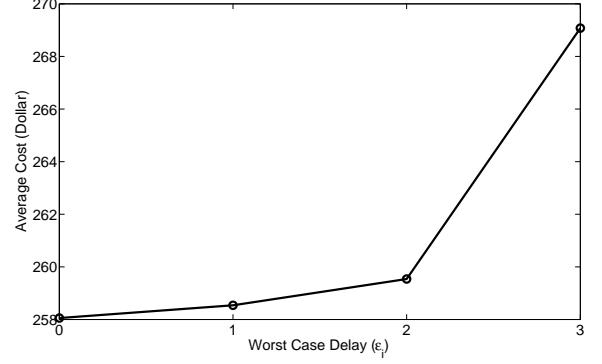


Fig. 6. The impact of ϵ_i on the cost saving

served in the low energy cost period.

VI. CONCLUSIONS

In this paper, we present an algorithm (LCMA) for the problem for distributed coordinated stochastic optimization of flexible energy resources in a smart grid setting. The total system cost can be reduced if more energy loads are elastic and can tolerate being served with some delay. Our algorithm is simple and was shown to be able to operate without knowing the statistical properties of the underlying dynamics in the system. With the increase of energy storage capacities, the performance of our algorithm is proved to be arbitrarily close to the optimal value. Moreover, our algorithm provides an explicit relationship between energy storage capacity, worst-case delay, and cost saving. Extensive numerical evaluations based on real-world data sets show the effectiveness of our approach.

APPENDIX A THE WORST-CASE DELAY FOR BUFFERED ELASTIC LOADS

Here we prove Lemma 1.

Proof: For all households i , consider any slot t for which $d_{i,2}(t) > 0$. We will show that this energy load $d_{i,2}(t)$ is served on or before time slot $t + \delta_i^{max}$ by contradiction. Suppose not, then during slots $\tau \in \{t+1, \dots, t + \delta_i^{max}\}$, it must be that $Q_i(\tau) > 0$. Otherwise, the energy load $d_{i,2}(t)$ would have been served before τ . Therefore, $1_{Q(\tau)>0} = 1$,

and from the update equation (26) of $Z_i(t)$, we have for all $\tau = \{t+1, \dots, t+\delta_{max}\}$:

$$Z_i(\tau+1) \geq Z_i(\tau) - y_i(\tau) + \epsilon_i.$$

Summing the above over $\tau = \{t+1, \dots, t+\delta_i^{max}\}$ yields:

$$Z_i(t+\delta_i^{max}+1) - Z_i(t+1) \geq - \sum_{\tau=t+1}^{t+\delta_i^{max}} y_i(\tau) + \delta_i^{max} \epsilon_i.$$

Rearranging the terms and using the facts that $Z_i(t+1) \geq 0$ and $Z_i(t+\delta_i^{max}+1) \leq Z_i^{max}$ yields:

$$\sum_{\tau=t+1}^{t+\delta_i^{max}} y_i(\tau) \geq \delta_i^{max} \epsilon_i - Z_i^{max}. \quad (42)$$

Since the energy loads $d_{i,2}(t)$ are queued in a FIFO manner and $Q_i(t+1) \leq Q_i^{max}$, it would be served on or before time $t+\delta_i^{max}$ whenever there are at least Q_i^{max} units of energy served during $\tau \in \{t+1, \dots, t+\delta_i^{max}\}$. Since we have assumed that the energy loads $d_{i,2}(t)$ are not served by time $t+\delta_i^{max}$, it must be that $\sum_{\tau=t+1}^{t+\delta_i^{max}} y_i(\tau) < Q_i^{max}$. Comparing this inequality with (42) yields:

$$Q_i^{max} > \delta_i^{max} \epsilon_i - Z_i^{max},$$

which implies that $\delta_i^{max} < (Q_i^{max} + Z_i^{max})/\epsilon_i$, contradicting the definition of δ_i^{max} in (27). ■

REFERENCES

- [1] A. Ipakchi and F. Albuyeh, "Grid of the future," *Power and Energy Magazine, IEEE*, vol. 7, no. 2, pp. 52–62, 2009.
- [2] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 3, pp. 381–388, Aug 2011.
- [3] M. Albadri and E. El-Saadany, "A summary of demand response in electricity markets," *Electric Power Systems Research*, vol. 78, no. 11, pp. 1989–1996, 2008.
- [4] H. Farhangi, "The path of the smart grid," *Power and Energy Magazine, IEEE*, vol. 8, no. 1, pp. 18–28, 2010.
- [5] United States Energy Information Administration. [Online]. Available: <http://www.eia.gov/>
- [6] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *Smart Grid, IEEE Transactions on*, vol. 1, no. 2, pp. 120–133, Sep 2010.
- [7] A. Mohsenian-Rad, V. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *Smart Grid, IEEE Transactions on*, vol. 1, no. 3, pp. 320–331, Dec 2010.
- [8] T. T. Kim and H. V. Poor, "Scheduling power consumption with price uncertainty," *Smart Grid, IEEE Transactions on*, vol. 2, no. 3, pp. 519–527, Sep 2011.
- [9] M. Neely, A. Tehrani, and A. Dimakis, "Efficient algorithms for renewable energy allocation to delay tolerant consumers," in *Smart Grid Communications (SmartGridComm'10), First IEEE International Conference on*, Washington, DC, Oct 2010, pp. 549–554.
- [10] A. Papavasiliou and S. Oren, "Supplying renewable energy to deferrable loads: Algorithms and economic analysis," in *Power and Energy Society General Meeting, 2010 IEEE*, Minneapolis, MN, Jul 2010.
- [11] E. Y. Bitar, R. Rajagopal, P. P. Khargonekar, K. Poolla, and P. Varaiya, "Bringing wind energy to market," *Power Systems, IEEE Transactions on*, 2012.
- [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*, Detroit, MI, Jul 2011, pp. 1–8.
- [13] L. Jiang and S. H. Low, "Multi-period optimal energy procurement and demand response in smart grid with uncertain supply," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, Orlando, FL, Dec 2011.
- [14] Y. Guo, M. Pan, and Y. Fang, "Optimal power management of residential customers in the smart grid," *Parallel and Distributed Systems, IEEE Transactions on*, 2012.
- [15] S. Barker, A. Mishra, D. Irwin, P. Shenoy, and J. Albrecht, "Smartcap: Flattening peak electricity demand in smart homes," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom'12)*, Lugano, Switzerland, Mar 2012.
- [16] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. San Rafael, CA: Morgan & Claypool Publishers, 2010.
- [17] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross-layer Control in Wireless Networks*. Now Publishers, 2006.
- [18] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramanian, "Optimal power cost management using stored energy in data centers," in *ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'11)*, San Jose, CA, Jun 2011, pp. 221–232.
- [19] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2000.
- [20] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*. Old Tappan, NJ: Prentice-Hall, 1989.
- [21] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge University Press, 2004.
- [22] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 8, pp. 1439–1451, Aug 2006.
- [23] Privacy on the Smart Grid, IEEE Spectrum. [Online]. Available: <http://spectrum.ieee.org/energy/the-smarter-grid/privacy-on-the-smart-grid>
- [24] NREL: Measurement and Instrumentation Data Center. [Online]. Available: <http://www.nrel.gov/midc/>

Micro grid scheduling policies, forecasting errors, and cooperation based on production correlation

Georgios Tsaousoglou

Department of Electrical and
Computer Engineering, National
Technical University of Athens,
Greece

Prodromos Makris

Computer Technology Institute and
Press (CTI), University Campus of
Patras, Rion,
Greece

Emmanouel Varvarigos

Department of Electrical and
Computer Engineering, National
Technical University of Athens,
Greece

Abstract—We present scheduling policies that can be used by Micro Grids (MGs) with the possibility of energy production, consumption and storage, in the context of a novel Smart Grid architecture. Assuming a Day-Ahead market, we investigate the management of resources so as to achieve higher profit for the MGs under various operating scenarios. The possibility of cooperation among MGs in the presence of forecast errors is also studied, demonstrating that it can lead to more intelligent and profitable operation of the MG resources. The profits of cooperation among MGs whose production patterns have positive, zero, or negative correlation are assessed.

Keywords—Micro Grids, scheduling policy, Micro Grid cooperation, forecast deviation, resources virtual integration

I. INTRODUCTION

Renewable energy sources (RES) play an increasingly important role in the energy market, with their integration in the future Smart Grid being the subject of much recent research [1]. Medium and small energy prosumers (producers and consumers at the same time) are important participants of a liberalized energy market. The formation of Micro Grid (MG) coalitions in order to enhance their market role [4],[5] also shows promising results. As the energy pricing model will no longer be based on a flat price, the times at which an MG buys or sells energy is important in optimizing its profit. In this context, energy storage management [7] is attracting new interest, given the important storage possibilities [6] offered by plug-in electric vehicles.

Regulation in many countries dictates that a certain percentage of the energy comes from RES, thus creating a demand for RES energy. The market for RES energy will probably be based on business models that are different than the feed-in tariff policies currently used in many countries. The major directions promoted by the EU are the increase of RES penetration and the liberalization of the energy market (directive 200/72/EC [12]). In a fully liberalized unsubsidized market, RES producers may find it hard to compete unless new business models are devised that will take into account the positive externalities of RES energy (namely green production and avoidance of CO₂ emission penalties, and production sites close to consumers with corresponding reductions in the investments needed for the distribution network), and try to return these benefits to the small

producers that create them. Since RES production is distributed, a decentralized market operation model is needed to help MGs have a more active role and participation in the energy market. Such a model creates new challenges for the Information and Communication Technologies (ICT) field [8]. As ICT is introduced in the energy grid, the virtualization of energy resources also becomes feasible, where a big energy prosumer is no longer necessarily formed by investing large capital on big prosumption facilities. Multiple small prosumers (MGs) can organize themselves in associations that participate as a single entity in the market, thus forming big energy prosumers, which we call Virtual Micro Grids (VMG). The traditional power production plant is expanded to the concept of the Virtual Power Plant, an entity that underpins the growing number of small production units [9], while the VMG will probably constitute a key actor that brings together RES producers in the new market models. Figure 1 displays the new approach, where MGs can pool energy resources together into VMGs that can be direct participants in a decentralized energy market. The VMG concept is a central idea in the ongoing Virtual Micro Grids for Smart Energy Networks (VIMSEN) project [13].

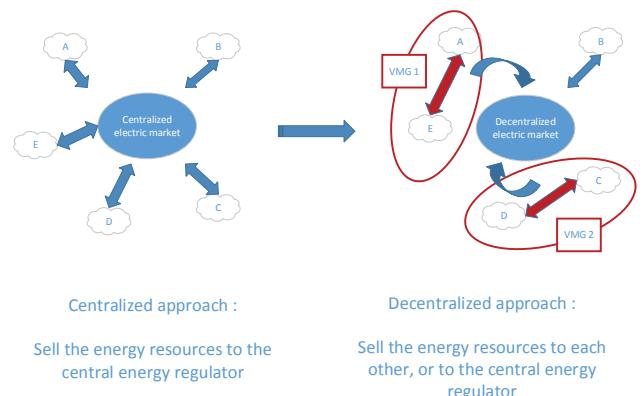


Fig. 1. VMGs and Decentralized Energy Market

The present paper focuses on the scheduling policies that can be used by MGs participating in the energy market when making their sell/buy/store decisions, as well as the advantages that can be obtained through the cooperation of multiple MGs in bigger coalitions (the VMGs). We first look

into the case of a single MG that is able to produce, consume and store energy, proposing and evaluating nine (9) scheduling policies that can be used to make its market (buy/sell/store) decisions. A Day-Ahead market is assumed and the costs/profits of an MG for the different resource management algorithms are evaluated, first by assuming that the actual daily prosumption pattern of the MG is the same as the forecasted one, and then by introducing a deviation vector to model inaccuracies in the forecasts. In the presence of forecast errors, the MG fails to meet its Service Level Agreement (SLA) and pays a penalty for this. The possibility of cooperation among Micro Grids is then studied, with the results demonstrating a more intelligent and profitable operation of the MG resources when they are pooled together. The main idea is that MG coalitions (VMGs), through mutual energy exchanges can help each other reduce the forecast error effects and the corresponding SLA deviations. The profits of cooperation are also studied in the case of positive, null and negative correlation among the production patterns of the cooperating MGs, giving insight on the criteria that should be used to cluster MGs into VMGs. We show that MGs whose production patterns are negatively correlated can gain important benefits from their cooperation, but the cooperation benefits also extend, even though reduced, to independent or even positively correlated MGs.

A) Contribution Points

The ways and the policies according to which MGs operate and make their decisions in order to benefit from this newly introduced framework are the subject of this paper. The main contribution points of the paper can be summarized as follows:

- A novel Smart Grid architecture is introduced, composing the framework in which the system model is developed.
- A variety of scheduling algorithms are proposed regarding the management of an MG's own resources in this particular model framework, in a single MG (non-interconnected with other MGs) scenario.
- A deviation vector is introduced, and the effect of forecast inaccuracies in the algorithms' performance is studied.
- A cooperative scenario is proposed, where the MGs are shown to gain profits by forming virtual MG cooperations (VMGs) and jointly making their buy/sell/store decisions.
- A correlation factor is introduced on the production profiles of the MGs forming a VMG, and the effect of the correlation among the cooperating MGs on the total profits is studied.

II. MODEL USED FOR THE PRICING, FORECAST AND STORAGE CAPABILITIES

A brief description of the system's architecture, being developed in the VIMSEN project, is depicted in Figure 2. The main actors in the system's architecture are:

- The Energy Market Operator, which may be a department of the Big Energy Producer (BEP) or an independent public or private organization.
- The VMG Aggregator (VMGA), which would be an association, a company, or simply a software collecting data from the MGs, making and broadcasting decisions to the MGs via the VMGA Portal, as well as operating the communication between the MGs and the MO.
- The Micro-Grids, which are small/medium size entities with possibilities of producing, consuming and storing energy. Examples are a factory, a photovoltaic park or a residence.

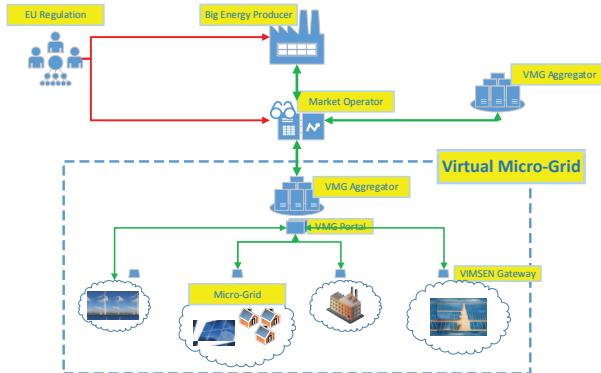


Fig. 2. The VIMSEN Architecture

The MGs are Smart-Grid integrated, equipped with smart meters and a VIMSEN Gateway to communicate their data to the VMGA, receive production day-ahead forecasts from a weather station, and generate predictions for their day-ahead consumption based on training algorithms and historical data[10].

The BEP creates a day-ahead, time-of-use pricing pattern, based on statistical data [11]. Currently, the BEP generally buys energy from small producers at a flat or close to flat price, set by feed-in tariff policies. In order to decrease its operational costs, the BEP decides to buy this energy at a peak-demand time so that the operation of supplementary low-efficiency production plants is avoided. Thus, a “special-price event” is decided where the BEP asks to buy energy at a certain time of the day-ahead schedule, offering a higher price for that particular time. The whole pricing pattern for buying and selling energy is broadcast through the Market Operator to the VMG Aggregators.

We take the perspective of a MG and focus on the MG and VMG level. Two scenarios are studied. The non-interconnected scenario where MGs exchange energy only with the BEP, while the interconnected scenario assumes a set of MGs (forming a VMG) that can exchange energy with each other. This will turn out to be useful for the MGs in meeting their SLA with the BEP in the presence of forecast errors on their prosumption profile.

III. PROBLEM FORMULATION

The *prosumption profile* of an MG is characterized by a triplet

$$MG = (X, Y, C),$$

where

$$X = (X^1, X^2, \dots, X^{24})$$

denotes the forecasted MG production pattern throughout a day at one hour intervals, with X^i being expressed in kWh,

$$Y = (Y^1, Y^2, \dots, Y^{24}),$$

denotes the estimated MG energy consumption pattern, and

C in kWh is the maximum energy storage capacity of the MG.

The day-ahead pricing pattern for buying/selling energy broadcasted by the MO to the VMG Aggregators is denoted as

$$P = (B, S)$$

where

$$B = (B^1, B^2, \dots, B^{24})$$

denotes the prices (€ per kWh) at which the MG can buy energy from the BEP at each time interval and

$$S = (S^1, S^2, \dots, S^{24})$$

the prices at which the MG can sell energy to the BEP. Each MG applies a scheduling policy (to be discussed in Section IV) that results in its day-ahead Market Decisions captured in the pair

$$MD = (E_b, E_s)$$

denoting the amount (kWh) an MG will buy/sell at each time:

$$E_b = (E_b^1, E_b^2, \dots, E_b^{24})$$

$$E_s = (E_s^1, E_s^2, \dots, E_s^{24})$$

The MG makes an SLA with the BEP to buy and sell these quantities at the specified times and prices. In the case of forecast errors, the MG may not be able to fulfill its SLA, at a certain hour i in which case it will have to pay a penalty. In our performance results the penalty is that if the production is less than the forecasted one by some D_i , the MG will have to buy this amount at price B^i and sell it at price S^i for a cost of $D^i(B^i - S^i)$. Other ways in which the penalty for breaking the SLA (e.g., buying at the spot price) can also be included.

When there are no forecast errors, the daily energy cost of the MG is captured by the *COST* parameter, defined as

$$COST = \sum_{i=1}^{24} [B^i \cdot E_b^i] - \sum_{i=1}^{24} [S^i \cdot E_s^i]$$

IV. SCHEDULING POLICIES AND STORAGE MANAGEMENT

In this section we look into scheduling and storage management policies than can be used by an MG (section IV.1) or an association of interconnected MGs (section IV.2).

IV.1 Case of single (non-interconnected) MGs

We first look into the non-interconnected case, focusing on a single MG. We propose nine algorithms that can be used by the MG to make its buy/sell/store decisions trying to satisfy its consumption with the least cost and sell energy to the BEP at the best price, so as to minimize its *COST*. The scheduling and storage management algorithms examined are the following:

1) Uncapacited: the algorithm does not make use of the storage capacity at all. When energy is needed the MG buys, and when energy is produced, it sells.

2) Selfie: the MG tries to be self-sufficient. When energy is produced, the MG stores it. When energy is needed, it consumes from the Storage and if this is empty it buys from the market as much as needed. The MG sells only when its storage capacity is full and can no longer store the energy it produces.

3) Sniper: The MG buys as much energy as can be stored before the "peak demand zone" and stores and sells it all at the special-price event where the sales price is high.

4) Stocky: The MG buys as much energy as it can store, before the "peak demand zone" and uses it to cover its consumption needs in the "peak demand zone" to avoid buying at a high price. The MG sells only when it has more energy than can be stored.

5) Stock 'n' Stock: a variation of Stocky, where the MG buys energy to fill up storage not only before the peak demand zone but also in the low demand zone when the price is low.

6) Conservative: The MG buys energy at the same points with the Stock 'n' Stock but it does not fill up the storage, always leaving a margin of 4 kWhs. Also in "special-price event" it sells only half of the storage.

7) Conservative 70%: The MG buys leaving a 30% margin and sells in the "special-price event" 70% of storage.

8) SmartStock: The MG buys energy at the same points (before the price goes higher) but only as much energy as it expects to be needed by the MG or that it expects to be able to sell. The MG sells in the special-price event only as much energy as it expects not to be needed by the MG in the peak demand zone.

9) SmartStockConservative: It works like SmartStock but is more conservative. It buys and sells half than SmartStock to leave a margin in case predictions are inaccurate.

IV.1.1 Effect of deviations between forecasted and realized production for a single MG

The preceding section considers a Day-Ahead market where an MG trades energy based on the forecast of the previous day. To model the prediction errors, we define a *deviation vector* D , as the difference between the vector \tilde{X} of actual hourly production values and the vector X of forecasted values,

$$D = (D^1, D^2, \dots, D^{24}) = X - \tilde{X}.$$

The cost is no longer that of *P*, as the MG is forced to buy/sell additional energy at a non-beneficial price compared to the scheduled policy to compensate for the inaccurate production forecast. Thus, the arrays E_b and E_s that define the scheduled buy and sell decisions (SLA) have to be modified to \tilde{E}_b and \tilde{E}_s , respectively, and the *COST* is modified accordingly:

$$\widetilde{COST} = \sum_{i=1}^{24} [B^i \cdot \tilde{E}_b^i] - \sum_{i=1}^{24} [S^i \cdot \tilde{E}_s^i].$$

The performance in the presence of forecasting errors is expected to be worse than when forecasts are accurate, $\widetilde{COST} > COST$.

IV.2 Case of MGs Interconnected into VMGs: Micro Grid cooperation and Virtual Storage Bank

We now turn our attention to the case where we have n MGs forming a Virtual Micro Grid (VMG). An MG is now able to use the available resources of other MGs in the same VMG so that the additional energy transfers due to non-conformance with the SLAs between the set of the cooperating MGs and the BEP (and thus the corresponding penalties) are minimized. The idea is to create a *virtual storage bank*, consisting of the sum of storage capacities of individual MGs, so that MGs are able to fulfill each other's needs in case of inadequacy of the individual resources. The prosumption profile of a VMG is defined by the component wise addition of the prosumption profiles of its constituent MGs

$$P_{VMG} = \sum_n (X^n, Y^n, C^n)$$

In this section, all MGs are assumed to operate according to the SmartStock algorithm (which is the algorithm of choice, based on the performance results in Section V). As the scheduling is based on forecasted values, the deviation vector of Section IV.1.1 results in the need for the MG to have to buy or sell additional energy to fulfill its Day-Ahead SLA agreement. This results in scheduling (SLA) deviations and the vector of additional energy quantities that have to be bought or sold by the MG is given by:

$$\begin{aligned} e^b &= \widetilde{E}_b^i - E_b^i \\ e^s &= \widetilde{E}_s^i - E_s^i \end{aligned}$$

respectively. When an MG operates as a single entity, energy equal to these deviations needs to be bought/sold by the MG at an additional cost of $B^i \cdot S^i$ per unit of power to fulfill its SLA.

When the MGs form VMGs and cooperate in trading energy, the algorithms running in the VMGA Gateway tries for the scheduling deviations in an MG to be compensated by another MG so that the high-cost energy exchange (SLA penalty) with the BEP is avoided. Thus, the objective of the VMG is to ensure that the overall energy exchanged (sold / bought) between the VMG and the BEP is almost equal to the quantities scheduled at each hour, so that the individual SLAs are met, i.e., that the total energy difference resulting from the scheduling deviations for a VMG should be close to zero at each hour (componentwise):

$$\begin{aligned} \sum_{k=1}^n [\widetilde{E}_b^{i,k}] - \sum_{k=1}^n [E_b^{i,k}] &\rightarrow 0 \\ \sum_{k=1}^n [\widetilde{E}_s^{i,k}] - \sum_{k=1}^n [E_s^{i,k}] &\rightarrow 0 \end{aligned}$$

IV.2.1. Grouping of the Micro Grids into Virtual Micro Grids according to the correlation of their production patterns

An important issue affecting the algorithms' performance and the MGs' ability to meet their SLAs through collaboration is related to the way MGs are grouped together into VMGs.

Useful in this context is the concept of "MG production correlation".

An MG A will be said to be positively correlated to a MG B when their production patterns are affected by weather and other conditions in the same way, or more mathematically, if their deviation vectors have strictly positive crosscorrelation,

$$E(D_A * D_B) > 0,$$

where $*$ denotes inner product and $E()$ expected value. An example of positively correlated MGs would be a set of solar parks located in nearby geographical areas, where a loss of sunshine would affect MG production patterns in similar ways. MG A will be said to be negatively correlated to MG B when

$$E(D_A * D_B) < 0$$

MG A will be said to be uncorrelated to MG B , when their production sources are independently affected,

$$E(D_A * D_B) = E(D_A) * E(D_B) = 0,$$

where we assume unbiased estimators, $E(D_A) = E(D_B) = 0$. In the performance results of Section V we examine the cases where a VMG consists of a) maximally positively correlated MGs, b) uncorrelated MGs and c) pairs of negatively correlated MGs.

V. PERFORMANCE EVALUATION RESULTS

In this section we evaluate the algorithms' performance under a variety of conditions and scenarios. The performance results for the single MG (non-interconnected) case are presented in section V.1 along with the effects of the deviation vector defined in V.1.1. The profits that can be obtained through the cooperation among MGs that form VMGs along with the role played by the correlation factor are investigated in section V.2.

V.1 Case of Single (non-interconnected) MG

The resource management algorithms of section IV.1 were implemented in MATLAB, using production, demand and storage capacity data derived from random uniform distributions within a range of realistic energy patterns.

More precisely, each MG profile (X, Y, C) was implemented by drawing from uniform distributions as follows:

- the forecast X^i for the energy produced by the MG during hour i , was uniformly distributed in the range 7-30 kWhs, independently for different i 's;
- the forecast Y^i for the energy consumed by the MG during hour i , was uniformly distributed in the range 7-30 kWhs, independently for different i 's;
- the Storage Capacity C of an MG, was chosen as a constant in the range of 0-30 kWhs, and was different for different MGs.

Regarding the prices used in our performance evaluation results we used datasets containing realistic data derived from the Greek National Energy Provider Company (PPC) pricing patterns, presented in Figure 3.

In each experiment, an algorithm takes as input the profile $MG=(X, Y, C)$ of the Micro Grid and the data for the day-ahead pricing $P=(B, S)$. The results we describe were averaged over 10,000 experiments performed. The brief average scoreboard in cost (euros per day) is presented in Table 1 for all the algorithms examined. SmartStock algorithm achieves the best

results in every scenario, significantly outperforming the Uncapacited, which interestingly is the policy applied in the majority of RES plants of today.

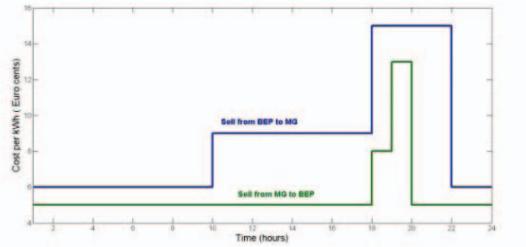


Fig. 3. Pricing patterns

TABLE I. AVERAGE COST PER 24H FOR ALL ALGORITHMS EXAMINED FOR THE CASE OF A SINGLE MG, WHEN THERE ARE NO DEVIATIONS BETWEEN THE FORECASTED AND THE ACTUAL PRESUMPTION PROFILE

Algorithms	Average Cost (€ per day)
Uncapacited	13.160
Selfie	1.445
Sniper	0.769
Stocky	1.307
DoubleStock	1.238
Conservative	1.026
ConservativePerCent	0.997
SmartStock	0.690
SmartConservative	1.019

V.1.1 Effect of deviations in production profile

We then evaluated the performance of the algorithms in the case the forecasted production profile differs from the production profile actually realized by the deviation vector D , considered to follow a random uniform distribution within a range of $\pm 25\%$. The results are presented in Table 2. The performance results obtained in this case are worse, as expected, except for the Uncapacited algorithm which could not be affected, as it was not making use of any actual strategy in the first place. The SmartStock algorithm still achieves the best performance.

V.2 Case of Multiple MGs Interconnected into VMGs

In this section we present the performance results obtained for the case where multiple MGs cooperate into forming a VMG. A VMG consisting of n cooperating MGs with different (randomly distributed) storage capacities, production and demand patterns was implemented within realistic pattern margins. The MGs compensate their energy needs either on their own, by exchanging energy with the BEP (as in the previous section), or by cooperating and applying the concept of the virtual storage bank (as a VMG). In each case the daily cost per MG was calculated resulting in two values, the average daily cost per MG when they do not cooperate, denoted as $Cost^{non-coop}$, and the average daily

cost per MG when they cooperate in a VMG, denoted as $Cost^{coop}$. The difference between these values gives the daily monetary profit that each MG gains on average by cooperating in VMGs. The percentage profit (“value of cooperation”) is defined as:

$$\text{Value of Cooperation} = \\ (Cost^{non-coop} - Cost^{coop}) \cdot 100 / Cost^{non-coop}$$

We are particularly interested in the criteria used for selecting the MGs that participate in the same VMG. To cluster MGs into VMGs we used the correlation between their production profiles, as discussed in section IV.2.1.

TABLE II. AVERAGE COST PER 24H FOR ALL ALGORITHMS FOR A SINGLE MG, UNDER RANDOM 25% DEVIATIONS BETWEEN THE FORECASTED AND THE ACTUAL PRESUMPTION PROFILE.

Algorithms	Average Cost
Uncapacited	13.160
Selfie	2.809
Sniper	0.906
Stocky	1.410
DoubleStock	1.346
Conservative	1.162
ConservativePerCent	1.126
SmartStock	0.827
SmartConservative	1.148

The Value of Cooperation is calculated for different numbers n of cooperating MGs and presented in Table 3. The results are plotted in the same graph for the three correlation cases in Figure 4. From the results it is demonstrated that negatively correlated MGs exhibit a profit of the order of 15.5-16.5%, when cooperating. This gain increases somewhat when n increases. This was expected since in the latter case the production of one MG serves as a “hedge” for the production of the other when forecast errors occur. The profit is significantly smaller in the case of independent MGs (starts at 6.45% when $n=2$ but increases rapidly with n) and approaches that of negatively correlated MGs for large n . Thus, a higher number of cooperating MGs results to a higher profit per MG when the MGs are independent. Interestingly and rather surprisingly, there is also a positive value of cooperation (rather modest, of the order of 2.1%) even for maximally positively correlated MGs indicating that such MGs can still benefit from cooperation. The gains of cooperation therefore also come from MGs pooling their resources together. When the MGs are positively or negatively correlated, increasing n also helps but has diminishing returns after a certain point. This means, that in terms of storage sharing, small MG coalitions are good enough when the MGs are positively or negatively correlated, but larger coalitions are needed when the MGs are independent.

TABLE III. VALUE OF COOPERATION (PROFIT PER CENT) FOR N MGs

Value of Cooperation (profit %)	n=2	n=6	n=10	n=20	n=40	n=60
Positive correlation	2.10	2.85	3.55	3.45	3.73	4.21
Null correlation	6.45	11.12	12.66	13.77	14.70	15.23
Negative correlation	15.45	15.77	16.33	16.39	16.42	16.60

VI. CONCLUSIONS

We proposed scheduling algorithms that can be used by an MG to benefit from the time of use market scheme, and studied the benefits that MG cooperation can provide for the case of multiple Micro Grids forming coalitions. A profit of up to about 16,6% can be gained from the MG cooperation.

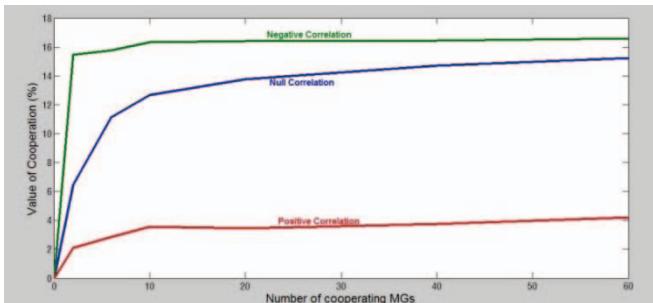


Fig. 4. Value of Cooperation (profit per cent) for n MGs

The benefits of cooperation are higher when the MGs forming a VMG have negatively correlated production patterns and they can also become equally high even when the MGs have independent production patterns, by increasing the number of participating MGs. The profit would be more significant if other techniques, such as Demand Response (DR) and Supply Response (SR), were applied. Future research will be carried out to investigate the profits that can be obtained by empowering, through cooperation, the negotiation power of MGs that could thus become a more significant player in the energy market.

ACKNOWLEDGMENT

The work presented in this paper has been undertaken in the context of the project VIMSEN (VIrtual Microgrids for Smart Energy Networks). VIMSEN is a Specific Targeted Research Project (STREP) supported by the European 7th Framework Programme, Contract number ICT-2013.6.1 - 619547.

REFERENCES

- [1] M. Alonso, H. Amaris, C. Alvarez-Ortega, Integration of renewable energy sources in smart grids by means of evolutionary optimization algorithms, *Expert Systems with Applications*, Vol. 39, 5513–5522, 2012.
- [2] I. Koutsopoulos, L. Tassiulas, Optimal Control Policies for Power Demand Scheduling in the Smart Grid, *IEEE Journal on Selected Areas in Communications*, Vol. 30, pp. 1049 – 1060, 2012.
- [3] J. Fossati, Unit Commitment and Economic Dispatch in Micro-Grids”, *Memoria de Trabajos de Difusion Cientifica y Tecnica*, Vol. 10, 2012.
- [4] W. Saad, Z. Han and H. V. Poor, Coalitional Game Theory for Cooperative Micro-Grid Distribution Networks, *IEEE International Conference on Communications Workshops* (2011) 1-5.
- [5] C. Wei, Z. Md. Fadlullah, N. Kato and A. Takeuchi , GT-CFS: A Game Theoretic Coalition Formulation Strategy for Reducing Power Loss in Micro Grids, *IEEE Trans. Parallel and Distributed Systems*, 2013.
- [6] R. M. Oviedo, Z. Fan, S. Gormus, P. Kulkarni, A residential PHEV load coordination mechanism with renewable sources in smart grids, *Electrical Power and Energy Systems*, 511–521, 2014.
- [7] I. S. Bayram et al, Local Energy Storage Sizing in Plug-in Hybrid Electric Vehicle Charging Stations Under Blocking Probability Constraints, Architectures and Models for the Smart Grid, 78-83, 2011.
- [8] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges, *IEEE Comm Surveys & Tutorials*, Vo. 15, 5-20, 2013.
- [9] L. Nikonowicz, J. Milewski, Virtual Power Plants – general review: structure, application and optimization, *Journal of Power Technologies*, 135–149, 2012.
- [10] L. G. Swan, V. I. Ugursal, Modeling of end-use energy consumption in the residential sector:A review of modeling techniques, *Renewable and Sustainable Energy Reviews*, Vol. 13, 1819–1835, 2009.
- [11] N. Amjady, Day-Ahead Price Forecasting of Electricity Markets by a New Fuzzy Neural Network, *IEEE Trans. on Power Systems*, Vol. 21(2), 2006.
- [12] Directive 2009/72/EC of the European Parliament and of the council of 13 July 2009 concerning common rules for the internal market in electricity.
- [13] <http://www.ict-vimSEN.eu/>

Demand Side Management in Smart Grid Using Heuristic Optimization

Thillainathan Logenthiran, *Student Member, IEEE*, Dipti Srinivasan, *Senior Member, IEEE*, and Tan Zong Shun

Abstract—Demand side management (DSM) is one of the important functions in a smart grid that allows customers to make informed decisions regarding their energy consumption, and helps the energy providers reduce the peak load demand and reshape the load profile. This results in increased sustainability of the smart grid, as well as reduced overall operational cost and carbon emission levels. Most of the existing demand side management strategies used in traditional energy management systems employ system specific techniques and algorithms. In addition, the existing strategies handle only a limited number of controllable loads of limited types. This paper presents a demand side management strategy based on load shifting technique for demand side management of future smart grids with a large number of devices of several types. The day-ahead load shifting technique proposed in this paper is mathematically formulated as a minimization problem. A heuristic-based Evolutionary Algorithm (EA) that easily adapts heuristics in the problem was developed for solving this minimization problem. Simulations were carried out on a smart grid which contains a variety of loads in three service areas, one with residential customers, another with commercial customers, and the third one with industrial customers. The simulation results show that the proposed demand side management strategy achieves substantial savings, while reducing the peak load demand of the smart grid.

Index Terms—Demand side management, distributed energy resource, evolutionary algorithm, generation scheduling, load shifting, smart grid.

I. INTRODUCTION

S MART GRID [1], [2] represents a vision of the future power systems integrating advanced sensing technologies, control methodologies and communication technologies at transmission and distribution levels in order to supply electricity in a smart and user friendly way. According to the U.S. Department of Energy's modern grid initiative report, the main characteristics [2] of a smart grid are consumer friendliness, hack proof self-healing, resistance for attack, ability to accommodate all types of generation and storage options, electricity market based efficient operation, high power quality, and optimal assets. This modern grid is prompted by several economical, political, environmental, social, and technical factors.

Demand side management [3], [4] is an important function in energy management of the future smart grid, which provides support towards smart grid functionalities in various areas such as electricity market control and management, infrastructure construction, and management of decentralized energy

Manuscript received July 22, 2011; revised November 08, 2011; accepted April 11, 2012. Date of publication June 08, 2012; date of current version August 20, 2012. This work was supported by National Research Foundation programme grant, NRF-2007EWT-CERP01-0954 (R-263-000-522-272). Paper no. TSG-00260-2011.

The authors are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576 (e-mail: logenthiran@nus.edu.sg; dipti@nus.edu.sg; u0705831@nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSG.2012.2195686

resources and electric vehicles. Controlling and influencing energy demand can reduce the overall peak load demand, reshape the demand profile, and increase the grid sustainability by reducing the overall cost and carbon emission levels. Efficient demand side management can potentially avoid the construction of an under-utilized electrical infrastructure in terms of generation capacity, transmission lines and distribution networks.

Smart pricing [5], [6] is a unique characteristic of smart grid made possible by usage of smart metering devices in the automatic metering infrastructure. It could lead to cost-reflective pricing based on the entire supply chain of delivering electricity at a certain location, quantity and period. When smart pricing is used with demand side management, control of the customer's energy usage will be influenced by real-time penalty and incentive schemes at all levels of the supply chain. However, the rationale behind the implementation of demand side management within the context of the smart grid is to promote the overall system efficiency, security and sustainability by maximizing the capacity of the existing infrastructure while facilitating the integration of low carbon technology into the system.

Demand side management also plays a significant role in electricity markets [7], [8]. Demand side management system will inform cluster's central controller about new load schedule and available load reduction capabilities for each time step of next day. Then, the central controller can place bids in the market such that some loads from the peak demand will be shifted. Profits made through this load demand side management will be reimbursed to customers of the cluster.

There are several demand side management techniques and algorithms used in the literature [4]–[6], [9]–[13]. Most of them are system specific [4]–[6], [10], [13] strategies, and some of which are not applicable to practical systems that have a wide variety of independent devices. Most of the techniques were developed using dynamic programming [13] and linear programming [5], [10]. These programming techniques cannot handle a large number of controllable devices from several types of devices which have several computation patterns and heuristics. The primary objective of the demand side management techniques presented in the literature is reduction of system peak load demand and operational cost. Although the utilities are capable of offering different incentives to respective customers for direct control [12]–[15] over selected loads by grouping the customers' loads, most of the methodologies used in the literature do not consider the criteria and objectives independently. Thus, it is difficult to employ these methods for demand side management of future smart grids which aim to provide the customers with greater control over their energy consumption. In a smart grid, the demand side management strategies need to handle a large number of controllable loads of several types. Furthermore, loads can have characteristics which spread over a few hours. Therefore, the strategies should be able to deal with all possible control durations of a variety of controllable loads.

In addition, the transformation of today's grid towards smart grid opens new perspectives on demand side management. First, a significant part of the generation in smart grid is expected to come from renewable energy resources such as wind and solar [16]. The unpredictability of these renewable energy sources makes power dispatch functions in a smart grid challenging. Such a scenario necessitates the use of load control methodologies. Next, the operation of smart grid requires a two way communication between central controller and various system elements. The designed demand side management system should therefore be able to handle the communication infrastructure between the central controller and controllable loads. The last, but not the least, criteria for deciding the optimal load consumption can vary widely. The criteria could be maximizing the use of renewable energy resources, maximizing the economic benefit by offering bids to reduce demand during peak periods, minimizing the amount of power imported from the main grid, or reducing peak load demand.

In this paper, a demand side management strategy is proposed for smart grid. The strategy is based on load shifting technique, which can handle a large number of devices of several types. A heuristic based evolutionary algorithm that can easily adapt its heuristics has been developed for solving the problem [15], [17]. Simulation studies were carried out on a smart grid which has different types of customers with a variety of loads. The remaining paper is organized as follows. Section II briefly explains the features of a suitable demand side management technique for future smart grid. Section III presents a demand side management strategy for smart grid, formulates load shifting technique mathematically, and briefly explains the proposed methodology. Section IV provides the details about simulation studies. Section V presents simulation results and discussion. Section VI concludes the paper.

II. DEMAND SIDE MANAGEMENT TECHNIQUES

Demand side management alters customers' electricity consumption patterns to produce the desired changes in the load shapes of power distribution systems. The changes in the final consumption profile will depend on the planning objectives and operation of the utility companies. Demand side management focuses [18] on utilizing power saving technologies, electricity tariffs, monetary incentives, and government policies to mitigate the peak load demand instead of enlarging the generation capacity or reinforcing the transmission and distribution network. To mitigate system instabilities brought about by increasing electricity demand, a suitable objective of demand side management activities could be to change the shape of the load demand curve by reducing the total load demand of the distribution system during peak periods, and shift these loads to be served during more appropriate times in order to reduce the overall planning and operational cost of the network. Such a scheme requires a sophisticated coordination between the network operators and customers.

The load shapes which indicate the daily or seasonal electricity demands of industrial, commercial or residential consumers between peak and off peak times can be altered by means of six broad methods [18]–[20]: peak clipping, valley filling, load shifting, strategic conservation, strategic load growth, and flexible load shape. Generally, these are the possible demand side management techniques that can be employed in future smart grids. These six demand side management techniques are illustrated in Fig. 1.

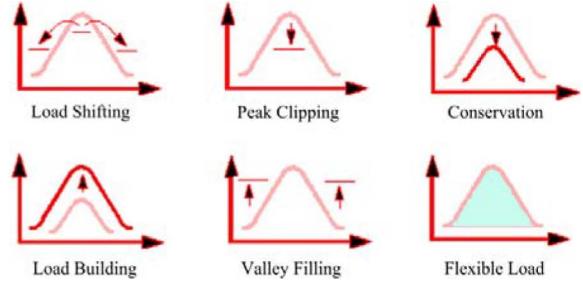


Fig. 1. Demand side management techniques.

Peak clipping and valley filling focus on reducing the difference between the peak and valley load levels to mitigate the burden of peak demand, and increase the security of smart grid. Peak clipping [18], [20] is a direct load control technique to make reduction of the peak loads, and valley filling constructs the off-peak demand by applying direct load control.

Load shifting [20] is widely applied as the most effective load management technique in current distribution networks. Load shifting takes advantage of time independence of loads, and shifts loads from peak time to off-peak time. Strategic conservation [18] aims to achieve load shape optimization through application of demand reduction methods directly at customer premises. The distribution management system has to consider this for longer term implications of demand reduction on network planning and operation.

Strategic load growth [18]–[20] optimizes the daily response in case of large demand introduction beyond the valley filling technique. It is based on increasing the market share of loads supported by energy conversion and storage systems or distributed energy resources. It is a planning and operations issue to balance the increasing demand with processes for constructing necessary infrastructure that accompanies load growth. The future smart grid has to provide the necessary infrastructure for strategic load growth. Flexible load shape [18]–[20] is mainly related to reliability of smart grid. Smart grid management systems identify customers with flexible loads which are willing to be controlled during critical periods in exchange for various incentives. Studies have to be conducted to identify the anticipated load shape which includes demand side activities forecasted over the planning horizon.

III. PROPOSED DEMAND SIDE MANAGEMENT STRATEGY

This paper presents a generalized day-ahead demand side management (DSM) strategy for the future smart grid. It uses load shifting as the primary technique that can be utilized by the central controller of the smart grid. Objective of the demand side management could be maximizing the use of renewable energy resources, maximizing the economic benefit, minimizing the power imported from the main distribution grid, or reducing the peak load demand. Smart grid manager designs an objective load curve according to the objective of the demand side management. The proposed optimization algorithm aims to bring the final load curve as close to the objective load curve as possible such that the desired objective of the DSM strategy is achieved. For example, if the objective of the demand side management is to reduce the utility bill, an objective load curve will be chosen such that it is inversely propositional to electricity market prices.

Fig. 2 shows the proposed architecture for the day-ahead demand side management strategy. According to the proposed architecture, the demand side management system receives the objective load curve as an input, and calculates the required load

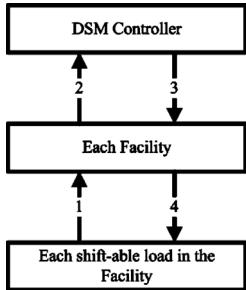


Fig. 2. Proposed architecture for DSM.

control actions in order to fulfill the desired load consumption. Therefore, the proposed algorithm is flexible in that it is completely independent from the criteria used to generate the objective load curve.

The demand side management is carried out at the beginning of a predefined control period which is typically a day. Then, the control actions are executed in real-time based on the results. Fig. 2 shows the exchange of information between the demand side management controller and each appliance during the real-time operation. This takes advantage of the communication capability of the smart grid. When a customer presses ON button of an appliance, the connection request is sent to the demand side management controller. The demand side management controller replies based on the results of demand side management technique that was carried out in advance. The reply is either the connection permitted or a new connection time.

A. Problem Formulation

The proposed demand side management strategy schedules the connection moments of each shiftable device in the system in a way that brings the load consumption curve as close as to the objective load consumption curve. Proposed load shifting technique is mathematically formulated as follows.

Minimize

$$\sum_{t=1}^N (P\text{Load}(t) - \text{Objective}(t))^2 \quad (1)$$

where $\text{Objective}(t)$ is the value of the objective curve at time t , and $P\text{Load}(t)$ is the actual consumption at time t .

The $P\text{Load}(t)$ is given by the following equation:

$$P\text{Load}(t) = \text{Forecast}(t) + \text{Connect}(t) - \text{Disconnect}(t) \quad (2)$$

where $\text{Forecast}(t)$ is the forecasted consumption at time t , and $\text{Connect}(t)$ and $\text{Disconnect}(t)$ are the amount of loads connected and disconnected at time t respectively during the load shifting.

$\text{Connect}(t)$ is made up of two parts: the increment in the load at time t due to the connection times of devices shifted to time t , and the increment in the load at time t due to the device connections scheduled for times that precede t . The $\text{Connect}(t)$ is given by the following equation:

$$\begin{aligned} \text{Connect}(t) = & \sum_{i=1}^{t-1} \sum_{k=1}^D X_{kit} \cdot P_{1k} \\ & + \sum_{l=1}^{j-1} \sum_{i=1}^{t-1} \sum_{k=1}^D X_{ki(t-1)} \cdot P_{(1+l)k} \end{aligned} \quad (3)$$

where X_{kit} is the number of devices of type k that are shifted from time step i to t , D is the number of device types, P_{1k} and $P_{(1+l)k}$ are the power consumptions at time steps 1 and $(1+l)$

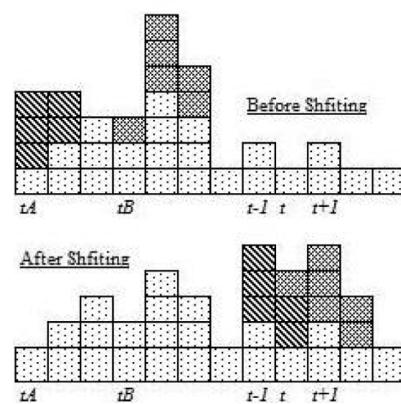
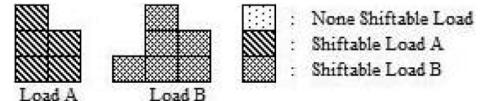


Fig. 3. $\text{Connect}(t)$.

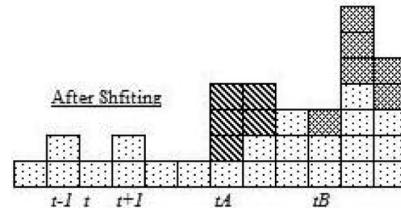
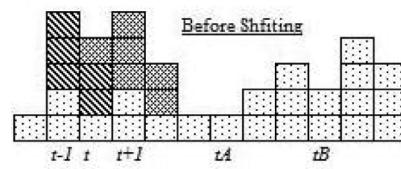
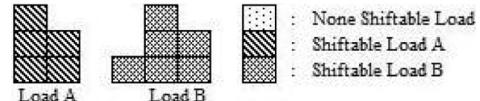


Fig. 4. $\text{Disconnect}(t)$.

respectively for device type k , and j is the total duration of consumption for device of type k .

Fig. 3 illustrates $\text{Connect}(t)$, where Load A is shifted from tA to $(t-1)$, and Load B is shifted from tB to t .

Similarly, $\text{Disconnect}(t)$ also consists of two parts: the decrement in the load due to delay in connection times of devices that were originally supposed to begin their consumption at time step t , and the decrement in the load due to delay in connection times of devices that were expected to start their consumption at time steps that precede t . The $\text{Disconnect}(t)$ is given by the following equation:

$$\begin{aligned} \text{Disconnect}(t) = & \sum_{q=t+1}^{t+m} \sum_{k=1}^D X_{ktq} \cdot P_{1k} \\ & + \sum_{l=1}^{j-1} \sum_{q=t+1}^{t+m} \sum_{k=1}^D X_{k(t-1)q} \cdot P_{(1+l)k} \end{aligned} \quad (4)$$

where X_{ktq} is the number of devices of type k that are delayed from time step t to q , m is the maximum allowable delay.

Fig. 4 illustrates $\text{Disconnect}(t)$, where Load A is shifted from $(t-1)$ to tA , and Load B is shifted from t to tB .

This minimization problem is subject to the following constraints:

The number of devices shifted cannot be a negative value.

$$X_{kit} > 0 \quad \forall i, j, k. \quad (5)$$

The number of devices shifted away from a time step cannot be more than the number of devices available for control at the time step.

$$\sum_{t=1}^N X_{kit} \leq Ctrlable(i) \quad (6)$$

where $Ctrlable(i)$ is the number of devices of type k available for control at time step i .

B. Proposed Algorithm

Demand side management algorithm for future smart grid needs to be designed to process a large number of controllable devices of several types. Furthermore, each type of controllable load can have different consumption characteristics which spread over few hours, and have several heuristics. Therefore, the designed algorithm should be able to handle these complexities. Linear programming and dynamic programming are commonly used in this field. However, both linear programming and dynamic programming algorithms cannot adequately handle these complexities. Evolutionary computation algorithms have shown potential for solving such complex problems in other areas [17]. These algorithms have several advantages over the traditional mathematical algorithms, in addition to providing near optimal results. Hence, in this paper, a heuristic based evolutionary algorithm is proposed, and has been developed to solve the problem [15], [17]. The proposed algorithm not only adapts the heuristics in the problem easily but also provides an efficient and cost effective solution to the problem.

One of the main advantages of the proposed algorithm is the flexibility in constructing and developing the algorithm which cannot be afforded by any other conventional approaches. The flexible nature of evolutionary algorithm allows implementation of features that model load demand patterns based on the lifestyles of the customers so that the inconvenience to the customers can be minimized. For example, consider two controllable loads: coffee maker and washing machine. They begin their consumptions in the morning. Coffee maker typically operates during the morning due to customers' lifestyle. Hence, the algorithm can take this into consideration, and try to shift coffee maker as early as possible in the morning. Washing machine has no such preference so that it can be shifted to a later time of the day.

Certain loads may have higher priority over other loads, which can also be taken into consideration by the algorithm so that these loads are shifted to the appropriate time steps according to their importance. These types of loads can be seen in the test cases considered in this paper, where the controllable devices have characteristics such that connection times of the devices can only be delayed, and not brought forward. Another main advantage of the proposed algorithm is the ability to handle large number of controllable devices of several types. The size of the problem affects only the length of the chromosomes of evolutionary algorithm.

The demand side management problem has characteristics such as, connection times of devices that can only be delayed and not brought forward, which can be expressed as

$$X_{kit} = 0 \quad \forall i > t. \quad (7)$$

The contract options stipulate the maximum allowable time delay for all devices and limit the possible number of time steps that devices can be shifted to, thus,

$$X_{kit} = 0 \quad \forall (t - i) > m \quad (8)$$

where m is the maximum permissible delay.

Taking the above into consideration, the maximum number of possible time steps N can be found using the following equation:

$$N = \left((24 - m) \times m + \sum_{n=1}^{m-1} n \right) \times k \quad (9)$$

where k is the number of different types of devices.

Chromosomes of the evolutionary algorithm represent the solutions to the problem. In this work, the chromosome is constructed as an array of bits. The length of the chromosome is directly related to the number of time steps N , which is given by the following equation:

$$\text{Length of chromosome} = N \times B \quad (10)$$

where B is the number of bits required to represent the number of devices that are shifted in each time step.

A population of chromosomes (i.e., 200) is randomly initialized by the numbers between upper and lower limits of each gene. A fitness function is chosen such that the algorithm achieves a final load curve as close to the objective load curve as possible, which is given as follows:

$$\text{Fitness} = \frac{1}{1 + \sum_{t=1}^{24} (PLoad(t) - Objective(t))^2}. \quad (11)$$

The main stages of the proposed evolutionary algorithm are shown in Fig. 5. While the algorithm is progressing, new populations of chromosomes are produced from the existing populations by genetic operators: single point crossover and binary mutation. A large crossover rate ensures faster convergence of the solution. Very large mutation rate may result in loss of good solutions from previous generations, and stop the algorithm with premature convergence. Best crossover rate (i.e., 0.9) and mutation rate (i.e., 0.1) were found experimentally for this problem. In addition, elitism is used to prevent the loss of good solutions that may emerge early on. A tournament based selection is used for reproduction. The algorithm is terminated when the stipulated number of generations (i.e., 500) is reached or when the magnitude in the change in fitness value does not vary more than a tolerance limit (i.e., 10^{-10}) for several (i.e., 50) subsequent generations.

IV. DETAILS OF THE TEST SMART GRID

To demonstrate the effectiveness of the proposed approach, the proposed demand side management strategy is tested on three different areas of a smart grid, each with different types of customers; namely residential, commercial and industrial customers. Electrical network diagram of the smart grid is shown in Fig. 6. The entire network operates at a voltage of 410 V. Each interconnection link including the link between the smart grid and the main grid has a resistance of 0.003 pu, a reactance of 0.01 pu, and maximum power transfer limit of 500 kVA. Length of the links in the residential microgrid is 2 km, while those in the commercial and industrial microgrids are 3 km and 5 km respectively. This network information is used for scheduling of resources without any congestion.

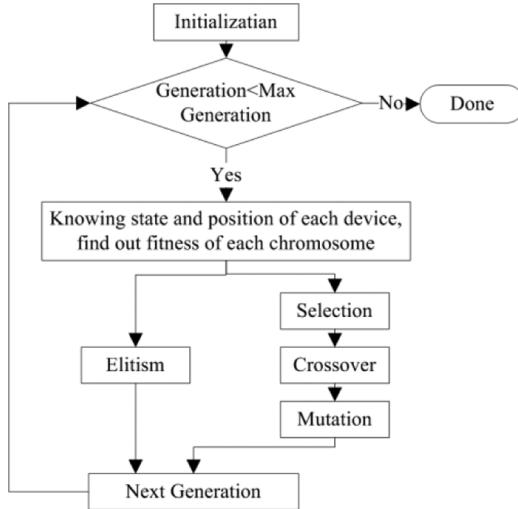


Fig. 5. Proposed evolutionary algorithm.

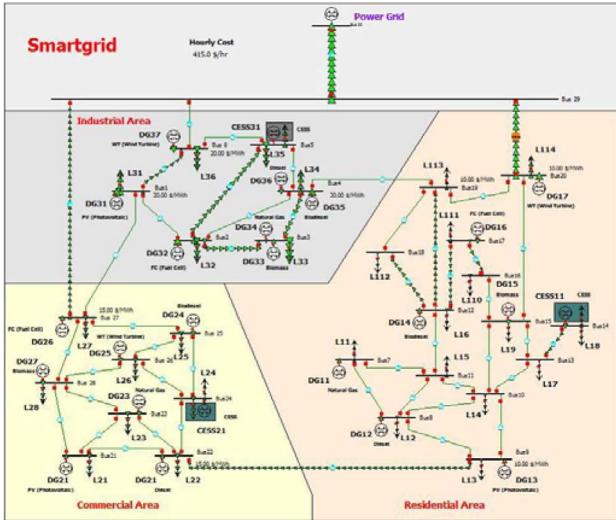


Fig. 6. Electrical network of the smart grid.

In this case study, the primary objective is to reduce the utility bills of consumers in these areas. Therefore, an objective load curve was chosen to be inversely proportional to electricity market prices. The same market prices were applied to all areas in the smart grid. Simulations were carried out with a maximum allowable delay of 12 h. It is known that the longer the delay, the better the performance of demand side management algorithm since the possible number of loads subjected to load shifting increases, resulting in improved results.

Forecasted hourly wholesale electricity prices and forecasted hourly load consumption of each area of the smart grid are given in Table I. The maximum load demands of residential, commercial, and industrial areas in this study are 1.5 MW, 2 MW, and 3 MW respectively. On a typical day, valley of the load consumption curve will be before the peak hours. If load shifting window is from 0 to 24 h, the peak load cannot be shifted to valley hours. In order to avoid this situation, the control period is changed from 8th hr of current day to eighth hour of the following day. This helps to better understand the effect of load shifting. Each area of the smart grid has different types of controllable devices, the details of which are given as follows.

TABLE I
FORECASTED LOAD DEMANDS AND WHOLESALE ENERGY PRICES

Time	Wholesale Price (ct/kWh)	Hourly Forecasted Load (kWh)		
		Residential Microgrid	Commercial Microgrid	Industrial Microgrid
8hrs-9hrs	12.00	729.4	923.5	2045.5
9hrs-10hrs	9.19	713.5	1154.4	2435.1
10hrs-11hrs	12.27	713.5	1443.0	2629.9
11hrs-12hrs	20.69	808.7	1558.4	2727.3
12hrs-13hrs	26.82	824.5	1673.9	2435.1
13hrs-14hrs	27.35	761.1	1673.9	2678.6
14hrs-15hrs	13.81	745.2	1673.9	2678.6
15hrs-16hrs	17.31	681.8	1587.3	2629.9
16hrs-17hrs	16.42	666.0	1558.4	2532.5
17hrs-18hrs	9.83	951.4	1673.9	2094.2
18hrs-19hrs	8.63	1220.9	1818.2	1704.5
19hrs-20hrs	8.87	1331.9	1500.7	1509.7
20hrs-21hrs	8.35	1363.6	1298.7	1363.6
21hrs-22hrs	16.44	1252.6	1096.7	1314.9
22hrs-23hrs	16.19	1046.5	923.5	1120.1
23hrs-24hrs	8.87	761.1	577.2	1022.7
24hrs-1hrs	8.65	475.7	404.0	974.0
1hrs-2hrs	8.11	412.3	375.2	876.6
2hrs-3hrs	8.25	364.7	375.2	827.9
3hrs-4hrs	8.10	348.8	404.0	730.5
4hrs-5hrs	8.14	269.6	432.9	730.5
5hrs-6hrs	8.13	269.6	432.9	779.2
6hrs-7hrs	8.34	412.3	432.9	1120.1
7hrs-8hrs	9.35	539.1	663.8	1509.7

TABLE II
DATA OF CONTROLLABLE DEVICES IN THE RESIDENTIAL AREA

Device Type	Hourly Consumption of Device (kW)			Number of Devices
	1st Hr	2nd Hr	3rd Hr	
Dryer	1.2	-	-	189
Dish Washer	0.7	-	-	288
Washing Machine	0.5	0.4	-	268
Oven	1.3	-	-	279
Iron	1.0	-	-	340
Vacuum Cleaner	0.4	-	-	158
Fan	0.20	0.20	0.20	288
Kettle	2.0	-	-	406
Toaster	0.9	-	-	48
Rice Cooker	0.85	-	-	59
Hair Dryer	1.5	-	-	58
Blender	0.3	-	-	66
Frying Pan	1.1	-	-	101
Coffee Maker	0.8	-	-	56
Total	-	-	-	2604

A. Residential Area

The devices subjected to control in the residential area have small power consumption ratings and short durations of operation. Table II shows device types that are subjected to load control and their consumption patterns. There are over 2600 controllable devices available in this area from 14 different types of devices.

B. Commercial Area

The devices subjected to load control in the commercial area have consumption ratings which are slightly higher than those in the residential area. The consumption patterns of the loads under the control are given in Table III. There are over 800 controllable

TABLE III
DATA OF CONTROLLABLE DEVICES IN THE COMMERCIAL AREA

Device Type	Hourly Consumption of Device (kW)			Number Devices
	1st Hr	2nd Hr	3rd Hr	
Water Dispenser	2.5	-	-	156
Dryer	3.5	-	-	117
Kettle	3.0	2.5	-	123
Oven	5.0	-	-	77
Coffee Maker	2.0	2.0	-	99
Fan/AC	3.5	3.0	-	93
Air Conditioner	4.0	3.5	3.0	56
Lights	2.0	1.75	1.5	87
Total	-	-	-	808

TABLE IV
DATA OF CONTROLLABLE DEVICES IN THE INDUSTRIAL AREA

Device Type	Hourly Consumption of Device (kW)						Number Devices
	1st Hr	2nd Hr	3rd Hr	4th Hr	5th Hr	6th Hr	
Water Heater	12.5	12.5	12.5	12.5	-	-	39
Welding Machine	25.0	25.0	25.0	25.0	25.0	-	35
Fan/AC	30.0	30.0	30.0	30.0	30.0	-	16
Arc Furnace	50.0	50.0	50.0	50.0	50.0	50.0	8
Induction Motor	100	100	100	100	100	100	5
DC Motor	150	150	150	-	-	-	6
Total	-	-	-	-	-	-	109

devices available for control in this area from 8 different types of devices.

C. Industrial Area

The number of devices available for control in the industrial area is the smallest among all three areas; however, the devices have largest consumption ratings and longest consumption periods. The consumption patterns of the devices in this area are given in Table IV. The reason for a small number of devices available for control can be attributed to the fact that most of the industrial loads are critical and cannot be subjected to load control. The control periods of the devices are similar to those in the other two areas. There are over 100 controllable devices belonging to 6 different types.

V. SIMULATION RESULTS AND DISCUSSION

Simulation results show that the proposed demand side management strategy has managed to bring the final consumption close to the objective load curve in all three cases. The proposed algorithm has efficiently handled the large number of controllable loads of several types, and adopts all heuristics in the smart grid.

The simulation results obtained for the residential area are given in Fig. 7. The utility bill of the residential area for the day reduces from \$2302.90 to \$2188.30 with demand side management strategy, resulting in about 5.0% reduction in the operating cost.

The results obtained for the commercial area are given in Fig. 8. The utility bill of the commercial area for the day reduces from \$3636.60 to \$3424.30 with demand side management strategy, which results in approximately 5.8% reduction in the operating cost.

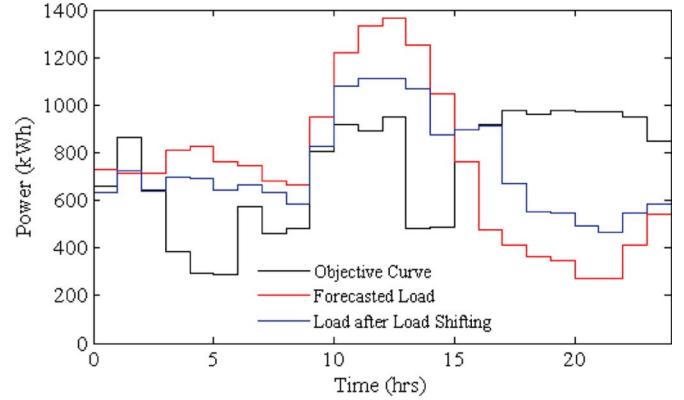


Fig. 7. DSM results of the residential area.

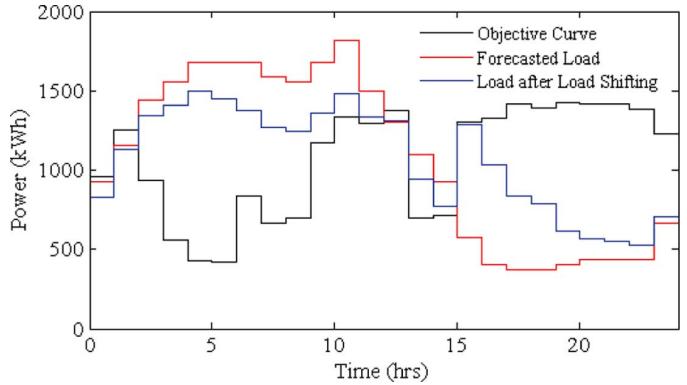


Fig. 8. DSM results of the commercial area.

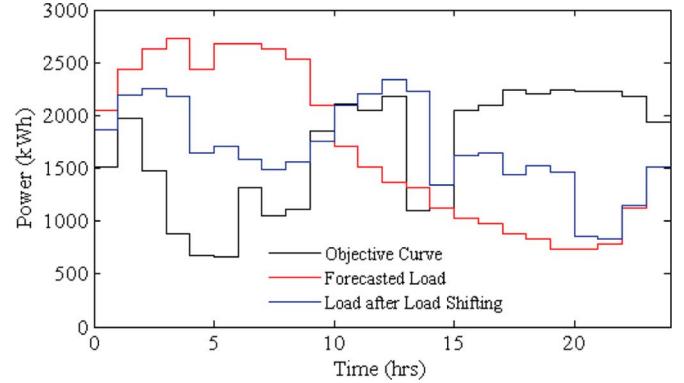


Fig. 9. DSM results of the industrial area.

The results obtained for the industrial area are given in Fig. 9. The utility bill of the industrial area without demand side management strategy is \$5712.00 for the day; whereas it is \$5141.60 with demand side management strategy, resulting in 10% reduction in the operating cost.

Table V summarizes the simulation results from the proposed demand side management strategy for these three areas of the smart grid. The approach has successfully managed to achieve the objective in all three areas, with considerable savings in the utility bills.

Typically, demand side management results are better when the number of devices available for control increases [3]–[15]. However, this may not be true in this case study because complexities of the devices under control pose restrictions. In this case study, even though the number of devices available for control is the least in the industrial area, the percentage reduction in the operating cost is the highest among all areas. On the

TABLE V
OPERATIONAL COST REDUCTION

Area	Cost without DSM (\$)	Cost with DSM (\$)	Percentage Reduction (%)
Residential	2302.90	2188.30	5.0
Commercial	3636.60	3424.30	5.8
Industrial	5712.00	5141.60	10.0

TABLE VI
PEAK DEMAND REDUCTION

Type of Area	Peak Load Without DSM (kW)	Peak Load With DSM (kW)	Peak Reduction (kW)	Percentage Reduction (%)
Residential	1363.6	1114.4	249.2	18.3
Commercial	1818.2	1485.2	333.0	18.3
Industrial	2727.3	2343.6	383.7	14.2

other hand, the residential area has the highest number of devices available for control in terms of quantity and variety, but the percentage reduction in the operating cost is not much as expected. This can be attributed to the fact that high power consumption of devices in the industrial area compared to much lower consumption of devices in the residential area. Additionally, even small load shifting of high power devices results in huge savings for the customers.

In a smart grid, it is expected that utilities will bundle up services and provide several contract options for customers. The amount of reimbursement taken by customers through such schemes will depend on how much inconvenience the customer is willing to undergo. In case of customers in residential area, level of tolerance is high for most of the devices. This means that generally the customers do not have strong preference regarding the time when the loads have to be consumed. On the other hand, customers in commercial and industrial areas are less willing to change their consumption patterns.

Effective demand side management provides benefits not only to the end users but also to the utilities. One of the main advantages is the reduction in peak load demands. Table VI shows the peak load demands with and without the proposed demand side management strategy for the three areas. It can be observed that the proposed demand side management strategy reduces the peak load demand for each area.

Reduction in the peak load demand improves grid sustainability by reducing the overall cost and carbon emission levels. Furthermore, this will lead to the avoidance of the construction of an under-utilized electrical infrastructure in terms of generation capacity, transmission lines and distribution networks.

Generation companies also stand to benefit from demand side management, as the reduction in peak load demand results in substantial cost savings since costly generators that are typically turned on to provide power during the peak load demand are no longer needed. When system peak load demand reduces, the operating cost of generators will therefore be reduced substantially. This would also result in increasing reserve generation capacity of the system.

In order to determine the cost saving for generation companies, the entire smart grid comprising of three areas is scheduled for the day with and without proposed demand side management strategy. Problem formulation and methodology of the generation scheduling are described in [21], [22].

Figs. 10–12 show the generation scheduling of residential, commercial, and industrial areas without proposed demand side

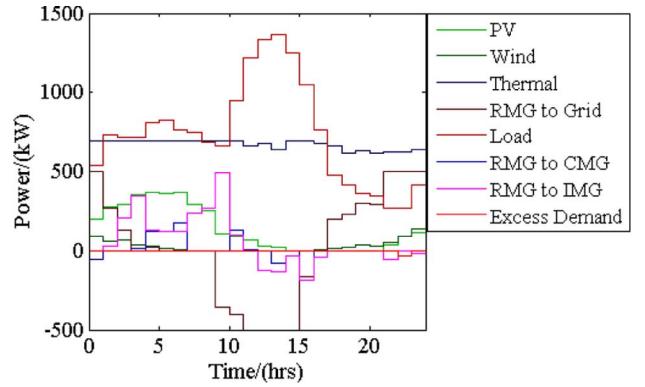


Fig. 10. Residential area without DSM.

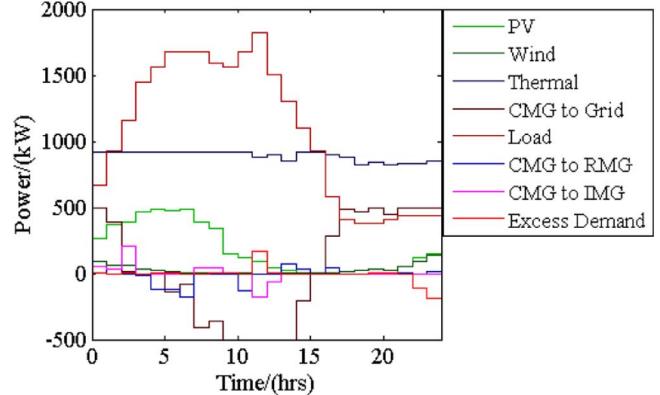


Fig. 11. Commercial area without DSM.

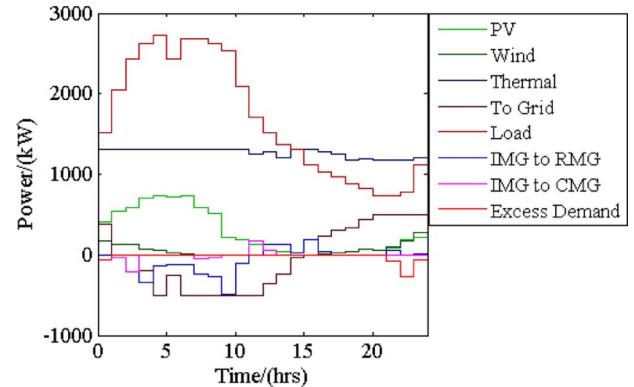


Fig. 12. Industrial area without DSM.

management strategy respectively. Fig. 13 shows power exchange among the areas. In this case, the industrial area spends \$695.51 with 346.34 kWh of load curtailment, and the commercial area spends \$551.33, whereas the residential area saves \$225.48 for the scheduling day. As a whole, the smart grid spends only \$53.70 with 346.34 kWh of load curtailment during the scheduling day.

Figs. 14–16 show the generation scheduling of residential, commercial, and industrial areas with proposed demand side management strategy respectively.

Fig. 17 shows power exchange among the areas. In this case, the industrial area spends \$252.15, and the commercial area spends \$157.52, whereas the residential area saves \$345.82 for the scheduling day. As a whole, the smart grid saves \$103.70 without any unsatisfied demand during the scheduling day.

Table VII shows the cost saving of each area from generation companies' point of view, and unsatisfied load demand of

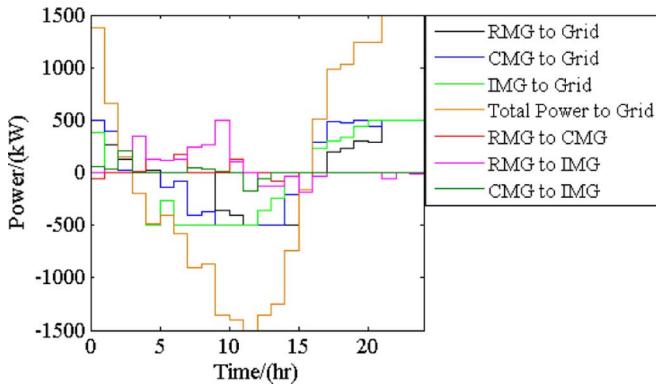


Fig. 13. Power exchanges without DSM.

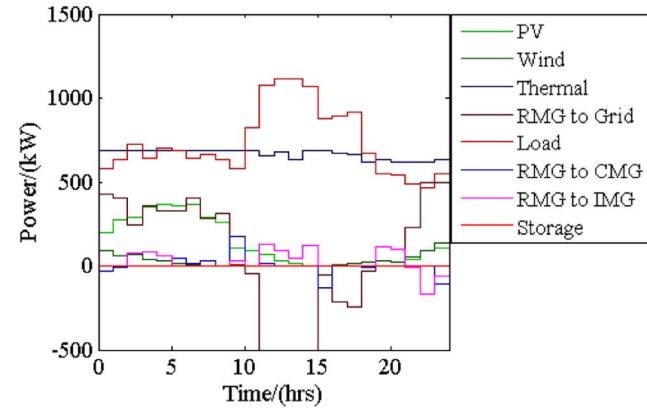


Fig. 14. Residential area with DSM.

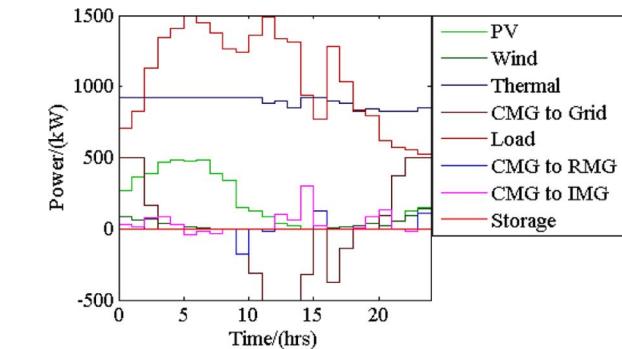


Fig. 15. Commercial area with DSM.

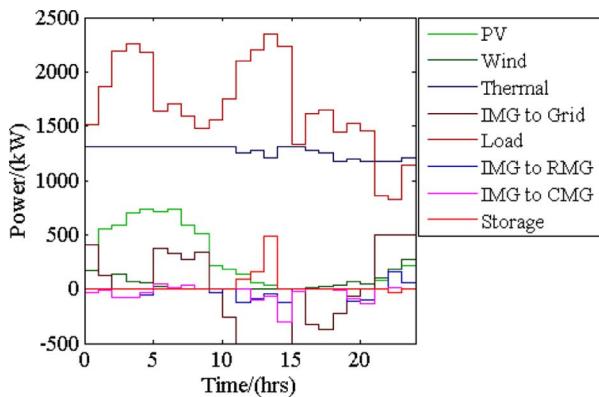


Fig. 16. Industrial area with DSM.

each area. The simulation results show that demand side management is indeed beneficial to both consumers and the utility companies. In the test case, customers achieve 5% to 10% cost savings, and generation companies archive a substantial saving with optimized generation scheduling. In addition, transmission

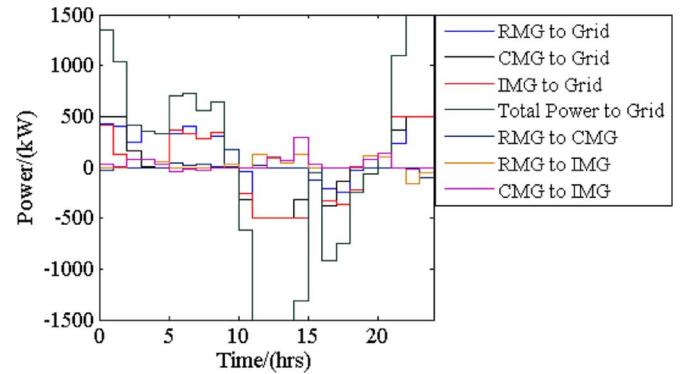


Fig. 17. Power exchanges with DSM.

TABLE VII
ADDITIONAL SAVING FROM GENERATION SCHEDULING

Type of Area	Additional Saving by DSM (\$)	Unsatisfied load (kWh)	
		Without DSM	With DSM
Residential	120.34	0	0
Commercial	393.81	0	0
Industrial	443.36	346.34	0
Whole Smart grid	157.4	0	0

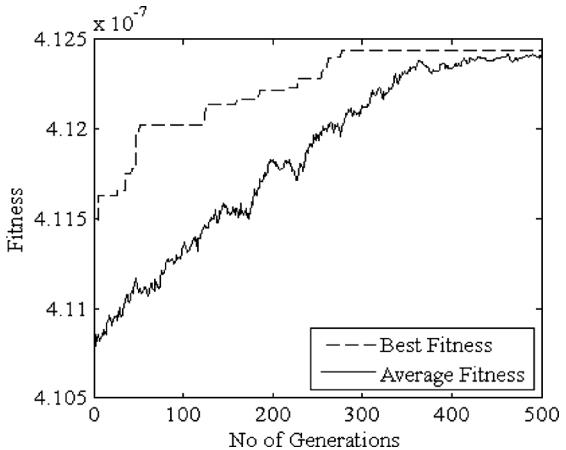


Fig. 18. Convergence characteristic of evolutionary algorithm.

companies achieve between 15% to 20% reduction in the network congestion. The above test cases are only isolated examples. The actual cost savings and peak reduction will be much higher when the proposed strategy is used for a large smart grid. The above test cases are just examples to show how the proposed demand side management strategy could be applied for smart grid operation.

The proposed demanded side management strategy is a generalized technique based on a day-ahead load shifting, and can be applied for large smart grids of the future. The inputs to the problem are: control period (i.e., number of time steps), discrete model of devices' load consumption pattern, power consumption at each time step and various uncertainties. The proposed evolutionary algorithm converges well, and it takes about 6 h for the test case comprising three areas. The convergence characteristic for the case study is shown in Fig. 18. Although the computation time appears large, it is acceptable as it solves a day-ahead demand side management strategy which is to be carried out 24 h before the time of operation.

Even though the proposed strategy provides good results, further work is needed for real time implementation. A real-time demand side management strategy, together with this proposed

day-ahead demand side management strategy and demand side management techniques such as load curtailment, load shedding and load shifting with small time span can be developed for real-time operation of smart grid. The nature of real-time demand side management problem shows that it is possible to develop only on a distributed operational platform. The authors plan to carry out further work on real-time demand response on a distributed artificial intelligent platform provided by multiagent system (MAS) [16], [23].

VI. CONCLUSION

Demand side management has potential to provide many benefits to the entire smart grid, particularly at distribution network level. This paper presents a demand side management strategy that can be employed in the future smart grid. The proposed strategy is a generalized technique based on load shifting, which has been mathematically formulated as a minimization problem. A heuristic based evolutionary algorithm is developed for solving the problem. Simulations were carried out on a smart grid which contains three different kinds of customers' areas. The simulation outcomes show that the proposed algorithm is able to handle a large number of controllable devices of several types, and achieves substantial savings while reducing the peak load demand of the smart grid.

REFERENCES

- [1] Q. Li and M. Zhou, "The future-oriented grid-smart grid," *J. Comput.*, vol. 6, no. 1, pp. 98–105, 2011.
- [2] P. Agrawal, "Overview of DOE microgrid activities," in *Proc. Symp. Microgrid*, Montreal, QC, Canada, 2006 [Online]. Available: http://der.lbl.gov/2006microgrids_files/USA/Presentation_7_Part1_Poonum-grawal.pdf
- [3] S. Rahman and Rinaldy, "An efficient load model for analyzing demand side management impacts," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1219–1226, Aug. 1993.
- [4] A. I. Cohen and C. C. Wang, "An optimization method for load management scheduling," *IEEE Trans. Power Syst.*, vol. 3, no. 2, pp. 612–618, May 1988.
- [5] K.-H. Ng and G. B. Sheble, "Direct load control-A profit-based load management using linear programming," *IEEE Trans. Power Syst.*, vol. 13, no. 2, pp. 688–694, May 1998.
- [6] F. C. Schweppe, B. Daryanian, and R. D. Tabors, "Algorithms for a spot price responding residential load controller," *IEEE Trans. Power Syst.*, vol. 4, no. 2, pp. 507–516, May 1989.
- [7] M. Shahidehpour, H. Yamin, and Z. Li, *Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management*. New York: Wiley–IEEE Press, 2002.
- [8] Z. N. Popovic and D. S. Popovic, "Direct load control as a market-based program in deregulated power industries," in *Proc. IEEE Bologna Power Tech Conf.*, Jun. 23–26, 2003, vol. 3, p. 4.
- [9] H. Lee and C. L. Wilkins, "A practical approach to appliance load control analysis: A water heater case study," *IEEE Trans. Power App. Syst.*, vol. PAS-102, no. 4, pp. 1007–1013, Apr. 1983.
- [10] C. N. Kurucz, D. Brandt, and S. Sim, "A linear programming model for reducing system peak through customer load control programs," *IEEE Trans. Power Syst.*, vol. 11, no. 4, pp. 1817–1824, Nov. 1996.
- [11] W. C. Chu, B. K. Chen, and C. K. Fu, "Scheduling of direct load control to minimize load reduction for a utility suffering from generation shortage," *IEEE Trans. Power Syst.*, vol. 8, no. 4, pp. 1525–1530, Nov. 1993.
- [12] H. G. Weller, "Managing the instantaneous load shape impacts caused by the operation of a large-scale direct load control system," *IEEE Trans. Power Syst.*, vol. 3, no. 1, pp. 197–199, Feb. 1988.
- [13] Y. Y. Hsu and C. C. Su, "Dispatch of direct load control using dynamic programming," *IEEE Trans. Power Syst.*, vol. 6, no. 3, pp. 1056–1061, Aug. 1991.
- [14] K. H. Ng and G. B. Sheble, "Direct load control-A profit-based load management using linear programming," *IEEE Trans. Power Syst.*, vol. 13, no. 2, pp. 688–694, May 1998.

- [15] L. Yao, W. C. Chang, and R. L. Yen, "An iterative deepening genetic algorithm for scheduling of direct load control," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1414–1421, Aug. 2005.
- [16] T. Logenthiran, D. Srinivasan, and A. M. Khambadkone, "Multi-agent system for energy resource scheduling of integrated microgrids in a distributed system," *Electr. Power Syst. Res.*, vol. 81, no. 1, pp. 138–148, 2011.
- [17] T. Back, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. New York: IOP Publ. and Oxford Univ. Press, 1997.
- [18] I. K. Mahajan, *Demand Side Management: Load Management, Load Profiling, Load Shifting, Residential and Industrial Consumer; Energy Audit, Reliability, Urban, Semi-Urban and Rural Setting*. Saarbrücken, Germany: LAP (Lambert Acad. Publ.), 2010.
- [19] D. P. Kothari, *Modern Power System Analysis*. New Delhi, India: Tata McGraw-Hill, 2003.
- [20] C. W. Gellings, *Demand-Side Management: Concepts and Methods*. Lilburn, GA: Fairmont, 1988.
- [21] T. Logenthiran, D. Srinivasan, A. M. Khambadkone, and H. N. Aung, "Scalable multi-agent system (MAS) for operation of a microgrid in islanded mode," in *Proc. Joint Int. Conf. PEDES & Power India*, Dec. 20–23, 2010, pp. 1–6.
- [22] T. Logenthiran and D. Srinivasan, "Short term generation scheduling of a microgrid," in *Proc. TENCON IEEE Region 10 Conf.*, Jan. 23–26, 2009, pp. 1–6.
- [23] T. Logenthiran and D. Srinivasan, "Management of distributed energy resources using intelligent multi-agent system," in *Multi-Agent Applications With Evolutionary Computation and Biologically Inspired Technologies: Intelligent Techniques for Ubiquity and Optimization*, S.-H. Chen, Ed. et al. Hershey, PA: IGI Global, 2011, pp. 208–231.



Thillainathan Logenthiran (S'07) received the B.Sc. degree in electrical and electronic engineering from University of Peradeniya, Sri Lanka, in 2005. Currently, he is working toward the Ph.D. degree in applications of intelligent multiagent systems for control and management of distributed power systems from National University of Singapore, Singapore.

His main areas of interest are distributed power systems, and applications of intelligent multiagent systems and computational intelligence techniques to power engineering problems.

Mr. Logenthiran is the first prize award winner of Siemens Smart Grid Innovation Contest, 2011. He was awarded in recognition and appreciation of his professional contribution on "Multi-Agent System for Operation of a Smart Grid" and "Autonomous Distributed Power System Restoration."



Dipti Srinivasan (SM'02) received the M.Eng. and Ph.D. degrees in electrical engineering from the National University of Singapore (NUS), Singapore, in 1991 and 1994, respectively.

She worked at the University of California at Berkeley's Computer Science Division as a Postdoctoral Researcher from 1994 to 1995. In June 1995, she joined the faculty of the Electrical & Computer Engineering department at the National University of Singapore, where she is an Associate Professor. From 1998 to 1999 she was a Visiting Faculty in the

Department of Electrical & Computer Engineering at the Indian Institute of Science, Bangalore, India. Her research interest is in the development of hybrid neural network architectures, learning methods, and their practical applications for large complex engineered systems, such as the electric power system and urban transportation systems.

Dr. Srinivasan is currently serving as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. She was awarded the IEEE PES Outstanding Engineer award in 2010.

Tan Zong Shun received the B.Eng. degree in electrical engineering from National University of Singapore (NUS), Singapore.

His research interests include power system operation and control, distributed generation, and applications of artificial intelligence in power systems.

Scheduling for Electricity Cost in Smart Grid

Mihai Burcea^{1,*}, Wing-Kai Hon², Hsiang-Hsuan Liu²,
Prudence W.H. Wong¹, and David K.Y. Yau³

¹ Department of Computer Science, University of Liverpool, UK
{m.burcea,pwong}@liverpool.ac.uk

² Department of Computer Science, National Tsing Hua University, Taiwan
{wkhon,hhliu}@cs.nthu.edu.tw

³ Information Systems Technology and Design, Singapore University of Technology
and Design, Singapore
david_yau@sutd.edu.sg

Abstract. We study an offline scheduling problem arising in demand response management in smart grid. Consumers send in power requests with a flexible set of timeslots during which their requests can be served. For example, a consumer may request the dishwasher to operate for one hour during the periods 8am to 11am or 2pm to 4pm. The grid controller, upon receiving power requests, schedules each request within the specified duration. The electricity cost is measured by a convex function of the load in each timeslot. The objective of the problem is to schedule all requests with the minimum total electricity cost. As a first attempt, we consider a special case in which the power requirement and the duration a request needs service are both unit-size. For this problem, we present a polynomial time offline algorithm that gives an optimal solution and show that the time complexity can be further improved if the given set of timeslots is a contiguous interval.

1 Introduction

We study an offline scheduling problem arising in “demand response management” in smart grid [7, 9, 18]. The electrical smart grid is one of the major challenges in the 21st century [6, 28, 29]. The smart grid uses information and communication technologies in an automated fashion to improve the efficiency and reliability of production and distribution of electricity. Peak demand hours happen only for a short duration, yet makes existing electrical grid less efficient. It has been noted in [4] that in the US power grid, 10% of all generation assets and 25% of distribution infrastructure are required for less than 400 hours per year, roughly 5% of the time [29]. *Demand response management* attempts to overcome this problem by shifting users’ demand to off-peak hours in order to reduce peak load [3, 12, 17, 20, 23, 25]. This is enabled technologically by the advances in smart meters [13] and integrated communication. Research initiatives in the

* Supported by EPSRC Studentship.

area include GridWise [10], the SeeLoadTM system [16], EnviroGridTM [24], peak demand [27], etc.

The smart grid operator and consumers communicate through smart metering devices. We assume that time is divided into integral timeslots. A consumer sends in a power request with the power requirement, required duration of service, and the time intervals that this request can be served (giving some flexibility). For example, a consumer may want the dishwasher to operate for one hour during the periods from 8am to 11am or 2pm to 4pm. The grid operator upon receiving all requests has to schedule them in their respective time intervals using the minimum energy cost. The *load* of the grid at each timeslot is the sum of the power requirements of all requests allocated to that timeslot. The *energy cost* is modeled by a convex function on the load. As a first attempt to the problem, we consider in this paper the case that the power requirement and the duration of service requested are both unit-size, a request can specify several intervals during which the request can be served, and the power cost function is any convex function.

Previous work. Koutsopoulos and Tassiulas [12] has formulated a similar problem to our problem where the cost function is piecewise linear. They show that the problem is NP-hard, and their proof can be adapted to show the NP-hardness of the general problem studied in this paper for which jobs have arbitrary duration or arbitrary power requirement (see elaboration in Section 6). They also presented a fractional solution and some online algorithms. Salinas et al. [25] considered a multi-objective problem to minimize energy consumption cost and maximize some utility. A closely related problem is to manage the load by changing the price of electricity over time, which has been considered in a game theoretic manner [3, 20, 23]. Heuristics have also been developed for demand side management [17]. Other aspects of smart grid have also been considered, e.g., communication [4, 14, 15], security [19]. Reviews of smart grid can be found in [7, 9, 18].

The combinatorial problem we defined in this paper has analogy to the traditional load balancing problem [2] in which the machines are like our timeslots and the jobs are like our power requests. The main difference is that the aim of load balancing is usually to minimize the maximum load of the machines. Another related problem is deadline scheduling with speed scaling [1, 31] in which the cost function is also a convex function, nevertheless a job can be served using varying speed of the processor. Two problems that are more closely related are the minimum cost maximum flow problem [5] with convex functions [21, 26] when we have unit power requirement and unit duration for each job; and the maximum-cardinality minimum-weight matching on a bipartite graph. Yet, existing algorithms for the problem cater for more general input [8, 11, 22, 30]. They are more powerful and have higher time complexity than necessary to solve our problem.

Our contributions. In this paper we study an optimization problem in demand response management in which requests have unit power requirement, unit duration, arbitrary timeslots that the jobs can be served, and the cost

function is a general convex function. We propose a polynomial time offline algorithm that gives an optimal solution. We show that the time complexity of the algorithm is $\mathcal{O}(n^2\tau)$, where n is the number of jobs and τ is the number of timeslots. We further show that if the feasible timeslots for each job to be served forms a contiguous interval, we can improve the time complexity to $\mathcal{O}(n\tau \log n)$.

Technically speaking, we use a notion of “feasible graph” to represent alternative assignments. After scheduling a job, we can look for improvement via this feasible graph. We show that we can maintain optimality each time a job is scheduled. For the analysis, we compare our schedule with an optimal schedule via the notion of “agreement graph”, which captures the difference of our schedule and an optimal schedule. We then show that we can transform our schedule stepwise to improve the agreement with the optimal schedule, without increasing the cost, thus proving the optimality of our algorithm.

Organization of the paper. Section 2 gives the definition of the problem and notions required. Section 3 describes our algorithm and its properties. In Section 4, we prove that our algorithm gives an optimal solution, while in Section 5 we prove its time complexity. We give some concluding remarks in Section 6.

2 Preliminaries

We consider an offline scheduling problem where the input consists of a set of unit-sized jobs $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$. The time is divided into integral timeslots $T = \{1, 2, 3, \dots, \tau\}$ and each job $J_i \in \mathcal{J}$ is associated with a set of feasible timeslots $I_i \subseteq T$, in which it can be scheduled. In this model, each job J_i must be assigned to exactly one feasible timeslot from I_i . The *load* $\ell(t)$ of a timeslot t represents the total number of jobs assigned to the timeslot. We consider a general convex cost function f that measures the cost used in each timeslot t based on the load at t . The total cost used is the sum of cost over time. Over all timeslots this is $\sum_{t \in T} f(\ell(t))$. The objective is to find an assignment of all jobs in \mathcal{J} to feasible timeslots such that the total cost is minimized. We first describe the notions required for discussion.

Feasible graph. Given a particular job assignment A , we define a *feasible graph* G which is a directed multi-graph that shows the potential allocation of each job in alternative assignments. In G each timeslot is represented by a vertex. If job J_i is assigned to timeslot r in A , then for all $w \in I_i \setminus \{r\}$ we add a directed edge (r, w) with J_i as its label.

Legal-path in a feasible graph. A path (t, t') in a feasible graph G is a *legal-path* if and only if the load of the starting point t is at least 2 more than the load of the ending point t' , i.e., $\ell(t) - \ell(t') \geq 2$. Note that if there is a legal-path in the feasible graph G , the corresponding job assignment is not optimal.

Agreement graph. We define an *agreement graph* $G_a(A, A^*)$ which is a directed multi-graph that measures the difference between a job assignment solution A and an optimal assignment A^* . In $G_a(A, A^*)$ each timeslot is represented by a vertex. For each job J_i such that J_i is assigned to different timeslots in A and A^* , we add an arc from t to t' , where t and t' are the timeslots that J_i

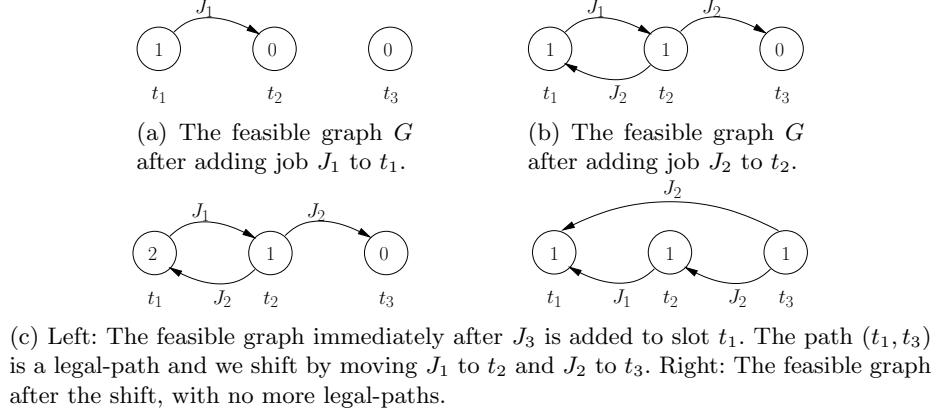


Fig. 1: Let $\mathcal{J} = \{J_1, J_2, J_3\}$, $T = \{t_1, t_2, t_3\}$, $I_1 = \{t_1, t_2\}$, $I_2 = \{t_1, t_2, t_3\}$, and $I_3 = \{t_1\}$. The number inside the vertices denotes their load. Suppose the algorithm schedules the jobs in order of their indices. (a) and (b) Jobs J_1 and J_2 are arbitrarily assigned their feasible minimum load slots. (c) A legal-path and the corresponding shift after assigning J_3 .

is assigned to by A and A^* , respectively. The arc (t, t') is labelled by the tuple $(J_i, +/-=)$. The second value in the tuple is “+” or “-” if moving job J_i from timeslot t to timeslot t' causes the total cost of assignment A to increase or decrease, respectively. The value is “=” if moving the job does not cause any change in the total cost of assignment A .

Observation 1. *By moving J_i from t_1 to t_2 the overall energy cost (i) decreases if $\ell(t_1) > \ell(t_2) + 1$, (ii) remains the same if $\ell(t_1) = \ell(t_2) + 1$, and (iii) increases if $\ell(t_1) < \ell(t_2) + 1$.*

Shifting. By Observation 1, existence of a legal-path implies that the assignment is not optimal and we can execute a *shift* and decrease the total cost of the assignment. Given a legal-path P , a shift moves each job corresponding to an arc e along P from the original assigned timeslot to the timeslot determined by e . More precisely, if the path contains an arc (r, w) with J as its label, then job J is moved from r to w . It is easy to see from Observation 1 that such a shift decreases the cost, implying that the original assignment is not optimal.

On the other hand, when there is no legal-path, it is not as straightforward to show that the assignment is optimal. Nevertheless, we will prove this is the case in Lemma 6.

3 Our Algorithm

The algorithm. We propose a polynomial time offline algorithm that minimizes the total cost (Figure 1 shows an illustration). The algorithm arranges the jobs in \mathcal{J} in arbitrary order, and runs in stages. At any Stage i , we have three steps:

- (1) Assign J_i to a feasible timeslot with minimum load, breaking ties arbitrarily;
- (2) Suppose J_i is assigned to timeslot t . We update the feasible graph G to reflect this assignment in the following way. If applicable, we add arcs from t labelled by J_i to any other feasible timeslots (vertices) of J_i ;
- (3) If there exists any legal-path in G from t to any other vertex t' , the algorithm executes a shift along the legal-path (see Section 2). At the end, the algorithm updates the feasible graph G to reflect this shift.

Invariants. In the next section, we show that the algorithm maintains the following two invariants. At the end of each stage:

- (I1) There is no legal-path in the resulting feasible graph;
- (I2) The assignment is optimal for the jobs considered so far.

Additional notations. To ease the discussion, in the remainder of the paper, we use $\ell'_i(t)$ to represent the load of timeslot t after adding J_i (but before the shift), $\ell_i(t)$ to represent the load of timeslot t at the end of Stage i , and $\ell'_i(s, t)$ and $\ell_i(s, t)$ to represent $\ell'_i(s) - \ell'_i(t)$ and $\ell_i(s) - \ell_i(t)$, respectively.

4 Correctness

Theorem 1. *Our algorithm finds an optimal assignment.*

Framework. Consider any stage. After Step (2), there may be a legal-path in the resulting feasible graph G . In Lemma 1, we show that if a legal-path exists in G after adding J_i to timeslot r , there is at least one legal-path starting from r . Suppose the algorithm chooses the legal-path (r, t) and executes the shift along this path in Step (3). In Lemma 3, we show that if there is no legal-path in the feasible graph G before adding a job, then after adding a job and executing the corresponding shift by the algorithm, the resulting feasible graph has no legal-paths. Therefore, Step (3) of the algorithm needs to be applied only once and there will be no legal-path left, implying that Invariant (I1) holds. In Lemma 6, we show that if there is no legal-path in a feasible graph G , the corresponding assignment is optimal and hence Invariant (I2) holds.

Proof of Invariant (I1)

Lemma 1. *Suppose that before adding job J_i to timeslot r the feasible graph G has no legal-path. If there is any legal-path after adding J_i , there is at least one legal-path starting from r .*

Proof. Assume that there is a legal-path (s, t) after assigning J_i to timeslot r , so that $\ell'_i(s, t) \geq 2$. If $r = s$, we have obtained a desired legal-path. Otherwise, $r \neq s$, there are two cases:

Case 1. G contains an (s, t) path before adding J_i . Since $r \neq s$, $\ell_{i-1}(s) = \ell'_i(s)$ and $\ell_{i-1}(t) \leq \ell'_i(t)$ (the latter inequality comes from the fact that r may be equal to t). This implies $\ell_{i-1}(s, t) \geq \ell'_i(s, t) \geq 2$, which contradicts the precondition that there is no legal-path before adding J_i . Thus, Case 1 cannot occur.

Case 2. G does not contain any (s, t) path before adding J_i . Since (s, t) becomes a legal-path after adding J_i , it must be the case that assigning J_i to timeslot

r adds some new edge (r, w) (with J_i as its label) to G , which connects an existing (s, r) path and an existing (w, t) path. We know that $\ell_{i-1}(s) - \ell_{i-1}(r) \leq 1$ because there is no legal-path before adding J_i . Also, $\ell'_i(s) = \ell_{i-1}(s)$ and $\ell'_i(r) = \ell_{i-1}(r) + 1$ because the new job J_i is assigned to r , with $r \neq s$. Hence, $\ell'_i(r, t) \geq \ell'_i(s, t)$, so that the (r, t) subpath is also a legal-path. \square

Lemma 2. *If before adding a job the feasible graph G does not have a legal-path, then after adding one more job there will be no legal-paths where the load of the starting point is at least 3 more than the load of the ending point. In other words, the load difference corresponding to any new legal-path, if it exists, is exactly 2.*

Lemma 2 will be proved in the full paper and we proceed with Invariant (II).

Lemma 3. *Suppose that G is a feasible graph with no legal-paths. Then after adding a job and executing the corresponding shift by the algorithm, the resulting feasible graph has no legal-paths.*

Proof. Suppose that there were no legal-paths in G after Stage $i-1$, but there is a new legal-path in G after assigning J_i . By Lemma 1, there must be one such legal-path (s, t) where s is the timeslot assigned to J_i , and without loss of generality, let it be the one that is selected by our algorithm to perform the corresponding shift. Let the ordering of the vertices in the path be $[s, v_1, v_2, \dots, v_k, t]$, and P denote the set of these vertices.

We define $In(r)$ to be the set of vertices w such that a (w, r) path exists before adding J_i , and $Out(r)$ to be the set of vertices w such that an (r, w) path exists before adding J_i . We assume that $r \in In(r)$ and $r \in Out(r)$ for the ease of later discussion. Similarly, we define $In''(r)$ to be the set of vertices w such that a (w, r) path exists after shifting, and we define $Out''(r)$ analogously. Given a set R of vertices, let $IN(R) = \bigcup_{r \in R} In(r)$ and $OUT(R) = \bigcup_{r \in R} Out(r)$. The notation $IN''(R)$ and $OUT''(R)$ are defined analogously.

Briefly speaking, we upper bound the load of a vertex in $IN''(P)$, and lower bound the load of a vertex in $OUT''(P)$, as any legal-path that may exist after the shift must start from a vertex in $IN''(P)$ and end at a vertex in $OUT''(P)$. Based on the bounds, we shall argue that there are no legal-paths as the load difference of any path after the shift will be at most 1. Note that after the shift, only the load of t is increased by one, whereas the load of any other vertex remains unchanged. Now, concerning the legal-path (s, t) , there are two cases:

Case 1. There was an arc from s to v_1 in the feasible graph G before adding J_i . In this case, it is easy to check that $IN''(P) \subseteq IN(P)$,[§] and $OUT''(P) \subseteq OUT(P) \cup OUT(I_i)$.[‡]

[§] Otherwise, let z be a vertex in $IN''(P)$ but not in $IN(P)$. Take the shortest path from z to some vertex in P after the shift. Then all the intermediate vertices of such a path are not from P . However, the jobs assigned to those intermediate vertices are unchanged, so that such a path also exists before the shift, and z is in $IN(P)$. A contradiction occurs.

[‡] Otherwise, let z be a vertex in $OUT''(P)$ but not in $OUT(P) \cup OUT(I_i)$. Take the shortest path that goes to z starting from some vertex in P after the shift. Then

Suppose that $\ell_{i-1}(s) = x$. Then, $\ell_{i-1}(t) = x-1$ because there is no legal-path before adding J_i but there is one after adding J_i . This implies $\ell_{i-1}(v_h) \leq x$ for any $h \in [1, k]$, or there was a legal-path (v_h, t) before adding J_i . The load of any vertex in $IN(P)$ is at most x or there was a legal-path entering t before adding J_i . The load of any vertex in $OUT(P)$ is at least $x-1$ or there was a legal-path leaving s before adding J_i . For any vertex r in I_i , $\ell_{i-1}(r) \geq x$, since $s \in I_i$ has the minimum load. This implies that the load for any vertex in $OUT(I_i)$ is at least $x-1$, or there was a legal-path leaving a vertex in I_i before adding J_i . Thus, after the shift, the load of any vertex in $IN''(P)$ is at most x , and the load of any vertex in $OUT''(P)$ is at least $x-1$, so no legal-paths will exist.

Case 2. There were no arcs from s to v_1 in the feasible graph G before adding J_i . In this case, J_i must be involved in the shift, so that the jobs assigned to s after the shift will be the same as if J_i was not added. Consequently, if there is still a legal-path after the shift, the starting vertex must be from $IN''(P \setminus \{s\})$, while the ending vertex must be from $OUT''(P \setminus \{s\})$. Similar to Case 1, it is easy to check that $IN''(P \setminus \{s\}) \subseteq IN(P \setminus \{s\})$ and $OUT''(P \setminus \{s\}) \subseteq OUT(P \setminus \{s\}) \cup OUT(I_i)$. Suppose that $\ell_{i-1}(s) = x$, so that $\ell'_i(s) = x+1$. Because adding J_i creates a new legal-path (s, t) , by Lemma 2, $\ell'_i(t) = \ell_{i-1}(t) = x-1$. Thus, the load of any vertex in $IN(P \setminus \{s\})$ is at most x , since there was no legal-path entering t before adding J_i . On the other hand, $\ell_{i-1}(v_1) \geq x$ otherwise job J_i would be assigned to v_1 . However, $\ell_{i-1}(v_1) \leq x$ or there is a legal-path (v_1, t) . Hence, $\ell_{i-1}(v_1) = x$. This implies that the load of any vertex in $OUT(P \setminus \{s\})$ is at least $x-1$, since there was no legal-path leaving v_1 before adding J_i . As for the vertices in $OUT(I_i)$, we can use a similar argument as in Case 1 to show that their load is at least $x-1$. Thus, after the shift, the load of any vertex in $IN''(P \setminus \{s\})$ is at most x , and the load of any vertex in $OUT''(P \setminus \{s\})$ is at least $x-1$, so no legal-path will exist. \square

Proof of Invariant (I2)

We now prove in Lemma 6 (the other key lemma for the correctness) that non-existence of legal-paths implies the assignment is optimal. The rough ideas are as follows. Consider an optimal assignment A^* (satisfying some constraints as to be defined). In Lemma 5, we show that there is a sequence of agreement graphs $G_a(A_1, A^*), G_a(A_2, A^*), \dots, G_a(A_k, A^*)$ where the cost is non-increasing every step, A_1 is the original assignment of jobs given by our algorithm, and A_k is an optimal assignment. We prove Lemma 6 by contradiction, assuming there is no legal-path in the feasible graph G but the assignment A is not optimal. We then consider the sequence of agreement graphs given in Lemma 5 and show that either there is no agreement graph in the sequence involving strict decrease of overall cost (which means A is already optimal) or that there is a legal-path in the feasible graph G , leading to a contradiction.

Note that Lemma 5 considers an optimal assignment A^* such that $G_a(A, A^*)$ is acyclic. The existence of such A^* is stated here and proved in the full paper.

all the intermediate vertices of such a path are not from P . If such a path does not involve vertices from I_i , then this path must exist before the shift, so that z is in $OUT(P)$. Else, z is in $OUT(I_i)$. A contradiction occurs.

Lemma 4. *There exists an optimal assignment A^* such that $G_a(A, A^*)$ is acyclic.*

Lemma 5. *Suppose A is not optimal and A^* is an optimal assignment such that $G_a(A, A^*)$ is acyclic. Then we can have a sequence of agreement graphs $G_a(A_1, A^*), G_a(A_2, A^*), \dots, G_a(A_k, A^*)$ such that $A_1 = A$, $A_k = A^*$, and the cost is non-increasing every step.*

Proof. Consider the agreement graph $G_a(A_i, A^*)$, for $i \geq 1$, starting from $A_1 = A$. In each step, from $G_a(A_i, A^*)$ to $G_a(A_{i+1}, A^*)$, one arc is removed. For $i \geq 1$, we consider in $G_a(A_i, A^*)$ any arc labelled with either a “-” or an “=” and we execute the move corresponding to this arc. Through this move, we remove one arc, and thus we do not introduce any new arcs. However, the $+/-/$ label of other arcs may change. If the resulting graph $G_a(A_{i+1}, A^*)$ does not contain any more “-” or “=” arcs, we stop. Otherwise, we repeat the process.

Note that the cost is non-increasing in every step. By the time we stop, if the resulting graph, say, $G_a(A_h, A^*)$, does not contain any more arcs, we have obtained the desired sequence of agreement graphs. Otherwise, we are left only with “+” labelled arcs in $G_a(A_h, A^*)$; however, in the following, we shall show that such a case cannot happen, thus completing the proof of the lemma.

Firstly, $\text{cost}(A_h) \geq \text{cost}(A^*)$ since A^* is an optimal assignment. Next, by Lemma 4, $G_a(A_1, A^*)$ is acyclic and the resulting graph $G_a(A_h, A^*)$ by removing all “-” and “=” labelled arcs is also acyclic. Thus, in $G_a(A_h, A^*)$, there must exist at least one vertex with in-degree 0 and one vertex with out-degree 0. We look at all such (v_1, v_i) paths in $G_a(A_h, A^*)$, where v_1 has in-degree 0 and v_i has out-degree 0. For any such (v_1, v_i) path, we show that by executing all moves of the path (i) the overall cost is increasing, and (ii) the labels of all arcs not contained in the (v_1, v_i) path remain “+”. After executing all moves of the path, all arcs of the (v_1, v_i) path are removed.

(i) Suppose the vertices of the path are $[v_1, v_2, \dots, v_i]$ and $\ell(v_1) = x$. As all arcs in (v_1, v_i) are labelled with “+” (i.e., the cost is increasing), $\ell(v_j) \geq x$, for $j > 1$. By executing all moves in the path, $\ell(v_1) = x - 1$, $\ell(v_j)$ is unchanged, for $1 < j < i$, and $\ell(v_i)$ is increased by one. Thus, the overall cost is increasing.

(ii) We show that the labels of all arcs not contained in the (v_1, v_i) path remain “+”. There may be out-going arcs from v_1 to other vertices not in the (v_1, v_i) path initially labelled by “+”. Before executing all the moves in the (v_1, v_i) path, the load of all other vertices is at least x as we assume $\ell(v_1) = x$. After the move, $\ell(v_1) = x - 1$ and out-going arcs from v_1 point to vertices with load at least x . Thus, an arc from v_1 to any other vertex denotes a further increase in the cost and the labels of the arcs do not change. For vertices v_j , for $1 < j < i$, the load of v_j remains unchanged and thus the labels of the arcs incoming to or outgoing from v_j remain the same. For v_i , there may be incoming arcs. Suppose $\ell(v_i) = y$ before executing all the moves in the (v_1, v_i) path. Then the load of all other vertices pointing to v_i is at most y and the arcs are labelled by “+”. After executing all the moves in the (v_1, v_i) path, $\ell(v_i) = y + 1$, and thus any subsequent moves from vertices pointing to v_i cause further increases in the cost, i.e., the labels do not change.

Thus, the overall cost is increasing. We repeat this process until there are no more such (v_1, v_i) paths. We end up with $\text{cost}(A_k) > \text{cost}(A^*)$, which contradicts the fact that $\text{cost}(A_k) = \text{cost}(A^*)$ as $A_k = A^*$. Thus, the case where we are left only with “+” labelled arcs in $G_a(A_h, A^*)$ cannot happen, and the lemma follows. \square

Lemma 6. *If there is no legal-path in the feasible graph G , the corresponding assignment is optimal.*

Proof. Suppose by contradiction there is no legal-path in the feasible graph G , but the corresponding assignment A is not optimal. Let $A^*, A_1 = A, A_2, \dots, A_k = A^*$ be the assignments as defined in Lemma 5. Note that each arc in the agreement graph $G_a(A_1, A^*)$ corresponds to an arc in the feasible graph G (since G captures all possible moves). Because the sequence of agreement graphs in Lemma 5 only involves removing arcs, each arc in all of $G_a(A_i, A^*)$ corresponds to an arc in G .

Suppose $G_a(A_j, A^*)$ is the first agreement graph in which a “–” labelled arc is considered between some timeslots t_a and t_b . If there is no such arc, then A is already an optimal solution (since the sequence will be both non-increasing by Lemma 5 and non-decreasing as no “–” labelled arc is involved). Otherwise, if there is such an arc in $G_a(A_j, A^*)$, we show that there must have existed a legal-path in the feasible graph G , leading to a contradiction. We denote by $\ell(A_i, t)$ the load of timeslot t in the agreement graph $G_a(A_i, A^*)$. Suppose $\ell(A_j, t_a) = x$, then $\ell(A_j, t_b) \leq x - 2$ as the overall energy cost would be decreasing by moving a job from t_a to t_b . If $\ell(A_1, t_a) = x$ and $\ell(A_1, t_b) \leq x - 2$ in the original assignment, then there is a legal-path in G , which is a contradiction. Otherwise, we claim that there are some timeslots u_{i_y} and v_{k_z} such that $\ell(A_1, u_{i_y}) \geq x$ and $\ell(A_1, v_{k_z}) \leq x - 2$, and there is a path from u_{i_y} to v_{k_z} in G . This forms a legal-path in G , leading to a contradiction.

To prove the claim, we first consider finding u_{i_y} . We first set $i_0 = j$ and $u_{i_0} = t_a$. If $\ell(A_1, u_{i_0}) \geq x$, we are done. Else, since $\ell(A_j, u_{i_0}) = x$ and $\ell(A_1, u_{i_0}) < x$, there must be some job that is moved to u_{i_0} before A_j . Let $i_1 < i_0$ be the latest step such that a job is added to u_{i_0} and the job is moved from u_{i_1} . Note that since this move corresponds to an arc with label “=”, $\ell(A_{i_1}, u_{i_1}) = x$ and $\ell(A_{i_1}, u_{i_0}) = x - 1$. If $\ell(A_1, u_{i_1}) \geq x$, we are done. Otherwise, we can repeat the above argument to find u_{i_2} and so on. The process must stop at some step $i_y < i_0$ where $\ell(A_1, u_{i_y}) \geq x$. Similarly, we set $k_0 = j$ and $v_{k_0} = t_b$, so that we can find a step $k_z < k_0$ such that $\ell(A_1, v_{k_z}) \leq x - 2$. Recall that since each arc in $G_a(A_1, A^*)$ corresponds to an arc in the feasible graph G and in all subsequent agreement graphs we only remove arcs, there is a path from u_{i_y} to v_{k_z} in G . Therefore, we have found a legal-path from u_{i_y} to v_{k_z} in G . \square

5 Time Complexity

We prove the time complexity of our algorithm in Theorem 2 and show that this can be improved for the case where the feasible timeslots associated with each job are contiguous.

Theorem 2. *We can find the optimal schedule in $\mathcal{O}(n^2\tau)$ time.*

Proof. We add jobs one by one. Each round when we assign the job J_i to timeslot t , we add arcs (t, w) labelled by J_i for all vertices w that $w \in I_i$ in the feasible graph. By Lemma 1, there is a legal-path starting from t if there is a legal-path after assigning J_i to timeslot t . When J_i is assigned to t , we start breadth-first search (BFS) at t . By Lemma 2, if there is a node w which can be reached by the search and the number of jobs assigned to w is two less than the number of jobs assigned to t , it means that there is a legal-path (t, w) . Then we shift the jobs according to the (t, w) legal-path. After shifting there will be no legal-paths anymore by Lemma 3. Finally we update the edges of the vertices on the legal-path in the feasible graph.

Adding J_i to the feasible graph needs $\mathcal{O}(|I_i|)$ time. Because $|I_i|$ is at most the total number of timeslots in T , $|I_i| = \mathcal{O}(\tau)$ where τ is the number of timeslots. The BFS takes $\mathcal{O}(\tau + n\tau)$ time because there are at most $n\tau$ edges in the feasible graph. If a legal-path exists after adding J_i and its length is l , the shifting needs $\mathcal{O}(l)$ time, which is $\mathcal{O}(\tau)$ because there are at most τ vertices in the legal-path. After the shift, at most $n\tau$ edges are updated in the feasible graph, taking $\mathcal{O}(n\tau)$ time. The total time for adding n jobs is thus bounded by $\mathcal{O}(n^2\tau)$. \square

We now consider the special case where each job $J_i \in \mathcal{J}$ is associated with an interval of contiguous timeslots $I_i = [\rho_i, \delta_i]$, for positive integers $\rho_i \leq \delta_i$, and each job J_i must be assigned to exactly one feasible timeslot s_i , for $\rho_i \leq s_i \leq \delta_i$. We give a sketch here, while the full proof can be found in the full paper.

Theorem 3. *We can find the optimal schedule in $\mathcal{O}(n\tau \log n)$ time for the case where the feasible timeslots associated with each job are contiguous.*

Proof (Sketch). For the special case, we use data structure techniques for the speed up. For each timeslot $t_i \in T$, we use two balanced binary search trees that contain the feasible intervals for all jobs assigned to t_i . For each job J_j with $I_j = [\rho_j, \delta_j]$ assigned to t_i , the first binary tree keeps the value of ρ_j , while the second binary tree keeps the value of δ_j . The binary trees are updated whenever a job is moved to and from t_i accordingly, and each such update takes $\mathcal{O}(\log n)$ time. We can query a minimum and a maximum value of the two trees, respectively, in order to establish the *directly reachable interval* of timeslot t_i , i.e., the other timeslots that jobs from t_i can be moved to. Because of the contiguous property of the feasible intervals, the set of timeslots is contiguous. We denote this interval of timeslots by $[\alpha_i, \beta_i]$ and we have that $\alpha_i \leq t_i \leq \beta_i$.

We further find the set of the ending vertices of all the paths of length at most $\tau - 1$ that start from t_i , which we call *reachable interval*. Note that the ending vertices of paths of length 2 from t_i can be found by checking the binary search trees of each timeslot in $[\alpha_i, \beta_i]$, which can then be used to find vertices at distance 3 from t_i and so on. Finding the reachable interval requires $\mathcal{O}(\tau)$ time. We can then identify any legal path in $\mathcal{O}(\tau)$ time.

In summary, adding a job to the feasible graph takes $\mathcal{O}(\log n)$ time. Finding the reachable interval and legal path takes $\mathcal{O}(\tau)$ time. Shifting of jobs along the

legal path found takes $\mathcal{O}(\tau \log n)$ time. Thus the time taken to add one job is bounded by $\mathcal{O}(\tau \log n)$. The overall time for adding all n jobs is thus bounded by $\mathcal{O}(n\tau \log n)$. \square

6 Conclusion

In this paper we study an offline scheduling problem arising in demand response management in smart grid. We focus on the particular case where requests have unit power requirement and unit duration. We give a polynomial time offline algorithm that gives an optimal solution. Natural generalization extends to arbitrary power requirement and arbitrary duration. The problem where requests have unit power requirement and arbitrary duration has been shown to be NP-hard [12] by a reduction from the bin packing problem. Using a similar idea, it can be shown that the problem where requests have arbitrary power requirement and unit duration is also NP-hard. An obvious research direction is to develop approximation algorithms for the general problem. It would be also interesting to consider online algorithms for the problem.

Acknowledgement: We would like to thank the reviewers for very helpful comments leading to improvement in the time complexities of our algorithms.

References

1. S. Albers. Energy-efficient algorithms. *Communication ACM*, 53(5):86–96, 2010.
2. Y. Azar. On-line load balancing. In A. Fiat and G. J. Woeginger, editors, *Online Algorithms*, volume 1442 of *LNCS*, pages 178–195. Springer, 1998.
3. S. Caron and G. Kesidis. Incentive-based energy consumption scheduling algorithms for the smart grid. In *IEEE Smart Grid Comm.*, pages 391–396, 2010.
4. C. Chen, K. G. Nagananda, G. Xiong, S. Kishore, and L. V. Snyder. A communication-based appliance scheduling scheme for consumer-premise energy management systems. *IEEE Trans. Smart Grid*, 4(1):56–65, 2013.
5. J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, Apr. 1972.
6. European Commission. European smartgrids technology platform. ftp://ftp.cordis.europa.eu/pub/fp7/energy/docs/smartgrids_en.pdf, 2006.
7. K. Hamilton and N. Gulhar. Taking demand response to the next level. *Power and Energy Magazine, IEEE*, 8(3):60–65, 2010.
8. D. S. Hochbaum and J. G. Shanthikumar. Convex separable optimization is not much harder than linear optimization. *J. ACM*, 37(4):843–862, Oct. 1990.
9. A. Ipakchi and F. Albuyeh. Grid of the future. *IEEE Power and Energy Magazine*, 7(2):52–62, 2009.
10. L. D. Kannberg, D. P. Chassin, J. G. DeSteese, S. G. Hauser, M. C. Kintner-Meyer, R. G. Pratt, L. A. Schienbein, and W. M. Warwick. GridWiseTM: The benefits of a transformed energy system. *CoRR*, nlin/0409035, Sept. 2004.
11. A. V. Karzanov and S. T. McCormick. Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM J. Comput.*, 26(4):1245–1275, Aug. 1997.

12. I. Koutsopoulos and L. Tassiulas. Control and optimization meet the smart power grid: Scheduling of power demands for optimal energy management. In *Proc. e-Energy*, pages 41–50, 2011.
13. R. Krishnan. Meters of tomorrow [in my view]. *IEEE Power and Energy Magazine*, 6(2):96–94, 2008.
14. H. Li and R. C. Qiu. Need-based communication for smart grid: When to inquire power price? *CoRR*, abs/1003.2138, 2010.
15. Z. Li and Q. Liang. Performance analysis of multiuser selection scheme in dynamic home area networks for smart grid communications. *IEEE Trans. Smart Grid*, 4(1):13–20, 2013.
16. Lockheed Martin. SEELoad™Solution. <http://www.lockheedmartin.co.uk/us/products/energy-solutions/seesuite/seeload.html>.
17. T. Logenthiran, D. Srinivasan, and T. Z. Shun. Demand side management in smart grid using heuristic optimization. *IEEE Trans. Smart Grid*, 3(3):1244–1252, 2012.
18. T. Lui, W. Stirling, and H. Marcy. Get smart. *IEEE Power and Energy Magazine*, 8(3):66–78, 2010.
19. C. Y. T. Ma, D. K. Y. Yau, and N. S. V. Rao. Scalable solutions of markov games for smart-grid infrastructure protection. *IEEE Trans. Smart Grid*, 4(1):47–55, 2013.
20. S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Basar. Dependable demand response management in the smart grid: A stackelberg game approach. *IEEE Trans. Smart Grid*, 4(1):120–132, 2013.
21. M. Minoux. A polynomial algorithm for minimum quadratic cost flow problems. *European Journal of Operational Research*, 18(3):377 – 387, 1984.
22. M. Minoux. Solving integer minimum cost flows with separable convex cost objective polynomially. In G. Gallo and C. Sandi, editors, *Netflow at Pisa*, volume 26 of *Mathematical Programming Studies*, pages 237–239. Springer, 1986.
23. A.-H. Mohsenian-Rad, V. 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.
24. REGEN Energy Inc. ENVIROGRID™SMART GRID BUNDLE. <http://www.regenenergy.com/press/announcing-the-envirogrid-smart-grid-bundle/>.
25. S. Salinas, M. Li, and P. Li. Multi-objective optimal energy consumption scheduling in smart grids. *IEEE Trans. Smart Grid*, 4(1):341–348, 2013.
26. P. T. Sokkalingam, R. K. Ahuja, and J. B. Orlin. New polynomial-time cycle-canceling algorithms for minimum-cost flows. *Networks*, 36(1):53–63, 2000.
27. Toronto Hydro Corporation. Peaksaver Program. http://www.peaksaver.com/peaksaver_THESL.html.
28. UK Department of Energy & Climate Change. Smart grid: A more energy-efficient electricity supply for the UK. <https://www.gov.uk/smart-grid-a-more-energy-efficient-electricity-supply-for-the-uk>, 2013.
29. US Department of Energy. The Smart Grid: An Introduction. <http://www.ee.energy.gov/SmartGridIntroduction.htm>, 2009.
30. L. A. Végh. Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. In *Proceedings of the 44th symposium on Theory of Computing*, STOC ’12, pages 27–40, New York, NY, USA, 2012. ACM.
31. F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced CPU energy. In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 374–382, 1995.

Optimized Job Scheduling Approach based on Genetic Algorithms in Smart Grid Environment

Firas Albalas, Wail Mardini and Dua'a Bani-salameh

Department of Computer Science, Jordan University of Science and Technology

Abstract: The advances in communications and information technologies have been playing a major role in all aspects of our lives. One of those majors' aspects that affect our daily lives is the power grids which lead to what we call Smart Grids. One of the major challenges in these grids is to optimize the consumption and resources. This paper presents an optimized job scheduling approach using genetic algorithm which provides a minimum cost for completing different tasks in a grid environment. In grid environment different independent appliances are sharing the same resources depending on the availability of resources and the need of these appliances to run. There are different job scheduling approached starting from typical strategies, Ant Colony (AC) and Genetic Algorithm (GA). In this paper we present a cost optimized Genetic Algorithm approach for appliances job scheduling by considering different parameters like job duration time, the resources availability and the job priority to start. The proposed approach is tested using a simulator written in c++ programming language. The results show that the total saving in cost is better than the previous approaches.

Keywords: Smart Grid, Job Scheduling, Genetic Algorithms, Sensor networks.

1. Introduction

Grid Computing technologies are considered as a new evolution in distributed heterogeneous systems [5]. It focuses on sharing and utilization of the available resources to different applications. It is used to solve complex problems such as scientific, engineering and business. Resources in grids are dynamic and the applications (appliances) can run at any time. Hence, the job scheduling is an important issue in grids. Due the nature of appliances connected to the grid, traditional task (job) scheduling is a time consuming, while the introduction of genetic algorithm to take the task of scheduling will shorten the task scheduling time and the cost and improvement of the grid performance [6].

Using electrical devices that consume electricity to run their operation, such as Dryer, Dishwasher, Air conditioner, Coffee maker embedded with sensors. These electrical devices have a diversity of power consumption amount and even the same device with different programs have different power consumption and if scheduled according to time of starting in, it will reduce the final cost. Each device is attached to a wireless sensor to gather information share it with the grid, these sensors should be deployed carefully in order not to impair the lifetime[16] of the grids which is represented by grids.

Each device has its own duration time and amount of energy consumption level for each run. Examples of appliances and the amount of power need to run in a specified duration time are shown in table 1[10].

According to [12] rechargeable battery (PHEV) can be

considered as an appliance because it needs to be charged when empty.

Table 1. Energy consumption and cycle durations of the appliances.[10]

Appliance	Energy consumption (kWh)	Duration (min)
Washer	0.89	30
Dishwasher	1.19	90
Dryer	2.46	60
Coffee Maker	0.4	10
PHEV	9.9	60
AC	1.5	60

Different appliances with different power consumption and different priority to run are the main components of grids. These appliances need to run in an optimized manner to make sure all appliances got the required power to run with the minimum cost.

In Grid environment jobs are scheduled by using traditional scheduling algorithms to achieve the goal of cost minimizing, some traditional scheduling algorithms are: First Come First Served (FCFS) and Shortest Job First (SJF) which will not perform well with grid environment. Modern algorithms are introduced to perform job scheduling in a professional way; it is Genetic algorithms.

Genetic algorithm is technique that used in computing field for propose search to find the best solution that are known as exact or approximate solution to approve the optimization and search problems. The genetic algorithms are categorized as global search heuristics, the operation model based on biological evaluation such as selection, crossover, and mutation.

The genetic algorithms approach is usually consisting of the following steps:

- Creation an initial population
- Computing the fitness of each individual
- While (not stopping condition) do
 - o Select parents from population.
 - o Execute crossover to produce offspring.
 - o Perform mutations.
 - o Compute fitness of each individual.
 - o Replace the parents by the corresponding offspring in new generation.
- Repeat

2. Related work

[2] have proposed system to monitor and control an office environment that connected with smart grid to schedule the tasks according to policies defined by the user.

They applied and tested the system in a living lab environment. The results showed show an interesting economic savings of an average of about 35%. And another interesting result is the power saving is about 10% to 20% in building that also contains renewable generation plants.

The authors in [12] proposed frame work for scheduling residential energy consumption. Authors claim that this proposed framework helps to decrease the cost of energy bill and minimize the waiting time to operate each device. The period that taken by authors to study the consumption is between September 1st 2009 to December 31st 2009 (122 days approximately four months). They used in their experiments variable number of electrical devices for each day from 10 to 25, which can be categorized into two parts: fixed consumption devices like lighting and electric stove the other part is varying consumption energy such as dishwasher and PHEV.

The results showed reducing the user cost with reduction of the peak time to average ratio in load demand.

The authors in [1] studied an offline scheduling task that depends on the user to send a request to run his appliance during a time slot and the proposed approach will find the best and the minimum cost for his request. The objective of this problem it is able to schedule all requests with minimum total electricity cost. For this reason, they proposed polynomial time offline algorithm in order to achieve the optimal solution and it is able to optimize the time complexity to $O(nT \log n)$ where the time complexity before the optimization is $O(n^2T)$.

In [13] the authors proposed new technique to group a set of jobs that have common features and then schedule these jobs by using an enhanced genetic algorithm, the group process depends on some features that such as mobility, resource availability and job completion time. The results show a performance improvement in job scheduling and minimize the job completion time

In [15] the authors developed new efficient method to get the fairness requirement for any operating system. They proposed a Distributed Weighted Round-Robin approach that based on genetic algorithms to distribute the tasks for CPU in a multiprocessor computer in a fair way. The scheduling complexity problem dependents on the number of processors (p), task processing time (T_i) and precedence constraints.

In [17] the authors mentioned that Genetic algorithms can be used for searching good solutions for different kind of problems, although this paper focus on find good solution for tailbiting codes, they give good indication about Genetic algorithms and their use in searching optimal solutions.

The experiments are applied on two computers: the first is sun ultra on 140 MHz with 64 MB RAMS and the second is Alpha station 600 on 333MHz and 256 MB RAM.

The Test problem consist with two problems are problem with 452 tasks which scheduled onto 20 processor and problem with 473 tasks which scheduled using 4 processors. They claimed that proposed approach shows achieve accurate proportional fairness and high performance for a diverse set of workloads.

In [8] the authors investigated the genetic algorithm (GA) in order to optimize sensor node's energy consumption. They used multi-objective algorithm that establish optimal number of sensor clusters with cluster heads and decreasing the cost of transmission.

The parameters that used to compute the performance of GA are affected through number of factors such as: the population size, the probability of mutation and crossover, and the method of replacement. Tables 1 and 2 indicate the WSN parameters and final GA parameters, respectively.

Table 2. WSN and GA parameters used in [6]

WSN PARAMATERS	
Parameter	Value
Node distribution	30*30 m ²
Initial population size	4
Bit for representing every sensor nodes	3
Battery capacity	(0,15)
Location of sink node	(0,0)
Number of sensor nodes	100, 200, 400, 600
GA PARAMATERS	
Parameter	Value
Mutation rate	0.004
Crossover rate	0.7
Max Generations	1000

They used MATLAB simulator to implement number of experiments with different values of these parameters. The experimental results showed that their approach can generate optimal clusters and minimizes the cost of transmission.

In power grid, generation capacity is considered as important topic to meet the peak hour load demand. The smart grid has been proposed by update the traditional grid to achieve new type of electricity grid with more reliability economically and sustainability electricity service.

The load demand through on-peak hours is much larger than the load demand through off-peak hours, so when the load demand is high, the power consumption cost will be high.

The previous work it was achieved one objective of this problem with minimize customer's cost. After that the authors in [Melike E. and Hussein M. 2011] resolved the load scheduling problem with multi-objective optimization problem (CMOP).

And they evaluated the performance of two algorithms are: load scheduling with EA (LSEA) and load scheduling with E-approximate (LSEA), that are implemented by using Matlab 2011b software on computer machine with certain specification , the capacity of CPU is 3.4 GHZ and 4 GB of the RAM memory.

The results showed the objectives that optimized like minimize the energy consumption cost and maximize the certain utility.

3. System Description

This paper considers a smart grid that connects a set of inelastic appliances power supply. The power supply in this system is of two types the ordinary power supply that came from Generation Company, and a battery bank at the consumers' end for energy storage. Because of the continuous charging and discharging of battery bank the operating life will be affected. To overcome this limitation a

controller can be used to control and regulate the charging process in the battery bank. Also an inverter is used to convert the DC in battery to an AC before supplying the appliances. A detailed description of system component is discussed in the following subsections.

3.1 Grid power supply

This power supply is the traditional power supply comes through wires to be used by consumers. This power supply cost is calculated by a smart meter as a part of the grid, and this power supply has two main prices; peak hours and off-peak hours.

3.2 Battery Bank

The battery bank is used in the consumer side to store power during the off – peak time and this power is used by the appliance as a priority at the peak time. We assumed that this power supply is fully charged at the first time used and the price of this power supply is added to the cost of the consumer power cost with an assumption of 10 years life of battery.

Most of the appliances met the amount of power in this power supply and as mentioned this power supply will be recharged at off-peak time with low price power supply, the discharge amount is limited to maximum of 2/3 of the power stored in this power supply.

3.3 Energy Demand

This proposed approach considers the power demand to at a particular time instance with a particular amount. If the demand is at off – peak hours it will be supplied by the grid on the other hand if the demand is at peak hours the priority to supply is for battery if the amount is partially or totally covers the demand otherwise the grid will cover the remaining demand or the whole demand depending on the battery status.

This optimization approach runs for 24 hours power demand. The main goal of this approach is to minimize the total cost so at any time the demand is requested the power supplies should give the demand requests either by battery or the grid power supply as discussed before.

4. Proposed Optimization Algorithm

Evolutionary Algorithms (EA) are a group of optimization and global search procedure which use the main three principles of natural evolution: natural selection, reproduction and diversity of the species [4]. Genetic Algorithm (GA) is part of Evolutionary algorithms which use the probabilistic rule translation, i.e stochastic algorithms.

This paper uses Genetic algorithms [14] to minimize the total cost of running a set of appliance in a home by creating the best order string (child) of running appliances.

In our work we consider the power cost as a real-time task scheduling problem where each task has a starting time (release time), duration time and end time. The task will start at starting time and take a duration time running and finishes at stop time, during this process the cost is calculated and added to the total cost for all appliances.

As finding the optimal solution in our proposed approach is an NP –hard because you need to find all possible solutions

and find the best. Thus in our proposed approach we cut the searching space by using the Evolutionary Algorithms (EA) especially the Genetic Algorithms (GA); discussed before; a global probabilistic search method based on natural selection and the population genetics, is used in [3].

The main goal of our job scheduling approach is to minimize the power cost after running the appliances in a period of time; in general the use of Genetic Algorithms (GA) is done in five steps.

1. Initialization

An initial population is randomly generated; the contents of each generated string should be unique. Each string is generated depending on some variables such as completion time, priority of execution and delay of starting.

2. Fitness functions(Evaluation)

At this step each string initialized in the previous step is evaluated. As the objective of our approach is to minimize the cost depending on the order of appliances run and completion time we used fitness function of time developed and used in [6].

3. Selection

At this step a selection operation is executed to select the higher two strings to be the parents for generating the new string (child), for selection the traditional roulette wheel selection is used to select the best strings(parents).

4. Crossover

At this step two parts of each string are exchanged to form new and hopefully best parents. Our proposed approach uses Blend crossover (BLX)[7]. Figure 1 shows an example of crossover operation.

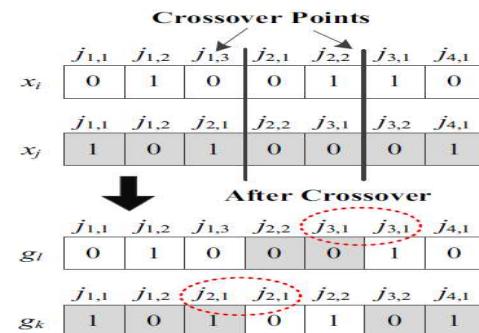


Figure 1.: Crossover Selection Example [6]

5. Mutation

This step is used to prevent the parent selection from randomness and local minimum.

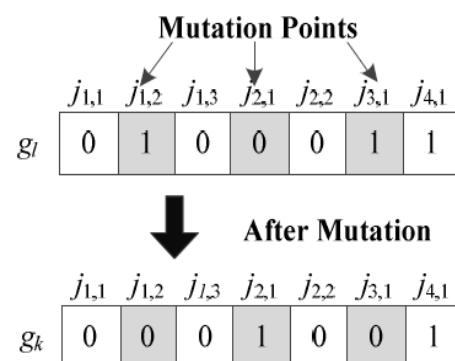


Figure 2. Mutation Method Example [6]

It works as a background step to ensure the diversity of genetic population. There are various types of mutation; Flipping, Interchanging, Reversing, Uniform mutation and non-uniform mutation. This paper uses non-uniform mutation used in [4]. Figure 2 shows an example of mutation operation.

The proposed work starts by collecting the events (jobs) from appliances through sensors to generate the initial populations then these populations are used by genetic algorithms and steps discussed before to find the best and optimized order of events to be run depending of a set of constraints discussed before. The new child generated from initial parents is tested and evaluated in the next section.

5. Experimental Results and analysis

In this section the performance of the proposed scheme is discussed by running different experiments. The main performance metric used to evaluate the proposed scheme versus the other schemes is the total cost in Jordan Dinar (JD) for running the home appliances over different periods of time.

Table 3 summarizes the parameters used during the simulation study. Our simulator is written using C++ programming language running under Ubuntu operating system.

Table 3. WSN and GA parameters used

Parameter	Value
Simulation Time	20 – 210 days
Number of devices	4 , 6 devices
Power Prices (Jordan)	On peak 0.62 \$/Kwh
	Off Peak 0.52 \$/Kwh

In this section we will discuss the results for the proposed scheduling scheme against the schemes proposed in[8][Firas Al Balas et al. 2016]

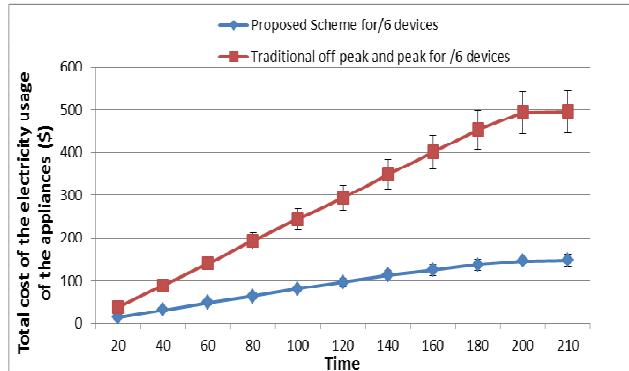


Figure 3. The total cost of the electricity consumed for 6 devices for proposed scheme and traditional power consumption

The comparison is done by running the proposed scheme on different scenarios including changing the number of appliances and also make some appliances repeat its power request more than once a day. The scenarios cover the average and high demand on power supply at homes to make

sure that the proposed scheme is able to run in high power demands.

Figure 3 show the result of the first experiment which include the comparison between the proposed scheme and results from [11]. This scenario runs using 6 devices running during off-peak and peak. The results show that there is a large saving in cost especially when run for a long time. This cost reduction is due to using genetic algorithms for scheduling by finding the best order of appliance running period that avoids peak time while trying to get benefit from battery bank during the peak hours.

Figure 4 show the result of the second experiment which include the comparison between the proposed scheme and results from [18]. In this scenario the 6 devices run during off-peak and peak hours with the support of Solar Photovoltaic (PV) Batteries in comparison with the proposed scheme. The results show that the proposed scheme gets a distinguished cost reduction because of using the best appliance running order without affecting the request time. This is also due to the optimization of off-peak time and battery bank in the proposed scheme.

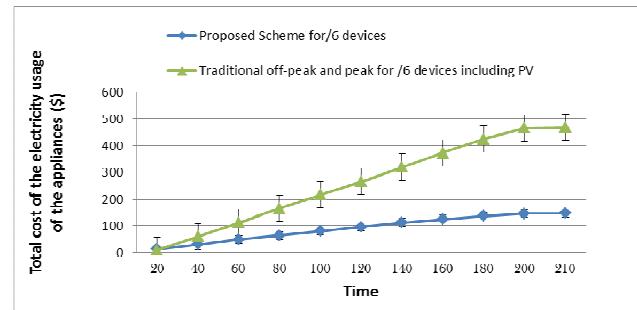


Figure 4. The total cost of the electricity consumed for 6 devices for proposed scheme and traditional power consumption with PV

Figure 5 show the third experiment results by comparing the traditional with proposed scheme by adding more devices to the scalability of the proposed scheme.

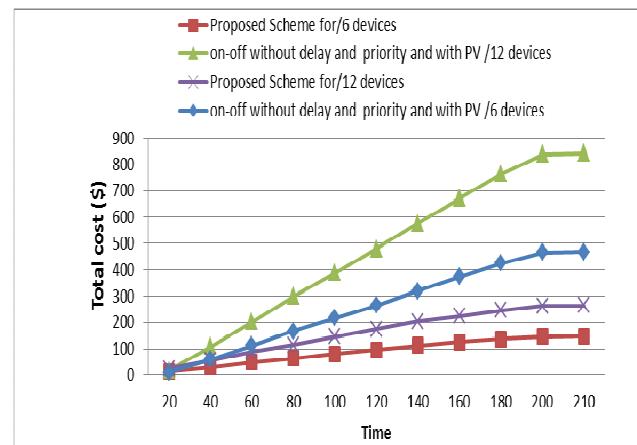


Figure 5. The total cost of the electricity consumed for 6 devices and 12 devices for proposed scheme and traditional power consumption with PV

It can be noticed that the increase of cost when doubling the number of devices is reasonable in the proposed scheme where in traditional power consumption it increases in high rates. The reason why the proposed scheme acts well in this

scenario is the use of genetic algorithm to find the best order of device during the off-peak and using the bank battery.

Figure 6 shows the experiment results by making high demand on power by allowing each device to make up to requests at the same day during off-peak and peak hours. This scenario is done to make sure that the proposed scheme can cope with the high demand and able to schedule these demands in a way with low power consumption which will at the end will reduce the total cost.

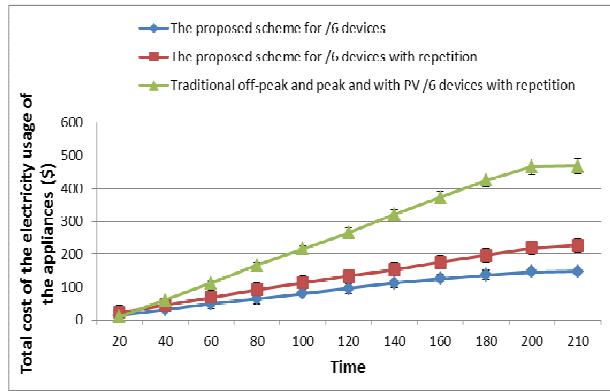


Figure 6. The total cost of the electricity consumed for 6 devices with repetition request for proposed scheme and traditional power consumption

6. Conclusion

In this work, a study of using genetic algorithms to schedule the appliances requests for running during off-peak and peak hours. The proposed scheme runs over 210 days for different number of devices and find the total cost at each time.

This proposed scheme shows an enhancement in saving the cost of power supply on different scenarios starting from changing the number of devices and repeating the demand request.

This proposed scheme proves that the correct use of scheduling the appliances at home will give a high saving in power consumption which at the end will save the cost.

7. Acknowledgement

This research is supported by Jordan University of Science and Technology / faculty of research. Grant number 74/2016

References

- [1] Burcea, M., Hon, W.-K., Liu, H.-H., Wong, P. W. H., & Yau, D. K. Y. "Scheduling for electricity cost in smart grid.: In P. Widmayer, Y. Xu, & B. Zhu (Eds.), *Combinatorial Optimization and Applications*, Vol. 8287, pp. 306-317. Zurich: Springer., 2013.
- [2] Georgievski,I., Degeler, V., Giuliano, G., Lazovik, A., Aiello, M., "Optimizing Energy Costs for Offices Connected to the Smart Grid", IEEE Transactions on Smart Grid, Vol: 2, Issue: 4, pp: 2273 – 2285,2012.
- [3] Goldberg, D.: Genetic algorithms in search, optimization, and machine learning, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©1989
- [4] Gundu, M. , Rajarajeswari, R.."Smart Grid Cost Optimization Using Genetic Algorithm." International Journal of Research in Engineering and Technology, Vol.: 3 Special Issue: 07, 2014
- [5] Ian, F., Carl, K.. "The Grid: Blueprint for a New Computing Infrastructure edited by I. Foster and C. Kesselman , 279-309. San Francisco: Morgan Kaufmann 1999.
- [6] Jiang, G, Wu, X., Yang, H., "Task Scheduling based on Genetic Algorithm in Mobile Grid" International Conference on Computer Science and Service System pp. 719–722 11-13 Aug. 2012.
- [7] Masato, T., Hajime, K., "A Crossover Operator Using Independent Component Analysis for Real-Coded Genetic Algorithms ", Proceedings of the IEEE Congress on Evolutionary Computation Seoul, Korea, pp 643–638 May 27-30, 2001
- [8] Melike, E., Hussein, M., "Wireless Sensor Networks for Smart Grid Applications." Saudi International Electronics, Communications and Photonics Conference (SIECPC). PP. 1 – 6 , 2011.
- [9] Melike E., Husseink, M. "Using wireless sensor networks for energy-aware homes in smart grids." IEEE Symposium on Computers and Communications, , pp. 456–458, 2010
- [10] Rainer S, "Synergy Potential of Smart Appliances" d2.3 of wp 2 From the Smart-A Project, 2008., University of Bonn, March 2009.
- [11] Melike, E., Hussein, M.O "Wireless Sensor Networks for Domestic Energy Management in Smart Grids." IEEE 25th Biennial Symposium on Communications: pp. 63-66. 2010
- [12] Mohsenian, A., Leon, A.."Optimal Residential Load Control With Price Prediction in Real-Time Electricity Pricing Environments." IEEE Transaction on Smart Grid.; vol. 1:pp. 120-133,Sep. 2010.
- [13] Sarvanan, G. ,Gopalakrisanan, V., "Improved Genetic Algorithm for Group-Based Job Scheduling in Mobile Grids." Journal of Theoretical and Applied Information Technology, Vol. 67 No.2, 2014.
- [14] SungHo, C., Taeweon S., HeonChang Y. "Genetic Algorithm based Scheduling Method for Efficiency and Reliability in Mobile Grid." IEEE Transactions on Parallel and Distributed Systems,pp.928– 934, 2009.
- [15] Tong ,L., Dan, B. , Scott, H.. "Efficient and Scalable Multiprocessor Fair Scheduling Using Distributed Weighted Round-Robin." Proceedings of the 14th ACM SIGPLAN symposium on Principles and practice of parallel programming, pp. 65-74, 2009.
- [16] F. Frattolillo, "A Deterministic Algorithm for the Deployment of Wireless Sensor Networks", International Journal of Communication Networks and Information Security, Vol.8, No.1, 2016.
- [17] P. Remlein, D. Szlapka, "Genetic Algorithm used in Search of good Tailbiting Codes", International Journal of Communication Networks and Information Security IJCNIS, Vol. 1, No. 3, December 2009.
- [18] F. Al Balas, Wail Mardini, Yaser Khamayseh, Dua'a Ah.K.Bani-Salameh (2016), "Improved Appliance Coordination Scheme with Waiting Time in Smart Grids," International Journal of Advanced Computer Science and Applications, vol. 7, no. 4, pp. 16-30, 2016

Heterogeneous Delay Tolerant Task Scheduling and Energy Management in the Smart Grid with Renewable Energy

Shengbo Chen, *Student Member, IEEE*, Prasun Sinha, *Senior Member, IEEE*, and Ness B. Shroff, *Fellow, IEEE*,

Abstract—The smart grid is the new generation of electricity grid that can efficiently utilize new distributed sources of energy (e.g., harvested renewable energy), and allow for dynamic electricity price. In this paper, we investigate the cost minimization problem for an end-user, such as a home, community, or a business, which is equipped with renewable energy devices when electrical appliances allow different levels of delay tolerance. The varying price of electricity presents an opportunity to reduce the electricity bill from an end-user’s point of view by leveraging the flexibility to schedule operations of various appliances and HVAC systems. We assume that the end user has an energy storage battery as well as an energy harvesting device so that harvested renewable energy can be stored and later used when the price is high. The energy storage battery can also draw energy from the external grid. The problem we formulate here is to minimize the cost of the energy drawn from the external grid while usage of appliances are subject to individual delay constraints and a long-term average delay constraint. The resulting algorithm requires some future information regarding electricity prices, but it achieves provable performance without requiring future knowledge of either the power demands or the task arrival process. Moreover, we analyze the influence of the assumption that energy can be sold from the battery to the grid. An alternative algorithm is proposed to take advantage of the ability to sell energy. The performance gap between our proposed algorithm and the optimum is shown to diminish as energy selling price approaches the electricity price.

I. INTRODUCTION

The next-generation electricity grid, known as the “smart grid”, provides both suppliers and consumers with full visibility and pervasive control over their assets and services in order to achieve economy and sustainability [2]. Being able to incorporate renewable energy sources (e.g., solar or wind) is one of the key objectives of the smart grid [3]. In addition, the utility companies are allowed to dynamically adjust the electricity price in order to control the power usage. For example, price of electricity increases during high demand periods, and decreases during low demand periods. Consumers

Manuscript received 31 September 2012; revised 13 March 2013.

Shengbo Chen is with Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, 43210, USA e-mail: chens@ece.osu.edu.

Prasun Sinha is with Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, 43210, USA e-mail: prasun@cse.ohio-state.edu.

Ness B. Shroff is with Department of Electrical and Computer Engineering and Computer Science and Engineering, The Ohio State University, Columbus, OH, 43210, USA e-mail: shroff@ece.osu.edu.

The preliminary version of this paper [1] appeared in the proceedings of the IEEE Conference on Decision and Control, 2012.

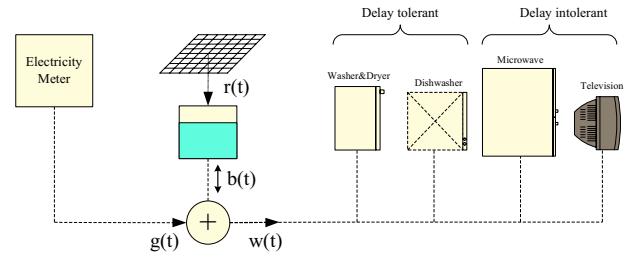


Fig. 1. Demand and Supply

thus can avoid the premium for using electricity at high price periods when they are aware of the price for some future period.

In this paper, we consider an end-user equipped with renewable energy devices in smart grid, where the electricity price is time varying. The renewable energy devices consist of an energy storage battery and an energy harvesting device. Renewable energy can be harvested and stored in the battery. We assume that the arrivals of demands for electrical appliances is a stochastic process (from now on, we use the terms appliance and task interchangeably). Fig. 1 shows some typical appliances at an end user. We assume that some tasks are delay tolerant, that is, they do not need to be activated immediately upon their arrival, such as washer and dish washer. They can be opportunistically scheduled when the electricity price is relatively low in order to reduce cost. For instance, if the price is high around 7pm and low around 2am, then some delay-tolerant tasks, such as dish washer, can be postponed to be scheduled around 2am. The power demand is met by drawing energy from the battery, or purchasing extra energy from the outside grid. We also allow the battery to charge energy from the grid, because the battery can purchase and store energy when the price is low, and discharge when the price is high.

In this work, we are interested in developing an optimal task scheduling algorithm that minimizes the total price cost of the energy drawn from the external grid subject to delay constraints. The customer has full control over all electricity appliances. The algorithm can exploit the delay flexibility and take advantage of time-varying prices.

A. State-of-the-art

In power networks, there have been some literature that has focused on scheduling delay-tolerant tasks. Koutsopoulos

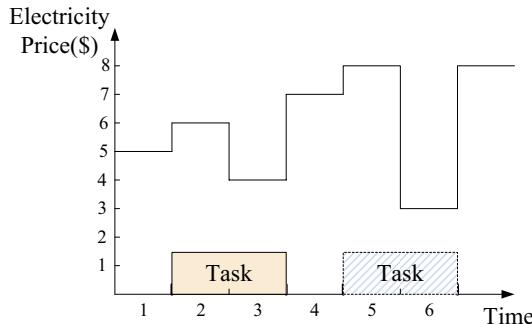


Fig. 2. An example of battery’s influence on task scheduling

and Tassiulas [4] investigate an off-line version and an on-line version of the task scheduling problem. The authors propose two algorithms under these two cases, respectively, and provide a provable performance bound. However, these finite-horizon problems are proven to be NP-hard, and the algorithms only achieve optimality when the delay constraint is arbitrarily loose. In [5], the authors develop an energy allocation algorithm to minimize the total electricity cost. However, they do not allow renewable energy to be saved for future use, which our work takes into account. In addition, we provide individual delay constraints to all tasks, instead of an universal worst-case delay constraint which is likely to be very large. Some works adopt dynamic programming techniques, e.g. [6]. They can achieve optimality only if the distribution of the power demand is known a priori. There are also some other works that have formulated problems using game theory, e.g., [7]. The authors in [8] [9] develop a scheduling scheme to achieve an optimized upper bound on the power peak load. In [10], we investigate an energy trading problem in the smart grid. The energy selling price is assumed to be a fraction of the electricity price. Under the assumption that the energy demand process is an exogenous input process, an asymptotically optimal energy trading scheme is developed. However, in this paper, energy usage is one of our control variables. In [1], we only consider task scheduling problem, yet we do not allow selling energy from the battery to the grid.

B. Our Contributions

In this paper, we address the task scheduling problem while tasks are subject to individual hard delay constraints and average delay constraints. To the best of our knowledge, it is the first work that takes into account these two different types of delay constraints in the area of smart grid. If there is no average delay constraint, a greedy algorithm could achieve the optimal solution. We, however, also take into account the average delay constraint, which is an important quality of service metric, but makes the problem challenging. Further, having a battery brings about significant differences. The reason is that the battery can draw energy from external grid when the electricity price is low and discharges energy when the price is high. Fig. 2 shows a simple example, where a task (the boxes illustrated in the figure) requires a service period of two slots. If there is no battery, we can see that the optimal

way is to schedule during time slot 2 and slot 3 (the red box), resulting in a total cost of 10 dollars. However, with the help of battery, we can store some energy in time slot 3 since the electricity price is low during this time slot. For simplicity of exposition, we assume that the maximum energy that is stored in one slot can be used to support up to one-slot service. Now, let us consider an alternate schedule, where the power demand during time slot 5 is met from the stored energy in slot 3 and the demand during time slot 6 is met from the external grid, as shown by the shadowed blue box. It can be seen that the total cost under this scheduling policy is 7 dollars, which is the optimal. Moreover, if energy can be sold from the battery to the grid, it implies higher flexibility in energy management and may lead to a further cost reduction.

We summarize our main contributions as follows:

- 1) We consider different types of delay constraints in our model. First, each task has a hard delay constraint, which cannot be violated. Further, there is a “dissatisfaction” function of delay for each task, and we require the long-term average dissatisfaction to be less than a threshold. This is a generalization of the average delay constraint.
- 2) We propose a simple algorithm that can achieve provable performance, which is within a bounded distance of the optimum. Note that our algorithm does not require future knowledge of the power demand and the task arrival process.
- 3) We revisit the cost minimization problem if selling energy from the battery to the grid is allowed. An alternative algorithm is proposed to take advantage of the ability to sell energy. The performance gap between our proposed algorithm and the optimum is shown to diminish as energy selling price approaches the electricity price.
- 4) We validate our algorithm using real electricity price traces to compute realistic savings. We show that our algorithm can indeed reduce cost under various system parameter settings.

Our paper is organized as follows: In Section II, we discuss our system model. In Section III, we formulate our cost minimization problem with various delay constraints. In Section IV, we develop our task scheduling algorithm and show its performance. Energy selling situation is discussed in Section V. After presenting simulation results in Section VI, we conclude our paper in Section VII.

II. SYSTEM MODEL

We consider a set of appliances connected to the external smart grid. Time is assumed to be slotted. The price of electricity is time varying and denoted by $P(t)$ in time slot t . As an example, Fig. 3 shows the average five-minute spot market prices for the Columbus area obtained from CAISO [11]. Let N_t represent the set of tasks that arrive in time slot t , while n_t represents the number of tasks in N_t , i.e., $n_t = |N_t|$, where $|\cdot|$ denotes the cardinality of a set. For simplicity of exposition, we assume that all tasks arrive in the beginning of each slot.

We note that there are two types of tasks, delay-tolerant and delay-intolerant tasks. Let c_i^t denote the required service

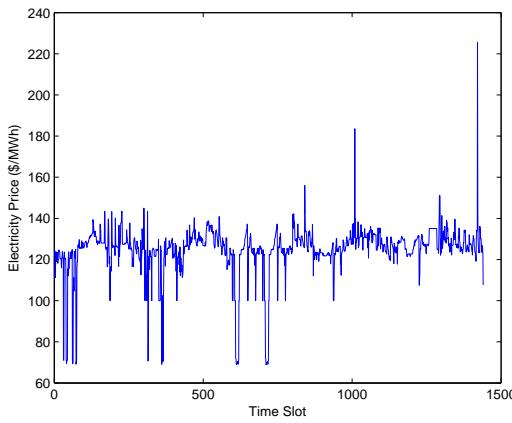


Fig. 3. 5-minute average spot market price during the week of 10/10/2011–10/14/2011 for Columbus Area from CAISO [11]

time for each task $i \in N_t$. Also, there is a deadline associated with each task $i \in N_t$, i.e., the maximum number of time slots allowed for finishing the job from its arrival time t , denoted by d_i^t . The deadline is a hard constraint, namely the task needs to be completed before time $t + d_i^t$. We call the task delay-intolerant if $c_i^t = d_i^t$, and delay-tolerant if $c_i^t < d_i^t$. For a delay-intolerant task, the only choice that we have is to activate it immediately upon its arrival. However, for delay-tolerant tasks, we can opportunistically schedule them in order to make use of the fluctuating nature of the electricity price. Our goal here is to find the optimal “postponing” time s_i^t so that the total cost is minimized subject to the delay constraints. Clearly, for the delay-intolerant tasks, we have to set $s_i^t = 0$. Let d_{max} denote the maximum delay allowed for any task, i.e., $d_{max} \triangleq \max_{t,i} d_i^t$. Note that $(c_i^t, s_i^t, d_i^t), \forall t, \forall i$ are integers. It is assumed that we have an accurate short-term estimation of the electricity price. More precisely, we know $\vec{P}_t \triangleq P(t), P(t+1), \dots, P(t+d_{max})$. It is worth pointing out that this is a reasonable assumption because the short-term estimation of electricity price can be obtained from the history [12].

Let $h(t)$ denote the harvested renewable energy in time slot t , and let $r(t)$ denote our energy storage decision, i.e., the actual energy that is stored into the battery. For simplicity of exposition, we assume that $r(t)$ amount of energy is stored in the battery at the end of slot t . First, it is convenient for us to assume that battery has infinite capacity. We will show later that our algorithm only requires a reasonable sized finite battery. A natural constraint of $r(t)$ is

$$r(t) \leq h(t). \quad (1)$$

The reason that we keep $r(t)$ and $h(t)$ different is due to some technical issues used in our proof. We assume that $[n_t, c_i^t, h(t)]$ is i.i.d. over slots.

Let $w(t)$ represent the total power demand in time slot t . We assume that each task $i \in N_t$ consumes energy at a constant rate π_i^t , namely the power consumption for task i stays the same during its activation period. In this paper, we only consider the case where the activation period of any task

TABLE I
NOTATIONS

Control var.	
s_i^t	Delay for task $i \in N_t$
$b(t)$	Energy drawn from (stored in) the battery
$r(t)$	Actual energy stored in the battery in slot t
Grid var.	
$P(t)$	Electricity price in time slot t
$g(t)$	Energy drawn from the grid in time slot t
Internal var.	
c_i^t	Required service time for task $i \in N_t$
d_i^t	Deadline for task $i \in N_t$
$w(t)$	Power demand in time slot t
$h(t)$	Harvested renewable energy in time slot t
$B(t)$	Battery level in time slot t
$\pi_i(t)$	Power consumption for task i
$U_i^t(s)$	Dissatisfaction function for delay s for task i

is a contiguous chunk of time, and we do not consider the case where the activation period of tasks can be interrupted and resumed. We notice that part of $w(t)$ is met by utilizing energy from the battery, while the other part will be drawn from the grid. Let $g(t)$ and $b(t)$ represent the amounts of energy that are drawn from the outside grid and the battery in time slot t , respectively. Because the supply always needs to balance the demand, we have $w(t) = g(t) + b(t)$ as shown in Fig. 1. In addition, we also allow the battery to charge energy from the grid, which means that $b(t)$ could be negative. In particular, the battery discharges/charges energy if we have $b(t) \geq 0$. We denote b_{max} as a maximal amount of energy either charging or discharging from the battery in one time slot. We use $B(t)$ to denote the battery level at the beginning of time slot t , and the energy dynamics can be formulated as follows:

$$B(t+1) = B(t) + r(t) - b(t). \quad (2)$$

Since we have $g(t) \geq 0$, it follows that $b(t) \leq w(t)$. Therefore, the constraints on $b(t)$ are given by

$$|b(t)| \leq b_{max} \quad (3)$$

$$b(t) \leq B(t), \quad (4)$$

$$b(t) \leq w(t), \quad (5)$$

where the second constraint means that the allocated energy from the battery should be less than or equal to the current available energy in the battery. For storing energy in the battery, $b(t)$ could be negative and thus Eqn. (4) and (5) do not constrain the storage process.

Note that $w(t)$ depends on the decisions made during time slot t up to time slot $t - d_{max} + 1$, we have

$$w(t) = \sum_{\tau=t-d_{max}+1}^t \sum_{i=1}^{n_\tau} \pi_i^\tau \mathbf{1}(\tau + s_i^\tau + c_i^\tau > t \& \tau + s_i^\tau \leq t), \quad (6)$$

where $\mathbf{1}(\tau + s_i^\tau + c_i^\tau > t \& \tau + s_i^\tau \leq t)$ is the indicator function. The term $\tau + s_i^\tau + c_i^\tau > t \& \tau + s_i^\tau \leq t$ means that a task i started before time slot t and will finish after time slot t . Thus, this task has energy consumption in time slot t .

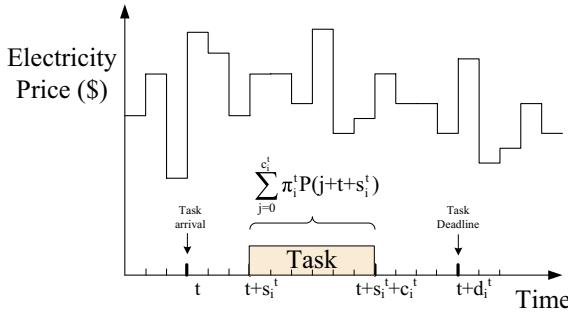


Fig. 4. Example of the scheduling of one task $i \in N_t$

Our goal is decide $(r(t), s_i^t, b(t))$ at each time slot such that the total price cost of the energy drawn from the external grid is minimized. We do not explicitly consider some practical issues, such as energy leakage in the battery or DC/AC conversion loss, but we can readily incorporate them into our model. We summarize the notations in Table I.

III. PROBLEM FORMULATION

Suppose that there is an increasing convex function $U_i^t(s)$, satisfying $U_i^t(0) = 0$, which reflects the dissatisfaction associated with delay s for task $i \in N_t$. The convexity models a typical user for whom the rate of increase in dissatisfaction increases with delay. Notice that $U_i^t(\cdot)$ is different for heterogeneous tasks. We assume that the long-term average dissatisfaction should be no greater than than some threshold α , that is,

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^{n_t} U_i^t(s_i^t) \leq \alpha. \quad (7)$$

For any task $i \in N_t$, since we have to finish it before the deadline, it yields

$$s_i^t + c_i^t \leq d_i^t.$$

Therefore, the constraint for the postponing time s_i^t is given by

$$0 \leq s_i^t \leq d_i^t - c_i^t. \quad (8)$$

Hence, the cost minimization problem can be formulated as

$$\begin{aligned} \text{Problem A: } & \min_{r(t), s_i^t, b(t)} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[g(t)P(t)] \\ & \text{s.t. (1), (2), (3), (4), (5), (7), (8),} \end{aligned} \quad (9)$$

where $P(t)g(t)$ represents the total price of the energy drawn from the grid during time slot t .

Since $g(t) = w(t) - b(t)$, we can rewrite Problem A as follows:

$$\begin{aligned} & \min_{r(t), s_i^t, b(t)} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[w(t)P(t) - b(t)P(t)] \\ & \text{s.t. (1), (2), (3), (4), (5), (7), (8).} \end{aligned} \quad (10)$$

Notice that $\lim_{T \rightarrow \infty} \sum_{t=1}^T w(t)P(t)$ represents the total cost of the power demand from the time horizon, while it

can be also derived by simply adding the cost for all tasks one by one. Thus, we have the following equation

$$\lim_{T \rightarrow \infty} \sum_{t=1}^T w(t)P(t) = \lim_{T \rightarrow \infty} \sum_{t=1}^T \sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + t + s_i^t), \quad (11)$$

where $\sum_{j=0}^{c_i^t-1} \pi_i^t P(j + t + s_i^t)$ is the cost of task $i \in N_t$ as depicted in Fig. 4.

Now, we can reformulate the optimization problem as follows:

$$\begin{aligned} \text{Problem B: } & \min_{r(t), s_i^t, b(t)} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^t + t) \right. \\ & \quad \left. - P(t)b(t) \right] \\ & \text{s.t. (1), (2), (3), (4), (5), (7), (8).} \end{aligned} \quad (12)$$

Now we focus on Problem B and adopt the Lyapunov optimization approach [5] to solve it.

IV. TASK SCHEDULING POLICY

In this section, we propose a task scheduling policy and show that its performance is within a bounded distance of the optimum as T tends to infinity.

A. Virtual Queue

Let us construct an auxiliary virtual queue $Q(t)$, whose input and output are $\sum_{i=1}^{n_t} U_i^t(s_i^t)$ and α respectively. The queueing dynamics is depicted as

$$Q(t+1) = \max\{Q(t) + \sum_{i=1}^{n_t} U_i^t(s_i^t) - \alpha, 0\} \quad (13)$$

Lemma 1: If the virtual queue is rate stable, i.e., $\limsup_{T \rightarrow \infty} Q(T)/T = 0$ with probability 1, then the constraint (7) is satisfied.

Proof: Suppose that the virtual queue is rate stable. Then we have

$$\limsup_{T \rightarrow \infty} \mathbb{E}[Q(T)]/T = 0. \quad (14)$$

Note that for any time T , by adding Eqn. (13) from slot 0 to slot $T-1$, the following inequality always holds:

$$Q(T) \geq Q(0) - T\alpha + \sum_{t=0}^{T-1} \sum_{i=1}^{n_t} U_i^t(s_i^t).$$

Dividing by T and taking expectation yields:

$$\mathbb{E}[Q(T)]/T \geq \mathbb{E}[Q(0)]/T - \alpha + \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{n_t} \mathbb{E}[U_i^t(s_i^t)].$$

We take the limsup for both sides, it yields:

$$\alpha \geq \limsup_{T \rightarrow \infty} \frac{1}{T} \left(\sum_{t=1}^T \sum_{i=1}^{n_t} \mathbb{E}[U_i^t(s_i^t)] \right).$$

■

B. Lower Bound the Minimum Cost

In this subsection, we will obtain a lower bound on the minimum cost of Problem **B**. The following lemma shows that the performance achieved by using a stationary and randomized algorithm forms a lower bound.

Let C^{opt} be the minimum cost to Problem **B**. And let \tilde{C} be the minimum cost to the following Problem **C**.

$$\text{Problem C: } \min_{r(t), s_i^t, b(t)} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^t + t) - P(t)b(t) \right]$$

s.t. (1), (2), (3), (7), (8).

Note that Problem **C** and **C** have the same objective function, but Problem **C** has fewer constraints. Thus, we know that C^{opt} is lower bounded by \tilde{C} , i.e., $\tilde{C} \leq C^{opt}$.

Lemma 2: \tilde{C} can be achieved by an optimal stationary and randomized policy, that is, the control action $(\tilde{r}(t), \tilde{s}_i^t, \tilde{b}(t))$ in each time slot is only a function of $[n_t, c_i^t, h(t)]$. In particular, we have

$$\mathbb{E} \left[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + t + \tilde{s}_i^t) - P(t)\tilde{b}(t) \right] = \tilde{C}, \quad (15)$$

$$\mathbb{E} \left[\sum_{i=1}^{n_t} U_i^t(\tilde{s}_i^t) - \alpha \right] \leq 0, \quad (16)$$

$$\mathbb{E} [\tilde{r}(t) - \tilde{b}(t)] \geq 0, \quad (17)$$

Proof: \tilde{C} is achieved over *all possible* control policies, not just stationary and randomized policies. However, we apply Theorem 4.5 in [13] in order to prove our result, that is, \tilde{C} can be achieved by a stationary and randomized policy $\tilde{b}(t)$.

We will show how to project Problem **C** to Eqn. (4.31)-(4.35) in [13]. First, $[n_t, c_i^t, h(t)]$ corresponds to the i.i.d. state $w(t)$. And $\frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^t + t) - P(t)b(t) \right]$ corresponds to $\bar{y}_0(t)$. Eqn. (16) means that the long-term average dissatisfaction achieved by the stationary policy is no greater than α . Eqn. (17) implies that the average allocated energy from the battery is no greater than the stored energy. Eqns. (16) and (17) imply the battery level $B(t)$ and the virtual queue $Q(t)$ are required to be mean rate stable. Therefore, the result that \tilde{C} can be achieved by an optimal stationary and randomized policy $(\tilde{r}(t), \tilde{s}_i^t, \tilde{b}(t))$ holds directly by applying Theorem 4.5 in [13]. \blacksquare

C. HTSA: Heterogeneous Task Scheduling Algorithm

We define the Lyapunov function $L(t) = \frac{1}{2}(Q(t)^2 + (B(t) - \theta)^2)$, where θ is a parameter specified later. The intuition behind it is that, by minimizing the drift of the Lyapunov function, we force $B(t)$ to approach θ . We also define several constants $n_{max} = \max_t n_t$, $h_{max} = \max_t h(t)$, $c_{max} = \max_{t,i} c_i^t$, and $U_{max} = \max_{t,i} U_i^t(d_i^t)$, where U_{max} reflects the maximum dissatisfaction among all tasks.

Let $Z(t) = (Q(t), B(t))$. The conditional Lyapunov drift is given by $\mathbb{E}\{(L(t+1) - L(t)|Z(t)\}$.

We will show some properties of the drift via the following lemma.

Lemma 3: The conditional Lyapunov drift satisfies that

$$\begin{aligned} \mathbb{E}\{(L(t+1) - L(t)|Z(t)\} &\leq \\ D + Q(t)\mathbb{E} \left[\sum_{i=1}^{n_t} U_i^t(s_i^t) - \alpha | Z(t) \right] \\ &+ (B(t) - \theta)\mathbb{E}[r(t) - b(t)|Z(t)], \end{aligned} \quad (18)$$

where $D \triangleq \frac{1}{2}(n_{max}^2 U_{max}^2 + \alpha^2 + r_{max}^2 + b_{max}^2)$.

Proof: We refer to Appendix for the proof. \blacksquare

By adding $V\mathbb{E}[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^t + t) - P(t)b(t)|Z(t)]$ on both sides of Eqn. (18), we have

$$\begin{aligned} &\mathbb{E}[(L(t+1) - L(t)|Z(t)] \\ &+ V\mathbb{E}[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^t + t) - P(t)b(t)|Z(t)] \\ &\leq D + Q(t)\mathbb{E} \left[\sum_{i=1}^{n_t} U_i^t(s_i^t) - \alpha | Z(t) \right] \\ &+ (B(t) - \theta)\mathbb{E}[r(t) - b(t)|Z(t)] \\ &+ V\mathbb{E}[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^t + t) - P(t)b(t)|Z(t)] \\ &= D - \alpha Q(t) + (B(t) - \theta)\mathbb{E}[r(t)|Z(t)] + \\ &\sum_{i=1}^{n_t} \mathbb{E}[Q(t)U_i^t(s_i^t) + V \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^t + t)|Z(t)] \\ &+ (\theta - B(t) - VP(t))\mathbb{E}[b(t)|Z(t)], \end{aligned} \quad (19)$$

where V is a control parameter.

We now describe our scheme, *heterogeneous task scheduling algorithm (HTSA)*. The idea of HTSA is to minimize the right-hand side (RHS) of Eqn. (19) subject to the energy-availability constraint (4) and (5).

Heterogeneous task scheduling algorithm (HTSA):

- In each time slot t , the harvested energy $r^*(t)$ is determined by

$$r^*(t) = \begin{cases} h(t), & \text{if } B(t) - \theta < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

- In each time slot t , the postponing time s_i^t for task $i \in N_t$ is determined by:

$$s_i^{t*} = \arg \min_{0 \leq s_i^t \leq d_i^t - c_i^t} Q(t)U_i^t(s_i^t) + V \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + t + s_i^t). \quad (21)$$

- In each time slot t , the battery charge/discharge is given by:

$$b^*(t) = \begin{cases} \min\{b_{max}, w(t)\}, & \text{if } \theta - B(t) - VP(t) < 0, \\ -b_{max}, & \text{otherwise,} \end{cases} \quad (22)$$

where $w(t)$ is determined by Eqn. (6).

Define a constant P_{max} as the highest electricity price, i.e., $P_{max} = \max_t P(t)$. By setting $\theta = b_{max} + VP_{max}$, from Eqn. (22), we can see that when $B(t) < b_{max}$, it always has $\theta - B(t) - VP(t) > 0$. In other words, the battery always draws energy from the grid, namely $b(t) = -b_{max}$, when the battery level is less than b_{max} . This implies that when the battery discharges, there is always enough energy in the battery, i.e., $B(t) > b_{max}$. Therefore, the energy constraint of Eqn. (4) is indeed *redundant*.

D. Performance Analysis

In this subsection, we will prove that HTSA achieves a performance that is within a bounded distance of the optimum via the following theorem.

Theorem 1: By setting $\theta = b_{max} + VP_{max}$ and $B(0) = \theta$, HTSA has the following property:

- 1) The battery level $B(t)$ satisfies:

$$B(t) \leq \theta + b_{max} + h_{max}. \quad (23)$$

- 2) There exists $M > 0$, such that $Q(t)$ is bounded by M for all t , where M is a constant.
- 3) The cost achieved by HTSA satisfies:

$$\begin{aligned} & \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^* + t) - P(t) b^*(t) \right] \\ & \leq C^{opt} + P_{max} b_{max} + \frac{D + (b_{max} + h_{max})^2}{V}. \end{aligned}$$

Proof: 1). We will use mathematical induction to prove it. i). We have $B(0) = \theta < \theta + b_{max} + h_{max}$. ii). Assume that $B(t) \leq \theta + b_{max} + h_{max}$. iii). For time slot $t+1$, let us consider two subcases. First, if $B(t) \leq \theta$, we can see that the maximum increased energy in the battery during one time slot is $h_{max} + b_{max}$, which is under the case $r(t) = h_{max}$ and $b(t) = -b_{max}$. Thus, we have $B(t+1) \leq \theta + b_{max} + h_{max}$. Second, if $B(t) > \theta$, from Eqn. (22), we can see that $b(t) > 0$ and $r(t) = 0$ when $B(t) > \theta$, that is, as long as the battery level is greater than θ , it discharges and there is no energy replenishment. Therefore, it follows that $B(t+1) \leq B(t) \leq \theta + h_{max} + b_{max}$. Hence, we conclude that $B(t+1) \leq \theta + h_{max} + b_{max}$, which means that under HTSA, the battery level is always bounded. Therefore, the required battery size is finite.

2). Without loss of generality, we assume that $U_i^t(1) \neq 0$ and denote $U_{min} \triangleq \min_{t,i} U_i^t(1)$. Note that $Q(t)U_i^t(s_i^t)$ is an increasing function of s_i^t and $U_i^t(0) = 0$. Consider Eqn. (21), when $s_i^{t*} = 0$, we have the value of Eqn. (21) to be $V \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + t)$. Thus, if we have $Q(t)U_i^t(1) \geq V \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + t)$, that is, the cost when $s_i^t = 1$ is higher than the cost when $s_i^t = 0$, it follows that $s_i^{t*} = 0$. This means that when $Q(t) > \frac{V c_{max} \pi_{max} P_{max}}{U_{min}}$, we have the input of $Q(t)$, i.e., $U_i^t(s_i^t)$, equals 0. Similarly to part 1), we can show that $Q(t) \leq M \triangleq \frac{V c_{max} \pi_{max} P_{max}}{U_{min}} + n_{max} U_{max}$.

3). Recall that HTSA minimizes the RHS of Eqn. (19). However, the existence of constraint Eqn. (5) has prevented $b(t)$ being selected in $(0, b_{max})$. Thus, the term $(\theta - B(t) - VP(t))\mathbb{E}[b(t)|Z(t)]$ is not maximized. We compare the stationary and randomized policy in Lemma 2 and HTSA. In particular, we have that

$$\begin{aligned} & D - \alpha Q(t) + (B(t) - \theta)\mathbb{E}[r^*(t)|Z(t)] + \\ & \sum_{i=1}^{n_t} \mathbb{E}[Q(t)U_i^t(s_i^{t*}) + V \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^{t*} + t)|Z(t)] \\ & + (\theta - B(t) - VP(t))\mathbb{E}[b^*(t)|Z(t)] \\ & \leq D - \alpha Q(t) + (B(t) - \theta)\mathbb{E}[\tilde{r}(t)|Z(t)] + \\ & \sum_{i=1}^{n_t} \mathbb{E}[Q(t)U_i^t(\tilde{s}_i^t) + V \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + \tilde{s}_i^t + t)|Z(t)] \\ & + (\theta - B(t) - VP(t))\mathbb{E}[\tilde{b}(t)|Z(t)] \\ & + b_{max}|\theta - B(t) - VP(t)|, \end{aligned} \quad (24)$$

where the last term is an upper bound on the term $(\theta - B(t) - VP(t))\mathbb{E}[b^*(t)|Z(t)]$, since we only need to consider the case $\theta - B(t) - VP(t) < 0$

From the fact that $B(t) < \theta + b_{max} + h_{max}$, we have $\theta \geq \theta - B(t) - VP(t) \geq -(b_{max} + h_{max} + VP_{max}) = -(\theta + h_{max})$. It follows that $|\theta - B(t) - VP(t)| \leq b_{max} + h_{max} + VP_{max}$.

Rearranging the RHS of Eqn. (24), it yields:

$$\begin{aligned} & D + Q(t)\mathbb{E} \left[\sum_{i=1}^{n_t} U_i^t(\tilde{s}_i^t) - \alpha |Z(t)| \right] \\ & + (B(t) - \theta)\mathbb{E}[\tilde{r}(t) - \tilde{b}(t)|Z(t)] \\ & + V\mathbb{E} \left[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} P(j + \tilde{s}_i^t + t) - P(t)\tilde{b}(t)|Z(t) \right] \\ & + b_{max}|\theta - B(t) - VP(t)| \\ & \leq D + V\tilde{C} + (b_{max} + h_{max} + VP_{max})b_{max}. \end{aligned} \quad (25)$$

where for the last inequality, we have used the following expressions:

$$\mathbb{E} \left[\sum_{i=1}^{n_t} U_i^t(\tilde{s}_i^t) - \alpha |Z(t)| \right] = \mathbb{E} \left[\sum_{i=1}^{n_t} U_i^t(\tilde{s}_i^t) - \alpha \right] \leq 0 \quad (26)$$

$$\begin{aligned} (B(t) - \theta)\mathbb{E}[\tilde{r}(t) - \tilde{b}(t)|Z(t)] &= (B(t) - \theta)\mathbb{E}[\tilde{r}(t) - \tilde{b}(t)] \\ &\leq (B(t) - \theta)\mathbb{E}[\tilde{r}(t)] \leq (b_{max} + h_{max})h_{max}. \end{aligned} \quad (27)$$

Eqn. (26) is derived from Eqn. (16) because $(\tilde{s}_i^t, \tilde{b}(t))$ is a stationary policy which is independent of $Z(t)$. Similarly, Eqn. (27) is from Eqn. (17) and $B(t) < \theta + b_{max} + h_{max}$.

Thus, combining Eqn. (24) and (25), we have

$$\begin{aligned} & \mathbb{E}[(L(t+1) - L(t))|Z(t)] \\ & + V\mathbb{E} \left[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^{t*} + t) - P(t)b^*(t)|Z(t) \right] \\ & \leq D + (b_{max} + h_{max})^2 + V\tilde{C} + VP_{max}b_{max} \\ & \leq D + (b_{max} + h_{max})^2 + VC^{opt} + VP_{max}b_{max}, \end{aligned} \quad (28)$$

where the last inequality holds because \tilde{C} is a lower bound of C^{opt} .

By taking the expectation with respect to $Z(t)$ on both sides of Eqn. (28) and take the summation from $t = 0$ to T , it yields that

$$\begin{aligned} & \mathbb{E}[(L(T+1) - L(0))] \\ & + V \sum_{t=1}^T \mathbb{E} \left[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^{t*} + t) - P(t)b^*(t) \right] \\ & \leq TD + T(b_{max} + h_{max})^2 + VTC^{opt} + VTP_{max}b_{max}. \end{aligned} \quad (29)$$

If we set $B(0) = \theta$, we have $L(0) = 0$. Rearranging Eqn. (29) and dividing by VT on both sides, we have

$$\begin{aligned} & \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[\sum_{i=1}^{n_t} \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + s_i^{t*} + t) - P(t)b^*(t) \right] \\ & \leq C^{opt} + P_{max}b_{max} + \frac{D + (b_{max} + h_{max})^2}{V}. \end{aligned} \quad (30)$$

Taking the limsup as $T \rightarrow \infty$ yields our result. ■

From part (2) in Theorem 1, since $Q(t)$ is bounded, combining with lemma 1, we can see that the average delay constraint, i.e., Eqn. (7), is satisfied.

Eqn. (30) shows that the cost induced by our algorithm is within a bounded distance of the optimum by setting the parameter V to be sufficiently large. It is worth pointing out that the algorithm does not require the future knowledge of the statistics of power demand and the task arrival process.

Discussion: Since our focus here is a family or a community, it is assumed here that the scheduling actions will not influence the electricity price. However, if the the scheduling policy is adopted for a large scale of power grid, it will lead to an impact on the electricity price, which will form the basis of our future work.

V. ENERGY SELLING

In this section we allow the system to sell energy back to the grid. Note that in our previous discussion, $g(t) \geq 0$ always holds due to the constraint in Eqn. (5), i.e., $b(t) \leq w(t)$. However, If we allow energy selling, this constraint is relaxed, which implies that $g(t) < 0$ is possible. We present our corresponding algorithm as follows:

Joint task scheduling and energy selling algorithm (JTSES):

- In each time slot t , the harvested energy $r^*(t)$ is determined by

$$r^*(t) = \begin{cases} h(t), & \text{if } B(t) - \theta < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (31)$$

- In each time slot t , the postponing time s_i^t for task $i \in N_t$ is determined by:

$$s_i^{t*} = \arg \min_{0 \leq s_i^t \leq d_i^t - c_i^t} Q(t)U_i^t(s_i^t) + V \sum_{j=0}^{c_i^t-1} \pi_i^t P(j + t + s_i^t). \quad (32)$$

- In each time slot t , the battery charge/discharge is given by:

$$b^*(t) = \begin{cases} b_{max}, & \text{if } \theta - B(t) - VP(t) < 0, \\ -b_{max}, & \text{otherwise.} \end{cases} \quad (33)$$

Notice that the only difference between JTSES and HTSA is $b(t)$, where $b(t)$ can be larger than $w(t)$ in JTSES.

Surprisingly, we can show that under this algorithm, our scheme can actually achieve asymptotic optimality as shown by the following theorem.

Theorem 2: If the user is allowed to sell energy to the grid at the price of $P(t)$, by setting $\theta = b_{max} + VP_{max}$ and $B(0) = \theta$, JTSES achieves a performance that could be arbitrarily close to the optimum as T tends to infinity.

Proof: Notice that when the constraint Eqn. (5) does not exist, $b(t)$ thus can be selected in $(0, b_{max})$. Therefore, RHS of Eqn. (19) is maximized by JTSES. It can be seen that the extra term $b_{max}|\theta - B(t) - VP(t)|$ in Eqn. (24) no longer exists. Following the same line of the proof of Theorem 1, it yields the conclusion. ■

Theorem 2 implies that the gap $P_{max}b_{max}$ diminishes if energy selling is allowed. Furthermore, it is worth pointing out that in JTSES the task scheduling and energy management have been decoupled due to the removed Eqn. (5).

In [10], we investigate an energy trading problem in the smart grid. The energy selling price is assumed to be $\beta P(t)$, where β is a constant between 0 and 1. Under the assumption that the energy demand process $w(t)$ is an exogenous input process, an asymptotically optimal energy trading scheme is developed. However, in this work, due to the fact that $w(t)$ is determined by the task scheduling decision, the asymptotic optimality is only achieved under the case when the energy selling price is equal to the energy buying price, i.e., $\beta = 1$. Furthermore, we can show the performance of JTSES under the selling price $\beta P(t)$ via the following theorem.

Theorem 3: If the energy selling price is $\beta P(t)$, by setting $\theta = b_{max} + VP_{max}$ and $B(0) = \theta$, JTSES achieves an average cost that is within a bound of $(1-\beta)P_{max}b_{max}$ of the optimum by setting V to be sufficiently large.

Proof: As discussed in Theorem 2, when $\beta = 1$, JTSES achieves the optimal value, which is denoted as J_1^* . Note that when $\beta < 1$, the optimum J_β^* will be higher since the benefit brought about by selling the energy to the grid is reduced, i.e., $J_\beta^* \geq J_1^*$.

Note that JTSES is sub-optimal for the case of $\beta < 1$, therefore the cost J_β achieved by JTSES for the case $\beta < 1$ is higher than J_β^* , i.e., $J_\beta \geq J_\beta^*$. Also note that the gap between the two cases $\beta = 1$ and $\beta < 1$ is bounded by $(1-\beta)P_{max}b_{max}$ under the same scheme JTSES, namely $J_\beta \leq J_1^* + (1-\beta)P_{max}b_{max}$. Thus, we have $J_\beta^* \leq J_\beta \leq J_1^* + (1-\beta)P_{max}b_{max} \leq J_\beta^* + (1-\beta)P_{max}b_{max}$, which proves our results. ■

VI. CASE STUDY

The remainder of the paper evaluates the algorithms presented in the previous section.

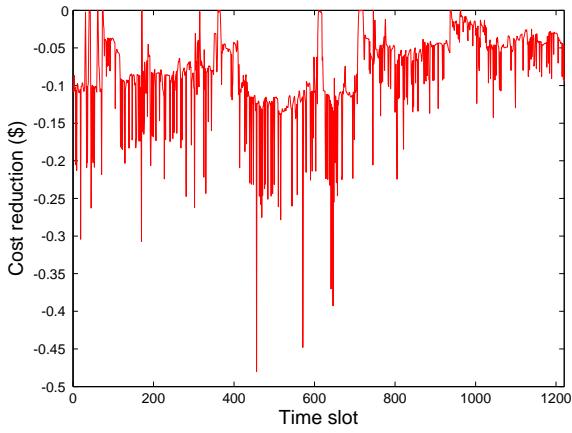


Fig. 5. Reduction in cost for Class I appliances

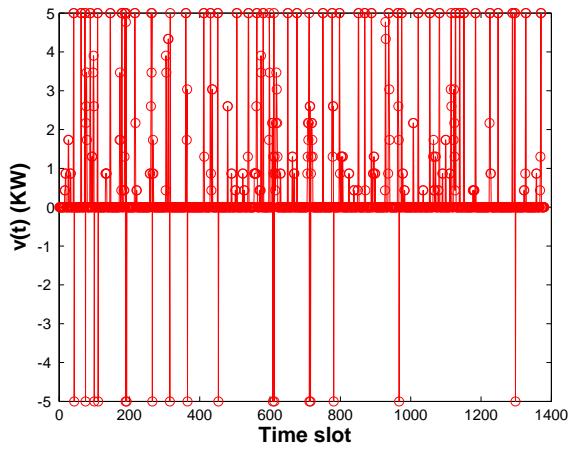


Fig. 6. Energy drawn from the battery in each time slot

A. Experiment Setup

We adopt the 5-minute average spot market prices for Columbus Area from CAISO [11]. The profile depicted in Fig. 3 shows the electricity price for the period 10/10/2011-10/14/2011. The arrival process of all tasks here are assumed to be Poisson process with different intensity λ_i , although Theorem 1 holds for any general arrival process. Without loss of generality, we consider four types of appliances in our simulations. The first three tasks are delay-tolerant, while the last one is delay-intolerant. The arrival intensities for these tasks are set to be 2, 0.5, 0.035 and 100, respectively. And the energy consumption rate for these tasks π_i^t are set to be 5.2kw, 3.5kw, 2.4kw and 60w. The “dissatisfaction” functions are assumed to be $U(x) = x^2$. The average delay constraint threshold α is set to be 10000, and the parameter V is set to be 100.

B. Performance Evaluation

We start by comparing our algorithm and a naive scheme, which activates the task immediately upon its arrival. Consider the first type of delay-tolerant task, which has a deadline of 100 slots, while the required service time is two slots. Fig. 5

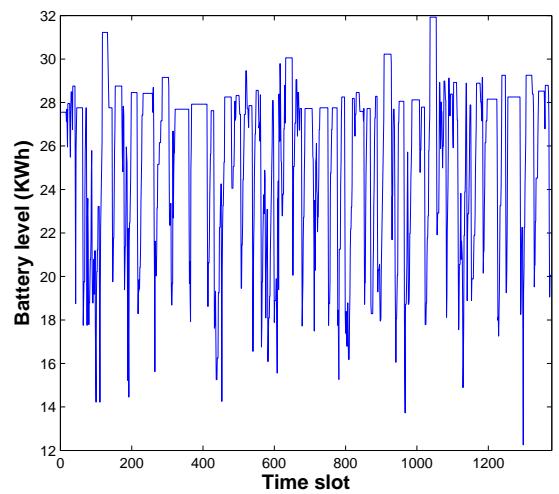


Fig. 7. Battery level in each time slot

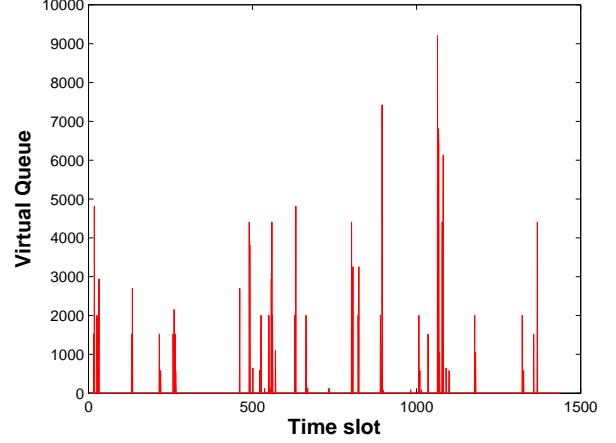


Fig. 8. Virtual queue length in each time slot

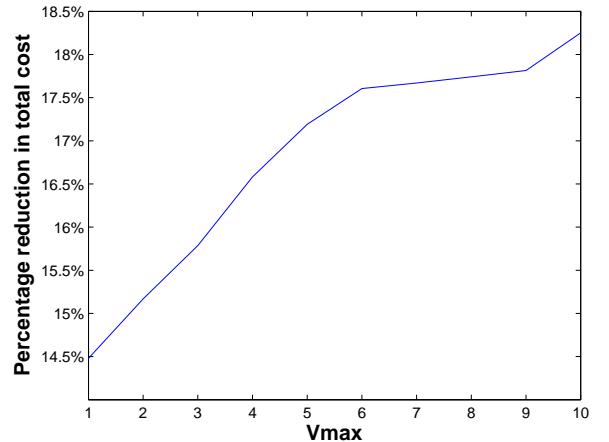


Fig. 9. Reduction in cost versus the battery size

shows the reduction in cost for scheduling this type of task using our algorithm. The total cost saved in these five days is

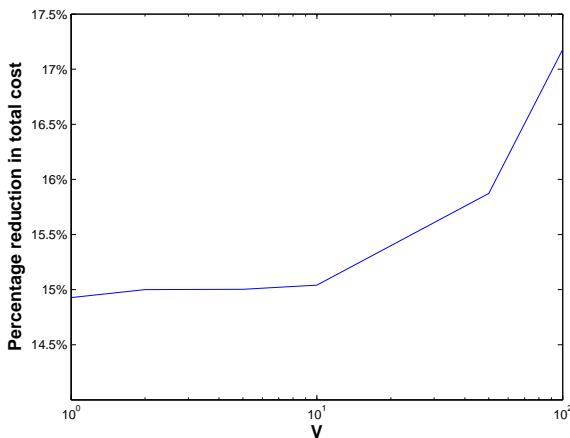


Fig. 10. Reduction in Cost versus parameter V

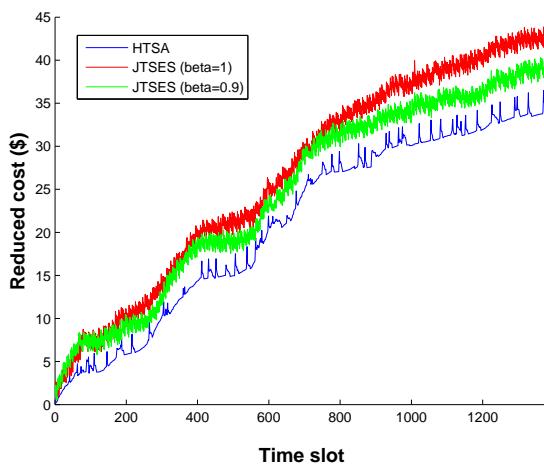


Fig. 11. Reduced Cost for both *HTSA* and *JTSES*

\$35.40, which is 19.82% of the total cost. If we extend the hard delay deadline to 200 slots, the corresponding percentage of saved cost increases to 27.20%. This is because if we have a less stringent delay constraint, we can gain more benefit.

Next, we will show how the battery influences the performance. The deadline for other two types of delay-tolerant tasks are set to be 10 and 20, respectively. We set the battery size to be $2b_{max} + VP_{max} + h_{max}$. Fig. 6 shows the energy drawn from the battery, i.e., $b(t)$, versus time in the whole period. Fig. 7 and Fig. 8 illustrate the energy level $B(t)$ and virtual queue length $Q(t)$, respectively, both of which are bounded. These results conform to our analytical result in the previous section.

Fig. 9 depicts the percentage reduction in cost versus b_{max} . We can see that the percentage reduction in cost increases as b_{max} grows. This is because a large battery maximum output can lead to a higher shaved cost which can be seen from Eqn. (22).

In Fig. 10, we illustrate the relationship between the percentage of reduced cost and the parameter V . It can be seen that when V is small, the reduced cost is less than the counterpart

when V is large. The reason is that the term $\frac{D+(b_{max}+h_{max})^2}{V}$ in Eqn. (30) cannot be neglected when V is small.

Fig. 11 shows the cost reduction under both *HTSA* and *JTSES* with different selling price. The parameter β in *JTSES* are assumed to be 1 and 0.9, respectively. We can see that when $\beta = 1$, i.e., the selling price is equal to the buying price, *JTSES* always outperforms *HTSA*. The reason is that selling energy is allowed in *JTSES*, which leads to further cost reduction. On the other hand, if β becomes smaller, the reduced cost also decreases. This observation is consistent with our theoretical result.

VII. CONCLUSION

In this paper, we investigate the cost minimization problem for an end-user, which is equipped with renewable energy devices when electrical appliances allow different levels of delay tolerance. The varying price of electricity implies an opportunity to reduce the electricity cost by utilizing the flexibility to schedule various appliances. We assume that the end user has an energy storage battery and an energy harvesting device so that harvested renewable energy can be stored and used when the price is high. The problem we formulate here is to minimize the cost of the energy from the external grid while usage of appliances are subject to individual delay constraints and a long-term average delay constraint. Our proposed algorithm, *HTSA*, requires some future information of the electricity price, but achieves provable performance without requiring future knowledge of either the power demands or the task arrival process. Further, when energy can be sold from the battery to the grid, we develop an alternate algorithm *JTSES*. The performance gap between *JTSES* and the optimum is shown to diminish as energy selling price approaches the electricity price.

APPENDIX

Proof: First, By squaring Eqn. (13) and noting that $\max[x, 0]^2 \leq x^2$, we have

$$\begin{aligned} & \frac{1}{2}Q(t+1)^2 - \frac{1}{2}Q(t)^2 \\ & \leq \frac{1}{2}\left(\sum_{i=1}^{n_t} U_i^t(s_i^t) - \alpha\right)^2 + Q(t)\left(\sum_{i=1}^{n_t} U_i^t(s_i^t) - \alpha\right) \\ & \leq \frac{1}{2}n_{max}^2 U_{max}^2 + \frac{1}{2}\alpha^2 + Q(t)\left(\sum_{i=1}^{n_t} U_i^t(s_i^t) - \alpha\right). \end{aligned}$$

Similarly, by Eqn. (2), we have

$$\begin{aligned} & \frac{1}{2}(B(t+1) - \theta)^2 - \frac{1}{2}(B(t) - \theta)^2 \\ & \leq \frac{1}{2}r(t)^2 + \frac{1}{2}b(t)^2 + (B(t) - \theta)(r(t) - b(t)) \\ & \leq \frac{1}{2}r_{max}^2 + \frac{1}{2}b_{max}^2 + (B(t) - \theta)(r(t) - b(t)). \end{aligned}$$

Thus, we have

$$\begin{aligned} L(t+1) - L(t) \\ \leq \frac{1}{2} n_{max}^2 U_{max}^2 + \frac{1}{2} \alpha^2 + Q(t) \left(\sum_{i=1}^{n_t} U_i^t (s_i^t) - \alpha \right) \\ + \frac{1}{2} r_{max}^2 + \frac{1}{2} b_{max}^2 + (B(t) - \theta)(r(t) - b(t)). \end{aligned}$$

Taking expectations on both sides conditioning on $Z(t)$, it yields the result. ■

REFERENCES

- [1] S. Chen, P. Sinha, and N. Shroff, "Scheduling heterogeneous delay tolerant tasks in smart grid with renewable energy," in *Proceedings IEEE Conference on Decision and Control (CDC)*, 2012. To appear.
- [2] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, pp. 18–28, 2010.
- [3] M. Liserre, T. Sauter, and J. Hung, "Future Energy Systems: Integrating Renewable Energy Sources into the Smart Power Grid Through Industrial Electronics," *IEEE Industrial Electronics Magazine*, vol. 4, pp. 18–37, 2010.
- [4] I. Koutsopoulos and L. Tassiulas, "Control and optimization meet the smart power grid scheduling of power demands for optimal energy management," *Computing Research Repository*, vol. abs/1008.3, 2010.
- [5] M. Neely, A. Tehrani, and A. Dimakis, "Efficient algorithms for renewable energy allocation to delay tolerant consumers," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 549–554, oct. 2010.
- [6] A. Papavasiliou and S. Oren, "Supplying renewable energy to deferrable loads: Algorithms and economic analysis," in *Power and Energy Society General Meeting, 2010 IEEE*, pp. 1–8, july 2010.
- [7] A.-H. Mohsenian-Rad, V. 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, jan. 2010.
- [8] T. Facchinetti and M. L. D. Vedova, "Real-Time Modeling for Direct Load Control in Cyber-Physical Power Systems," *IEEE Transactions on Industrial Informatics*, vol. 7, pp. 689–698, 2011.
- [9] Z. Li, P.-C. Huang, A. K. Mok, T. Nghiem, M. Behl, G. Pappas, and R. Mangharam, "On the feasibility of linear discrete-time systems of the green scheduling problem," in *Proceedings of the 2011 IEEE 32nd Real-Time Systems Symposium, RTSS '11*, (Washington, DC, USA), pp. 295–304, IEEE Computer Society, 2011.
- [10] S. Chen, P. Sinha, and N. B. Shroff, "Energy Trading in the Smart Grid: From End-users Perspective." <http://www.ece.osu.edu/~chems/cheninfocom13.pdf>. preprint.
- [11] "California iso open access same-time information system (oasis) hourly average energy prices." <http://oasis.caiso.com>.
- [12] M. He, S. Murugesan, and J. Zhang, "Multiple Timescale Dispatch and Scheduling for Stochastic Reliability in Smart Grids with Wind Generation Integration," *Computing Research Repository*, vol. abs/1008.3, 2010.
- [13] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.



Prasun Sinha (S'99-M'03-SM'10) received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology Delhi, New Delhi, India, in 1995, the M.S. degree in computer science from Michigan State University (MSU), East Lansing, in 1997, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign (UIUC) in 2001. Currently, he is an Associate Professor with the Department of Computer Science and Engineering, The Ohio State University (OSU), Columbus. From 2001 to 2003, he was a Member of Technical Staff with Bell Labs, Holmdel, NJ. His research focuses on ubiquitous networking. His awards includes the OSU Lumley Research Award in 2009, the NSF CAREER Award in 2006, the UIUC Ray Ozzie Fellowship in 2000, the UIUC Mavis Memorial Scholarship in 1999, and the MSU Distinguished Academic Achievement Award in 1997.



Ness B. Shroff received his Ph.D. degree from Columbia University, NY in 1994 and joined Purdue university immediately thereafter as an Assistant Professor. At Purdue, he became Professor of the school of ECE in 2003 and director of CWSA in 2004, a university-wide center on wireless systems and applications. In July 2007, he joined the ECE and CSE departments at The Ohio State University, where he holds the Ohio Eminent Scholar Chaired Professorship of Networking and Communications. His research interest are in fundamental problems in communication, sensing, social, and cyberphysical networks. Dr. Shroff is a Fellow of the IEEE, an NSF CAREER awardee, and a number of his conference and journals papers have received best paper awards.



Shengbo Chen received his B. E. and M. E. degrees in the department of Electronic Engineering from Tsinghua University, Beijing, China in 2006 and 2008, respectively. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering at The Ohio State University. His research interests include resource allocation in rechargeable sensor networks, scheduling policy design in smart grid. He is a student member of the IEEE.

SMART GRID COST OPTIMIZATION USING GENETIC ALGORITHM

Gundi Anny Mary¹, R.Rajarajeswari²

¹M. Tech Student, Electrical and Electronics Engineering, SRM University, Chennai

²Assistant Professor, Electrical and Electronics Engineering, SRM University, Chennai

Abstract

Formerly, energy had been inexpensive and management of energy was efficient and was limited to elementary considerations. In the current scenario, due to a rapid increase in demand, complexity of the electrical network, probability of contingency and electricity cost have equally increased. In the recent past, Smart Grids are proven to be the best way to minimize these problems in an easier and smart way. Smart grid is defined as an electric network which has information technology fused to it. This paper proposes a way to reduce the total electricity cost in a smart grid using Genetic Algorithm. The system considered has renewable energy and battery banks apart from the grid to meet the demand. Short term time averaged electricity cost is formulated as an objective for optimization by GA with discharge of battery, energy from the grid to charge battery and meet load etc. as decision variables. The optimization problem is run for a 24 hours data of renewable input, real-time electricity price and load using MATLAB software; and the obtained results are furnished.

Keywords: Smart grid, optimization of grid, Genetic Algorithm optimization, real time pricing, energy storage.

-----***-----

1. INTRODUCTION

Smart grid is a network with optimization techniques which minimizes network losses, voltage levels, increases reliability, and improves management by using real - time measurements. It is a system that uses sensors and has computing control and ability to integrate the users connected to it. The smart grid is about operating transmissions and monitoring to ease the connection and the action of generators. It is beneficial at both the grid side and demand side in terms of energy efficiency. The essential feature of the computerized smart grid is its automation technology, which adjusts and alters a single or millions of equipments from a salient position. Smart grid would also make the integration of the intermittent renewable energy sources and electric vehicles on to the grid easier.

There are two kinds of energy demands in smart grid, namely elastic and inelastic energy demands. This paper considers a smart grid with an inelastic demand, renewable sources of energy and energy storing battery banks; and proposes to minimize the short term time averaged cost of electricity by considering the discharge of battery, energy drawn from grid to meet the load, energy drawn from grid to charge batteries and a control variable – which decides the amount of renewable energy to be used – as four decision variables. The multi-variable single-objective optimization is performed using Genetic Algorithm in MATLAB software for a real-time data of electricity-price, renewable energy in-put and the total energy demand of the utilities. Optimal values of the time averaged electricity cost for different battery storage capacities are obtained.

The detailed modeling of the smart grid system considered is given in the next section, followed by the stochastic problem formulation, the detailed optimization algorithm, system data and the results obtained.

2. SYSTEM DESCRIPTION

This paper considers a smart grid system which also includes renewable energy generation by the consumers and a battery bank at the consumers' end for energy storage; apart from the consumers linked to the power grid. The schematic of the system considered [18] is shown in Fig. 1. The renewable energy generated is stored in the energy storage device i.e., battery bank. The battery operating life will be affected as it gets charged and discharged continuously. To prevent the battery from this damage of overcharging and over-discharging a controller is used. The controller controls the action of battery and regulates the charging and discharging of the battery. An inverter is connected to convert DC to AC before supplying it to the appliances. The detailed description of each component of the system is given in the following subsections.

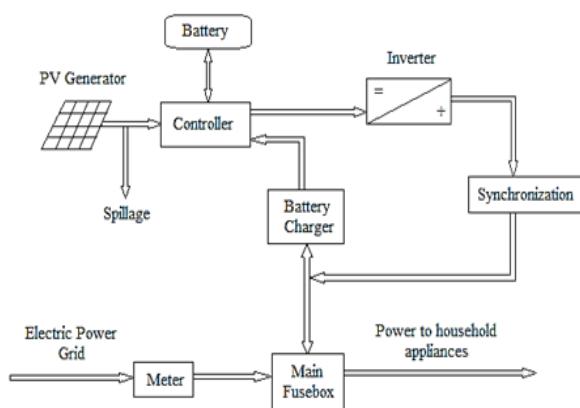


Fig -1: Schematic of the Smart Grid

2.1 Renewable Energy Generation

The renewable energy source considered in this system is solar energy. The detailed modeling of the photovoltaic panels is not dealt with in this paper and hence it is assumed that the output from the renewable source of energy is available for each time slot t (i.e., 1 hour). $S(t)$ is considered to be the amount of renewable energy generated in the time slot t . The amount of renewable energy used is decided by a decision variable, which is described in the next section and the amount of renewable energy generated in a day is also plotted in the further sections.

2.2 Battery Bank – Energy Storage

A battery bank is used at the consumer side to store the renewable energy generation; and use the stored energy at any particular time of energy requirement. In general, unused batteries discharge some amount of energy with the passing of time. For simplicity, this aspect of the battery is neglected. The amount of charge present in the battery is described by a term $SOC(t)$, which is the State of Charge of the battery at a time slot t . The SOC level of a battery in this paper is denoted by $B(t)$.

Most of the inelastic load, which is considered in this paper, is met by the battery. The battery is charged by a portion of the renewable energy generated by the consumer and also by the energy bought from the grid – whenever necessary and when the electricity price is low. The latter condition is attempted to be met by making the amount of power drawn from the grid to charge battery [$G_b(t)$] a decision variable in the optimization.

The amount of energy discharged from the battery, in each time slot t , to meet the load is taken as $D(t)$, which is limited by the maximum discharge rate D_{max} . The detailed equations for computing the SOC level for the next time step is given in the next section.

2.3 Energy Demand

This paper considers the demand at the consumer side to be inelastic, i.e., the energy demand arises only at particular instants of time and particular durations; and the energy demand at that particular time has to be met instantaneously. This is attempted to be done primarily by batteries; and when the battery is unable to meet the demand, energy is bought from the grid to directly meet the demand. The amount of energy bought from the grid to meet the demand is given by $G_l(t)$.

The optimization algorithm is executed for a 24 hour demand, which is furnished in the further sections. As the objective of the optimization is to minimize the electricity cost, energy to meet the unmet load is to be bought from the grid only when necessary and mostly when the electricity price is low. Like in the case of battery charging from the grid power, the above mentioned condition is attempted to be satisfied by considering $G_l(t)$ as a decision variable for the optimization.

3. PROBLEM FORMULATION

As mentioned earlier, the energy demand in smart grid can broadly be classified into two – elastic and inelastic. The energy demand appliances in household which are elastic are air conditioner, dish washer, heater etc, while some other energy demands of residential households are inelastic, such as lighting, television, computers etc. This paper considers inelastic energy demand, the energy requests of which must be met exactly in the time slot t (i.e., only whenever necessary). The energy demand, battery SOC level, electricity price, and renewable energy generation can be directly monitored by the splitter controller [18]. Another controller is used to determine the portion of renewable energy to be stored into the battery, which also monitors the status of renewable energy generation and battery SOC level.

3.1 Renewable Energy Generation

Assuming that the renewable energy generation, $S(t)$ is first stored in the battery before it can be used in the next time slot; a controller is used to regulate the portion $\gamma(t)$ of the generated renewable energy to be stored into battery for each slot t in order to prevent battery overflow. The remaining portion of renewable energy generated is spilled. Hence, we limit the control variable $\gamma(t)$ by

$$0 \leq \gamma \leq 1 \quad (1)$$

Moreover, the amount of renewable energy generation $S(t)$, is limited by a maximum value S_{max} . This is mathematically expressed as,

$$0 \leq S(t) \leq S_{max} \quad (2)$$

In order to utilize the time diversity of electricity price, it is assumed that in each time slot t , the amount of energy that can be drawn from the power grid to recharge the battery bank is $G_b(t)$. The state of charge (SOC) level $B(t)$ in the battery evolves according to the following equation:

$$B(t+1) = B(t) - D(t) + \gamma(t) S(t) + G_b(t) \quad (3)$$

Where, $D(t)$ is the amount of energy that is discharged from battery to supply demand in slot t . The SOC level in each time step is limited by the following constraint.

$$D(t) \leq B(t) \leq B_{\max} \quad (4)$$

Where, B_{\max} is the battery's maximum capacity. The amount of energy discharged from battery is further limited by its maximum discharge level D_{\max} , i.e.,

$$0 \leq D(t) \leq D_{\max} \quad (5)$$

The energy amount that can be drawn from the electric power grid to recharge battery for one time slot is also bounded $G_{b,\max}$ i.e.,

$$0 \leq G_b(t) \leq G_{b,\max} \quad (6)$$

The time-varying electricity price, $C(t)$, is sent to the consumer's smart meter by the utility company at the beginning of each time slot t . The cost of using renewable energy generated by the consumer itself is taken as zero. $G_l(t)$ is considered to be the power drawn from the electric power grid to directly supply the energy demand in slot t . Since the total electricity drawn from the electric power grid is the sum of the energies drawn to charge battery and meet the demand, the electricity cost for each time slot t is given by $[G_b(t)+G_l(t)]C(t)$.

3.2 Control Objective

Aiming at minimizing the total electricity cost for the customers, the short-term time averaged electricity cost [18], as described in equation (7), is considered as the optimization objective to be minimized.

$$P = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{C(t) [G_l(t) + G_b(t)]\} \quad (7)$$

The inelastic energy demand [18] generated in time slot t is given by the following equation

$$A_{\text{ine}}(t) = G_l(t) + D(t) \quad (8)$$

The problem can thus be formulated as the following stochastic optimization objective

$$\min_{D(t), G_b(t), G_l(t), \gamma(t)} P = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \{C(t) [G_l(t) + G_b(t)]\} \quad (9)$$

Subject to the following constraints

$$B(t+1) = B(t) - D(t) + \gamma(t) S(t) + G_b(t)$$

$$D(t) \leq B(t) \leq B_{\max}(t)$$

$$G_l(t) + D(t) = A_{\text{ine}}(t)$$

$$0 \leq D(t) \leq D_{\max}$$

$$0 \leq G_l(t) \leq G_{l,\max}$$

$$0 \leq G_b(t) \leq G_{b,\max}$$

$$0 \leq \gamma(t) \leq 1$$

$$X_{\text{ine}}(t) = B(t) - V_{\text{ine}} * C_{\max} - D_{\max} \quad (10)$$

Where, V_{ine} is a control parameter and $X_{\text{ine}}(t)$ is the shifted version of battery SOC level $B(t)$ which is used to ensure that the constraint on the SOC level of battery is satisfied. The shifted version of battery SOC level for the next time step is computed by the following equation

$$X_{\text{ine}}(t+1) = X_{\text{ine}}(t) - D_{\max} + \gamma(t) * S(t) + G_b(t) \quad (11)$$

The decision variables considered for the optimization are: $D(t)$ – the amount of energy discharged from the battery; $G_l(t)$ – the power drawn from the grid to directly supply the load; $G_b(t)$ – the power drawn from grid to charge the battery; and $\gamma(t)$ – the control parameter which decides the amount of renewable energy to be stored in battery. The optimization is performed using Genetic Algorithm (GA).

4. OPTIMIZATION ALGORITHM

There are two distinct types of optimization algorithms widely used today, namely – deterministic and stochastic algorithms. Deterministic algorithms use specific rules for moving from one solution to other; while stochastic algorithms use probabilistic translation rules. Genetic Algorithm (GA) is a direct, parallel, stochastic method for global search and optimization, which is used extensively for varied applications. GA is a part of the group of Evolutionary Algorithms (EA), which use the three main principles of the natural evolution: natural selection, reproduction and diversity of the species.

This paper proposes the minimization of the time averaged electricity cost with the decision variables as mentioned in the

previous section. The detailed algorithm of the multi-variable single objective Genetic Algorithm is described below.

1. Initialization:

A random initial population (size N) of individuals, x , is created within the bounds of each variable.

2. Evaluation:

Once the population is initialized or an offspring population is created, the fitness function (objective) is computed.

3. Selection:

Selection allocates more copies of those solutions with higher fitness values and thus imposes the survival-of-the-fittest mechanism on the candidate solutions. Many selection procedures namely roulette-wheel selection, stochastic universal selection, ranking selection and tournament selection

This paper uses Rank + Roulette wheel selection. The individuals are sorted based on their fitness values and are assigned a rank accordingly. The selection probability for each individual is calculated according the following non-linear function:

$$P = \beta(1 - \beta)(\text{rank} - 1) \quad (12)$$

There β is a user defined coefficient. The traditional roulette wheel selection is now used to select the best fit individuals.

4. Recombination:

Recombination, also called crossover, combines parts of two or more parental solutions to create new, possibly better solutions (i.e. offspring).

This paper uses Blend crossover (BLX) – α crossover operator. Considering two parents from the selection process, the offspring obtained after the crossover is given by

$$x_i^{(1,t+1)} = (1 - \gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)} \quad (13)$$

Where, $\gamma_i = (1 + 2\alpha)u_i - \alpha$, in which u_i is a random number between 0 and 1.

5. Mutation:

While recombination operates on two or more parental chromosomes, mutation locally but randomly modifies a solution.

This paper uses non-uniform mutation, according to which the mutated offspring is given by

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + \tau \left(x_i^{(U)} - x_i^{(L)} \right) \left(1 - r_i^{(1-t/t_{\max})^b} \right) \quad (14)$$

where τ takes a boolean value -1 or 1, each with a probability of 0.5, t_{\max} is the maximum number of allowed generations, and $b = 0.5$ is a user defined parameter.

6. Combined Population:

To preserve elitism, both parent and offspring population is combined and then sorted based on their fitness values. The first N individuals are chosen for the next generation.

7. Iterate:

Repeat steps 2–6 until termination condition is met (i.e., until fitness value converges).

5. INPUT DATA

The electricity price data used in this paper is collected from the California Independent System Operator CAISO consisting of an hourly average price $C(t)$ for 24 hours[19] and is illustrated in Fig. 3. The average renewable energy data used is taken from the Measurement and Instrumentation Data Center (MIDC) at National Renewable Energy Laboratory [20] and is shown in Fig- 4.

Depending upon the energy consumed by appliances during each time slot t , the inelastic load for 24 hours [21] is shown in Fig- 2.

Considering the parameters $D_{\max} = 30\text{KW}$, $G_{l,\max} = 30\text{ KW}$ and $G_{b,\max} = 20\text{ KW}$ [18], optimization using GA is performed and the results are furnished in the next section.

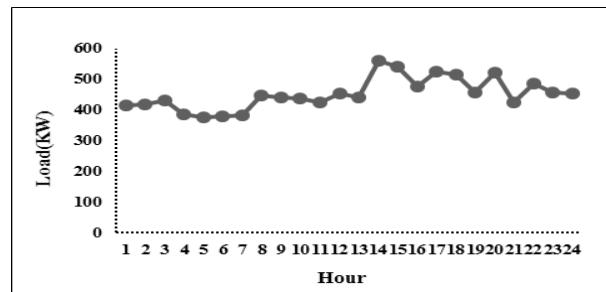


Fig- 2: Average hourly load for 24 hours

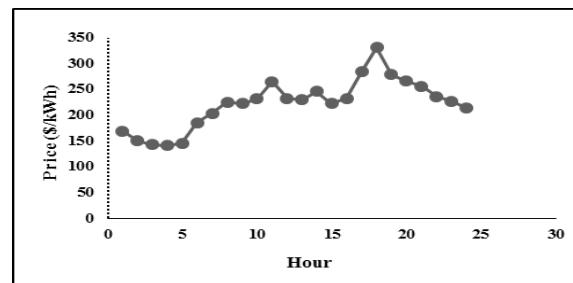
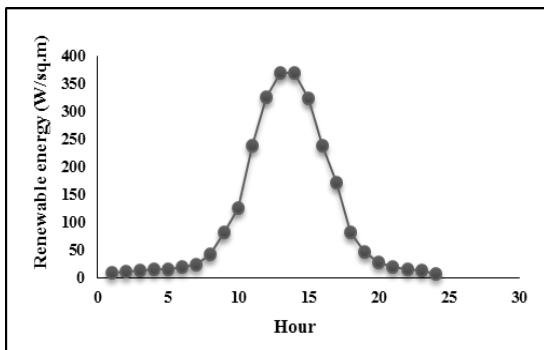


Fig- 3: Average hourly market price for 24 hours

**Fig-4:** Average hourly renewable energy for 24 hours

6. PERFORMANCE EVALUATION

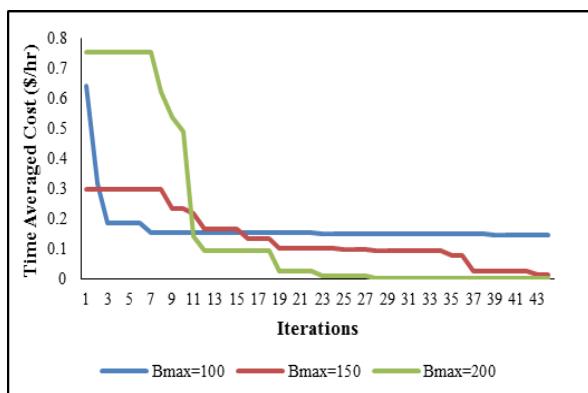
In this paper multi-variable, single objective Genetic Algorithm is proposed and simulated for the 24 hours data of renewable energy, electricity price and inelastic energy demand mentioned in the previous section. The algorithm is simulated using MATLAB to optimize the short-term time averaged electricity cost for different values of B_{max} . The best fitness values for $B_{max}=\{100,150,200\}$ KW-slot is tabulated in Table 1 and is illustrated in Fig- 5.

It can be observed from the results, that the total electricity cost is highest when $B_{max}=100$ and is lowest when $B_{max}=200$. It can thus be seen that the total electricity cost for consumers is reduced by increasing the capacity of the battery banks.

Table -1: Fitness values for different B_{max} value for inelastic energy demand

Iterations	Time Averaged Electricity Cost		
	$B_{max}=100$ KW	$B_{max}=150$ KW	$B_{max}=200$ KW
1	0.640871	0.298791	0.753732
2	0.317684	0.298791	0.753732
3	0.185219	0.298791	0.753732
4	0.185219	0.298791	0.753732
5	0.185219	0.298791	0.753732
6	0.185219	0.298791	0.753732
7	0.152103	0.298791	0.753732
8	0.152103	0.298791	0.621267
9	0.152103	0.232559	0.538477
10	0.152103	0.232559	0.488802
11	0.152103	0.21807	0.141082
12	0.152103	0.166326	0.091407
13	0.152103	0.166326	0.091407
14	0.152103	0.166326	0.091407

15	0.152103	0.166326	0.091407
16	0.152103	0.13321	0.091407
17	0.152103	0.13321	0.091407
18	0.152103	0.13321	0.091407
19	0.152103	0.100094	0.025175
20	0.152103	0.100094	0.025175
21	0.151068	0.100094	0.025175
22	0.151068	0.100094	0.025175
23	0.150033	0.100094	0.008972
24	0.149702	0.098631	0.008972
25	0.149702	0.095954	0.008972
26	0.148998	0.095954	0.008617
27	0.148998	0.095954	0.008617
28	0.148998	0.093998	0.000338
29	0.148998	0.093998	0.000338
30	0.148998	0.093884	0.000338
31	0.147963	0.093884	0.000338
32	0.147963	0.093338	0.000338
33	0.147963	0.091928	0.000338
34	0.147963	0.091815	0.000338
35	0.146929	0.075791	0.000338
36	0.146929	0.075256	0.000338
37	0.146929	0.025582	0.000338
38	0.146929	0.025582	0.000338
39	0.146535	0.025582	0.000338
40	0.145894	0.025582	0.000338
41	0.145894	0.025582	0.000338
42	0.145894	0.025582	0.000338
43	0.145303	0.012678	0.000338
44	0.145303	0.012678	0.000338

**Fig- 5:** Comparison of total cost for different B_{max} values with inelastic energy demand.

7. CONCLUSIONS

In this paper, the optimization of total electricity cost for the consumers with inelastic load in a smart grid is achieved by minimizing the short-term time averaged electricity cost. The inelastic load being time specific is supplied mostly by batteries; which is charged by the renewable energy generation and the grid. The optimization aims at charging the battery when electricity price is less and discharging the battery when electricity price is high. Optimization is performed by multi-variable, single objective Genetic Algorithm for a 24 hour data of real-time electricity price, renewable energy generation and total inelastic energy demand of the utilities. An optimized minimum cost is obtained for three cases of different battery capacities, and it is observed that the total electricity cost is lower for higher battery capacity and vice versa.

REFERENCES

- [1] Genetic Algorithms by Kumara Sastry, David Goldberg University of Illinois, USA Graham Kendall University of Nottingham, UK.
- [2] An Introduction to Genetic Algorithms by Paul Charbonneau.
- [3] D. Linden and T. B. Reddy, Battery Power and Products Technology, vol. 5, no. 2, pp. 10–12, March/April 2008.
- [4] M. Winter and R. Brodd, Chemical Reviews, vol. 104, 4245–4270, 2004.
- [5] Tackling Real Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis by F. HERRERA, M. LOZANO & J.L. VERDEGAY Department of Computer Science and Artificial Intelligence University of Granada, Avda. Andalucia 38, 18071 Granada, Spain.
- [6] Time-Varying Retail Electricity Prices: Theory and Practice by Severin Borenstein.
- [7] California ISO Open Access Same-Time Information System (OASIS), <http://oasis.caiso.com/>, 2012.
- [8] NREL: Measurement and Instrumentation Data Center, <http://www.nrel.govmidc/>, 2012.
- [9] R. Urgaonkar, B. Urgaonkary, M.J. Neely, and A. Sivasubramaniam, "Optimal Power Cost Management Using Stored Energy in Data Centers," Proc. ACM Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS '11), June 2011.
- [10] Dynamic Programming and Optimal Control Volume I THIRD EDITION by P. Bertsekas Massachusetts Institute of Technology.
- [11] Dynamic Programming and Optimal Control 3rd Edition, Volume II by Dimitri P. Bertsekas Massachusetts Institute of Technology.
- [12] Resource Allocation and Cross Layer Control in Wireless Networks by Leonidas Georgiadis, Michael Neely and Leandros Tassiulas.
- [13] Optimal Power Cost Management Using Stored Energy in Data Centers by Rahul Urgaonkar, Bhuvan Urgaonkar, Michael J. Neely, Anand Sivasubramanian.
- [14] Genetic Algorithm Performance with Different Selection Strategies in Solving TSP, Proceedings of the World Congress on Engineering 2011 Vol II WCE 2011, July 6 - 8, 2011, London, U.K by Noraini Mohd Razali, John Geraghty.
- [15] R. L. Wang, A genetic algorithm for subset sum problem, Neurocomputing, vol.57, pp.463-468, 2004.
- [16] Selection Schemes, Elitist Recombination and Selection Intensity by Dirk Thierens Department of Computer Science Utrecht University.
- [17] BASIC- A genetic algorithm for engineering problems solution, Institute of Chemical Engineering, Bulgarian Academy of Sciences by Elisaveta G. Shopova, Natasha G. Vaklieva-Bancheva.
- [18] Optimal Power Management of Residential Customers in the Smart Grid by Yuanxiong Guo, Miao Pan and Yuguang Fang.
- [19] California ISO Open Access Same-Time Information System (OASIS), <http://oasis.caiso.com/>, 2012.
- [20] NREL: Measurement and Instrumentation Data Center, <http://www.nrel.govmidc/>, 2012.
- [21] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Energy-Information Transmission Tradeoff in Green Cloud Computing," Proc. IEEE Globe Com '10, Mar. 2010.

BIOGRAPHIES



Gundi. Anny Mary received B. Tech degree in Electrical and Electronics Engineering from Jawaharlal Nehru Technological University Kakinada Andhra Pradesh, India, in May 2012 is pursuing M. Tech degree in Power Systems from SRM University, Tamilnadu, India, 2012-14. Her research interests includes stochastic network optimization and smart grids.



R. Rajarajeswari received her B.E and M.E degrees in Electrical and Electronics Engineering from Manonmaniam Sundaranar University of Technology, Tamilnadu and Anna University of Tamilnadu, India, in July 2001 and April 2006, respectively. She has published 1 international and 4 national papers in reputed journals and conferences.

Smart Hybrid Battery Enclosure BE13200



Fan

Dual variable speed fans controlled by charge/discharge rate.

IP54 rated

Can be installed inside or outside.

Prewired

For safe and fast installation.

Front panels

Powder coated, durable and die cast aluminium covers.

Capacity

Up to 13.2kWh.

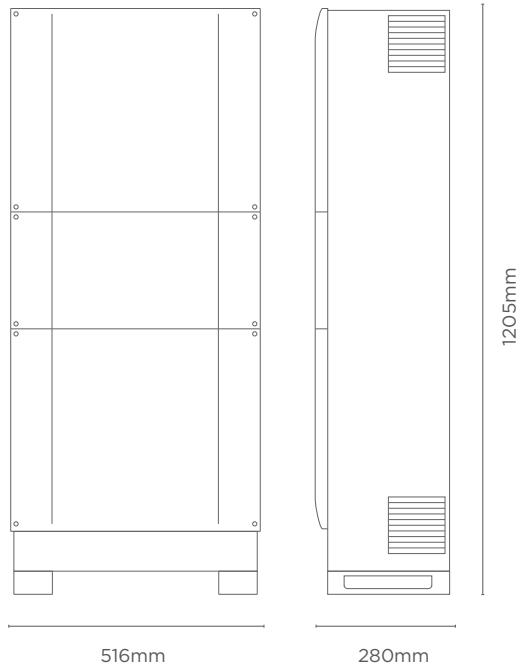
Energy for the future

Redback Technologies has a vision to enable every household and business to be entirely powered by low cost renewable energy all day, every day. The Redback Smart Hybrid System is the platform which facilitates all energy users to participate in the network of the future through clean, efficient and smart energy management. Redback Technologies, helping the world switch on renewable energy today for a cleaner tomorrow.

Energy flow



Dimensions



Specifications

Battery enclosure	BE13200
Number of battery units	Up to 4 x 19" rack mountable battery packs
Storage capacity	Up to 13.2kWh (4 x 3.3kWh LG batteries) / 9.6kWh (4 x 2.4kWh Pylon Tech batteries)
Battery voltage	48V DC nominal / 60V DC maximum
Battery chemistry	Lithium-ion with BMS
Access type	Removable front panels
Cable specification	
Battery cable rating	4 x 65A
Battery cable type	8 AWG (8.36mm ²)
Battery cable termination (battery enclosure)	Surlok amphenol connector
Battery cable termination (inverter)	Amphenol H4 (65A)
BMS cable type	Depends on battery type
BMS cable termination	Refer to battery enclosure installation manual
Ventilation specification	
Ventilation type	Passive and active cooling
Ventilation control	Smart temperature control
Number of fans	2
Fan power	48V DC / 0.13A per fan
Fan activation temperature	Variable depending on charge/discharge
Incoming ventilation aperture	288cm ² with washable filter
Outgoing ventilation aperture	288cm ² with washable filter
Passive airflow rate	30cm ³ /min
Active airflow rate	320cm ³ /min
General Data	
External dimension (W x H x D)	W 516mm x H 1205mm x D 280mm (with feet)
Mounting and weight - empty	32kg rear fixing
Mounting and weight - with batteries	130kg typical
Ambient temperature range	Based on battery specification
Environmental protection rating	IP54 - protected from rain, splashing and spraying
Noise emissions	Less than 25dB
Warranty	5 Years
Construction	Powder coated steel chassis
Finish	Sealed, powder coated front covers and chassis
Supply	Ships pre-assembled
Maintenance	Externally serviceable dust filters

POWERWALL

Tesla Powerwall 2 is a fully-integrated AC battery system for residential or light commercial use. Its rechargeable lithium-ion battery pack provides energy storage for solar self-consumption, load shifting, backup, and off-grid use.

Powerwall's electrical interface provides a simple connection to any home or building. Its revolutionary compact design achieves market-leading energy density and is easy to install, enabling owners to quickly realise the benefits of reliable, clean power.



PERFORMANCE SPECIFICATIONS

AC Voltage (Nominal)	208 V, 220 V, 230 V, 100/200 V, 120/240 V
Feed-In Type	Single Phase
Grid Frequency	50 Hz
Total Energy ¹	14 kWh
Usable Energy ¹	13.5 kWh
Real Power, max continuous ²	5 kW (charge and discharge)
Apparent Power, max continuous ²	5 kVA (charge and discharge)
Imbalance for Single-Phase Loads	100%
Power Factor Output Range	+/- 1.0 adjustable
Depth of Discharge	100%
Internal Battery DC Voltage	50 V
Round Trip Efficiency ^{1,3}	> 90%
Warranty	10 years

¹Values provided for 25°C, 3.3 kW charge/discharge power.

²Values region-dependent.

³AC to battery to AC, at beginning of life.

COMPLIANCE INFORMATION

Safety	UL 1642, UL 1741, UL 1973, UL 9540, UN 38.3, IEC 62109-1, IEC 62619, CSA C22.2.107.1
Grid Standards	Worldwide Compatibility
Emissions	FCC Part 15 Class B, ICES 003, EN 61000 Class B
Environmental	RoHS Directive 2011/65/EU, WEEE Directive 2012/19/EU, 2006/66/EC
Seismic	AC156, IEEE 693-2005 (high)

MECHANICAL SPECIFICATIONS

Dimensions	1150 mm x 755 mm x 155 mm
Weight	125 kg
Mounting options	Floor or wall mount

ENVIRONMENTAL SPECIFICATIONS

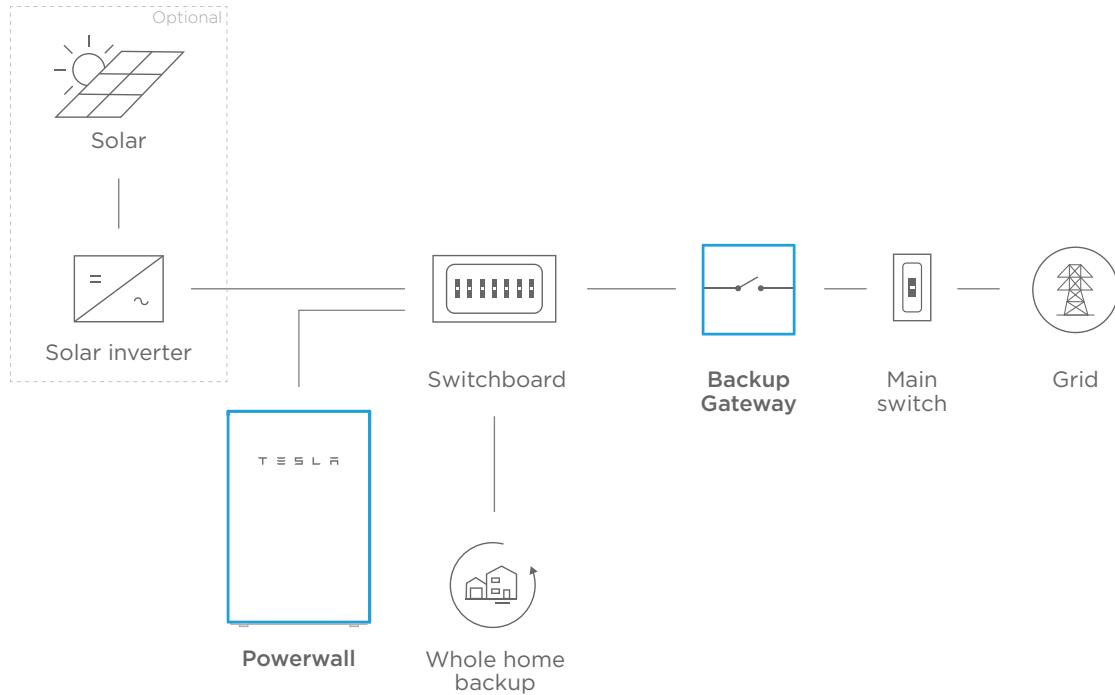
Operating Temperature	-20°C to 50°C
Operating Humidity (RH)	Up to 100%, condensing
Maximum Altitude	3000 m
Environment	Indoor and outdoor rated
Ingress Rating	IP67 (Battery & Power Electronics) IP56 (Wiring Compartment)
Noise Level @ 1m	< 40 dBA at 30°C

BACKUP GATEWAY SPECIFICATIONS

Dimensions	691 mm x 378 mm x 129 mm
Weight	16.4 kg
Disconnect Current	200 A
Ingress Rating	IP44
User Interface	Tesla App
Connectivity	Wi-Fi, Ethernet, 3G ⁴
AC Meter	Revenue grade
Operating Modes	Support for solar self-consumption, load shifting, backup, and off-grid use
Backup Operation	Automatic disconnect for seamless backup transition
Modularity	Supports up to 10 AC-coupled Powerwalls

TYPICAL SYSTEM LAYOUTS

WHOLE HOME BACKUP



PARTIAL HOME BACKUP

