Energy Provisioning in Stand-alone and Grid-Connected Solar Powered Networks

ENERGY PROVISIONING IN STAND-ALONE AND GRID-CONNECTED SOLAR POWERED NETWORKS

BY

MOHAMMAD SHEIKH ZEFREH, B.Sc., M.Sc.

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2013)	McMaster University
(Electrical & Computer Engineering)	Hamilton, Ontario, Canada

TITLE:	Energy Provisioning in Stand-alone and Grid-Connected Solar Powered Networks
AUTHOR:	Mohammad Sheikh Zefreh M.Sc., (Electrical Engineering) Isfahan University of Technology, Isfahan, Iran
SUPERVISOR:	Professor Terence D. Todd
NUMBER OF PAGES:	xviii, 126

Abstract

Solar energy is a clean and abundant renewable energy source which is currently used in many types of photovoltaic (PV) designs. In practical PV systems, solar panels are used to harvest solar energy and convert it into a usable form of electricity. Due to the intermittent nature of solar energy input however, battery storage, in combination with solar panels, must be used to provide an uninterrupted source of power.

The process of assigning solar panel and battery configurations for a PV system is referred to as energy resource provisioning. Unfortunately, energy provisioning costs are still relatively high, and this is one of the main obstacles that inhibits the adoption of solar power for many applications. These costs however, can be substantially reduced through cost-efficient resource provisioning methods. The focus of this thesis is on the development of efficient algorithms and energy management methods that will reduce energy provisioning costs in solar powered systems.

First, we consider resource provisioning in solar powered wireless mesh networks. In practical solar powered systems, there are usually restrictions in the way that the mesh nodes can be positioned, and this results in a time-varying and node-dependent attenuation of the available solar energy. Unfortunately, conventional resource provisioning methods cannot take this into account and therefore the deployed system may be unnecessarily expensive. In this part of the thesis, the resource provisioning problem is considered from this point of view. We first review conventional resource provisioning mechanisms and give an example which shows the value of introducing positional solar insolation awareness. A Position Aware Provisioning (PAP) algorithm is then introduced that takes known positional variations into consideration when performing the energy provisioning. Simulation results show that reductions in total network provisioning cost can be obtained using the proposed methodology compared to conventional algorithms.

In the second part of the thesis, we consider communication infrastructure that is operated from the power grid with a solar powered addition. Resource provisioning and energy management algorithms are introduced to minimize the capital expenditure (CAPEX) and operating expenditure (OPEX) costs. We first derive lower bounds on the costs using a linear programming (LP) formulation where solar components are sized using solar insolation and projected loading data. A variety of different node configurations are considered. Three energy scheduling algorithms are then introduced to optimize online OPEX costs, namely, Grid Purchase Last (GPL), Solar Load Optimization (SLO) and Solar Load Simulation (SLS) algorithms. Simulation results show the extent to which a solar powered add-on can reduce total cost.

Finally, we consider solar powered systems where part of their energy demands are deferrable, up to some maximum tolerable delay. The objective is to exploit the flexibility of deferrable energy demands in a way that decreases the total provisioning cost. A mixed integer linear optimization program is derived which gives a lower bound on the provisioning cost. A Delay Aware Provisioning (DAP) algorithm is then proposed to determine practical cost-efficient energy provisioning. The performance of DAP is compared to the provisioning bound and the conventional Stand-alone Node Provisioning (SNP) algorithm. Results are presented which show the significant provisioning cost savings that can be obtained.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Terence D. Todd, who patiently provided advice, guidance and encouragement throughout my doctoral program at McMaster University. It has been an honor to do my Ph.D. under his supervision.

I extend my warm thanks to the members of my Ph.D. supervisory committee and to the members of the examining committee for reading my thesis and for their valuable suggestions. My sincere thanks also goes to Dr. George Karakostas for his helpful comments on Chapter 4.

I would like to thank Dr. Amir A. Sayegh and Dr. Ghada H. Badawy whose helpful comments and assistance kept me going at the beginning of my Ph.D. research. I would also like to acknowledge the members of the Wireless Networking Group at McMaster University for sharing their valuable information and for providing a friendly working environment. I take this opportunity to particularly thank Hadi Meshgi, Abdulla A. Hammad, Hanan Hassanein, Naby Nikookaran, Maryam Mohseni and Abdulelah Alganas for their valuable help during my Ph.D.

My Ph.D. would not have been possible without my training at Isfahan University of Technology in Iran. I owe a great deal of gratitude to the many scholars at this university for encouraging me to pursue my doctorate.

It is difficult for me to express the level of encouragement and emotional support that

was consistently provided by my parents, whom I greatly respect.

Finally, thank you to my wife, Somayeh, for her selfless support, love and confidence in me.

Abbreviations

AC	Alternative Current
BG	Battery/Grid Configuration
BS	Basestation
BUP	Bandwidth Usage Profile
CAPEX	Capital Expenditure
CPN	Continuously Powered Node
DAP	Delay Aware Provisioning Algorithm
DC	Direct Current
EARTH	Energy Aware Radio and Network Technologies Project
ER	Energy Revenue
GO	Grid Only Configuration
GPL	Grid Purchase Last Algorithm
GSMa	GSM Association

IEAP	Iterative Energy Aware Provisioning
IF	Solar Insolation Factor
LP	Linear Programming
MILP	Mixed Integer Linear Program
OPEX	Operating Expenditure
PAP	Position Aware Provisioning Algorithm
PBG	Solar Panel/Battery/Grid Configuration
PG	Solar Panel/Grid Configuration
PHEV	Plug-in Hybrid Electric Vehicles
PV	Photo-Voltaic
RSD	Relative Standard Deviation
SLO	Solar Load Optimization Algorithm
SLS	Solar Load Simulation Algorithm
SNP	Stand-Alone Node Provisioning
SNR	Signal to Noise Ratio
SO	Solar Only Configuration
SPP	Shortest Path Resource Provisioning
TOU	Time of Use

WLAN	Wireless Local Area Networks
WMN	Wireless Mesh Network

Contents

List of Tables

List of Algorithms

List of Figures

Chapter 1

Introduction

1.1 Overview

Solar energy is a promising alternative compared to non-renewable and fossil fuel energy sources. Due to the nature of solar energy however, a combination of solar panels and battery storage must be used to exploit solar energy in most practical applications. The process of assigning solar panel and battery configurations to a photovoltaic node is referred to as energy resource provisioning. Unfortunately, solar panels and batteries are still relatively expensive, and for this reason, these costs are often an impediment to the use of solar energy in many applications. In this thesis we consider methods for decreasing the provisioning costs in wireless solar powered mesh networks and solar powered infrastructure.

1.2 Energy Provisioning in Stand-alone and Grid-Connected Solar Powered Networks

We focus on solar panel positioning in wireless solar powered mesh networks, nodes that are powered by a combination of solar and power grid energy, and solar powered nodes that have deferrable energy demands. In these types of systems we present provisioning algorithms which can decrease the costs of solar powered implementations. The proposed methods are summarized as follows.

- In the first part of the thesis, solar node positioning in wireless solar mesh networks is considered. In practical systems there are usually restrictions in the way that the solar nodes can be positioned, and this results in a time-varying and node-dependent attenuation of the available solar energy. Unfortunately, conventional resource provisioning methods cannot take this into account and therefore the deployed system may be unnecessarily expensive. In this part of the thesis, we consider the resource provisioning problem from this point of view. We first give an example which shows the value of introducing positional solar insolation awareness. A provisioning algorithm is then introduced that takes known positional variations into consideration when performing the energy provisioning. Compared with conventional methods, this takes the relative costs of provisioning each node into account when satisfying the desired design load objective. A variety of results are then presented which show that reductions in total network provisioning cost can be obtained using the proposed methodology compared to conventional algorithms. The proposed algorithm also performs very well compared with a linear programming formulation which gives lower bounds on the node resource provisioning cost assignments.

- In the second part of the thesis we consider solar powered communication infrastructure. This type of infrastructure is already common in situations where conventional power grid connections are unavailable or prohibitively expensive. The cost of solar power components will eventually decrease to the point where they can be routinely used as an add-on for infrastructure that is operated from the power grid. In this part of the thesis, we consider the operating and capital expenditure costs of providing these types of solar powered additions. The capital expenditure (CAPEX) costs are those associated with provisioning the solar power components and are selected using an offline design optimization. Once the solar add-on is designed and deployed, the node starts its online operation phase where it incurs ongoing operating expenditure (OPEX) costs. These costs are associated with the purchase of power grid energy when the node is running an online energy scheduling algorithm. We first derive lower bounds on the costs using a linear programming (LP) formulation, where the solar power components are sized using historical solar insolation traces and projected loading data. Different node add-ons are considered, which result in various configurations, including, solar panel/battery/grid (PBG), solar panel/grid (PG), and battery/grid (BG) scenarios. Three energy scheduling algorithms are then introduced to optimize online OPEX costs, namely, Grid Purchase Last (GPL), Solar Load Optimization (SLO) and Solar Load Simulation (SLS) algorithms. A variety of results are presented that show the extent to which a solar powered add-on can reduce total cost. These results also show that the SLO and SLS algorithms give results that are close to the lower bounds in many situations. The case where revenue can be derived from returning unused energy to the power grid is also considered.

- Finally, in the last part of the thesis, the energy provisioning problem is considered for solar nodes that have deferrable energy demands. The objective is to reduce the capital cost of a stand-alone solar powered system via management of its energy consumption. Energy demand is split into either "non-deferrable" or "deferrable" components. While non-deferrable demands are defined as the part of the load that must be satisfied immediately, deferrable demands are those that are delay tolerant, i.e., they do not need to be satisfied immediately upon their request and their activation time can be delayed from several minutes to several hours. The offline problem is formulated using a Mixed Integer Linear Programming (MILP) optimization. A realizable algorithm called Delay Aware Provisioning Algorithm (DAP) for offline provisioning is then introduced. Simulation results indicate that using the proposed algorithm will lead to a significant improvement in the total provisioning cost.

1.3 Thesis Organization

This thesis is organized as follows.

Chapter 2 introduces the fundamentals of solar energy and photovoltaic systems. The process of solar energy creation is presented starting from the creation of photons in the Sun to converting solar radiation to electrical power in the solar panels. The components needed to harvest and use solar energy are also discussed.

In Chapter 3 the problem of solar node positioning and its effect on the total network provisioning cost is considered. The motivation for the Position Aware Provisioning (PAP) algorithm is presented, and the problem is formulated via a linear optimization. Two simplified versions of energy aware provisioning, called IEAP and LAP are presented, and the conventional Shortest Path Provisioning (SPP) algorithm is introduced. The performance of the PAP algorithm is compared with SPP, IEAP, LAP and the provisioning bound using computer simulation.

In Chapter 4 energy provisioning and operating costs in hybrid solar powered infrastructure is considered. An optimization problem is formulated for different configurations, and three online energy scheduling algorithms named GPL, SLO and SLS are presented. Extensive simulation results evaluate and compare the performance of the proposed online scheduling algorithms and the provisioning bound.

The last contribution of the thesis is presented in Chapter 5 where solar powered nodes with deferrable energy demands are considered. The problem is formulated through a mixed integer and linear program, and the Delay Aware Provisioning (DAP) algorithm is introduced. The performance of DAP is compared with the conventional Stand-alone Node Provisioning (SNP) algorithm using several simulation experiments.

The conclusions of the thesis and possible future work are given in Chapter 6.

Chapter 2

Background

2.1 Introduction

In this chapter an overview is presented of solar radiation, photovoltaic (PV) technology, and some solar powered applications. The chapter starts with an introduction to the fundamentals of solar energy, followed by an overview of photovoltaic cells. It then explains the basics of the photovoltaic effect, and briefly discusses some of the other components needed for practical PV systems. Finally, stand-alone and power grid connected PV systems are considered, including a discussion of solar powered wireless mesh networks.

2.2 Solar Energy Fundamentals

The Sun is the most abundant and sustainable source of energy, and provides over 150,000 TW of power to the Earth. About half of this power reaches the Earth's surface, and the rest is absorbed or reflected back into outer space. However, only a small fraction of this amount would be enough to satisfy world-wide energy demands (Camacho *et al.*, 2012).

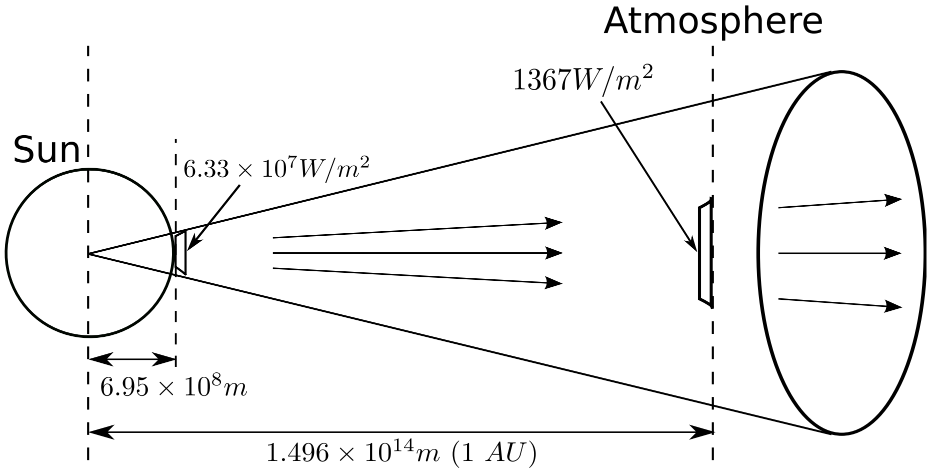


Figure 2.1: Energy Divergence from the Sun to the Earth

The Earth receives its solar radiation from nuclear fusion reactions accruing at the core of the Sun, where hydrogen atoms are fused into helium. The Sun's core is mostly composed of helium (65%), while hydrogen is reduced to 35% because of it's consumption in fusion reactions. The intense gravity at the core of the Sun provides a large enough force for these fusion reactions, which results in a conversion rate of about 700 million tons of hydrogen to helium each second. This results in a 15 million degree Celsius core temperature. At these temperatures, photons are emitted, and it takes about 100,000 years for them to reach the surface of the Sun, but only 8 minutes to travel the 149.6 million kilometers to the Earth (Camacho *et al.*, 2012).

The rate at which solar energy reaches a unit area at the Earth is called *solar irradiance* or *solar insolation* and is measured in W/m^2. The integral over time of solar irradiance is called *solar radiation* or *solar irradiation* and is measured in J/m^2. Often, solar irradiance is also referred to as solar radiation with the same units (W/m^2) (Camacho *et al.*, 2012).

The solar energy outside the atmosphere is called *extraterrestrial radiation*. Since the total power emitted by the Sun's surface is about 63.3 MW/m^2, and considering the radius of the Sun and its average distance with the Earth, the extraterrestrial radiation power falling on 1 m^2 of surface area is 1367 W which is known as the *solar constant* (Figure 2.1)(Camacho *et al.*, 2012).

When the solar radiation passes through the Earth's atmosphere, it is attenuated by reflection into space, absorbtion in the atmosphere, and scattering. The direct component of the solar radiation and the scattered component that reaches the Earth's surface are called *direct solar radiation* and *diffuse radiation*, respectively. To measure the received solar radiation on the Earth's surface, *Air Mass 1.5 standard* (AM1.5) is defined by the American Society for Testing and Materials (ASTM). In addition to defining the parameters for the atmosphere's impurity, the standard assumes that the receiving surface is an inclined plane at a 37^o tilt toward the equator, and the solar zenith angle is $48^o19'$. (Stine and Geyer, 2001), (Paulescu *et al.*, 2013).

2.3 Photovoltaic Cells

Solar radiation can be converted directly into electricity using photovoltaic (PV) cells. PV cells are constructed from a variety of semiconductors, but mostly from silicon (Si), cadmium sulphide (CdS), copper sulphide (Cu_2S) and gallium arsenide (GaAs). Cells are usually grouped into modules that produce electrical current when illuminated. A solar panel is a set of solar modules that are electrically connected in series or parallel to produce larger voltages or currents, and are mounted on a supporting structure (Kalogirou, 2009). To understand the operation of PV cells we first briefly review the basics of semiconductors.

It is well known that atoms consist of a nucleus and its orbiting electrons. Electrons of an isolated atom, according to quantum mechanics, can have only specific discrete or quantized energy levels and can occupy only certain orbits. Atoms have certain energy levels called *energy bands*. Electrons are free to leave some energy bands, and they cannot leave some others. The electrons in the *valence band* are the only ones that interact with other atoms. Some electrons in the valence band may possess a lot of energy, which enables them to jump into a higher band called the *conduction band*. These electrons are responsible for the conduction of electricity and heat.

Silicon is an example of a semiconductor that has four electrons in its valance band, and where each valance electron is bonded with a valance electron in the neighboring atoms. If silicon is impurified (doped) with an element like phosphorus which has five valance electrons, four of them are bonded with four valance electrons in the neighboring silicon atoms, and the remaining electron orbits around the phosphorus nucleus at very low temperatures, but diffuses through the crystal at ordinary temperatures. This creates an n-doped silicon crystal. On the other hand, if silicon impurified by an element such as boron, which has only three valance electrons, a p-type silicon is obtained. This incomplete bond has the ability to capture an electron. The captured electron leaves behind a “positive hole” that can be diffused through the crystal at ordinary temperatures. When n- and p-type semiconductors are joined together, i.e., a p-n junction is created, the excess electrons from the n-type diffuse across the p-n junction to the p-side to fill the holes in the p-type material, and the holes from the p-type material diffuse to the n-type side. The result is a net negative charge on the p-side and a net positive charge on the n-side of the p-n junction, which creates an electric field across the junction (Kalogirou, 2009). Solar cells are made of a p-n junction and can convert radiated solar energy into electrical current using the photovoltaic effect which is briefly discussed in the following section.

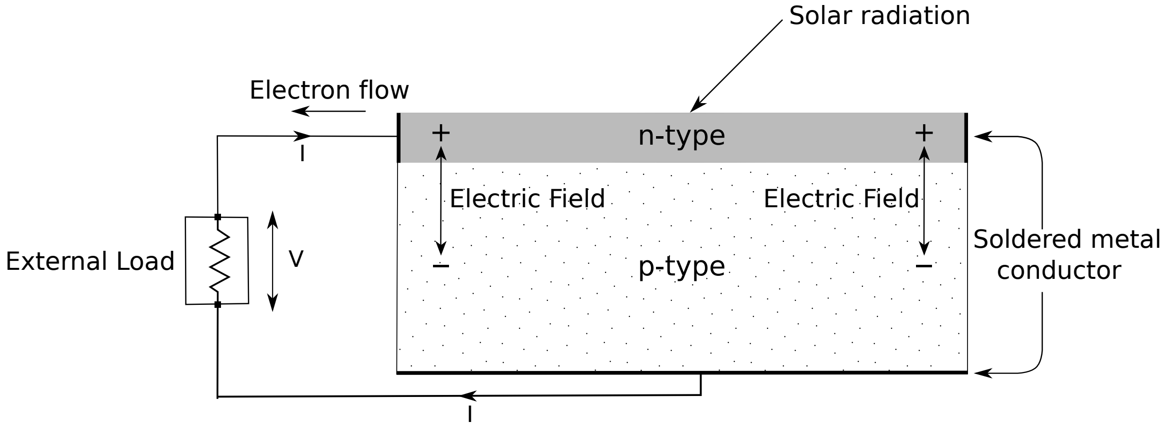


Figure 2.2: Photovoltaic Effect in a PV Cell

2.3.1 Photovoltaic Effect

When a photon hits a solar cell, it can be transmitted through the cell, reflected or absorbed. If the photon is absorbed by a valence electron, its energy is increased by the energy of the photon, and if this energy exceeds the band gap, the electron will jump into the conduction band, where it can move freely. The electron can then be moved by the electric field across the p-n junction, resulting in the flow of electrons, which will continue as long as the solar cell is illuminated (Kalogirou, 2009).

The energy contained in a photon, E_{ph}, is related to the wavelength of the light, λ, as follows.

$$E_{ph} = \frac{hC}{\lambda} \tag{2.1}$$

where h is Planck's constant and is 6.625×10^{-34} J.s, and C is the speed of light which is 3×10^{8} m/s (Kalogirou, 2009). If a PV cell is made of silicon, which has a band gap of 1.11eV, from Equation (2.1), photons with a wavelength less than 1.12μm are useful for creating electrical flow. Since the majority of solar energy that reaches the Earth is concentrated in the spectrum with wavelengths ranging from 0.2 μm to 1.2 μm, it means that a significant amount of solar radiation can be converted into electricity. Note that when a photon is absorbed by a valence electron, regardless of the intensity of the photon energy relative to the band gap energy, only one electron can be freed. This is why photovoltaic cells have a relatively low efficiency (Kalogirou, 2009).

2.3.2 Structure of PV Cells

A photovoltaic (PV) cell is mainly constructed of photovoltaic materials (p-n junctions), metal grids, an anti-reflection coating and the materials that support the cell. Metal grids in a PV cell are used to collect the electrical current. To minimize the reflection of sunlight from the surface of PV cells, an anti-reflection coating is typically applied. As a result, PV cells range in colors from black to blue.

The operating voltage of a single PV cell is about 0.5 V which is not usable for many applications. To generate a more useful voltage, PV cells are normally grouped into modules and are encapsulated with various materials to protect the cells and electrical connectors from environmental effects. PV modules can last for more than 25 years, and therefore their supporting structures must be designed with this in mind (Kalogirou, 2009).

The efficiency of PV cells is defined as the ratio of the output power of the solar cell to the incident light power. Efficiency is normally measured at a temperature of 25 ^oC and with incident light with a solar irradiation of 1000W/m^2. While the maximum theoretical efficiency for a single-junction solar cell is about 33.7% (Shockley and Queisser, 1961), this value for multi-junction types is 86.8% (Dimroth *et al.*, 2005).

2.4 Other PV System Components

In addition to solar panels, other components such as batteries, charge controllers and inverters are typically required for many PV applications. In the following an overview to each of these components is presented.

The most common battery types that are used in PV systems are deep-cycle lead-acid batteries. These can be divided into flooded or valve-regulated types. Flooded (or wet) batteries generally need more maintenance than the valve-regulated versions, but with adequate care, can last longer. Batteries in PV systems must often be able to accept repeated deep charging and discharging, and although car batteries have the same appearance, they are not designed for this type of use. Batteries can also be classified by their *nominal capacity*, which is the number of Ampere-hours (Ah) that can be maximally extracted from the battery under certain discharge conditions. Battery efficiency, state of charge and lifetime are other important properties. The ratio of the charge extracted during discharge over the amount of charge needed to restore the initial state of charge is defined as the battery efficiency. State of charge of the battery is defined as the ratio of the present capacity of the battery over the nominal capacity, and the lifetime of a battery is usually defined as the number of charge-discharge cycles that the battery can sustain before loosing 20% of its nominal capacity (Kalogirou, 2009).

Charge controllers are used in PV systems to protect the battery from over and undercharging by regulating the power from the PV modules. Some charge controllers can optimize the operating voltage of the PV module independently of the battery voltage so that the PV can operate with maximum output power (Hansen *et al.*, 2001).

PV modules produce direct current (DC), and therefore, a (DC-to-AC) inverter is required if the system has to provide alternating current. Similarly, an (AC-to-DC) inverter is needed if AC energy must be extracted from a power grid connection and stored locally. In addition to transferring energy, inverters also adjust the voltage levels, e.g., most PV modules provide DC power at 12 volts, whereas AC household power typically operates at 240 or 120 volts (Hansen *et al.*, 2001).

2.5 Applications of PV Systems

The most common applications for PV systems include remote site electrification, powering remote monitoring devices, water pumping electric motors, charging electrical vehicles, and powering wireless communication equipment. PV systems can be connected to the power grid or they can be designed to operate independently. From this viewpoint they can be categorized as *stand-alone PV systems* or *power grid connected PV systems* which we discuss in the following sections.

2.6 Stand-alone PV Systems

Stand-alone PV systems are typically used in remote areas where there is no access to the power grid, or when connecting to it would be very costly. In comparison with fuel generators which can also be used in remote areas, PV systems are often preferred since they are independent of a fuel supply, and require relatively low maintenance. In stand-alone PV systems, the produced energy can be directly consumed (e.g., in water pumping application), or it can be stored in batteries so that it can be used during times when there is insufficient solar radiation (e.g., at night). A stand-alone PV system can also be operated in conjunction with other stand-alone power sources such as fossil fuel (e.g., diesel fuel or gas) or other types of renewable energies (e.g., wind) (Sayegh *et al.*, 2008).

2.6.1 Solar Powered Wireless Networks

Solar powered wireless networks, as examples of stand-alone PV systems, are emerging in many wireless infrastructure applications. For example, the Green Wi-Fi organization has launched several projects around the world whose aim is to bring Internet access to schools in developing countries via solar powered networking (Green WiFi, 2011). In San Fransisco, California, 1,100 solar powered Wi-Fi bus stops will be unveiled by 2013, and in a similar project called Park Wi-Fi, which was run by the city of St. Louis Park in Minnesota, 400 solar powered wireless access points were installed to implement a high-speed Internet service delivered over a citywide wireless network (Pires, 2007). Google has funded a project at the University of Nigeria to implement solar powered wireless access to provide internet access to 40,000 students (UNN, 2010). In (Ab-Hamid *et al.*, 2011) a solar powered long range Wi-Fi network model is studied to connect nearby villages within a 10 km radio distance from the Internet center. This network has been implemented in a remote village in Borneo by connecting six villages to the telecenter for Internet access. In the Quail Ridge Wireless Mesh Network (QuRiNet) project, a wireless solar powered mesh network consisting of 34 wireless nodes is being developed that provides the backbone for collecting ecological and environmental data (Wu *et al.*, 2011).

2.7 Power Grid Connected PV Systems

In power grid connected PV systems, there is a fixed connection to the public electricity grid. The main components in these systems are the same as those used in stand-alone PV systems except that an inverter is always necessary for interaction between the PV system and the power grid. In these systems, energy storage units (e.g., batteries) can be used to:

(i) smooth out the fluctuations of the output power of the solar panels fed into the grid, (ii) consume the stored energy during the times that the energy demand from the power grid is high (“peak shaving”), (iii) store surplus solar energy during the times that energy pricing is low, so that it can be used when energy pricing is high. In addition, extra energy can be sold to the power grid or stored in batteries for future use (Ru *et al.*, 2012).

Grid connected PV systems can be either decentralized or centralized. Decentralized PV systems, which are the subject of Chapter 4 of this thesis, usually operate at low power consumption levels and can be installed on rooftops or integrated into building facades. On the other hand, centralized PV systems typically operate at much higher power levels (in the mega-Watt range) and are usually installed on unused land, but in some cases building installation is possible (Goetzberger and Hoffmann, 2005).

2.8 Energy Provisioning in PV Systems

Providing an uninterrupted source of power had been an objective in the design of many PV systems. The process of assigning solar panel and battery configurations so that this is accomplished is referred to as *energy resource provisioning*. Methodologies for this have been considered in the past literature and will be briefly summarized.

In (Narvarte and Lorenzo, 2000) it was shown that the accuracy of different photovoltaic sizing methods are bounded by statistical laws and in (Maghraby *et al.*, 2002) three methods for sizing stand-alone PV systems were compared. The results show that the most accurate provisioning can be obtained by simulation using historical solar insolation traces. Based on these results, most PV sizing papers adopt this approach using hourly solar insolation data. This is the approach that will be used in this thesis. In (Balouktsis *et al.*, 2006) a method of sizing stand-alone PV systems with respect to satisfying the load demand, the component cost, and the battery discharge depth was presented.

A methodology that produces the optimal design for a hybrid power system for stand-alone or grid connected applications has been proposed in (Chedid and Rahman, 1997), and in (Borowy and Salameh, 1996) a method for calculating the optimum battery and solar panel size for a stand-alone hybrid (wind/PV) system was developed. Historical data for wind speed and solar irradiance for every hour over 30 years were used to perform the provisioning. This information was used to compute the average power generated by the wind turbine and the PV module for each hour. For a given load characteristic, wind turbine, and a desired outage probability, the optimum number of batteries and PV modules was determined so that energy provisioning cost was minimized. Rather than assuming a constant system load, Reference (Saengthong and Premrudeepreechacharn, 2000) takes variable energy loading into account in the sizing of solar powered systems.

Genetic algorithms were used in (Xu *et al.*, 2005) to find the optimal resource assignment for a hybrid powered system. This reference defines a mixed multiple-criteria integer programming problem which optimizes the types and sizes of wind turbine generators, the tilt angles and sizes of PV panels, and the battery capacities. In (Lopez and Agustin, 2005) a hybrid optimization using genetic algorithms was proposed in the design a PV-diesel system. The algorithm obtains the optimal configuration of PV panels, batteries and diesel generator, and minimizes the total net cost over the useful lifetime of the system.

Energy provisioning in PV systems is the focus of this thesis. In Chapters 3 and 5 energy provisioning in stand-alone PV systems is considered, and in Chapter 4 this problem is considered in power grid connected PV systems.

2.9 Conclusions

In this chapter, background information on photovoltaic systems was reviewed. The discussion began by an introduction to the fundamentals of solar energy. This included a discussion of solar radiation, the photovoltaic effect and some basic information relating to solar cells. We then discussed applications of PV systems, including stand-alone and power grid connected types. Finally, energy provisioning in PV systems was discussed.

In the next chapter, the problem of energy provisioning in solar powered wireless mesh networks is addressed. In Chapters 4 and 5, the problem will be extended to power grid connected PV systems, and stand-alone PV systems which have deferrable energy demands.

Chapter 3

Energy Provisioning in Green Mesh Networks Using Positional Awareness

3.1 Introduction

Energy resource provisioning in solar powered wireless mesh networks is done by determining a bandwidth usage profile (BUP) for each mesh node, and then by assigning resources that are based on results obtained using historical meteorological data for the geographic location where the network is to be deployed (Farbod and Todd, 2006).

Solar resource provisioning is complicated by the problem of node positioning and orientation. In photo-voltaic systems, the orientation of the solar panels determines the amount of solar energy collected. In the northern hemisphere for example, solar panels are ideally placed in an open area, pointed directly south, and sloped to an angle slightly less than the node's geographic latitude, so that solar energy collection is maximized (Sayegh *et al.*, 2008). Unfortunately, this requirement places restrictions on the location and the flexibility with which the nodes may be placed. Since the positioning and orientation of the nodes may be constrained by other factors, panels may suffer from significant time-varying orientation and shadowing losses compared with optimally-positioned ones. These effects lead to a wide variation in the efficiency with which the nodes can collect solar energy, and conventional resource provisioning algorithms do not take this into consideration. This may lead to an unnecessarily expensive network deployment.

In this chapter we propose a position aware provisioning algorithm (PAP) which can give significant reductions in total solar resource provisioning costs. The chapter first discusses conventional resource provisioning methods and gives an illustration which shows why a conventional approach may not work very well. The PAP Algorithm is then introduced that takes known positional variations into consideration using link-weight assignments which are motivated by the relative provisioning costs associated with using the various mesh node links. A variety of results are presented and compared with previous approaches which show that significant reductions in provisioning costs can be obtained. The proposed algorithm also performs very well compared with a linear programming formulation which gives lower bounds on the node resource provisioning cost assignments (Zefreh and Todd, 2012), (Zefreh *et al.*, 2010).

The remainder of the chapter is organized as follows. In Section 3.2 a brief overview is given of related work. Following this, in Section 3.3 we discuss the issues of offline versus online network design and operation. In Section 3.4 the node energy flow model is discussed and the topic of resource provisioning is introduced. A lower bound on the minimum cost network node provisioning is then derived in Section 3.5 which is used as a comparison with the proposed position aware provisioning algorithm introduced in this chapter. Methods based on extensions to conventional stand-alone photo-voltaic node provisioning methods are then introduced in Section 3.6. This includes both the Shortest Path Resource

Provisioning (SPP) and Link Aware Resource Provisioning (LAP) algorithms. An example is then given in Section 3.7 which motivates the value of using positional awareness when doing resource provisioning. The Position Aware Provisioning (PAP) Algorithm is proposed in Section 3.8 and in Section 3.9 we present an iterative algorithm which can incorporate online energy aware routing into the provisioning process. Example performance results are then presented in Section 3.10 which show significant improvements over the conventional algorithms. In Section 3.11 we then discuss issues such as provisioning resiliency and solar insolation factor error. Finally, in Section 3.12 the chapter presents some conclusions of this work.

3.2 Related Work

Resource allocation and outage control in solar powered wireless mesh networks were first considered in (Farbod and Todd, 2006). Algorithms were proposed which can prevent node outage by introducing a bandwidth deficit when the node battery energy drops below a precomputed threshold. In this way, outage is avoided by adaptively reducing the level of service offered to the end user stations. In (Badawy *et al.*, 2010) a mechanism for mesh network resource provisioning was introduced based on energy awareness, and a genetic approach was used for battery assignment. This method is highly complex, and cannot be used for our problem since the objective in (Badawy *et al.*, 2010) was not to minimize total provisioning cost. However, in this chapter we do compare our results to a modified version of the energy aware algorithm proposed in (Badawy *et al.*, 2010) which can be used to reduce total provisioning costs.

There has also been recent work relating to the use of energy aware routing in energy sustainable networks. In (Wieselthier *et al.*, 2002), the residual energy is incorporated in the local cost metrics, and it is shown that using this technique it is possible to distribute the traffic load among nodes and increase network lifetime. In (Zeng *et al.*, 2006), routing protocols are proposed which make decisions based on packet progress, the residual battery energy level and harvested solar energy. A version of their cost function is used in Section 3.11. In (Ma and Yang, 2011), battery charging and discharging is modeled, and an online routing algorithm that considers battery discharge losses was proposed. In this algorithm, routes are frequently changed to let nodes with lower energy in their battery recover themselves and hence increase network lifetime. The problem of routing in networks with renewable energy sources is addressed in (Lin *et al.*, 2005) (Lin *et al.*, 2007) and a new link cost function is proposed that considers nodes with energy replenishment. The link cost incorporates available energy in the battery, rate of energy replenishment and required transmission power. It is shown through simulations that the performance of the algorithm is good in terms of maximum throughput. It should be noted that the energy aware routing algorithms discussed above required knowledge of the solar panel and battery sizes, and for this reason it is difficult to incorporate them directly into the energy provisioning process.

3.3 Offline Design Versus Online Operation

The design, deployment, and operation of an energy sustainable mesh network includes two separate phases. The first consists of *offline network design*, which is the topic of our work. This is followed by the *online operation* phase where the mesh nodes are activated and carry online traffic. We will briefly discuss these two phases and the restrictions that they impose on node energy provisioning procedures.

In the *offline design* phase the network topology and node energy provisioning configurations are determined. We assume that the network topology and an aggregate bandwidth usage profile (BUP) has been given. In offline design, site surveys are normally conducted to ascertain the final location and positioning of the nodes. Following this, the energy provisioning of the network is done, which consists of assigning solar panel and battery configurations to the nodes. This is the focus of this chapter.

In order to accommodate the assumed bandwidth usage profile, traffic routing must be determined so that node loading and energy provisioning can be found. It is important to note that at this time, *no reasonable algorithms have been proposed which incorporate conventional online energy-aware routing algorithms into this process.*[1] Online energy aware routing uses inputs such as residual battery levels and charge replenishment rates, which cannot be known until the nodes are provisioned.[2] This is a "chicken-and-egg" problem that makes the use of energy aware routing algorithms in energy provisioning very difficult. As an illustration of this problem, in Section 3.9 we will give an example of an iterative method using energy aware routing which leads to inconsistent performance.

Once the network has been designed and node energy provisioning has been determined, the network can be deployed and the *online operation* phase can begin. In this case, routing can include energy-awareness using inputs such as residual battery levels and charge replenishment rates, since the solar panel and battery sizes have now been determined and have been known since the network commenced online operation.

This discussion highlights the restrictions on routing which are possible during the offline design phase. In the offline case, solar panel and battery assignments must be made using path selection mechanisms that ensure satisfaction of the BUP, but cannot know the solar panel and battery size. In the online case, energy aware routing may be used.

[1] This difficulty and restriction is discussed in more detail in Section 3.9

[2] It is not possible to know the current battery level at a node unless the sizes of the battery and solar panel at the node were known when the network commenced operation. This is not known until the provisioning process is completed.

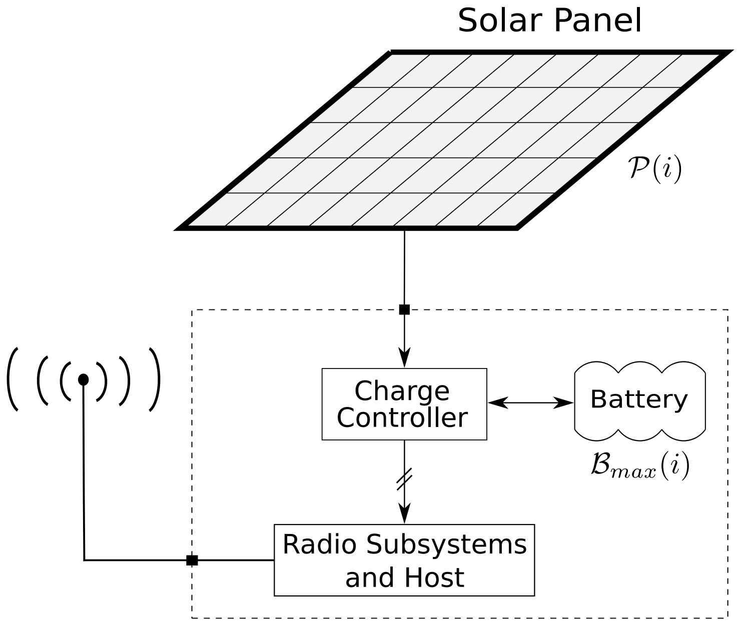


Figure 3.1: Solar Powered WLAN Mesh Node

3.4 Node Energy Flow and Solar Resource Provisioning

In this section we briefly review the energy flow model used at the solar powered mesh nodes and introduce the resource provisioning problem. An example node is shown in Figure 3.1. The solar panel and battery are connected to the node through a charge controller which provides protection from battery over and under-charging. Photo-voltaic system modelling is normally done in discrete time with intervals of duration Δ between time modelling instants. It is well accepted that sufficient accuracy is obtained using $\Delta = 1$ hour time increments (Farbod and Todd, 2006)(Maghraby *et al.*, 2002)(Khatib *et al.*, 2012)(Shrestha and Goel, 1998). In the node energy flow model we define $\mathcal{B}(i,k)$ as the residual battery energy stored at Node i at discrete time instant k, and $\mathcal{B}_{max}(i)$ is defined to be the node's battery capacity. $\mathcal{B}_{outage}(i)$ is defined to be the maximum allowed depth of discharge based on safety and battery life considerations of Node i. When $\mathcal{B}(i,k) < \mathcal{B}_{outage}(i)$, the charge controller will automatically disconnect the electrical load and the node will experience a radio outage.

We define $\mathcal{E}_{sp}(i,k)$ as the energy output of the solar panel at Node i over time increment $[(k-1)\Delta, k\Delta]$. If we assume that $\mathcal{L}(i,k)$ is the load energy demand over the same time period, then we can write that (Farbod and Todd, 2006) (Safie, 1989)

$$\mathcal{B}(i,k) = \text{min}\{\max[\mathcal{B}(i,k-1) + \mathcal{E}_{sp}(i,k) - \mathcal{L}(i,k), \mathcal{B}_{outage}(i)], \mathcal{B}_{max}(i)\}. \tag{3.1}$$

Equation 3.1 is a simple recursion that finds the battery energy at time k to be that at time $k-1$, plus the energy received from the solar panel, minus the load energy expended, over that time period. This expression uses the well accepted linear charge model but can easily be modified to account for factors such as battery charging efficiency and temperature dependence (Sayegh, 2008). It is well known for example, that battery capacity and charging efficiency vary with temperature, and this effect can be taken into consideration (Farbod and Todd, 2006)(Sayegh, 2008). For temperature dependence, a table can be used so that when Equation 3.1 is evaluated, the temperature index returns values that modify the battery capacity and charging efficiency.[1] In the case of time-varying positional solar attenuation we can, by definition, write that

$$\mathcal{E}_{sp}(i,k) \triangleq \mathcal{P}(i)\ \beta(i,k)\ \mathcal{E}^*(k) \tag{3.2}$$

where $\mathcal{P}(i)$ is the size (i.e., area) of Node i's solar panel and $\mathcal{E}^*(k)$ is the per unit area energy availability for an optimally oriented solar panel during time increment k. Sample

[1] The energy provisioning algorithms to be presented only rely on there being a causal energy recursion that permits the computation of current battery energy based on past history.

functions of historical values of $\mathcal{E}^*(k)$ are available from meteorological databases.[1] Equation 3.2 also contains the *solar insolation factor* term, $\beta(i,k)$, which gives the attenuation due to non-optimal positioning of the node, and where $0 \leq \beta(i,k) \leq 1$. Note that in practical network deployments the solar insolation factor can easily result in usable energy values that are one to two orders of magnitude less than the optimum value due to node positioning effects.[2]

In Equation 3.1 the value for $\mathcal{L}(i,k)$ is obtained based on a power consumption model for the mesh node that uses the average traffic flow rate over the associated time period to compute the expended energy (Farbod and Todd, 2006). We assume a standard exponential distance-dependent path loss propagation model to compute the transmission component of this energy consumption (Rappaport, 1996). Although this common model is used, the methodology in our work is applicable to any propagation model. This is discussed in detail in Section 3.5.

In order for outage-free operation, a solar powered node must have a sufficiently large battery and solar panel without over-provisioning the network. This is done using historical solar insolation traces for the location that the network is to be deployed. It is assumed that the network topology is given, consisting of a set of N nodes identified by an index in the set $\mathcal{N} = (1, 2, ..., N)$. The design is done based on a supplied bandwidth usage profile (BUP) and the input solar insolation trace, for a contiguous deployment time period $\mathcal{T}_{\mathcal{L}} = (k_{min}\Delta, k_{max}\Delta)$, where k runs over the set $\mathcal{K} = (k_{min}, k_{min}+1, ..., k_{max})$. The usage profile consists of a multicommodity traffic flow matrix $\mathcal{U} = [u_{sd}(k)]$, where

[1] Solar insolation data is available in the USA from the U.S. Department of Energy (U.S. Department of Energy, 2012). In Canada it can be obtained from The Meteorological Service of Canada (The Meteorological Service of Canada, 2012). These databases include samples spanning several decades of continuously collected measurements for hundreds of geographic locations.

[2] We restrict our attention to predictable insolation factors, such as those which are due to known orientation losses. For this reason, errors in the solar insolation factors are expected to be very small. However in Section 3.11.2 we also discuss how insolation factor errors may affect node provisioning.

$u_{sd}(k) \geq 0$ indicates the aggregate bandwidth requirement from Node s to Node d during time period $((k-1)\Delta, k\Delta)$.[1] In solar powered networks, it has been shown that node provisioning is highly insensitive to short-term fluctuations in node loading (Sayegh *et al.*, 2008), i.e., knowing the details of a node's 24-hour energy use does not provide much more information than knowing the same node's *average* 24-hour energy use. For this reason the temporal packet transmission activities over each Δ time period can be summarized by normalized flow rates. The solar insolation sample function consists of an input trace over the same time period whose values are given by $\mathcal{E}^*(k)$ for all $k \in \mathcal{K}$.

The energy resource provisioning problem can now be stated. Given the network topology, the BUP, and $\mathcal{E}^*(k)$ for all $k \in \mathcal{K}$, the problem is to find the solar resource provisioning assignments that minimize the total provisioning cost. This must be done subject to preventing node outage over all $k \in \mathcal{K}$. The node provisioning cost is given by the sum of the solar panel and battery cost at the node, and is defined in the next section, where a lower bound on provisioning cost is formulated.

3.5 Resource Provisioning Cost Bound

In this section, a linear program is formulated to compute lower bounds on the cost of network resource provisioning. Each solar powered Node i has an initial battery energy, $\mathcal{B}(i, k_{min}) = \mathcal{B}_{max}(i)$, where $i \in \mathcal{N}$. The traffic flow on Link (i, j) during time interval $[(k-1)\Delta, k\Delta]$ is defined to be $l_{ij}(k)$. For convenience we normalize the maximum possible flow on each link to unity, i.e., $0 \leq l_{ij}(k) \leq 1$ for all $i, j \in \mathcal{N}, k \in \mathcal{K}$. Flow continuity for

[1] A bandwidth usage profile is assumed in many other resource allocation problems and is typically very application dependent. As in other resource allocation problems, to account for BUP uncertainty, estimated BUPs can include a safety margin.

each Node i can then be written in the usual way as

$$\sum_{j\in\mathcal{N}} l_{ij}(k) + \sum_{d\in\mathcal{N}} u_{id}(k) = \sum_{h\in\mathcal{N}} l_{hi}(k) + \sum_{s\in\mathcal{N}} u_{si}(k) \qquad \forall i \in \mathcal{N} \text{ and } \forall k \in \mathcal{K} \tag{3.3}$$

where $u_{sd}(k)$ are the entries in the BUP as part of the input to the problem.

As in other mesh network studies (Sayegh *et al.*, 2008) we assume that there is a demand-based contention-free operation on the links, i.e., overhead due to contention-based media access control for example, is not present. In addition, it is assume that there is no conflict in assigning the links to each node. The total energy that Node i consumes for radio operation is

$$\mathcal{L}(i,k) = \Big(\sum_{j\in\mathcal{N}_i} Rl_{ji}(k) + \sum_{m\in\mathcal{N}_i} T_{i,m}l_{im}(k) + S\Big(1 - \sum_{h\in\mathcal{N}_i} l_{hi}(k) - \sum_{l\in\mathcal{N}_i} l_{il}(k)\Big)\Big)\Delta \qquad \forall i \in \mathcal{N}, k \in \mathcal{K} \tag{3.4}$$

In this expression, $\mathcal{N}_i$ is the set of nodes which have links to Node i, $T_{i,j}$ is the power consumption due to transmission on link (i, j), R is the power consumption for packet reception at a link, and S is the power consumption when the air interface is in a doze mode due to power saving. The first term in Equation 3.4 is the amount of energy that Node i uses for reception, and the second is that required for transmission.

The last term is the energy consumed in doze mode. Note that in this formulation we have assumed directional links, but other options are possible.

The cost of provisioning a node consists of the sum of the battery and solar panel costs.

Therefore, the total cost for Node i, $C_T(i)$, is given by

$$C_T(i) = C_B(\mathcal{B}_{max}(i)) + C_P(\mathcal{P}(i)), \tag{3.5}$$

where $C_B(b)$ and $C_P(p)$ are the battery and solar panel costs for sizes b and p, respectively. As in many other studies, we assume that these are linear functions of b and p, i.e., $C_T(i) = \gamma_B \mathcal{B}_{max}(i) + \gamma_P \mathcal{P}(i)$ (Sayegh *et al.*, 2008). The objective function can therefore be written as $\sum_{i \in \mathcal{N}} C_T(i)$ and the lower bound on the network energy provisioning cost is given by the following linear program (LP).

$$\underset{\mathcal{B}_{max}(i),\mathcal{P}(i)}{\text{minimize}} \quad \gamma_B \sum_{i\in\mathcal{N}} \mathcal{B}_{max}(i) + \gamma_P \sum_{i\in\mathcal{N}} \mathcal{P}(i) \tag{LP–PAP}$$

$$\text{subject to} \quad \mathcal{B}(i,k) \le \mathcal{B}(i,k-1) + \mathcal{E}_{sp}(i,k) - \mathcal{L}(i,k), \quad \forall i \in \mathcal{N}, k \in \mathcal{K} \setminus k_{min} \tag{3.6}$$

$$\mathcal{E}_{sp}(i,k) = \mathcal{P}(i)\, \beta(i,k)\, \mathcal{E}^*(k), \quad \forall i \in \mathcal{N}, k \in \mathcal{K} \tag{3.7}$$

$$\mathcal{B}(i,k) \le \mathcal{B}_{max}(i), \quad \forall i \in \mathcal{N}, k \in \mathcal{K} \tag{3.8}$$

$$\mathcal{B}_{outage}(i) \le \mathcal{B}(i,k), \quad \forall i \in \mathcal{N}, k \in \mathcal{K} \tag{3.9}$$

$$0 \le l_{ij}(k) \le 1, \quad \forall i,j \in \mathcal{N}, k \in \mathcal{K} \tag{3.10}$$

$$0 \le \mathcal{B}_{max}(i), \quad \forall i \in \mathcal{N} \tag{3.11}$$

$$0 \le \mathcal{P}(i), \quad \forall i \in \mathcal{N}. \tag{3.12}$$

In this formulation, $l_{ij}(k)$ was previously defined in Equation 3.3 and $\mathcal{L}(i,k)$ was given in Equation 3.4. In the interest of brevity we have not re-written these equations. LP–PAP minimizes the total provisioning cost subject to constraints 3.6 to 3.12. Constraint 3.6 states that the stored battery energy at time k must not exceed that which was available at time $k-1$, plus the energy that which was obtained from the panel, minus the energy consumption during this time interval. Constraint 3.8 indicates that the battery energy cannot exceed the battery capacity at the node and constraint 3.9 indicates that the node must maintain a minimum battery level in order to prevent node outage. Constraint 3.10 incorporates the usual link capacity constraint and constraints 3.11 and 3.12 ensure non-negative battery and panel provisionings. Note that constraint 3.6 is not active at time $k = k_{min}$, and this, in conjunction with constraint 3.8 permits the optimization to initialize a full battery.

It is important to note that although the above LP provides a lower bound on the provisioning cost, *it does not provide any solution to the provisioning problem for a realizable system.* The reason is that the LP is free to use its inputs in a non-causal fashion and may therefore route traffic using knowledge of future loading and solar insolation values. Because of this, a system provisioned using the results of the LP will generally not be energy sustainable when causal online path selection algorithms are used with the same loading and solar insolation inputs. For this reason, more sophisticated provisioning algorithms must be used which is the subject of Sections 3.6 to 3.8. Before proceeding, we first show the following result which provides an important theoretical connection between the lower bound and Equation 3.1.

Theorem 1. *A feasible minimum cost solution to the offline optimization given in LP–PAP can be obtained when the energy recursion given in Equation 3.1 is satisfied.*

Proof. The proof consists of starting with a solution obtained from LP–PAP. We then show that with feasible modifications to the $\mathcal{B}(i, k)$ variables, Equation 3.1 can be satisfied for the same optimum values of $\mathcal{B}_{max}(i)$ and $\mathcal{P}(i)$ for all i and k.

Assume that we have a solution to the optimization in LP–PAP, i.e., the values of $\mathcal{B}_{max}(i)$ and $\mathcal{P}(i)$ for all i minimize the objective function. Because of constraint 3.9, the "max" in Equation 3.1 must clearly be satisfied for all i and k. Also, because of inequalities 3.6 and 3.8, $\mathcal{B}(i,k)$ must satisfy

$$\mathcal{B}(i,k) \leq M(i,k) \triangleq \min\{\mathcal{B}_{max}(i), \mathcal{B}(i,k-1)+\mathcal{E}_{sp}(i,k)-\mathcal{L}(i,k)\} \qquad \forall i \in \mathcal{N}, k \in \mathcal{K} \backslash k_{min} \tag{3.13}$$

For a particular Node i, assume the same initial value of $\mathcal{B}(i, k_{min})$ and iterate over the values of $k = \{k_{min}+1, k_{min}+2, \ldots, k_{max}\}$ in order, checking if $\mathcal{B}(i,k) = M(i,k)$. When this is the case then Equation 3.1 is clearly satisfied in the optimum solution for these values of i and k. We proceed in this manner until we find the first value of k where $\mathcal{B}(i,k) < M(i,k)$, if one exists. Denote this value of k as k'. We now modify the value of $\mathcal{B}(i,k')$ by replacing it with $\mathcal{B}'(i,k') = M(i,k')$, i.e., energy which the LP had discarded is instead placed in the battery and therefore Equation 3.1 holds for k'. The value of $M(i,k'+1)$ is then updated to $M(i,k'+1) = \min\{\mathcal{B}_{max}(i), \mathcal{B}(i,k') + \mathcal{E}_{sp}(i,k'+1) - \mathcal{L}(i,k'+1)\}$. It can easily be seen that the value of $\mathcal{B}'(i,k')$ still satisfies all of the optimization constraints for the same objective function values since all we have done is to arrange for inequality 3.6 to be satisfied with equality. Also, since the update to $M(i,k'+1)$ is non-decreasing, the constraints for $k' < k$ are still satisfied with this new value of $\mathcal{B}(i,k')$. By continuing in this way the values of $\mathcal{B}(i,k)$ can be modified so that Equation 3.1 is satisfied with the same optimum cost values for $\mathcal{B}_{max}(i)$ and $\mathcal{P}(i)$ for all i. Repeating this process for each value of i completes the proof. The proof also holds if we fully charge the batteries at time k_{min}, i.e., $\mathcal{B}(i,k_{min}) = \mathcal{B}_{max}(i)$ for all i before beginning the iterations.[1] The above result is important in that it means that storing energy in the battery in accordance with

[1] Intuitively, this result is obtained by finding those times where LP–PAP has discarded energy when there is still room left in the battery. Our modification to $\mathcal{B}(i,k')$ instead, tops up the battery whenever possible which is what would be done in any practical system using Equation 3.1. The optimization may not always do this because it has non-causal knowledge of future solar insolation and node workload, and can discard energy that it knows it will not need.

Equation 3.1 (as would normally be the case in a practical system) does not necessarily restrict it to sub-optimal performance, i.e., the optimum bound computed in LP–PAP is also satisfied by Equation 3.1.

3.6 Conventional Resource Provisioning

In this section we discuss approaches for solar powered mesh node provisioning that are based on conventional methodologies. Two algorithms are discussed, namely, Shortest Path Provisioning (SPP) and Link Aware Provisioning (LAP). These algorithms are used later in this chapter for comparisons with the proposed Position Aware Provisioning Algorithm (PAP). Both SPP and LAP make use of the Stand-Alone Node Provisioning (SNP) Algorithm first described in (Farbod and Todd, 2006) which is an implementation of classical PV node provisioning (Maghraby *et al.*, 2002). SNP considers the stand-alone node case, where given a temporal solar insolation input trace and a temporal bandwidth usage profile (BUP), an iterative simulation using Equation 3.1 is used to find the cost-optimal battery and solar panel assignments (Sayegh, 2008).[1] The SNP Algorithm is embedded into the operation of both the SPP and LAP Algorithms as described below.

3.6.1 Shortest Path Provisioning (SPP) Algorithm

The SPP Algorithm was first discussed in (Badawy *et al.*, 2010) and is shown in Algorithm 3.1. The description in Algorithm 3.1 applies to both SPP and LAP (to be discussed

[1] Using the BUP and solar insolation inputs, SNP simulates the battery energy of the system for $k = \{k_{min}, k_{min}+1, \ldots, k_{max}\}$, in order, using Equation 3.1. It starts from an initial provisioning and when the node experiences an outage, the provisioning is incremented and the process is restarted. The optimum battery and panel combination is obtained via a search procedure combined with this iteration (Farbod and Todd, 2006).

in Section 3.6.2), the difference being how the link weights are computed in Step 4. This is shown in the algorithm by having $C_x(i,j,k)$ take on either $C_{SPP}(i,j,k)$ or $C_{LAP}(i,j,k)$ for SPP or LAP, respectively.

We are given the network topology, solar insolation data traces, and the bandwidth usage profile (BUP) for a desired design lifetime for which solar insolation traces are given. SPP uses Dijkstra's Shortest Path Algorithm to route the BUP flows using hop count as the link cost, which is similar to the routing used in Open Shortest Path First (OSPF) routing protocol. For this reason the link cost between Nodes i and j is given by $C_{SPP}(i,j,k) = 1$ if the nodes are within range, and ∞ otherwise. This initialization takes place in Steps 2 to 6 in Algorithm 3.1. Dijkstra's Algorithm is then used to route the flows for each time epoch in Step 7 which gives multicommodity flow rates for all time. Once the flows are known for a given k, the node energy consumptions are computed in Step 9. Finally, the SNP Algorithm from Reference (Farbod and Todd, 2006) is used in Step 12 to obtain solar panel and battery assignments. In a practical system, the battery and panel sizes obtained may be rounded up to the nearest commercially available size.

3.6.2 Link Aware Provisioning (LAP) Algorithm

In this section we introduce the Link Aware Provisioning Algorithm (LAP). LAP is similar to SPP except that it uses different link costs. The objective is to route traffic such that the path costs follow minimum energy consumption routes. Besides fixed overhead, energy consumption consists of that consumed for packet transmission and reception. Accordingly, the link weight, $C_{LAP}(i,j)$, for Link (i,j) is the power consumption needed for Node i to communicate over this link to Node j, i.e., that required to overcome path loss with the required SNR. As in the SPP Algorithm, Dijkstra's Algorithm is used to route the

Algorithm 3.1. Energy Provisioning Algorithm Template

```
for all $k_{min} \leq k \leq k_{max}$ do
    for all $1 \leq i \leq N$ do
        for all $1 \leq j \leq N$ do
            Initialize network link weights, $C_x(i, j, k)$.
      (e.g., $C_x(i, j, k) = C_{SPP}(i, j, k)$ for SPP, or, $C_x(i, j, k) = C_{LAP}(i, j, k)$ for LAP.)
        end for
    end for
    Use Dijkstra's Algorithm with link costs, $C_x(i, j, k)$, and the Bandwidth Usage Profile
  (BUP) to route each flow.
    for all $1 \leq i \leq N$ do
        Compute $\mathcal{L}(i, k)$ using Equation 3.4.
    end for
end for
Use SNP from Reference (Farbod and Todd, 2006) to determine $\mathcal{P}(i)$ and $\mathcal{B}_{max}(i)$ using solar
insolation sample functions.
```

BUP during each time epoch, and the provisioning is done as before.

3.7 Motivation for Position Aware Provisioning

In this section we give a simple example which illustrates that both SPP and LAP Algorithms may perform poorly where a deployment results in different node solar insolation factors, i.e., where $\beta(i, k) \neq \beta(j, k)$ for $i \neq j$. Consider the simple topology shown in Figure 3.2 where a source node S transmits packets to a destination node D. In this example the distance between source and relay nodes, and between relay and destination nodes is assumed to be equal and it is set to $d = 500$ m and power control is used to achieve a -70dBm received signal strength. There are two routes that Node S can use. The first is a direct transmission to Node D, i.e., route S-D. The second is a multi-hop route through relay node R, i.e., route S-R-D. For this simple single-flow traffic case we are interested in finding the minimum cost node provisioning for these two route selection options. For this

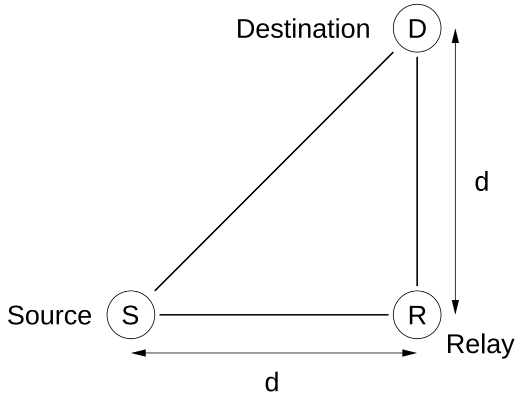


Figure 3.2: Three Node Provisioning Example. Node S can use either the direct path to Node D (i.e., path S-D) or a relay path through Node R (i.e., path S-R-D).

example we will assume that power consumption is dominated by node transmit power, although similar conclusions follow from including all power consumption components. From the assumed propagation model, the total network transmit power for route S-R-D is given by

$$P^{S-R-D} = \frac{2P_r}{G}(\frac{d}{d_0})^{\alpha}, \tag{3.14}$$

and for direct transmission the result is

$$P^{S-D} = \frac{P_r}{G}(\frac{\sqrt{2}d}{d_0})^{\alpha}, \tag{3.15}$$

where G is a constant which depends on the antenna characteristics and the average channel attenuation, d_0 is a far-field antenna reference distance, and α is the exponential path loss exponent (Rappaport, 1996). We have assumed that the nodes use transmit power control so that the receive power is the same for each link. It is easy to see that $P^{S-R-D} < P^{S-D}$ when $\alpha > 2$, i.e., multi-hop routing consumes less total energy in this case. Therefore, for this arrangement, if all nodes have the same per unit area solar energy source, transmitting over the multi-hop path (i.e., S-R-D) will lead to a lower total provisioning cost. In this case the LAP Algorithm would choose this route and will result in a lower provisioning cost than the SPP Algorithm, which would choose the S-D route. This example is an illustration of the following informal result: *When the per unit area solar insolation for the nodes on two routes is the same, then the minimum total cost provisioning will occur by using the path with the least total energy consumption.*[1] This conclusion is a result of the linearity of resource provisioning versus power consumption noted in (Sayegh *et al.*, 2008). A consequence of this result is that if power consumption is dominated by transmission and reception activity, then choosing an alternate path with the same hop length and the same solar insolation factors does not change the solar resource provisioning cost. This is true even if the average loading on the nodes along the two paths is different!

We now consider what happens if the solar insolation factors at the nodes in Figure 3.2 are different due to positional variations. The graph in Figure 3.3 shows the total provisioning cost of the network as the solar insolation factors of Nodes S and R, i.e., $\beta(S,k)$ and $\beta(R,k)$, are varied. This is done for the two routing scenarios discussed previously, i.e., S-R-D and S-D, corresponding to the SPP and LAP Algorithms defined in Section 3.6. Node D has a fixed solar insolation factor (IF) of 1.

In these results we have assumed that $\alpha = 3$, $\mathcal{B}_{outage}(i) = 0$ for all i, and that power control is used on the links so that a received signal strength of -70 dBm is obtained, as before. In this example we assume that the solar insolation factors are constant, i.e., independent of k. Note that for different values of the solar insolation factors, the nodes are re-provisioned using the SPP and LAP algorithms to create the plot shown. First consider the case where Node R is optimally positioned and Node S is not, i.e., $\beta(R,k) = 1$ for all k, at the left front of the figure. In this case the graph shows that using the S-D route is far more expensive than the S-R-D route. Over a wide range of solar insolation factor values,

[1] This is when there are equal solar insolation factors for nodes along the two routes.

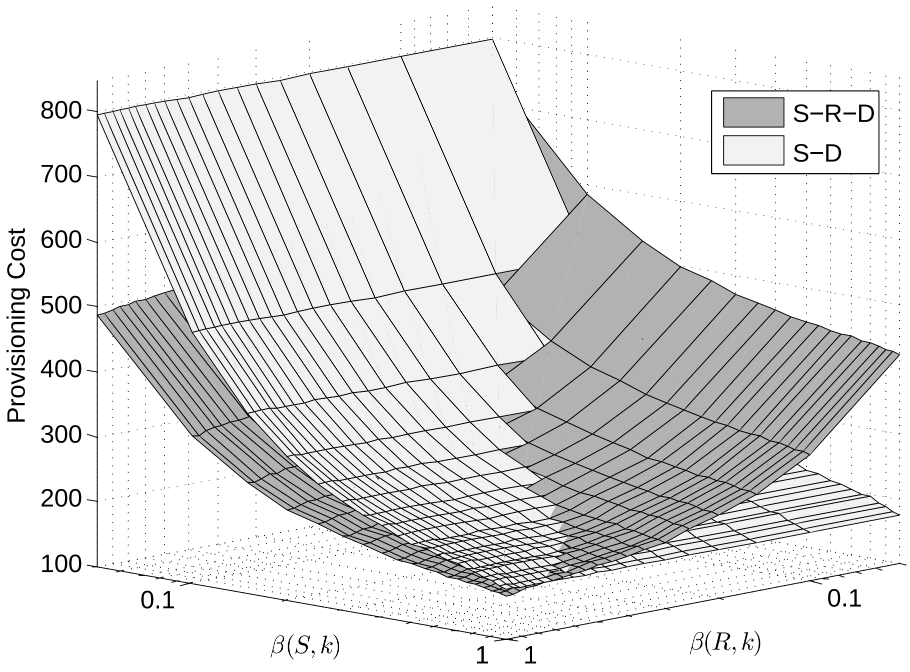


Figure 3.3: Resource provisioning cost for the network shown in Figure 3.2. The plot shows the total provisioning cost of the network as the solar insolation factors of Nodes S, i.e., $\beta(S,k)$ and R, i.e., $\beta(S,k)$, are varied. This is done for two S-to-D routing scenarios, i.e., S-R-D and S-D. In this example the insolation factors are constant, i.e., $\beta(S,k)$ and $\beta(R,k)$ are the values shown for all k. Also, $\beta(D,k) = 1$ for the entire graph.

the relative costs are more than double, which is significant. So in this scenario, the SPP Algorithm, which bases provisioning on shortest hop routes, gives a much more expensive provisioning than the LAP Algorithm which would use route S-R-D.

Now consider the case in Figure 3.3 where we vary the solar insolation factor of Node R, while $\beta(S,k) = 1$ for all k, at the right front of the plot. In this case, the opposite is true, i.e., resource provisioning based on the S-D path is significantly better than that based on using the S-R-D route. In this case the relative cost reductions are even more significant especially at low values of the insolation factor, $\beta(R,k)$. This scenario clearly shows that the most energy efficient routes do not translate into the most cost effective solar provisioning. Therefore the least cost provisioning in this case is attained by the SPP Algorithm, i.e., opposite to the situation we had previously. The intersection of the two plotted surfaces in Figure 3.3 shows the breakpoints between the provisioning costs using the SPP and LAP algorithms.

This example illustrates the fact that *in order to obtain the minimum cost resource provisioning, the algorithm cannot simply use the SPP or LAP approaches. Instead, the algorithm must be aware of the solar insolation factors, i.e., the $\beta(i,k)$'s and the relative costs on the two surfaces shown in Figure 3.3, that will result from node positioning.* This conclusion motivates the Position Aware Provisioning Algorithm (PAP) introduced in the next section. In the remainder of this chapter we assume that the solar insolation factors are due to node placement and orientation which are known when the network is planned during its offline site survey.

3.8 Position Aware Provisioning (PAP)

In this section a position aware provisioning algorithm is proposed that uses the motivation introduced in the previous section. In the proposed algorithm a position aware link cost is assigned to each link. The idea is to define a link cost that is proportional to the provisioning cost that the link would induce in a provisioned node. This is motivated and described in detail as follows.

The provisioning cost for a node consists of solar panel and battery costs. Therefore, we can say that the link cost between Nodes i and j at time k, $C_l(i,j,k)$, *contributes* to the provisioning cost as follows.

$$C_l(i,j,k) = C_b(i,j,k) + C_p(i,j,k) \tag{3.16}$$

where $C_b(i,j,k)$ and $C_p(i,j,k)$ are the contributions to battery and panel costs during this time. Based on commercial data sheets for solar panels, there is a linear relationship between panel size and its cost, i.e., $C_p = \gamma_P \mathcal{P}$, where $\mathcal{P}$ is the solar panel size (i.e., area) (Sayegh *et al.*, 2008).[1] Similarly, the cost of a lead acid battery is about $C_b = \gamma_B \mathcal{B}_{max}$ (Sayegh *et al.*, 2008). Therefore, the variation of link cost with respect to consumption power, P_c, can be expressed as

$$\frac{dC_l}{dP_c} = \frac{\partial C_l}{\partial C_b}.\frac{dC_b}{dP_c}\bigg|_{C_p} + \frac{\partial C_l}{\partial C_p}.\frac{dC_p}{dP_c}\bigg|_{C_b} = \frac{dC_b}{dP_c}\bigg|_{C_p} + \frac{dC_p}{dP_c}\bigg|_{C_b}, \tag{3.17}$$

$$\approx \gamma_B \frac{d\mathcal{B}_{max}}{dP_c}\bigg|_{C_p} + \gamma_P \frac{dP}{dP_c}\bigg|_{C_b}. \tag{3.18}$$

This follows from Equation 3.16 and the above discussion, i.e., since the battery and solar panel have linear power relationship, we have that $\frac{dC_l}{dP_c} \approx \gamma_B C_1 + \gamma_P C_2$. But we know that while the solar panel size is inversely proportional to solar insolation factor, the battery size is fairly insensitive. So, C_1 can be viewed as a constant and C_2 can be written as $C_2 \approx C_2'/\beta$, and therefore $\frac{dC_l}{d\mathcal{P}_c} \approx \gamma_B C_1 + \gamma_P C_2'/\beta$, and from (Sayegh *et al.*, 2008), $\gamma_B = 3.4$ and $\gamma_P = 6.7$. This gives $\frac{dC_l}{dP_c} \approx 3.4C_1 + 6.7C_2'/\beta$ and for the solar insolation data available from the geographic region considered, $C_1 = 90$ and $C_2' = 10$ give us good cost

[1] To simplify the expressions, we have removed the variable dependencies on i, j and k.

approximations. As a result,

$$\Delta C_l(i,j,k) \cong (\gamma_B C_1 + \frac{\gamma_P C_2'}{\beta(i,k)})\Delta P_c(i,j,k). \tag{3.19}$$

When $P_c(i,j,k) = 0$ then $C_l(i,j,k) = 0$ and thus an estimation of the link cost function for Link (i,j), $C_l(i,j,k)$, is given by

$$C_l(i,j,k) \cong (\gamma_B C_1 + \frac{\gamma_P C_2'}{\beta(i,k)})P_c(i,j,k). \tag{3.20}$$

This good approximation for single node provisioning cost has been verified over a wide range of parameter settings by using the SNP Algorithm (Farbod and Todd, 2006) to compute actual provisioning costs and then by comparing them to the above approximation. This link cost assignment is used in the PAP Algorithm proposed in the next section.

3.8.1 Position Aware Provisioning (PAP) Algorithm

The position aware provisioning algorithm uses the same template as that shown in Algorithm 3.1, except for the link weight computations discussed in Section 3.8. The link weights are now a function of time due to their dependence on the node solar insolation factors. In this case the link costs computed at Step 4 in Algorithm 3.1 are given by $C_x(i,j,k) = C_{PAP}(i,j,k) \triangleq (\gamma_B C_1 + \frac{\gamma_P C_2'}{\beta(i,k)})P_c(i,j,k)$ using Equation 3.20. The rest of the algorithm, which uses the solar insolation inputs and $\mathcal{L}(i,k)$ is the same as that discussed previously. As before, once the network flows are known the SNP Algorithm from Reference (Farbod and Todd, 2006) is used in Step 12 to obtain the solar panel and battery assignments.

Algorithm 3.2. Iterative Energy Aware Provisioning (IEAP) Algorithm

```
1: Solve the linear program bound in LP–PAP.
2: Initialize the network configuration by setting the node energy provisionings to the solution
   obtained in Step 1.
3: for all $k \in \{k_{min}, k_{min}+1, \ldots, k_{max}\}$ do
4:     Simulate the system at time $k$ for all $i \in \mathcal{N}$ with Equations 3.1, 3.2, 3.3 and 3.4 using online
   energy aware routing.
5:     if $\mathcal{B}(i,k) < \mathcal{B}_{outage}(i)$ for any value of $i$, i.e., outage occurs at a Node $i$ then
6:         Halt the simulation.
7:         Increase Node $i$ energy provisioning, i.e., $\mathcal{B}_{max}(i) \leftarrow \mathcal{B}_{max}(i) + \delta_B(i)$ and $\mathcal{P}(i) \leftarrow$
   $\mathcal{P}(i) + \delta_P(i)$.
8:         Restart the simulation at Step 1 using the updated network energy provisioning values.
9:     end if
10: end for
11: The final solar panel and battery values are the energy provisioning algorithm output.
```

3.9 Energy Provisioning Using Online Energy Aware Routing

In Section 3.3 we briefly discussed the problem of incorporating conventional energy aware routing algorithms into the offline provisioning process. We are only aware of one method for accomplishing this based on an iterative design procedure discussed in this section, referred to as Iterative Energy Aware Provisioning (IEAP). IEAP however, while incorporating energy aware routing, does not perform very well in general situations and is included to help illustrate this problem.

The algorithm simulates the operation of the system using the provided inputs and halts when node outage occurs. The energy provisionings for the node in question are then increased and the process is repeated until feasible provisionings are obtained. A detailed description of the algorithm is shown in Algorithm 3.2 and is described as follows.

The linear program from LP–PAP is first solved which gives a set of node provisionings which lower bound the provisioning cost. The solar panel and battery configurations are

Parameter	Value
Receive Signal Strength	-70 dBm
α	3
$\mathcal{B}_{outage}(i)\ \forall i$	0
$\mathcal{B}(i, k_{min})\ \forall i$	$\mathcal{B}_{max}(i)$
S	1 mW
R	500 mW
γ_B	3.4
γ_P	6.7
Location	Toronto, Canada (6 weeks, 01/01/80)

Table 3.1: Default Parameters for the Examples

then initialized using this solution. This is shown in Steps 1 and 2 of Algorithm 3.2. The network is then simulated in time using Equations 3.1, 3.2, 3.3 and 3.4. A conventional online energy aware routing algorithm can be used at this step. This is shown in Steps 1 and 4. During this process, if a node experiences outage, the simulation halts and the energy provisioning of the node is increased (Steps 5, 6 and 7). This is done by adding solar panel and battery increments defined by $\delta_P(i) \triangleq \delta P(i)$ and $\delta_B(i) \triangleq \delta \mathcal{B}_{max}(i)$, respectively, and where δ is a fixed step size factor. The simulation step is then restarted from $k = k_{min}$ and the process is repeated (Step 8). Once the simulation completes without any node outage the final panel and battery values are the algorithm output (Step 11). In Section 3.10 we present some results for the IEAP Algorithm using the energy aware routing algorithm from (Zeng *et al.*, 2006) which is known to perform well in renewable energy situations.

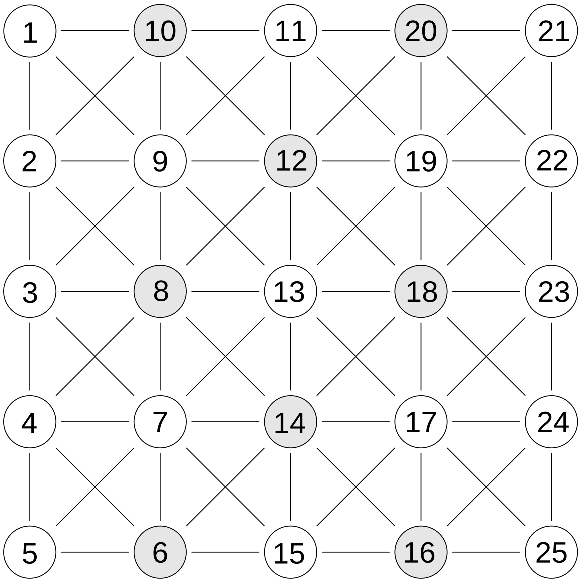


Figure 3.4: 25 Node Mesh Example.

3.10 Energy Provisioning Results

A large variety of experiments have shown that the PAP Algorithm can perform significantly better than that of the SPP and LAP Algorithms. In this section we present some examples of these results that illustrate various tradeoffs and conclusions. However, it should be noted that the relative cost performance comparison of the three algorithms is highly situation dependent. In these results we use the same parameter values as those used in the examples given in Section 3.7, e.g., we assume distance dependent exponential path loss using the default parameters given in Table 3.1.

We first consider a mesh network consisting of 5×5 nodes as shown in Figure 3.4 where the distance between horizontally and vertically adjacent nodes is 500m. The links in the figure show the nodes which are within communication range. In the initial set of experiments we assume that network carries transit flows where the BUP consists of the following large source-destination bandwidth flows: $(1, 25)$, $(2, 24)$, $(4, 22)$, $(5, 21)$, $(6, 23)$

Node ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
PAP	.5	.2	0	.2	.5	.2	1.1	0	1.1	.2	.6	0	1.1	0	.5	0	1.1	0	1.1	0
LAP	.2	.5	0	.7	.2	1.2	.5	.5	.3	.9	.3	.5	.5	.5	.3	.3	0	.3	.5	.3
SPP	.5	.2	0	.5	.5	.7	.6	0	1.1	.5	0	.6	1.1	.3	0	0	1.1	.3	.5	0

Table 3.2: Average Node Loading Example for SPP, LAP and PAP Algorithms.

and $(10, 23)$. For convenience, in our results the flow rates are expressed as a fraction of the link capacity and all nodes are assumed to be solar powered. Several simulations for different conditions are done, and for each case, the provisioning costs of the SPP, LAP and PAP Algorithms are compared.

In the first experiment, we assume that the solar insolation factors for Nodes 6, 8, 10, 12, 14, 16, 18 and 20, are smaller by 10% than the rest of the nodes. We refer to these as *shadowed nodes* and are shown as shaded in Figure 3.4. The traffic flows of all sources are set to be equal. For the solar insolation inputs, a profile was used consisting of 30 years of solar insolation data for Toronto, Canada, using 6 week duration sample functions starting at the beginning of January 1980.[1]

To illustrate the path selection effects, the average load on the first twenty nodes when the links are heavily utilized is shown in Table 3.2. SPP uses the shortest paths regardless of the solar energy capability of the nodes. For this reason it has included most of the shadowed nodes (i.e., Nodes 6, 10, 12, 14 and 18) when other options are available. On the other hand, the LAP Algorithm uses minimum energy routes which may also involve

[1] In temperate regions, provisioning is often done using worse-case monthly solar insolation inputs. This is because node provisionings that are outage-free spanning worse-case winter months result in year-round outage-free operation.

shadowed nodes. Conversely, the PAP Algorithm includes the solar factors in its path selection thus avoiding shadowed nodes as much as possible. This is clearly apparent in Table 3.2. While shadowed nodes are heavily selected by the LAP and SPP Algorithms, the PAP load on these nodes is much lighter. We can see that there is no load on shadowed Nodes 8, 12, 14, 16, 18, 20, and Nodes 6 and 10 are lightly loaded by the PAP Algorithm.

Note that even if PAP has to incorporate a shadowed node in its chosen route, the route is such that shadowed nodes need to transmit over shorter hops to consume less power. For this reason, in the PAP Algorithm, nodes that are in the neighbourhood of a shadowed node have to relay traffic for their neighbours. For example, since Nodes 17 and 19 are in the neighbourhood of shadowed Node 18, PAP avoids the shadowed node and diverts traffic to its neighbours, i.e., Nodes 17 and 19, as would be expected. As another example, consider the flow from Node 10 to Node 23. When the flow rate is high, the SPP Algorithm chooses the shortest path 10-12-18-23 which includes three shadowed nodes, 10, 12 and 18. Among these, Nodes 10 and 12 transmit over diagonal hops which consume more energy than when horizontal or vertical links are used. The path selected by LAP is 10-9-8-13-18-23, which is clearly the minimum energy path. However, from a provisioning cost viewpoint, this route contains shadowed Nodes 8 and 18, which leads to higher provisioning costs. On the other hand, PAP uses path 10-11-19-23 which has the same hop count as the SPP route but avoids shadowed Nodes 12 and 18, and avoids hops 10-12 and 12-18. This example shows that the PAP Algorithm is making path selections which should lead to lower provisioning costs.

The network provisioning costs of SPP, LAP, IEAP and PAP along with the cost lower bound for the example given are shown in Figure 3.5. In this figure we have plotted total provisioning cost versus normalized traffic flow rates, which were obtained by scaling up

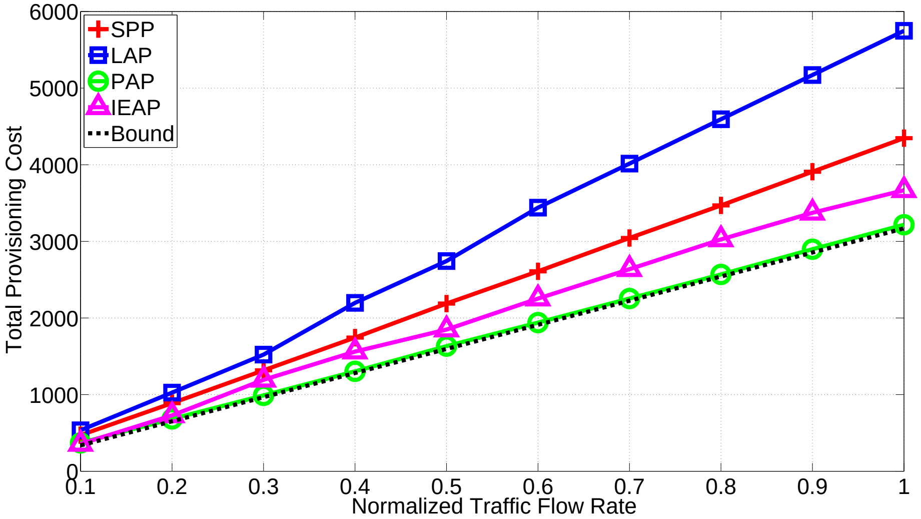


Figure 3.5: Network Provisioning Costs for SPP, LAP, IEAP, and PAP Algorithms for the 25 Node Mesh Network. The provisioning cost lower bound is also included.

the BUP. As can be seen in this figure, the network provisioning cost when a high flow rate and the PAP algorithm is used is about 26% less than when SPP is used, and we have about 44% improvement when we compare it with LAP. The provisioning cost bound which is derived in Section 3.5 is tracked very closely by the PAP Algorithm, which indicates that there is very little room for improvement using more sophisticated methods. We have found that this is always the case for the regular mesh network examples that we have considered. This is caused by the fact that the path selection options at each node are relatively limited and the PAP Algorithm does a very good job of choosing minimum cost paths. In this particular example it can be seen that the SPP Algorithm significantly out-performs LAP, but as will be seen, the extent to which this happens is very situation dependent. In Figure 3.5 we have also shown results for the IEAP algorithm using the battery and solar charging aware routing described in (Zeng *et al.*, 2006). In this case the algorithm performs well compared with SPP and LAP but is unable to find provisionings that are as good as PAP. However, in this particular example the results obtained using IEAP are reasonable.

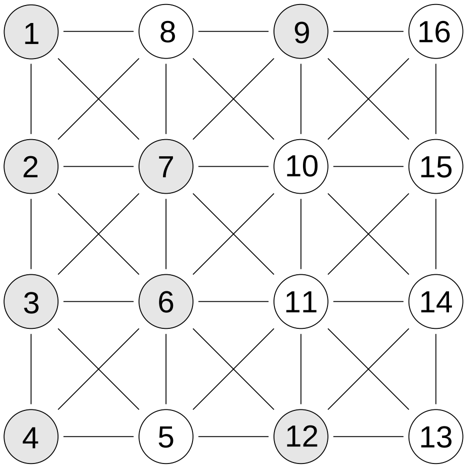


Figure 3.6: 16 Node Mesh Example

We also include results which compare the algorithms for a 16 node mesh network shown in Figure 3.6 using the same assumptions as before. The nodes which are labeled as 1, 2, 3, 4, 6, 7, 9 and 12 are assumed to be positioned such that their solar insolation factors are 10% of the other nodes. The distance between nodes and path loss exponent are the same as in the previous case. In this example there are 4 heavy source-destination flows with sources 1, 4, 6 and 7 and corresponding destinations 13, 16, 13 and 16, respectively.

In Figure 3.7 the network provisioning cost for each of the algorithms and the cost provisioning bound are shown. It can be seen that at low flow rates LAP and SPP have roughly the same cost until 0.5, and after that the performance of LAP is better than SPP. On the other hand, the provisioning cost for the PAP Algorithm is much lower across the entire range. For example, at the 50% point, the provisioning cost for PAP is about 50% lower than the other two. This ratio is about the same for higher flow rates. In the limit at the right hand side of the graph, PAP is better than SPP for 49.6%, and LAP provides a network which its cost is 143% of the cost of the same network provisioned using PAP.

The breakpoints shown in Figure 3.7 are due to flow bottlenecks. For example, when

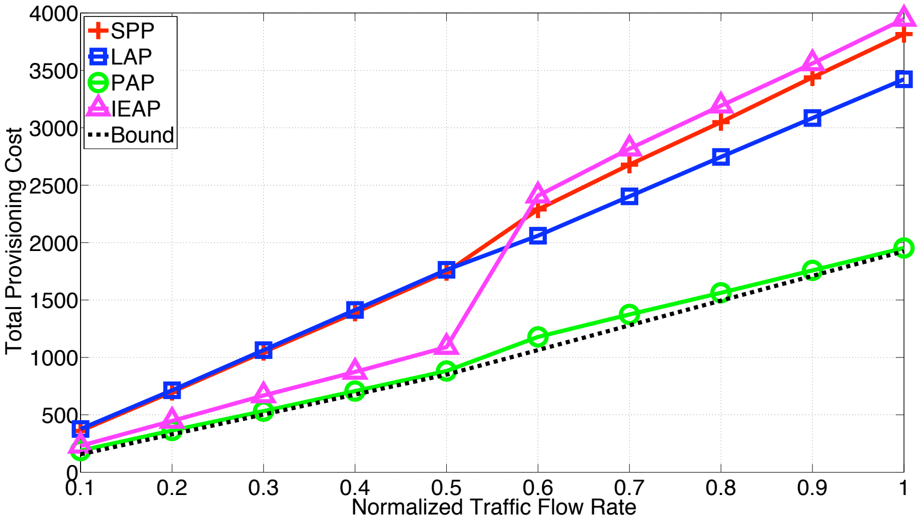


Figure 3.7: Network Provisioning Costs for SPP, LAP, IEAP, and PAP Algorithms for the 16 Node Mesh Network. The provisioning cost lower bound is also included.

the flow rates are 0.5, the selected path by PAP algorithm for Flows 1-13 and 4-16 are 1-8-10-11-14-13 and 4-5-11-10-15-16, respectively. These flows have Link 10-11 in common. When the flow rate is increased above 0.5, this forces the algorithms to use additional routes and therefore the new paths are 1-8-10-11-14-13 and 4-5-11-15-16. The new route for flow 4-16 uses Link 11-15 instead of 11-10-15 which increases the total provisioning cost, and consequently there is a jump in the provisioning cost curve. This is also true for SPP and IEAP. However, the reason that there is no jump for LAP is that in this example, the total cost for the alternative routes that LAP uses have the same cost as before.

In Figure 3.7 we have also included results for IEAP using the routing as in Figure 3.5. In this example the algorithm performs well compared with SPP and LAP at low values of flow rate, but is *significantly worse* when the flow rates are higher, i.e., 0.6 and greater. This type of inconsistent result is typical of those which were obtained with IEAP. The behaviour can be traced to the incorporation of energy awareness in the offline routing algorithm, i.e., when a node outage is found, energy provisioning at the node is increased.

As this repeatedly happens, this may eventually lead to radically different routing decisions due to the energy state awareness of the routing algorithm. This results in new routes that do not require the increased provisionings that were set for the nodes on the original routes. As a result of this, the provisioning costs become unnecessarily high, and in other results that we ran it performed worse than SPP and LAP. This highlights the problem of incorporating energy aware routing algorithms into the offline provisioning process that make decisions based on the instantaneous energy stored at the nodes. In view of these results, we will not include curves for IEAP in our remaining results but will instead quote some of the costs obtained.

It can be seen from Figure 3.7 that the PAP Algorithm again performs very well compared with the provisioning cost bound derived in Section 3.5. As before, this shows that in this example, PAP is generating resource provisioning which are very close to being optimum. This is typical of what we have found in other mesh examples that we have tested.

3.10.1 Random Hybrid Network Examples

We have also considered the network provisioning cost examples for hybrid networks, i.e., those consisting of combinations of non-solar and solar powered mesh nodes. An example is given which uses the same topologies and parameters as in the 25 node mesh as before. In this example, 8 nodes are randomly selected as shadowed nodes with reduced solar insolation factors by the same as that in first experiment. From the remaining nodes, 4 are randomly chosen as continuously powered nodes (CPNs). Network provisioning is done for 100 different random selections of CPNs and shadowed nodes, and the average provisioning costs for the SPP, LAP and PAP algorithms are shown in Figure 3.8. Note

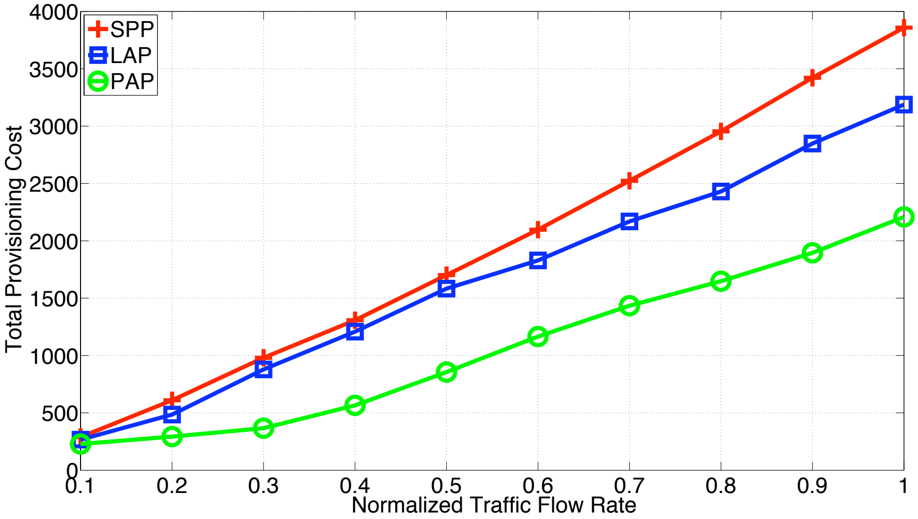


Figure 3.8: Average Network Provisioning Cost for the Hybrid Network Example

that since we are generating random network configurations we have not included cost provisioning bounds in these results.

The LAP and PAP algorithms are clearly better able to take advantage of the continuously powered nodes, and it can be seen that they give less provisioning costs than SPP. Furthermore, the PAP algorithm takes the insolation factor of solar nodes into account and, consequently, its resultant provisioning cost is less than LAP. For example, it can be seen that when the flows are heavy, the provisioning costs for the PAP are lower by about 23% when we compare it with LAP, and 37% less than SPP. These values, when the flow is about half that value, are 49% and 52%, respectively. In this example, the results for IEAP were much worse and gave provisioning costs which were roughly twice that of SPP across the traffic flow rate range.

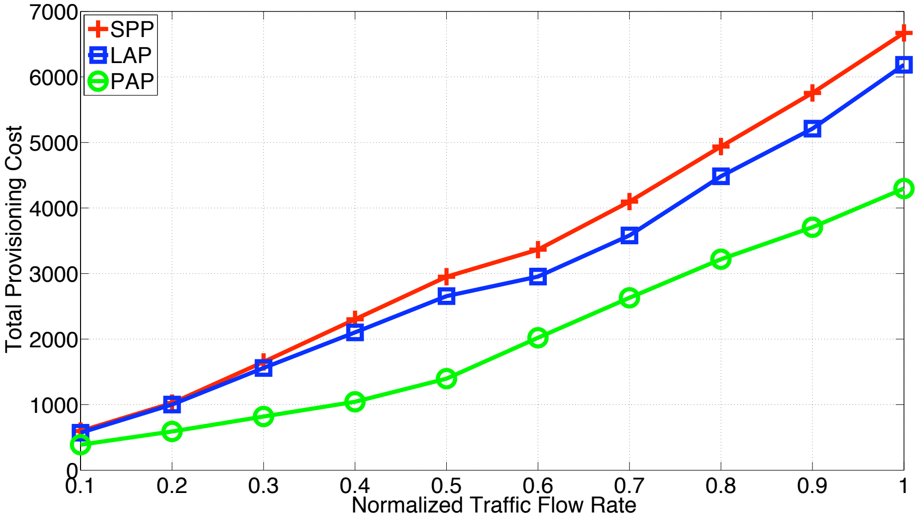


Figure 3.9: Average Network Provisioning Cost for Random Solar Insolation Factor Assignments

3.10.2 Random Insolation Factor Example

We have also considered cases that include random selections of solar insolation factors. In this example, we use the 25 node mesh in Figure 3.4. The same flow sources and destinations are assumed, but in this case a random solar insolation factor in the interval $(0, 1]$ is assigned to each node. The algorithms are run for 100 different selections using the same parameters as before. The average of the results obtained are shown in Figure 3.9. Again, we see that the performance of the PAP Algorithm is superior than LAP and SPP. As in the previous example, the results for IEAP roughly twice that of SPP across the traffic flow rate range.

3.10.3 Random Topology Example

In this experiment, the performance of PAP in comparison with SPP and LAP is evaluated for random topologies. To do this, a square area with side length of 2000m is considered

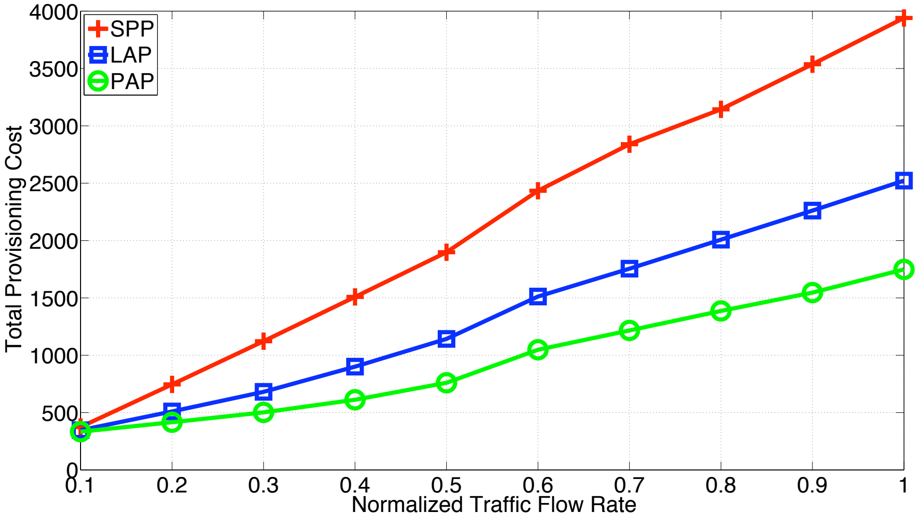


Figure 3.10: Average Network Provisioning Costs for 100 Random Topologies

and 25 nodes are randomly distributed. The source and destination nodes are as the mesh network in Figure 3.4, i.e., the destinations for source nodes 1, 2, 4, 5, 6 and 10 are 25, 24, 22, 21, 23 and 23, respectively. All other parameters are as before. Again, Nodes 6, 8, 10, 12, 14, 16, 18 and 20 are shadowed nodes and their solar insolation factor is 10% of the others. The minimum distance from each node to all other nodes is computed and the transmission range is set to the maximum of all minimum distances. Using this transmission range, if the network is connected, then the provisioning is done, otherwise a new random topology is generated. This is done for 100 topology selections, and for each, the SPP, LAP and PAP algorithms are applied.

The average provisioning costs versus different flow rates are shown in Figure 3.10. The figure indicates that the proposed position aware provisioning algorithm performance is superior to SPP and LAP in terms of network provisioning cost. It can be seen that in general, LAP, which chooses routes with minimum energy is less expensive than SPP.

We have also considered examples which include a “sink node” where mesh nodes relay all traffic to and from the sink. This type of scenario may be common when there is a single point of presence through which the mesh nodes communicate. One example consisted of a single sink node at the center of a 19 hexagon arrangement with shared edges, and 54 solar powered mesh nodes were placed at the hexagon vertices. The distance between adjacent nodes in the same row or column is given by 300m, the maximum transmission range is 600m, and 28 of the nodes were taken to be shadowed. A variety of different loading conditions were considered and the results that were obtained in this case showed very similar relative algorithm comparisons as that obtained before, i.e., the provisioning costs for the PAP Algorithm were significantly less than the other two. In these experiments the differences between SPP and LAP are quite large. SPP makes too much use of shadowed nodes and LAP also uses shadowed nodes but those that are shadowed tend to transmit over shorter hops. This tends to result in lower provisioning costs compared to SPP. Since the main conclusions are similar as that before, we have chosen not to include the graphs.

3.11 Discussion

3.11.1 Provisioning Resiliency

The results presented in Section 3.10 show that significant cost advantages are possible when positional awareness is taken into consideration. However, it is possible that a network provisioned for lower cost may be less able to handle traffic flow which is different from that for which the network was provisioned. This is an obvious trade-off which was considered in Reference (Badawy *et al.*, 2010) and is expected to be true when a reduced cost network is operated online compared with a more expensive deployment. In this section we show an example that this is not always the case when traffic loading is uniformly

Overload (%)	5	10	15	20	25	30	35	40	45	50
SPP	271	260	245	236	125	120	117	113	104	101
PAP	120	115	108	104	100	96	93	89	87	84
LAP	97	92	88	84	81	78	75	72	70	67

Table 3.3: Algorithm Provisioning Robustness Example: Network Lifetime vs. Traffic Overload

scaled.

In this example, a 5 × 5 mesh network is considered with the same sources and destinations as in the first set of results in Section 3.10. In this experiment, nodes with odd labels have a fixed solar insulation factor equal to 0.3 and the value for even nodes is set to be 0.9. A traffic flow rate equal to 0.6 is set for all sources and destinations, and the network is first provisioned using PAP, LAP and SPP provisioning algorithms.

For each network provisioning we then used an online battery aware algorithm which routes traffic in each time interval based on Dijkstra's Algorithm using the link weights given in (Zeng *et al.*, 2006). This algorithm was chosen because it incorporates both the node energy state and the solar insolation renewal rate. In (Zeng *et al.*, 2006) this link cost metric was found to produce good results for networks operating with renewable energy sources. The traffic load was then scaled to values up to 50% larger than that given in the bandwidth usage profile. This results in network outage and the network lifetime is measured for the SPP, LAP and PAP provisioned networks. The results are shown in Table 3.3 which gives the network lifetime for the three provisioning algorithms versus the percentage overload. It is clear that the network provisioned using the SPP algorithm is superior to the other two. For example, at a 10% overload the PAP network lifetime is only about 44% of that for SPP. At the same overload level, the LAP designed network is at about 35%.

These values change to roughly 78$\%$ and 64$\%$ when the overload is increased to 40$\%$.

These results confirm our intuition that in some cases a more precisely provisioned network may be less able to adapt to unforeseen bandwidth scenarios. This should be taken into account when defining the bandwidth usage profile that is used in the design process. If network usage parameters can be predicted well in advance, then significant cost savings can be obtained by position-aware provisioning combined with small bandwidth margins. If bandwidth flows differ significantly from that used in the provisioning, then in practical deployments this may lead to higher bandwidth deficits when an outage control algorithm is used. This may be perfectly acceptable in many applications.

3.11.2 Solar Insolation Factor Error

The use of positional awareness in provisioning depends on site survey results that include the solar insolation factors. The results that we have shown so far indicate that there can be major cost advantages when this information is taken into account. However, in practice there may be errors which occur when collecting the solar insolation factor data. These errors will clearly lead to provisioning differences compared to that which would take place in the error free case. In this section we present some results which show that when solar insolation factor errors are introduced, this may lead to network outages when the network is operated online. This possibility should be taken into account during the design process.

We have done a variety of experiments which illustrate the effects of solar insolation factor error, and a short representative example is given. The mesh network used in Section 3.11.1 was considered when there are errors in the estimated solar insolation factors, i.e., $\beta(i,k)$'s. Given the three network provisioning algorithms, we routed traffic using the corresponding offline algorithms. The case was considered where the estimated insolation factors (IFs) include a random Gaussian error component, with mean $\mu_{i,k} \triangleq \beta(i,k)$ and standard deviation, σ, i.e., the mean value of insolation factor is the actual solar insolation factor.

The network is operated using Dijkstra's Algorithm with the link costs defined in the PAP, LAP and SPP Algorithms using the same inputs as in Section 3.11.1. The network lifetime vs the average Relative Standard Deviation (RSD) is considered, where RSD is defined as the normalized standard deviation, i.e., $\sigma/\mu_{i,k}$. The results are shown in Table 3.4. As can be seen, the network lifetime decreases with the variation in IF error, as one would expect, although the table is showing a large range of RSD. In the case we have shown here, the network provisioned using the PAP algorithm is the most resilient to IF error, followed by the LAP and SPP algorithms, respectively. However, this comparison cannot be generalized in that the results are highly situation dependent. However, this and other examples not shown clearly indicate the need for accurate IF estimation when truly minimum cost provisioning is required.

The results of this section are a confirmation of the tradeoffs associated with minimum cost provisioning. In some cases a more precisely provisioned network, or a network whose BUP or IFs cannot be reasonably estimated, may not perform as expected or may be less able to adapt to unforeseen bandwidth scenarios. This should be kept in mind when defining the BUP that is used in the design process. If network usage can be predicted well in advance, then significant cost savings can be obtained, but if bandwidth flows differ significantly from that used in the provisioning, then in practical deployments this may lead to higher bandwidth deficits when an outage control algorithm is used.

RSD Error	5	10	15	20	25	30	35	40	45	50
PAP	904	610	417	353	321	293	268	249	232	219
LAP	592	430	325	294	270	248	225	208	193	181
SPP	377	332	297	276	257	236	215	199	185	174

Table 3.4: Algorithm Provisioning Insolation Factor Error Example: Network Lifetime vs. Relative Standard Deviation Error.

3.12 Conclusions

In this chapter a position aware provisioning methodology for solar powered wireless mesh networks was proposed. Unlike previous methods for assigning solar panel and battery configurations, the new methodology takes into account the positional variations in the per unit area ability for a deployed node to harvest solar power. It was shown that when this is taken into account, reductions in the provisioning costs are possible. The problem was first formulated as a linear programming optimization which gives a lower bound on the total network provisioning cost. A provisioning methodology (PAP) that takes position aware routing into consideration was then proposed. The performance of the proposed algorithm was compared with conventional provisioning methods, i.e. Shortest Path Provisioning (SPP) and a proposed Link Aware Provisioning (LAP) algorithm. Results for a 5×5 mesh network show that the total provisioning cost savings when PAP is used can be as high as 44% than in the case where LAP is used, and is 26% when compared with SPP. The reductions in the provisioning cost for the 4×4 mesh network are about 50%. Some other sample scenarios, e.g. hybrid networks, random networks, network with random insolation factors and networks with a centralized receiver, show that the proposed algorithm can significantly decrease the total network provisioning cost. Results were also presented which illustrate provisioning resiliency and the effects of solar insolation factor error. As one would expect, as the cost of provisioning is reduced, it may make the network less able to accommodate unforeseen loading conditions. This effect should be taken into account when the network is designed, otherwise it may be forced to reduce its offered performance to prevent outage.

Chapter 4

Energy Provisioning and Operating Costs in Hybrid Solar Powered Infrastructure

4.1 Introduction

The cost of solar components will eventually decrease to the point where they may be commonly used as an add-on for grid powered communications infrastructure. In a hybrid powered node of this type, the issue of node energy provisioning is a key consideration. In this chapter we consider a methodology that can be used to assess the costs of installing and operating a hybrid powered node solar add-on. The offline problem is formulated as a linear program (LP) which provides lower bounds on the operating expenditure (OPEX) and capital expenditure (CAPEX) costs. A variety of configurations are considered including conventional solar add-ons which give, solar panel/battery/grid, battery/grid, and panel/grid systems. These designs are compared on the basis of total CAPEX and OPEX costs. Three energy scheduling algorithms are introduced which operate on the online system and reduce the costs associated with power grid purchases. They are referred to as, Grid Purchase Last (GPL), Solar Load Optimization (SLO), and Solar Load Simulation (SLS). A variety of results are presented that show the conditions under which a solar powered add-on can reduce the total cost. These results also show that the SLO and SLS algorithms give performance that is close to the lower bounds in many situations (Zefreh *et al.*, 2013).

The rest of the chapter is organized as follows. In Section 4.2 we briefly review previous work that is related to our work. Then, in Section 4.3 the problem is introduced in detail including a discussion of the OPEX and CAPEX costs under consideration. In Section 4.4 we then carefully introduce the offline and online design phases associated with our problem. This is followed, in Section 4.5, by an introduction to the node energy flow model used throughout this chapter. In Section 4.6, lower bounds are formulated for the cost of solar configuration add-ons using linear programming (LP) formulations. Online algorithms are then introduced in Section 4.7, and simulation results are presented in Section 4.8. Finally, in Section 4.9, the conclusions are given.

4.2 Related Work

To move towards green wireless infrastructure, many researchers and manufacturers are now focusing on improved energy efficiency. The intent is that better designs will reduce both capital expenditures (CAPEX), and ongoing operational expenditure (OPEX) costs. For example, the Energy Aware Radio and Network Technologies (EARTH) project is a European research effort started in 2010 with partners from 15 countries. Their goal is to achieve a reduction in the energy consumption of mobile networks by 50% (Gruber *et al.*, 2009). Green Radio (Green Radio, 2007) (Han *et al.*, 2011), Green Touch (Green Touch, 2010), ELBA and Class-S (ELBA and Class S Projects, 2007) are similar projects that have been recently launched, which focus on improving energy efficiency of power amplifiers and other communication subsystem components.

In addition to improving networking components, infrastructure power management can also significantly decrease the total energy consumption in many applications, such as cellular networks. To improve energy efficiency for example, basestations (BSs) that are experiencing light traffic loading can be switched off, and other BSs can be coordinated to replace the removed nodes. In (Bu *et al.*, 2012), Coordinated MultiPoint, CoMP, is used to extend the coverage of the active BSs, and to ensure acceptable service quality in the cells whose basestations have been deactivated. In (Louhi, 2007), an overview is given of improved transmitter efficiency, system features, fresh air cooling, renewable energy sources and energy saving during low traffic periods.

Solar energy is being considered and used to power basestations by many cellular network operators. In 2008, the GSM association (GSMa), consisting of about 800 mobile operators, initiated a program to use renewable energy sources such as solar and wind, to power 118,000 BSs. The aim is to provide communications services in rural and remote areas where connections to the electricity grid are either unavailable or financially prohibitive (GSM Association, 1995). In (Chowdhury and Aziz, 2012), the design of a basestation that is powered by solar and diesel energy is analyzed, and in (Yu and Qian, 2009) a hybrid solar-wind powered basestation is described. A hybrid system consisting of solar, wind and diesel energy was designed in (Nema *et al.*, 2010).

In the research and projects cited above, the basestation use of solar and wind power is mainly in scenarios where there is no access to power grid connections, and therefore it is assumed that the basestation is completely powered by renewable energy. In (Han and Ansari, 2012) a hybrid configuration of grid and solar energy is considered and it is assumed that the cellular network consists of highly loaded BSs (HBS) that are powered by grid electricity, and solar powered BSs that are lightly loaded (LBS). An algorithm is proposed that minimizes the maximal energy depleting rates of the LBSs, therefore enabling more users to be served with green energy. The optimum number of green base stations in a cellular network is analyzed in (Zheng *et al.*, 2012).

To better manage energy consumption, smart grid technology is appearing. In smart grids, in addition to power delivery, a two-way information flow is established between energy providers and clients. The provider declares its energy pricing and the client can manage its energy usage based on these advertisements (Hashmi *et al.*, 2011). This feature has been considered in many published papers which improve power grid performance and at the same time decrease client energy costs. Typically the provider charges more during peak energy hours, which helps to motivate the clients to modify their energy consumption profiles. This in turn, relieves stress on the power grid (Joe-Wong *et al.*, 2012)(Conejo *et al.*, 2010).

4.3 Problem Introduction

We consider a single outdoor communications node which provides some kind of fixed infrastructure functionality[1]. It is assumed that the node is connected to the electric power grid, but that the energy needed for continuous operation may be supplemented with a solar powered add-on. For this reason, the final system is referred to as a hybrid-powered

[1] There are many applications, such as micro-cellular basestations, vehicular roadside units, sensor network back-haul infrastructure, etc.

system, an example of which is shown in Figure 4.1. The solar add-on includes solar panels that have been positioned to absorb the maximum possible solar energy, and an associated battery bank that can store both solar energy, and optionally, draw energy from the power grid connection. The electrical inputs and outputs are interconnected through an *Energy Controller* which, among other things, provides protection from battery over and under-charging. There will also be an AC/DC converted block (not shown in the figure) between the power grid and the energy controller. Further system details and the energy flow model used are given below in Section 4.5.

In this chapter we are interested in the solar add-on deployment and operating costs of the node shown in Figure 4.1. The total cost under consideration consists of the sum of Capital Expenditure (CAPEX) and Operating Expenditure (OPEX) costs over some defined time period[1]. There are many different CAPEX and OPEX scenarios. In our case we assume that the node is already deployed with a power grid connection, and the CAPEX cost of interest is that associated with a solar power add-on to the existing system. In this case, a significant CAPEX cost consists of solar panel and battery provisioning and installation costs. Note that in our work we do not consider the installation cost component, but this can easily be included in our formulations, if desired. Once the solar add-on is installed, the solar energy used is "free" and does not contribute to on-going OPEX costs. For this reason, the major OPEX cost considered, is that of power grid electricity purchases. Clearly, the OPEX and CAPEX costs are dependent, i.e., a higher investment in solar CAPEX cost can reduce ongoing OPEX costs, and vice versa. In the basic model, we therefore assume that the total cost consists of the sum of the solar panel and battery provisioning costs, plus the ongoing energy costs incurred over a specified time period.

[1] CAPEX consists of one-time costs typically associated with the deployment itself, such as installation costs and equipment purchases. OPEX costs consist of on-going operating costs such as those associated with power grid energy purchases.

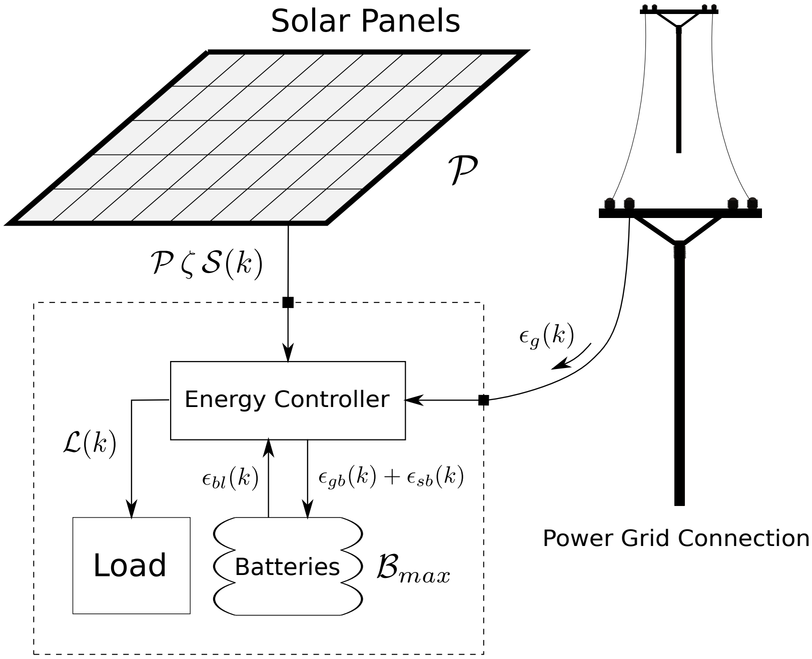


Figure 4.1: Grid Powered Node with Solar Powered Add-on. The figure shows the solar panel/battery/grid (PBG) configuration. The load energy needed during time epoch k is $\mathcal{L}(k)$. The solar panel size and battery capacity are given by $\mathcal{P}$ and $\mathcal{B}_{max}$, respectively. During time period k, solar energy $\mathcal{P}\,\zeta\,\mathcal{S}(k)$ is available, where $\mathcal{S}(k)$ is the per unit area solar insolation and ζ is the panel efficiency. An energy, $\epsilon_g(k)$, may also be purchased from the power grid and stored in the battery or applied to the load, i.e., $\epsilon_g(k) = \epsilon_{gb}(k) + \epsilon_{gl}(k)$. In the configuration shown, energy $\epsilon_{gb}(k)$ and $\epsilon_{sb}(k)$ are placed in the battery from the power grid and the solar panel, respectively. $\epsilon_{bl}(k)$ is drawn from the battery to the load in time period k. In some configurations the node may derive revenue by selling surplus energy back to the power grid.

4.4 Design and Operation Methodology

In this chapter we consider CAPEX and OPEX costs which involve both *offline design* and *online operation* of the node to which solar components are to be added. To avoid confusion, it is important to clearly distinguish between these two phases and how they relate to our algorithms and experiments.

In the *offline design* phase, for each particular solar add-on configuration, the energy provisioning for that configuration is determined. This is done by using historical solar insolation traces for the node's geographic location, combined with estimates of the node's energy usage requirements. The design is assessed from a total CAPEX and OPEX cost viewpoint using the offline input data. When this phase is completed, the solar add-on configuration has been designed, which sets its CAPEX cost.

Once a particular solar configuration has been designed and included in the node, it can be used for *online operation* experiments. In a practical system this phase would continue indefinitely once the solar add-on has been installed. As would be the case in a real system, in our work the online experiments are done using solar insolation traces and node energy loading which are *different* from those which were used in the offline design phase. This procedure allows us to accurately assess the combination of offline and online design that is done in practice.

Based on the above discussion, the design and experimental procedures used in our work are briefly summarized as follows.

- The process starts with input data which includes historical solar insolation traces for the node deployment region, and estimated node energy loading values. In our case we use solar insolation data from the years 1967-1978. An offline design is performed based on minimizing the sum of projected CAPEX and OPEX costs over a target deployment amortization time period. This is done using linear programming formulations discussed in Section 4.6 and gives us the solar add-on configuration to be deployed.

- The designed solar add-on is now deployed and subjected to online experiments which must include a real-time node energy scheduling algorithm which tries to minimize ongoing OPEX costs. The energy scheduling algorithms are introduced in Section 4.7. In our experiments we use different solar insolation traces and node loading values, as would be the case in a practical deployment. In our case, the solar insolation data used is from the years 1979-1990.

- The costs are compared for different configurations and with lower bounds computed using the online experimental inputs. This is done in Section 4.8.

In the next section we formalize these definitions by reviewing the energy flow model used in this chapter.

4.5 Hybrid Node Energy Flow Model

Modeling in photo-voltaic systems is normally done in discrete time using time epochs of duration Δt between time modeling instants[1]. We are interested in the operation of the system over some contiguous time period $\mathcal{T} = [0, K\Delta t]$, where K is a large integer, and for convenience we have started the system at time $t = 0$. In discrete time we define $\mathcal{K} = [1, 2, \ldots, K]$, where each $k \in \mathcal{K}$ corresponds to one Δt time epoch.

For expository purposes, we consider the solar panel/battery/grid (PBG) configuration shown in Figure 4.1. The node energy flow model accounts for the production, purchase, and use of energy over $\mathcal{T}$. We define $\mathcal{B}(k)$ as the residual energy in the battery at the end of time epoch k. Input energy flow to the battery during time period k consists of energy harvested from the solar panels, $\epsilon_{sb}(k)$, and energy purchased from the power grid, $\epsilon_{gb}(k)$[2]. During the same time epoch, energy $\epsilon_{bl}(k)$ is drawn from the battery and consumed in the

[1] Since our experiments will span very long time periods, it is convenient for us to express time in units of hours. It is also well accepted that for solar provisioning purposes, excellent accuracy may be obtained using $\Delta t = 1$ hour time increments (Maghraby *et al.*, 2002)(Farbod and Todd, 2006)(Khatib *et al.*, 2012)(Shrestha and Goel, 1998).

[2] The notation for the energy flow variables is $\epsilon_{xy}(k)$, where x and y are the energy source (e.g., solar (s), grid (g)) and destination (e.g., battery (b), load (l)), respectively.

load. $\mathcal{B}_{min}$ and $\mathcal{B}_{max}$ are defined to be the minimum allowed battery level, and the battery capacity, respectively. $\mathcal{B}_{min}$ is typically based on safety and battery life considerations, which is enforced by the energy controller in Figure 4.1.

Using these definitions, the energy in the battery at the end of time epoch k can be written as

$$\mathcal{B}(k) = \min\{\max\{\mathcal{B}(k-1) + \eta_b^+(\epsilon_{sb}(k) + \epsilon_{gb}(k)) - \epsilon_{bl}(k), \mathcal{B}_{min}\}, \mathcal{B}_{max}\} \quad \forall k \in \mathcal{K} \tag{4.1}$$

where η_b^+ is the charging efficiency of the battery (Sobu and Wu, 2012). Equation (4.1) is a simple recursion that finds the battery energy at time k to be that at time $k-1$, plus the energy received from the solar panels and the power grid, minus the energy supplied from the battery to the load over that time period. Equation (4.1) uses the well known linear energy flow model that is commonly used for photovoltaic node energy provisioning (Maghraby *et al.*, 2002)(Badawy *et al.*, 2010)(Sayegh, 2008). We have added variables to the conventional model to account for the presence of the power grid connection. Note that this recursion ensures that $\mathcal{B}_{min} \leq \mathcal{B}(k) \leq \mathcal{B}_{max}$, $\forall k \in \mathcal{K}$, as required. Typically, we assume that the battery is fully charged at the beginning of its operation, and therefore we define $\mathcal{B}(0) = \mathcal{B}_{max}$.

The load energy required during time epoch k is given by $\mathcal{L}(k)$. This can be supplied from a combination of sources: 1) directly from the solar panels without storage in the battery, 2) from energy purchased from the power grid, and, 3) from energy drawn from the battery. This gives the following load equation,

$$\mathcal{L}(k) = \epsilon_{sl}(k) + \epsilon_{gl}(k) + \eta_b^- \, \epsilon_{bl}(k) \quad \forall k \in \mathcal{K} \tag{4.2}$$

where $\epsilon_{sl}(k)$ and $\epsilon_{gl}(k)$ are the direct solar energy and power grid energy consumed in the load, and η_b^- is the battery discharge efficiency. Finally, the maximum energy that can be drawn from the battery in time epoch k must not exceed that which was available at the start of the interval, i.e.,

$$\epsilon_{bl}(k) \leq \mathcal{B}(k-1) \quad \forall k \in \mathcal{K} \tag{4.3}$$

Assuming that $\epsilon_s(k)$ and $\epsilon_g(k)$ are the total harvested solar and purchased power grid energies in time epoch k, we must have that

$$\epsilon_s(k) = \epsilon_{sb}(k) + \epsilon_{sl}(k) \quad \forall k \in \mathcal{K} \tag{4.4}$$

$$\epsilon_g(k) = \epsilon_{gb}(k) + \epsilon_{gl}(k) \quad \forall k \in \mathcal{K}. \tag{4.5}$$

That is, the solar energy available for storage and the load is equal to the amount harvested. Similarly, the power grid energy applied to battery storage and the load, equals that purchased. The term, $\epsilon_s(k)$, can be written as

$$\epsilon_s(k) = \mathcal{P}\,\zeta\,\mathcal{S}(k) \quad \forall k \in \mathcal{K}. \tag{4.6}$$

where $\mathcal{P}$, ζ and $\mathcal{S}(k)$ are the solar panel size (i.e., area), efficiency, and the per unit area solar energy availability for an optimally oriented solar panel during time period k, respectively. Sample traces of historical values for $\mathcal{S}(k)$ which are used in this chapter are available from meteorological databases.

The above definitions define the node energy flow, inputs, outputs and constraints. In the next section, we formulate the total CAPEX and OPEX cost for the node and derive lower bounds that are used for offline design, and for comparisons with the online systems running energy scheduling algorithms.

4.6 Hybrid Node Total Cost Bounds

Our objective is to minimize the total CAPEX and OPEX cost for the node over the time period $\mathcal{T}$. Using the energy flow model from Section 4.5, this can be found using the linear programs (LPs) formulated in this section. Different versions are given for various node configurations: 1) solar panel/battery/grid (PBG), 2) solar panel/grid (PG), 3) battery/grid (BG), and, 4) PBG with energy revenue (ER).

4.6.1 Solar Panel/Battery/Grid (PBG) Configuration

This is the basic configuration shown in Figure 4.1 which includes a solar panel and battery add-on. The inputs to the problem are given by the set of n-tuples

$$\mathcal{I} = \{(c_b, c_p, c_g(k), \mathcal{L}(k), \mathcal{S}(k), \eta_b^+, \eta_b^-, \zeta, \mathcal{B}_{min}\} \quad \forall k \in \mathcal{K} \tag{4.7}$$

where c_p and c_b are the per unit solar panel and battery prices, respectively, and $c_g(k)$ is the power grid energy purchase price during time epoch k.

The LP finds a lower bound on the minimum total CAPEX and OPEX costs over the set of optimization variables defined by the n-tuples

$$\mathcal{V} = \{(\mathcal{B}_{max}, \mathcal{B}(k), \mathcal{P}, \epsilon_g(k), \epsilon_{gb}(k), \epsilon_{gl}(k), \epsilon_s(k), \epsilon_{sb}(k), \epsilon_{sl}(k), \epsilon_{bl}(k))\} \quad \forall k \in \mathcal{K}. \tag{4.8}$$

The LP finds the optimal battery and solar panel sizing, the power grid energy purchases for all time, and how the input energy is stored and applied to the load[1]. The optimization,

[1] For the purpose of deriving cost bounds and for the offline designs, we model the solar panel and battery sizes as continuous variables. In the online case, these are rounded up to the nearest available panel and battery sizes. An alternative formulation is to introduce integer variables into LP–PBG representing the numbers of panels and batteries. This will result in an integer linear program (ILP) which will, in general, have

referred to as LP–PBG, is first given, and then described below.

$$\begin{aligned}
\underset{\mathcal{V}}{\text{minimize}} \quad & c_b\, \mathcal{B}_{max} + c_p\, \mathcal{P} + \sum_{k\in\mathcal{K}} c_g(k)\, \epsilon_g(k) && && \text{(LP–PBG)}\\
\text{subject to} \quad & \mathcal{B}(k) \le \mathcal{B}(k-1) + \eta_b^+ (\epsilon_{sb}(k) + \epsilon_{gb}(k)) - \epsilon_{bl}(k) && \forall k \in \mathcal{K} && (4.9)\\
& \mathcal{L}(k) \le \epsilon_{sl}(k) + \epsilon_{gl}(k) + \eta_b^-\, \epsilon_{bl}(k) && \forall k \in \mathcal{K} && (4.10)\\
& \epsilon_{sb}(k) + \epsilon_{sl}(k) \le \epsilon_s(k) && \forall k \in \mathcal{K} && (4.11)\\
& \epsilon_{gb}(k) + \epsilon_{gl}(k) \le \epsilon_g(k) && \forall k \in \mathcal{K} && (4.12)\\
& \epsilon_{bl}(k) \le \mathcal{B}(k-1) && \forall k \in \mathcal{K} && (4.13)\\
& \mathcal{B}_{min} \le \mathcal{B}(k) && \forall k \in \mathcal{K} && (4.14)\\
& \mathcal{B}(k) \le \mathcal{B}_{max} && \forall k \in \mathcal{K} && (4.15)\\
& \epsilon_s(k) = \mathcal{P}\, \zeta\, \mathcal{S}(k) && \forall k \in \mathcal{K} && (4.16)\\
& \mathcal{B}(0) = \mathcal{B}_{max} && && (4.17)\\
& 0 \le \epsilon_g(k), \quad 0 \le \epsilon_s(k), \quad 0 \le \epsilon_{gl}(k), \quad 0 \le \epsilon_{gb}(k), && && \\
& 0 \le \epsilon_{sl}(k), \quad 0 \le \epsilon_{sb}(k), \quad 0 \le \epsilon_{bl}(k) && \forall k \in \mathcal{K} && (4.18)\\
& 0 \le \mathcal{P}, \quad 0 \le \mathcal{B} && && (4.19)
\end{aligned}$$

The first two terms in the objective consist of the solar panel and battery CAPEX costs, and the third term is the sum of the OPEX costs of power grid purchases over $\mathcal{T}$. The LP finds the minimum sum of OPEX and CAPEX costs by selecting the best combination of battery and panel sizes, and the amount of energy that should be purchased during each time interval. Most of the terms in LP–PBG come from straightforward modifications to

exponential worse-case complexity. LP–PBG is a relaxation of this formulation with polynomial complexity which gives us a lower bound on cost, and a reduced complexity solution suitable for online heuristics.

the energy flow equations from Section 4.5, as discussed below.

Inequality (4.9) expresses Equation (4.1) as an inequality constraint, as does (4.10) for Equation (4.2). Inequality (4.11) ensures that the solar energy stored and expended in the load during interval k cannot exceed that provided by the solar panel. Similarly, inequality (4.12) performs the same function for purchased power grid energy. Constraints (4.13) to (4.19) are the same that those discussed in results (4.3) to (4.6), plus the obvious non-negativity constraints on the energy and panel/battery sizes. This includes the constraint that batteries are fully charged at the beginning of their operation, i.e., Equation (4.17).

The above LP considers supplementing a grid-powered system with solar powering, which includes battery storage and solar panels. However, depending on limitations and requirements, battery/grid (BG) and solar panel/grid (PG) systems can also be considered. It is also easy to include battery charging/discharging rate constraints in LP–PBG. However, we have found that for the inputs and parameters considered in this chapter, these constraints are not active, and in the interests of simplicity, they have not been included in the above formulation.

Before continuing with our description of other node configurations, we first show the following result, which provides theoretical motivation for our Grid Purchase Last (GPL) energy scheduler, which will be introduced in Section 4.7.1.

Theorem 1. *Consider the basic solar panel and battery (PBG) configuration when $\eta_b^+ = \eta_b^- = 1$. If grid energy pricing is fixed, $c_g(k) = c_g$ for all k, then a feasible minimum cost solution for LP–PBG can always be obtained when $\epsilon_{gb}(k) = 0$ for all k. That is, in order to achieve the optimum offline bound, it is never necessary to store power grid energy in the battery.*

Proof. It is assumed that the battery is ideal, i.e., $\eta_b^+ = \eta_b^- = 1$, and it will be shown that even in this case, it is never necessary to store power grid energy in the battery in order to achieve the offline cost bound. We start from a given optimal solution, and we manipulate it into a more structured one, before we prove the theorem. First, notice that if $\epsilon_{gb}(k) > 0$ in (4.9), then it must be the case

$$\mathcal{B}(k) = \mathcal{B}(k-1) + \epsilon_{gb}(k) + \epsilon_{sb}(k) - \epsilon_{bl}(k) \tag{4.20}$$

otherwise we could reduce $\epsilon_{gb}(k)$ without becoming infeasible and get a solution better than the optimal. Notice that if also $\epsilon_{bl}(k) > 0$, we can apply the following changes for some $\epsilon > 0$, without changing the objective value or violating the feasibility of the solution:

$$\epsilon_{gb}(k) := \epsilon_{gb}(k) - \epsilon, \quad \epsilon_{bl}(k) := \epsilon_{bl}(k) - \epsilon, \quad \epsilon_{gl}(k) := \epsilon_{gl}(k) + \epsilon.$$

After these changes $\epsilon_{gb}(k)$ or $\epsilon_{bl}(k)$ becomes 0. Then (4.20) is replaced by

$$\mathcal{B}(k) = \mathcal{B}(k-1) + \epsilon_{gb}(k) + \epsilon_{sb}(k) \tag{A}$$

or by

$$\mathcal{B}(k) = \mathcal{B}(k-1) + \epsilon_{sb}(k) - \epsilon_{bl}(k). \tag{B1}$$

The rest (when $\epsilon_{gb}(k) = 0$) of the constraints (4.9) are of the form

$$\mathcal{B}(k) \leq \mathcal{B}(k-1) + \epsilon_{sb}(k) - \epsilon_{bl}(k) \tag{4.21}$$

and by decreasing $\mathcal{B}(k-1)$ while still maintaining (4.13), we get either

$$\mathcal{B}(k) = \mathcal{B}(k-1) + \epsilon_{sb}(k) - \epsilon_{bl}(k) \ \wedge \ \mathcal{B}(k-1) > \epsilon_{bl}(k) \tag{C1}$$

or

$$\mathcal{B}(k) \leq \mathcal{B}(k-1) + \epsilon_{sb}(k) - \epsilon_{bl}(k) \ \wedge \ \mathcal{B}(k-1) = \epsilon_{bl}(k). \tag{C2}$$

Note that the constraints in group (B1) can be distributed (relaxed to an inequality) into groups (C1), (C2), depending on whether $\mathcal{B}(k-1) > \epsilon_{bl}(k)$ or $\mathcal{B}(k-1) = \epsilon_{bl}(k)$ respectively, i.e, we end up with an optimal solution that satisfies (A), (C1), (C2).

Suppose that k_{max} is the largest time index in (4.9), and k the biggest index of any constraint (4.9) from group (A). Then any sequence of constraints (4.9) after time index k are of the following general form in terms of the group they belong to:

index	$\dots$	k	$k+1$	$\dots$	$k+i$	$k+i+1$	$\dots$
group	$\dots$	(A)	(C1)	$\dots$	(C1)	(C2)	$\dots$

We note that from what follows it will become apparent how to handle the special cases $i=0$ (no (C1) constraints following k) and $k+i = k_{max}$ (only (C1) constraints, if any, following k). Also note that $\mathcal{B}(l-1) > \epsilon_{bl}(l) \geq 0, k+1 \leq l \leq k+i$. This implies that $\mathcal{B}(k+i) = \epsilon_{bl}(k+i+1) > 0$, because otherwise $\mathcal{B}(k+i) = 0 = \mathcal{B}(k+i-1) + \epsilon_{sb}(k+i) - \epsilon_{bl}(k+i) > 0$, a contradiction (the same contradiction arises in the case $i=0$, because $\mathcal{B}(k) = \mathcal{B}(k-1) + \epsilon_{gb}(k) + \epsilon_{sb}(k) > 0$).

Let

$$\epsilon_1 := \text{min}_{k+1 \leq l \leq k+i}\{\mathcal{B}(l-1) - \epsilon_{bl}(l)\}$$

and let

$$\epsilon := \text{min}\{\epsilon_1, \epsilon_{gb}(k), \epsilon_{bl}(k+i+1)\}.$$

Obviously, $\epsilon > 0$. Then if we perform simultaneously the following changes, we maintain feasibility and we reduce $\epsilon_{gb}(k)$ by ϵ:

$$\begin{aligned}
\epsilon_{gb}(k) &:= \epsilon_{gb}(k) - \epsilon, \\
\mathcal{B}(k) &:= \mathcal{B}(k) - \epsilon, \\
\mathcal{B}(k+1) &:= \mathcal{B}(k+1) - \epsilon, \\
&\dots \\
\mathcal{B}(k+i) &:= \mathcal{B}(k+i) - \epsilon, \\
\epsilon_{bl}(k+i+1) &:= \epsilon_{bl}(k+i+1) - \epsilon, \\
\epsilon_{gl}(k+i+1) &:= \epsilon_{gl}(k+i+1) + \epsilon.
\end{aligned}$$

Notice that after these changes the composition of groups (A), (C1), (C2) have changed: the constraint (4.9) at time index k will move to group (C1) or (C2) (in case $\epsilon = \epsilon_{gb}(k)$), or a constraint (4.9) between indices $k+1$ and $k+i$ will move to group (C2) (in case $\epsilon = \epsilon_1$, or in case $\epsilon = \epsilon_{bl}(k+i+1)$, because then $\mathcal{B}(k+i) = 0$ and, hence, (i) all the constraints (4.9) at indices $k+i, k+i-1, \dots, k$ must move to group (C2), and (ii) $\epsilon_{gb}(k) = 0$, in order to avoid a contradiction).

We continue in a similar fashion, until group (A) becomes empty.

□

4.6.2 Battery/Grid (BG) Configuration

In a BG configuration, electricity from the power grid can be purchased and stored for future use. This may be advantageous when power grid costs are time dependent, e.g., when $\epsilon_g(k)$ is not the same for all k. During off-peak hours, when power grid pricing is reduced, battery storage for future use may lead to reduced OPEX costs. By setting $\mathcal{P} = 0$ in LP–PBG we can obtain the offline lower cost bound for the battery/grid case.

4.6.3 Solar Panel/Grid (PG) Configuration

The panel/grid configuration is another deployment option. In this case, the capability of storing electricity in off-hours is eliminated, and all the input energy from the solar panels is applied directly to the load whenever possible. Since energy pricing is normally more expensive during daytime hours, there is a good correlation between solar energy availability and higher grid pricing.

To formulate the panel/grid problem, LP–PBG can be modified as follows.

$$\begin{aligned}
\underset{\mathcal{P},\,\epsilon_g(k)}{\text{minimize}} \quad & c_p\,\mathcal{P} + \sum_{k\in\mathcal{K}} c_g(k)\,\epsilon_g(k) && && \text{(LP–PG)}\\
\text{subject to} \quad & \mathcal{L}(k) \le \mathcal{P}\,\zeta\,\mathcal{S}(k) + \epsilon_g(k) && \forall k \in \mathcal{K} && (4.22)\\
& 0 \le \epsilon_g(k) && \forall k \in \mathcal{K} && (4.23)\\
& 0 \le \mathcal{P} && && (4.24)
\end{aligned}$$

In this case, the LP selects the best solar panel size and power grid energy purchases. This is subject to providing sufficient load energy during each time epoch.

4.6.4 Energy Revenue (ER) Configuration

We also consider the case where the PBG configuration sells unused energy back to the power grid. In this case we define $\epsilon_r(k)$ to be the energy sold to the power grid during time epoch k. The objective function in LP–PBG will now become

$$c_b \, \mathcal{B}_{max} + c_p \, \mathcal{P} + \sum_{k \in \mathcal{K}} c_g(k) \, \epsilon_g(k) - \sum_{k \in \mathcal{K}} r_g(k) \, \epsilon_r(k) \tag{4.25}$$

where $r_g(k)$ is the unit price *paid* by the power grid for surplus energy at time k. We also introduce two new optimization variables, $\epsilon_{bg}(k)$ and $\epsilon_{sg}(k)$, which are battery-to-grid and solar-to-grid energy transfers during time interval k. Inequality (4.9) now becomes

$$\mathcal{B}(k) \leq \mathcal{B}(k-1) + \eta_b^+(\epsilon_{sb}(k) + \epsilon_{gb}(k)) - \epsilon_{bl}(k) - \epsilon_{bg}(k) \quad \forall k \in \mathcal{K} \tag{4.26}$$

and expressions (4.11) and (4.13) will change to

$$\epsilon_{sb}(k) + \epsilon_{sl}(k) + \epsilon_{sg}(k) \leq \epsilon_s(k) \quad \forall k \in \mathcal{K} \tag{4.27}$$

and

$$\epsilon_{bl}(k) + \epsilon_{bg}(k) \leq \mathcal{B}(k-1) \quad \forall k \in \mathcal{K} \tag{4.28}$$

We must also have that

$$\epsilon_r(k) \leq \epsilon_{sg}(k) + \eta_b^- \, \epsilon_{bg}(k) \quad \forall k \in \mathcal{K} \tag{4.29}$$

Given these changes to LP–PBG, we can easily compute a lower bound on cost for the energy revenue configuration.

The offline LPs formulated above have access to all past and future input data, and therefore, they provide only lower bounds on cost, which are in general, not achievable in a practical system, i.e., if the minimum cost solar provisioning (i.e., $\mathcal{P}$ and $\mathcal{B}_{max}$) is used, the bound cannot generally be realized. For this reason, online energy scheduling algorithms are needed which control energy flow, purchases and storage decisions in real time. In the following section, three online algorithms are proposed and performance results are compared with the lower bounds in Section 4.8. As described in Section 4.4, the system is first configured using the LPs defined in this section. This sets the CAPEX costs. The purpose of the online algorithms is then to reduce OPEX costs as much as possible, by proper energy scheduling, given the node configuration.

4.7 Online Energy Scheduling Algorithms

When the system is operated online, the solar configuration and its associated CAPEX costs, have already been determined. The objective of the online algorithms is then to schedule node energy use with minimum OPEX costs. We first consider the PBG configuration discussed in Section 4.6.1. The algorithms to be discussed are also applicable for the BG configuration.

4.7.1 Grid Purchase Last (GPL) Algorithm

The Grid Purchase Last (GPL) algorithm is based on the discussion at the end of Section 4.6.1 and the result which motivates the purchase of power grid energy as a last resort. This is also motivated by the idea that when energy is bought and stored before it is actually needed, the chance of energy wastage is increased due to battery overflow. Accordingly, the GPL algorithm defers any power grid purchases as long as possible. The details of it's operation are shown in Algorithm 4.1.

In step 2, if there is sufficient solar energy to power the load, this is the option taken.

Algorithm 4.1. Grid Purchase Last (GPL) Algorithm

```
for all $k \in [1, 2, \cdots, \infty)$ do
    if $\mathcal{L}(k) \leq \epsilon_s(k)$ then
        Supply load, $\mathcal{L}(k)$, using solar energy and place any residual energy in the battery, i.e., $\epsilon_{sb}(k) = \epsilon_s(k) - \mathcal{L}(k)$.
    else if $\epsilon_s(k) < \mathcal{L}(k) \leq \epsilon_s(k) + \eta_b^- \mathcal{B}(k-1)$ then
        Supply $\mathcal{L}(k)$ using both solar and battery energy, i.e., $\epsilon_{bl}(k) = \epsilon_s(k) - \mathcal{L}(k)$.
    else
        Supply $\mathcal{L}(k)$ using using available solar and battery energy plus a grid power purchase, i.e., $\epsilon_{gl}(k) = \epsilon_s(k) + \eta_b^- \mathcal{B}(k-1) - \mathcal{L}(k)$.
    end if
end for
```

In this case any residual solar energy, $\epsilon_s(k) - \mathcal{L}(k)$, is made available for storage in the battery. If there is insufficient solar energy, then energy is also drawn from the battery to make up the shortfall, as shown in step 5. Finally, only if solar and battery reserves are insufficient energy is purchased from the power grid (step 7). In this case the purchase is just enough to supply the energy shortfall.

The GPL algorithm is very simple and is likely to be the sort of default algorithm used in many practical situations. However, there are many scenarios where its performance is not very good. In the following, two other algorithms are introduced which take into account solar insolation factors, traffic loading and grid energy pricing. In order to accomplish this we must have predictions of future solar insolation values.

4.7.2 Solar Load Optimization (SLO) Algorithm

The SLO Algorithm uses predictions of input solar insolation values, $\mathcal{S}(k)$, over a future window of duration $w\Delta t$ as a basis for its energy scheduling decisions. The predictions that we use are based on the algorithm first proposed in (Ali *et al.*, 2010), and is briefly reviewed in Appendix A. This is used to predict both future solar energy, $\mathcal{S}(k)$ and the

Algorithm 4.2. Solar Load Optimization (SLO) Algorithm

```
for all $k \in [1, 2, \cdots, \infty)$ do
    Use the prediction from Equations (A.1) to (A.5) and LP–PBG for $k+1 \leq i \leq k+w$ with the updated objective (4.30) to find the target variables including $\epsilon_g(i)$ for $i = k, k+1, \ldots, k+w$.
    Implement the energy flow in accordance with the solution obtained in step 2. If $\mathcal{L}(k)$ is higher than its prediction, or if $\mathcal{S}(k)$ is lower, then draw the additional energy needed from the battery, i.e., increase $\epsilon_{bl}(k)$ above its predicted value. If this is insufficient, purchase additional energy from the power grid, i.e., increase $\epsilon_{gl}(k)$ to make up the shortfall.
end for
```

load, $\mathcal{L}(k)$. Using these *predicted* values, the algorithm solves a linear program over the next w time epochs, which gives estimated values of $\epsilon_g(k)$ to use for power grid energy purchases.

More formally, at time k the algorithm estimates the values of $\mathcal{S}(i)$ and $\mathcal{L}(i)$ for $i = k+1, k+2, \ldots, k+w$, and uses these estimates with the known value of $\mathcal{B}(k)$ in LP–PBG to determine the estimated future values of $\epsilon_g(j), \epsilon_{gb}(j), \epsilon_{gl}(j), \epsilon_{sb}(j), \epsilon_{sl}(j)$, for $k \leq j \leq k+w$. Note however, that the objective function in LP–PBG is replaced by

$$\sum_{i=k}^{i+w} \epsilon_g(k)\, c_g(i) \tag{4.30}$$

since the battery and panel sizes are already determined, and are no longer variables of the LP. The only remaining variables are the amounts of energy that should be purchased from the grid in each future time epoch. the CAPEX costs are already determined. The above result gives estimates for the variables indicated, however, the “next step” values for $k+1$ are the only ones used. This is because the entire process is repeated and the estimates are updated for each Δt period.

The algorithm is run at the end of every Δt time epoch. At each run the algorithm introduced in Appendix A is used in step 2 to predict the available solar energy and the power consumption in the next $w\Delta t$ hours, i.e., for $i = k+1, k+2, \ldots, k+w$. The LP Solution gives a set of grid energy variables, $\epsilon_g(j)$ for $j = k, k+1, \ldots, k+w$, that should be purchased over this time period. Since the algorithm is run every Δt, only the result for the current time period, k, is purchased.

4.7.3 Solar Load Simulation (SLS) Algorithm

The SLO Algorithm involves solving an LP at each time epoch, which is undesirable. In this section we consider an alternative with the same motivation but does not have this requirement. This algorithm is referred to as the Solar Load Simulation (SLS) Algorithm and is shown in Algorithm 4.3. The idea behind the algorithm is that when a grid energy purchase is needed, the algorithm is willing to purchase the energy early, provided that there is room in the battery and the purchase price is lower. As in SLO, predicted values for solar insolation and load are used for a window of w time epochs extending into the future.

In Step 3 the predicted values in Step 2 are used to find the battery energy levels at each time epoch over next w intervals. This is accomplished by simulating the battery energy recursion (4.1), subject to Equations (4.2) to (4.6) using the GPL Algorithm. We then find the first time period, p, where a future power grid energy purchase is predicted, i.e., step 4. If the current energy purchase price (at time k) is less than all other time periods between k and p, and purchasing energy would not overflow of the battery for $i \in \{k+1, \ldots, p\}$, then we move the grid energy purchase from time p to k. This is shown in steps 5 and 6. Note that this early energy purchasing is with respect to the current available energy in the battery. The variables used in the recursion are then updated.

Algorithm 4.3. Solar Load Simulation (SLS) Algorithm

```
1: for all $k \in [1, 2, \cdots, \infty)$ do
2:     Obtain the predictions from Equations (A.1) to (A.5) for $i \in \{k+1, k+2, \ldots, k+w\}$.
3:     Using the predicted values from step 2, and known values of available solar energy and
       battery energy level at time $k$, simulate the system over $\{k, \ldots, k+w\}$ using the battery energy
       recursion (4.1), and (4.2) to (4.6). When doing this, use the GPL Algorithm.
4:     Find $p = \min j : 0 < \epsilon_g(j)$ for all $j \in \{k+1, \ldots, k+w\}$.
5:     if $c_g(k) < c_g(i)$ and $\mathcal{B}(i) < \mathcal{B}_{max}$ for all $i \in \{k+1, \ldots, p\}$ then
6:         $\epsilon_g(k) := \epsilon_g(k) + \min(\epsilon_g(p), \mathcal{B}_{max} - \mathcal{B}(k))$.
7:     end if
8: end for
```

4.8 Simulation Results

The offline and online results are obtained as follows. For a particular solar add-on configuration, historical solar insolation data and estimates of the node loading over a target time period are used to obtain a good energy provisioning (i.e., solar panel and battery configuration) for the node. The time period used for this are the years 1967-1978 taken from solar insolation data for Boston, MA. This is done by using the appropriate LP from Section 4.6, for the desired node configuration. This result determines the CAPEX cost of the solar add-on. Once this has been obtained, we subject the configured node to online experiments. For this purpose we use solar insolation data from the years 1979-1990 for the same geographic location. The node energy loading assumed in this case is correlated to the original, as discussed below.

A wide variety of experiments have been performed that illustrate the performance of the proposed algorithms. In this section we present some examples of these results that illustrate various tradeoffs and conclusions. We have also evaluated the algorithms for New York, Atlanta, Phoenix and Seattle, cities which have different climate conditions than Boston, for the same period of time.

Although node temporal loading is application dependent, for the examples in this chapter we use a model obtained from the observations in (Correia *et al.*, 2010) that show a 24hr sinusoidal periodicity in averaged basestation power consumption. Accordingly, we define the average power usage of the load by P_L. In addition, the minimum and maximum value of energy consumption of the node occurs at 5 a.m. and 5 p.m., respectively. Based on the results presented in (Correia *et al.*, 2010) (Oh *et al.*, 2011) (Frenger *et al.*, 2011), which show a time-correlated daily periodicity in energy use, we adopt the following node power consumption model where $P(k)$ is the power consumption at the start of time epoch k.

$$P(k) = P_L(1 + \frac{1}{3}\cos(\frac{\pi}{12}(k\Delta t + 7))) + \gamma_n \quad \forall k \in \mathcal{K} \tag{4.31}$$

where γ_n is a normally distributed random variable with mean of 0 and standard deviation of 70 W, which is used to model random perturbations in the loading. Note that in Equation (4.31), $k = 0$ corresponds to midnight at the start of the first day considered in $\mathcal{T}$. The actual node energy requirements are therefore given by $\mathcal{L}(k) = P(k)\,\Delta t$.

A *Time of Use* (TOU) pricing model is used which has two pricing modes. It is assumed that electricity price is on-peak from 7 a.m. to 6 p.m., and the price during these hours is 1.5 times the average, and the off-peak price is half the average (Yang *et al.*, 2012). Unless otherwise stated, the unit battery and panel specifications as well as other parameter values are given in Table 4.1. The default value for P_L is set to 1450 W (Arnold *et al.*, 2010).

For comparison purposes we include three different node powering configurations:

1. Operation completely from power grid energy (Grid-Only (GO)), i.e., no solar add-on.

Solar Panels	
Size	$1.65 \times .992\,\mathrm{m}^2$
ζ	0.1497
Cost	$197 CAN

Default Parameters	
w	24 hours
H	3 hours
D	20 days
π	0.2
Δt	1 hour

Batteries	
Capacity	150 Ah
η_b^+, η_b^-	0.92
Cost	$229 CAN

Table 4.1: Default Parameter Settings

2. Operation completely from the solar add-on, i.e., it is provisioned for energy sustainable (Solar Only (SO)) operation.

3. The hybrid case where the node is powered by a suitable combination of solar and grid energy, i.e., PBG.

The intent is to design a minimum cost system for the *online time period*, 1979-1990, and to compare the total costs for various configurations and parameters. The plotted lower bounds on cost are obtained using the *actual online data* for 1979-1990, and solving the appropriate LP in Section 4.6. As discussed previously, the data from 1967-1978 are used to determine the CAPEX cost configuration of the system in the offline design phase. Once this is done, the actual values of traffic loading and solar insolation for the intended deployment years of 1979-1990 are used in the algorithms that were introduced in Section 4.7 for an online computation of the required energy and OPEX costs.

The hybrid scenario can be divided into three categories. In the first, both batteries and panels are used, i.e., the PBG case from Section 4.6.1. In the second case, only a solar panel is used to augment node energy requirements, i.e., the PG case from Section 4.6.3. Finally we have the battery/grid case, i.e., BG case from Section 4.6.2 in which no solar energy is used. In this case electricity can be stored in the battery and used later. In the following, simulation results for each category are presented.

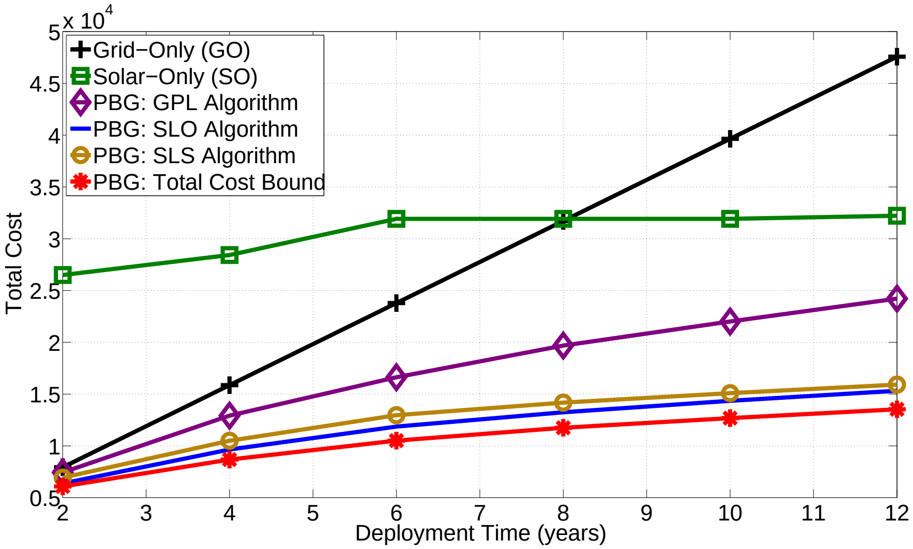


Figure 4.2: Example Comparison of GO, SO and PBG Configurations. Total cost is plotted versus the add-on (amortization) deployment time, assuming an energy pricing of 0.3 $/kWh.

4.8.1 Solar Panel, Battery and Grid (PBG) Case

In the first experiment, the performance of the PBG configuration is compared with the cases where we operate the system with grid-only (GO) powering, and solar-only (SO) powering. The GO case is the original system without any addition, and the second is with a solar/battery add-on which is provisioned to be energy sustainable, i.e., no grid purchases are needed. Results are evaluated for deployment time periods ranging from 2 to 12 years. The average energy price varies over a wide range from 0.1 to 0.5 $\$/kWh$.

In the first set of results shown in Figure 4.2, the average energy price is set to 0.3 $/kWh, with the daily fluctuation discussed above, and the performance of the algorithms in terms of the total cost, i.e., the sum of CAPEX and OPEX costs, is compared for an add-on deployment time from 2 to 12 years. It can be seen that for this set of parameters, the cost break-point between a GO and SO configuration is about 8 years. Clearly the SO

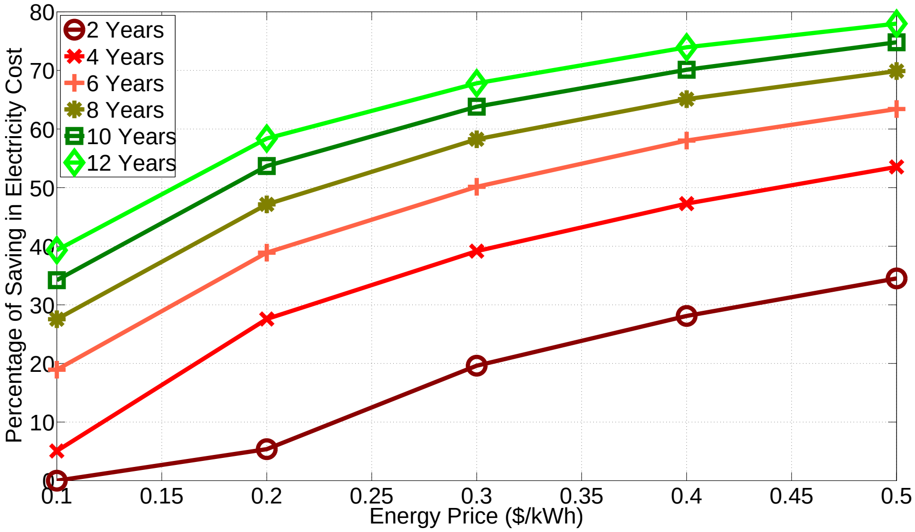


Figure 4.3: Percentage of Cost Savings using PBG versus GO Configurations. This is plotted versus energy pricing for different CAPEX amortization periods.

case incurs a high initial CAPEX cost but no ongoing OPEX costs, compared to the GO case which is exactly the opposite. An interesting result is that by properly combining the two configurations, we can obtain a system with significantly less total cost. From Figure 4.2, the GO scenario is up to about 100% more expensive than the PBG configuration, even when the Grid Purchase Last (GPL) algorithm is used. However, the GPL scheduler has relatively poor performance compared with SLO and SLS. This is because it does not take into account the daily grid power pricing differences. In this case, where we have used practical numbers, the cost of the system using GO is about 210% higher than that obtained with the SLO and SLS algorithms. In results that we have not shown, we find that when the grid pricing is fixed, i.e., $c_g(k) = c_g$ for all k, the GPL scheduler performance is about the same as SLO and SLA. This observation is consistent with the theoretical result obtained for the offline optimization in Theorem 1.

Figure 4.3 summarizes the cost saving percentage obtained by using the PBG configuration compared with the Grid-Only case for different energy pricing. This information is plotted for different CAPEX amortization periods. In this case we have used the SLO scheduler rather than GPL for the previously mentioned reasons. It can be seen that the hybrid system will lead to between 40% to 78% savings in the total cost over a wide range of energy prices. Note that when the deployment time is as short as two years and electricity pricing is as low as 0.1 $/kWh, it is probably not worth investing in a solar power add-on. In this case the optimizer prefers to continue using only grid energy. Notice that there is a “saturation effect” in Figure 4.3. For larger time periods, the percentage improvement as a function of energy pricing in going to a solar add-on, flattens out. This is caused by the decreasing use of grid energy to power the node as we move to the right in the figure. Eventually the node will be configured to be fully solar powered and costs will then be independent of increases in grid energy pricing. Clearly. Figure 4.3 makes a good case for solar powered add-on configurations as future energy pricing increases.

4.8.2 Solar Panel/Grid (PG) Configuration

In the next set of results, the node add-on consists only of solar panels, and its performance is compared with the GO configuration. Using a PG configuration is interesting because solar energy output obviously occurs only during daytime hours. However, this is a good match since this is exactly when node energy consumption and grid energy pricing tends to be higher. In these experiments, LP–PG has been used to determine the optimum cost for variable energy pricing and for different deployment periods.

The total cost bound for PG along with its online results in comparison with the Grid-Only case are shown in Figure 4.4. It can be seen that there is very little difference between the bound and the online results. This is because there is no intelligent energy scheduling required, i.e., any available solar energy is immediately used, and the power grid fills in

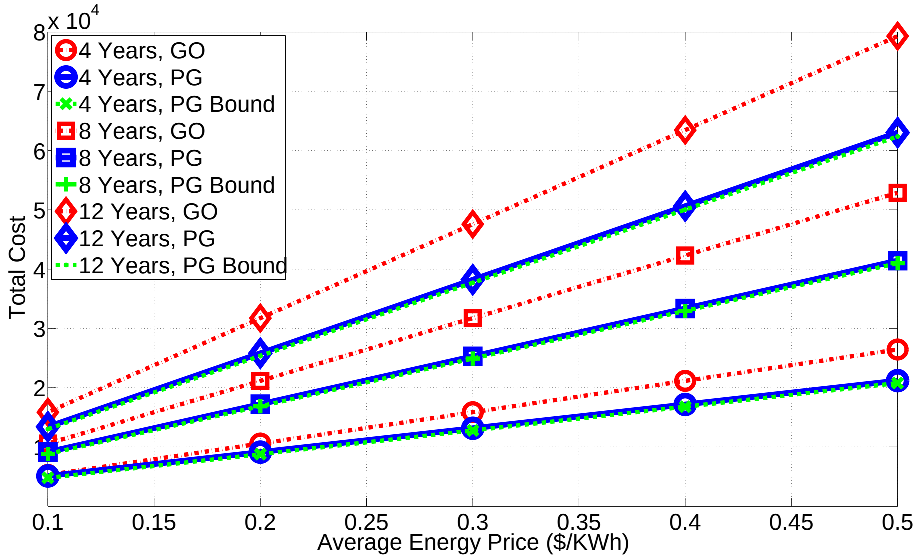


Figure 4.4: Total Cost (OPEX and CAPEX) for a Panel/Grid (PG) Configuration Compared with the Grid Only (GO) case.

any shortfall. Similarly, if the solar energy is higher than that needed, the surplus is lost. It can be seen from this figure that the savings in cost are much more modest than the cases that we considered previously. For example, when the deployment time is more than about 4 years, up to 21% savings is achievable over a wide range of energy prices. Only when energy pricing becomes very high is this arrangement reasonable. Too much of the available solar energy is lost and cannot be applied during non-daytime hours.

4.8.3 Battery/Grid (BG) Configuration

The converse case to PG is to use a battery only, BG, node add-on. In this case all consumed energy will be that from the power grid. This configuration makes no sense if energy pricing is fixed, i.e., when $c_g(k) = c_g$ for all k. When this is not the case however, the node may choose to store energy purchases made at lower prices for use later. This makes sense since, in practice, elevated pricing tends to occur over long time periods during peak usage hours.

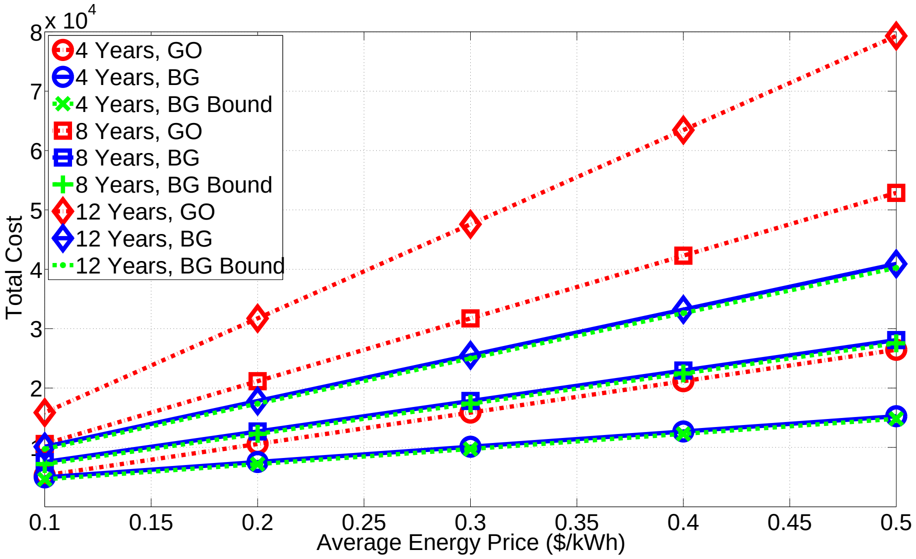


Figure 4.5: Total cost (OPEX and CAPEX) for the Battery/Grid (BG) Configuration Compared to the Grid Only (GO) Case.

The SLP Algorithm is used to determine battery provisioning cost, and this result, along with the total cost bound are compared with the GO case in Figure 4.5. In this case we see that the cost improvements due to the add-on are much more significant than in the PG configuration. For example, up to almost 50% cost savings can be obtained in this method for the energy price range considered before. This improvement over the BG case is due to the fact that the “energy time shifting” can occur over large time periods and is not subject to the energy loss associated with the PG configuration’s inability to store energy. Clearly, for the parameters that we are considering, BG provides better cost saving performance than PG. This scenario could change however, if, for example, the relative unit costs of the battery and solar panel were to dramatically change from their current values.

4.8.4 Energy Revenue (ER) Configuration

In this section it is assumed that the node add-on is configured to sell unused energy back to the power grid. The LP introduced in Section 4.6.4 is used to determine the CAPEX and provisioning costs using the modified version of LP–PBG as discussed in that section. The CAPEX configuration of the add-on therefore takes into account the revenue scenario. We then present results using the SLO algorithm and this modification to the LP. The online algorithm therefore schedules grid energy purchases, but also determines if surplus energy should be sold back to the grid at each time epoch.

Simulation results when the average energy price for purchasing electricity differs from the selling price are shown in Figure 4.6. The average energy purchase price is 0.3 $/kWh, and the add-on amortization time is set to 10 years. It is clear from this figure that permitting energy revenue strongly affects the economics of the node add-on. The GO case is the top curve shown which is significantly more expensive than the PBG configuration. In this figure, separate curves have been shown for the add-on CAPEX and selling profit values. With a revenue capability, the overall cost is seen to decrease as the energy selling price increases. It can be seen that if the selling price increases, the optimizer aggressively increases CAPEX expenditures. Eventually, if the selling price becomes sufficiently high, the node can turn a profit by investing in solar panels and battery storage, and by time-shifting energy purchases.

4.8.5 Example for Different Locations

A wide variety of results have been obtained for different locations, and compared for New York, Atlanta, Phoenix, Seattle and Boston. Among these, Boston and new York have almost the same climate. Seattle has a marine climate, with a relatively high fraction of

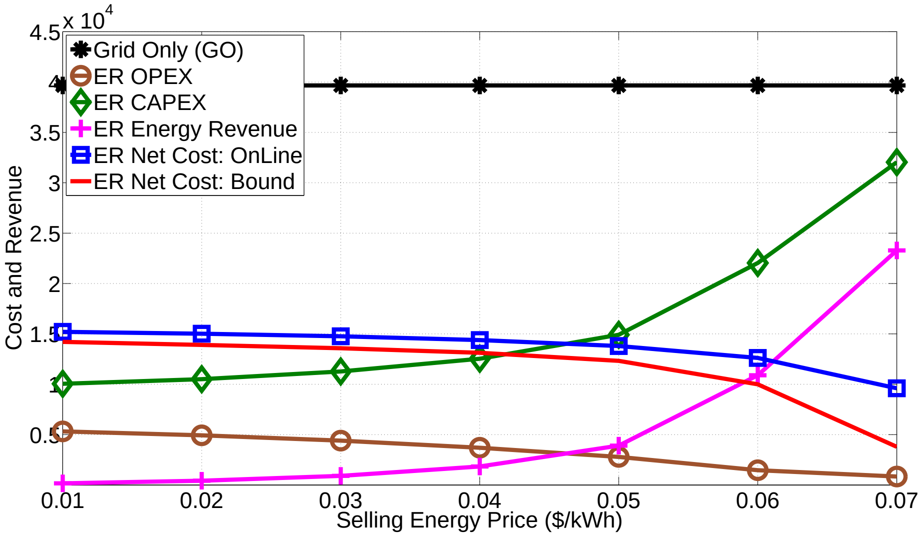


Figure 4.6: Energy Revenue Example with PBG Configuration

overcast days that significantly affect CAPEX provisioning costs. Atlanta has a temperate climate, and Phoenix has a subtropical arid climate with extremely hot summers and warm winters, with an abundance of solar energy.

As a final example, the total costs for a deployment period of 10 years are shown in Figure 4.7 using the PBG configuration. Phoenix clearly has the lowest total cost and Seattle, as expected, has the most expensive solar add-on provisioning costs. It can be seen that Phoenix and Atlanta have the best cost savings, followed by New York and Boston. Clearly, the economics of a solar add-on are highly location dependent.

4.8.6 Discussion

The development and results introduced in this chapter provide a methodology and strong indications for the future use of hybrid grid and solar powered infrastructure. However, as noted in Section 4.3, one should be careful at assessing all of the costs associated with providing solar powered additions. For example, if installation costs become significant,

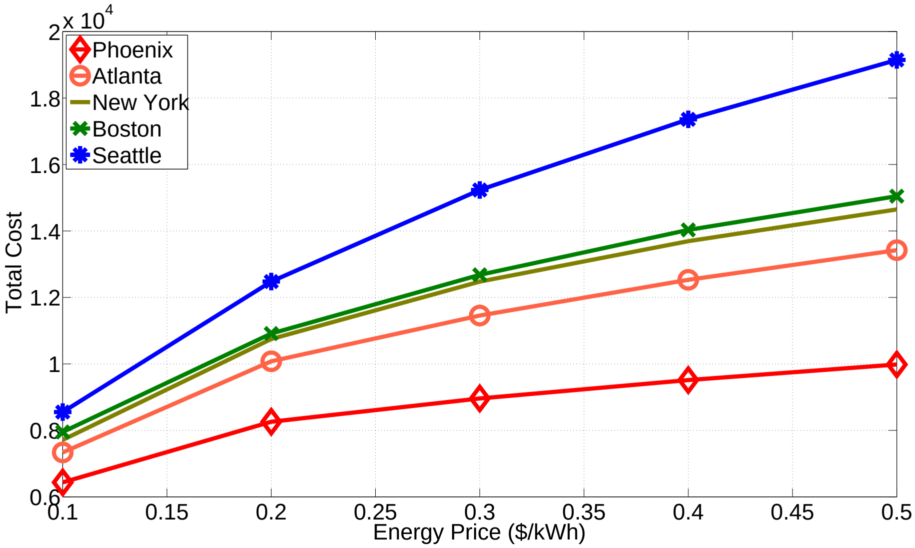


Figure 4.7: Cost Comparisons For Different Locations using the PBG Configuration.

then these should be taken into account when making cost comparisons. This may very well result in shifts to the cost curves which will affect the timing of cost break-points. These types of costs can easily be incorporated into the formulations.

4.9 Conclusions

In this chapter, we have considered the operating and capital costs of providing a solar powered add-on to power grid operated communications infrastructure nodes. Both capital (CAPEX) costs and operating (OPEX) costs were considered. CAPEX costs are associated with provisioning the solar panel and battery components of the add-on, and are selected using an offline design method. The operating (OPEX) costs are those associated with power grid energy purchases when the node is online. Online energy scheduling algorithms were also introduced. Lower bounds on the costs were also derived using linear programming (LP) formulations, where solar components are sized using solar insolation and projected loading data.

To compare the performance of the proposed algorithms, solar insolation traces for the city of Boston, MA. were used. Simulation results show that when comparing with the grid energy only case, using the proposed energy management scheduling algorithms results in a 40% to 78% deduction in the total cost. This is achieved when a combination of batteries and panels are used to supplement purchased power grid energy. The improvements obtained are 21% and 48% for the panel/grid and battery/grid configuration cases, respectively. Simulation results were also performed for other cities with different solar insolation profiles. Results show that the algorithms achieve better cost improvements when solar power is more plentiful.

Chapter 5

Energy Provisioning in Stand-alone Solar Powered Systems with Deferrable Load

5.1 Introduction

In this chapter we show that in a stand-alone PV system where part of the energy demand is deferrable, it is possible to reduce the total provisioning cost by careful energy management. A single solar powered node is considered where its energy (load) demands can be divided into non-deferrable and deferrable components. Non-deferrable loads are referred to as the part that must be satisfied immediately upon their request, and deferrable loads are those that are delay tolerant, i.e., their satisfaction can be postponed up to a predetermined deadline. Electric vehicles, heating, ventilation, and air conditioning equipment are examples of power consumers with deferrable energy demand which can defer their activation from several minutes to several hours.

The energy control problem is formulated using a Mixed Integer Linear Programming (MILP). This is done for the specified deployment time, and its solution results in a lower bound on the energy provisioning cost. An algorithm called Delay Aware Provisioning (DAP) for offline resource provisioning is then introduced and its performance is evaluated using several simulation scenarios.

5.2 Related Work

A substantial portion of the energy demands of residential, commercial and industrial consumers are time flexible. This fact creates an opportunity to manage loading in a way which increases the performance of the system. Load management in power systems has been discussed since the early 1980's (Ipakchi and Albuyeh, 2009) and various management methods including peak shaving, peak shifting and direct load management have been addressed in the literature and implemented by many utilities.

In (Vukojevic *et al.*, 2012) deferrable loads are considered as a way to decrease the effect of overloading in transformers. An algorithm was proposed to increase the expected transformer lifetime by shifting deferrable loads to off-peak hours. The unpredictability and non-controllable fluctuation of renewable power sources is studied in (Papavasiliou and Oren, 2010). In order to mitigate this issue, the problem is presented in the form of a stochastic dynamic program and two algorithms are proposed for supplying renewable power to time-flexible electrical loads. The idea is that the requests from flexible consumers are received and regulated by a central scheduler which determines the allocation of renewable resources to consumers subject to their deadlines.

An issue with renewable energy resources is the effect of their high penetration on the power grid. With current growth rates of solar energy generation, it is expected that solar PV power plants will be a significant component of power systems in the near future. High solar energy generation and input into the power system during periods of low energy demand may cause problems for the power grid. This issue has been addressed and studied in (Steffel *et al.*, 2012), and as a solution the authors have suggested that consumers with delay tolerant loading shift their demands to peak PV output intervals.

Deferrable loads have also been used to decrease the energy bills for consumers. In (Alizadeh *et al.*, 2012) Time of Use (TOU) pricing of energy is discussed and the problem of minimizing the operating cost for consumers that have deferrable loads is formulated and solved. In (Shinwari *et al.*, 2012) a water-filling based method for scheduling power consumption is proposed, such that the overall demand becomes relatively constant, which will reduce the cost of electricity generation. The load is divided into soft and hard load components. To reduce the peak-to-average load ratio and flatten the overall energy consumption profile, soft loads are allocated to time intervals with less hard loads.

In the remainder of this chapter we will show that in addition to all the above advantages of deferrable loads, it is possible to decrease the provisioning cost of an off-grid solar powered system by properly scheduling its energy use.

5.3 Motivation for Delay Aware Provisioning

The participation of deferrable loads in the total energy consumption of a household consumer is expected to grow significantly in the coming years as a result of the increasing penetration of Plug-in Hybrid Electric Vehicles (PHEV) and smart appliances (Gan *et al.*, 2013). According to (Ipakchi and Albuyeh, 2009), to drive with an electric vehicle for

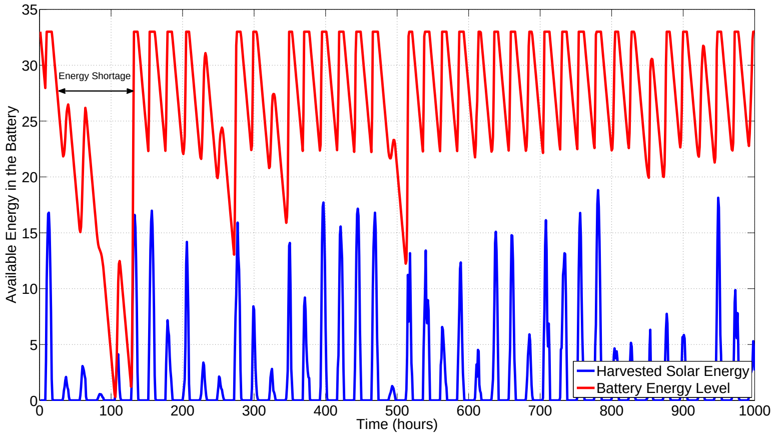


Figure 5.1: Battery Energy Level and Input Solar Energy vs. Time.

30-40 miles, the vehicle requires 7-10 kWh of energy. In other words, most plug-in vehicles need 0.2-0.3 kWh of charging power for a mile of driving. When the charging rate of a PHEV is 1.4 kW it takes about 8 hours to fully charge its batteries. During its charging time the PHEV consumes more than double the average household electricity consumption[1].

Now suppose that a system, such as the one mentioned above, is powered by solar energy (i.e., a single solar powered node). Therefore, a combination of solar panels and batteries are used to provision the node. We assume that the batteries and panels are characterized by the same parameters as the ones described in Chapter 3. Suppose that the node has to supply a 1000 W load in each hour over 1000 hours of deployment time. Solar insulation data for the city of Toronto, starting from January 1980, is used. Accordingly, energy provisioning of the node can be done based on the method that is explained in (Farbod and Todd, 2006). The available energy in the battery at each time interval as well as input solar energy is plotted and shown in Figure 5.1. It can be seen from this figure that the input

[1] Note that this information is for sedan types of PHEVs. The consumption power for other types of PHEVs with higher charging rates, when they are charged in fast charging mode, would be between three to about fifteen times than that of a typical household energy consumption rate (Ipakchi and Albuyeh, 2009).

solar energy to the node is decreased over several time intervals, and therefore the node has encountered an energy shortage in these intervals (the first one is shown in the figure). Since the node has to continuously supply the load, the battery energy level is dropped when the input solar energy is low. Therefore, in order for the node to operate outage free, the battery must have enough capacity to compensate for solar energy shortage, which in turn may lead to high provisioning costs.

Now suppose that part of the load is deferrable, i.e., the corresponding demands are delay tolerant and do not need to be satisfied immediately upon their request. It is apparent that deferrable demands in the intervals with energy shortage can be postponed until a time that the node receives sufficient solar energy for their fulfillment. Energy drops in the battery would be minimized by this process, and therefore, a smaller battery size would be required. As a result, the total provisioning cost would be decreased.

5.4 Energy Flow Model

In this section we briefly review the energy flow model used at an isolated solar powered node and where solar is the only power source. The node could be a house, for example, that has a significant delay tolerant energy demand from its appliances and other electrical equipment. Plug-in Hybrid Electric Vehicles (PHEV) are examples of this kind of equipment where their charging can be postponed for several hours. An example node is shown in Figure 5.2. The energy model is the same as the one explained in Section 3.4 except that in this chapter, we assume that the energy controller not only controls the battery charging state but also schedules the energy demands. We assume that the operation of the system is done over a contiguous time period that can be defined using a set of time intervals $\mathcal{K} = [1, 2, \ldots, k_{max}]$, where the duration of each is equal to one hour.

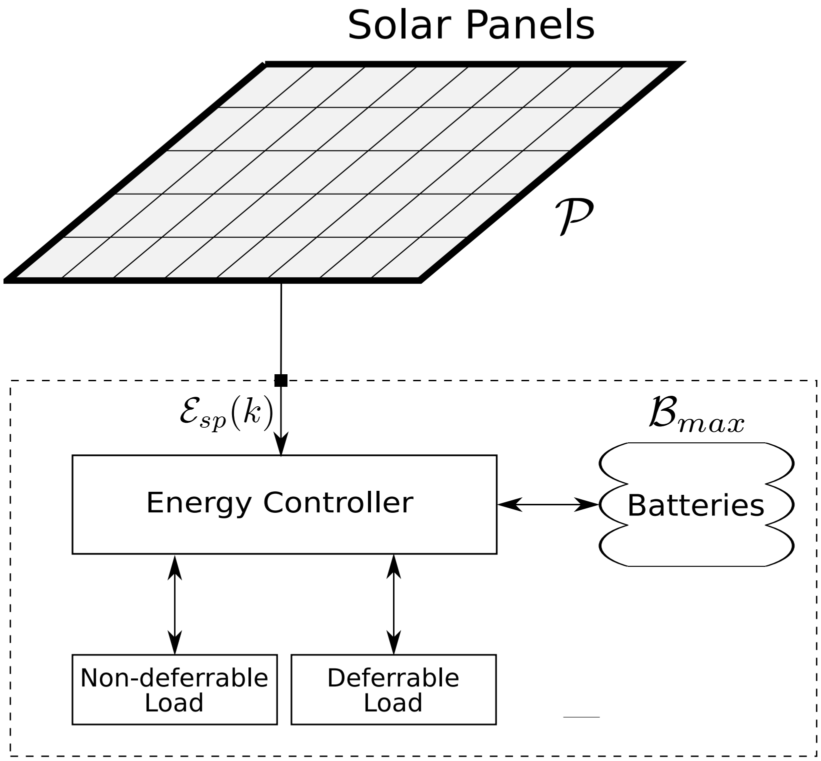


Figure 5.2: Solar Powered Infrastructure with Deferrable Loading.

In this chapter, we assume that the solar panel is optimally oriented and therefore Equation (3.2) can be written as follows

$$\mathcal{E}_{sp}(k) = \mathcal{P}\mathcal{E}^{*}(k) \tag{5.1}$$

where $\mathcal{P}$ is the area of the panel and $\mathcal{E}^{*}(k)$ is the per unit area energy availability at each time interval $k \in \mathcal{K}$ at the geographical location of the solar node.

Battery energy level $\mathcal{B}(k)$ at time interval k is given by Equation 3.1, and we represent it as follows

$$\mathcal{B}(k) = \text{min}\{\max[\mathcal{B}(k-1) + \mathcal{E}_{sp}(k) - \mathcal{L}(k), \mathcal{B}_{outage}], \mathcal{B}_{max}\} \tag{5.2}$$

where $\mathcal{L}(k)$ is the load energy demand over time interval $k \in \mathcal{K}$, and B_{max} and B_{outage} are the maximum battery capacity and the maximum allowed depth of discharge, respectively.

Using this recursion, in the next section we explain and formulate the problem of solar node provisioning when the load energy demand is a combination of deferrable and non-defferable loads.

5.5 Problem Formulation and Provisioning Bound

In this section we formulate the problem of solar node provisioning when all or part of the energy demands are delay tolerant. Our objective is to exploit the flexibility of supplying deferrable loads to minimize the total provisioning cost. We assume that we have a sequence of d_{max} demands over the designated time period $\mathcal{K} = [1, 2, \ldots, k_{max}]$. The set of demands is given by $\mathcal{D} = \{1, 2, 3, ..., d_{max}\}$ where $i \in \mathcal{D}$ is the ith demand in the sequence of demands. The demand type could be either deferrable or non-deferrable which is declared to the energy controller unit upon its request. If demand $i \in \mathcal{D}$ is deferrable, its maximum tolerable delay is also sent to the controller. We assume that each demand can be supplied in one time interval, and define the binary variable $b(i, k)$ to indicate the supplying of demand $i \in \mathcal{D}$ at time interval k as follows.

$$b(i,k) = \begin{cases} 1 & \text{if demand } i \text{ is supplied at time interval } k \\ 0 & \text{otherwise,} \end{cases} \tag{5.3}$$

for all $i \in \mathcal{D}$ and $k \in \mathcal{K}$.

We define $a(i) \in \mathcal{K}$ to be the arrival time of demand i. Hence the delay in satisfying demand i can be obtained from the following equation

$$\tau(i) = \sum_{k=1}^{k_{max}} k.b(i,k) - a(i) \quad \forall i \in \mathcal{D} \tag{5.4}$$

where $\tau(i) \geq 0$.

Assuming that $\tau^*(i)$ is the maximum delay that demand $i \in \mathcal{D}$ can tolerate, the following constraint should be met.

$$\tau(i) \leq \tau^*(i) \quad \forall i \in \mathcal{D} \tag{5.5}$$

Note that since each demand i has to be supplied only once, we have the following constraint.

$$\sum_{k=1}^{k_{max}} b(i,k) = 1 \quad \forall i \in \mathcal{D} \tag{5.6}$$

Let $E(i)$ be the required energy to supply the deferrable demand $i \in \mathcal{D}$, and let $\mathcal{L}_{nd}(k)$ be the non-deferrable load at time interval k which the node has to supply immediately. Therefore, the total load that the node supplies at each time interval k can be defined as follows.

$$\mathcal{L}(k) = \sum_{i=1}^{d_{max}} b(i,k).E(i) + \mathcal{L}_{nd}(k) \quad \forall k \in \mathcal{K} \tag{5.7}$$

Now we can develop an optimization which derives a lower bound on the provisioning cost where the demand is a combination of non-deferrable and deferrable demands. The LP finds the optimal battery and solar panel sizing and determines how deferrable loads have to be scheduled such that the minimum provisioning cost is achieved. The inputs to the problem are given by the set of n-tuples

$$\mathcal{I} = \{\gamma_P, \gamma_B, a(i), \tau^*(i), E(i), \mathcal{E}^*(k), \mathcal{B}_{min}, \mathcal{L}_{nd}(k), \mathcal{L}^*, k_{max}, d_{max}\} \quad \forall i \in \mathcal{D}, \quad \forall k \in \mathcal{K} \tag{5.8}$$

and the problem variables are defined by the following n-tuples.

$$\mathcal{V} = \{\mathcal{B}_{max}, \mathcal{B}(k), \mathcal{P}, \mathcal{L}(k), \mathcal{E}_{sp}(k), b(i,k), \tau(i)\} \quad \forall i \in \mathcal{D}, \quad \forall k \in \mathcal{K} \tag{5.9}$$

The optimization problem, referred to as Linear Program for Delay Aware Provisioning (LP-DAP), is given below.

$$\underset{\mathcal{V}}{\text{minimize}} \quad \gamma_B.\mathcal{B}_{max} + \gamma_P.\mathcal{P} \tag{LP–DAP}$$

$$\text{Subject to:} \quad \mathcal{B}(0) = \mathcal{B}_{max} \tag{5.10}$$

$$\mathcal{B}_{outage} \leq \mathcal{B}(k) \qquad \forall k \in \mathcal{K} \tag{5.11}$$

$$\mathcal{B}(k) \leq \mathcal{B}_{max} \qquad \forall k \in \mathcal{K} \tag{5.12}$$

$$\mathcal{B}(k) \leq \mathcal{B}(k-1) + \mathcal{E}_{sp}(k) - \mathcal{L}(k) \qquad \forall k \in \mathcal{K} \tag{5.13}$$

$$\mathcal{E}_{sp}(k) = \mathcal{P}\mathcal{E}^*(k) \qquad \forall k \in \mathcal{K} \tag{5.14}$$

$$b(i,k) \in \{0,1\} \qquad \forall i \in \mathcal{D},\ \forall k \in \mathcal{K} \tag{5.15}$$

$$\sum_{k=1}^{k_{max}} b(i,k) = 1 \qquad \forall i \in \mathcal{D} \tag{5.16}$$

$$\tau(i) = \sum_{k=1}^{k_{max}} k.b(i,k) - a(i) \qquad \forall i \in \mathcal{D} \tag{5.17}$$

$$\tau(i) \leq \tau^*(i) \qquad \forall i \in \mathcal{D} \tag{5.18}$$

$$\mathcal{L}(k) = \sum_{i=1}^{d_{max}} b(i,k).E(i) + \mathcal{L}_{nd}(k) \qquad \forall k \in \mathcal{K} \tag{5.19}$$

$$\mathcal{L}(k).\Delta \leq \mathcal{L}^* \qquad \forall k \in \mathcal{K} \tag{5.20}$$

$$0 \leq \mathcal{L}(k),\ 0 \leq \mathcal{B}(k),\ 0 \leq \mathcal{P},\ 0 \leq \tau(i) \qquad \forall i \in \mathcal{D},\ \forall k \in \mathcal{K} \tag{5.21}$$

LP–DAP minimizes the total provisioning cost for a solar powered node that has deferrable load subject to constraints (5.10) to (5.21). Parameters γ_B and γ_P in the objective function are battery and panel unit prices, respectively. We assume that the battery is fully charged at the beginning of its operation, and Equation (5.10) shows this condition. Constraint (5.11) indicates that the node must maintain a minimum battery level in order to prevent node outage, and Inequality (5.12) indicates that the battery energy cannot exceed the battery capacity. Therefore, these two inequalities put upper and lower limits on the battery energy level. Constraint (5.13) states that the stored battery energy at time k must not exceed that which was available at time $k-1$, plus the energy which was obtained from the panel, minus the energy consumption during this time interval. Note that the recursion (5.2) is reformulated by constraints (5.12) to (5.11) using Theorem 1 in Chapter 3. Constraints (5.14) to (5.19) have already been defined and explained. We assume that there is a limitation on the amount of load that the node can support. Inequality (5.20) shows this constraint by using $\mathcal{L}^*$ as the maximum load that the node supports at each time interval k. Constraint (5.21) ensures non-negative values for the variables that are used in LP–DAP.

The lower bound on the provisioning cost is obtained using LP–DAP. However, since this LP does not provide an online algorithm, we need to introduce one. This algorithm is proposed in the next section.

5.6 Delay Aware Provisioning (DAP) Algorithm

In this section a provisioning algorithm for a stand-alone PV system when the energy demand consists of a combination of deferrable and non-deferrable demands is introduced. The proposed Delay Aware Provisioning (DAP) Algorithm is shown in Algorithm 5.1. In the first step, given the input data set $\mathcal{I}$ in (5.8), the algorithm uses the Stand-alone Node Provisioning (SNP) algorithm introduced in (Farbod and Todd, 2006) to determine the optimum values of battery and panel sizes. Note that although the demand set consists of deferrable and non-deferrable demands, in the first step we do provisioning as if all the demands are non-deferrable.

In the next steps, the DAP algorithm starts with the optimum values of battery and panel, $\mathcal{B}_{max}^{ini}$ and $\mathcal{P}^{ini}$, achieved from Step 1. It fixes the panel size to $\mathcal{P}^{ini}$ and tries to find a smaller battery size by using the flexibility of deferrable demands. This is done by putting a threshold T_r on the available energy in the battery. We measure the difference between available energy in the battery $\mathcal{B}(k)$ and the threshold. We have called this energy $\mathcal{B}_{DAP}(k)$, which is our energy budget, and is used to test if this amount of energy is sufficient to satisfy demands over the current time interval. Initially the threshold value is set to $\mathcal{B}_{max}^{ini}$ in Step 2 which means that $\mathcal{B}_{DAP} = 0$ at the beginning of the operation of the algorithm.

Since the demand set includes non-deferrable demands that should be satisfied immediately, the satisfaction priority is given to this kind of demand. Note that the deferrable demand $i \in \mathcal{D}$ can be treated as non-deferrable if it has not been satisfied for $\tau^*(i)$ time intervals, where $\tau^*(i)$ is the maximum delay that demand i can tolerate.

To find the best battery size, the DAP algorithm does two kinds of iterations. The first set of iterations is over the threshold value, and the second, which is nested in the first iteration set, is the iteration over all time intervals $k \in \mathcal{K}$. In Step 3 the DAP algorithm

Algorithm 5.1. Delay Aware Provisioning (DAP) Algorithm

```
Given: Input data set $\mathcal{I}$
 1: Use SNP from Reference (Farbod and Todd, 2006) to determine $\mathcal{P}^{ini}$ and $\mathcal{B}^{ini}_{max}$ using
    input data set $\mathcal{I}$
 2: Set $Tr \leftarrow \mathcal{B}^{ini}_{max}$
 3: while there is any unsatisfied demand in the buffer do
 4:     Set $Tr \leftarrow Tr - \delta_{Tr}$
 5:     for all $1 \leq k \leq k_{max}$ do
 6:         Update battery energy level $\mathcal{B}^{ini}(k)$ using Equation 5.2.
 7:         Set $\mathcal{B}_{DAP}(k) \leftarrow \mathcal{B}^{ini}(k) - Tr$
 8:         while there is any non-deferrable demand do
 9:             if $\mathcal{B}_{DAP}(k)$ is sufficient to satisfy the demand then
10:                 Satisfy the demand and update $\mathcal{B}_{DAP}(k)$.
11:             else
12:                 Go to Step 4
13:             end if
14:         end while
15:         while there is any deferrable demand do
16:             if $\mathcal{B}_{DAP}(k)$ is sufficient to satisfy the demand then
17:                 Satisfy the demand and update $\mathcal{B}_{DAP}(k)$.
18:             else
19:                 Put the demand in the buffer
20:             end if
21:         end while
22:     end for
23: end while
24: Return $\mathcal{B}_{DAP}$
```

iterates over all possible thresholds until the node can satisfy all its energy demands. In each iteration, we decrease the threshold value Tr by a small amount δ_{Tr}. It turns out that $\mathcal{B}(k)_{DAP}$ is increased by δ_{Tr} to start a new iteration with a higher value of $\mathcal{B}_{DAP}(k)$. This step is shown in Step 4. In Step 5, the iteration over time intervals starts. At each time interval, the battery energy level $\mathcal{B}^{ini}(k)$ is updated in Step 6 using input solar energy. Consequently, in Step 7, $\mathcal{B}_{DAP}(k)$ is updated. Considering $\mathcal{B}_{DAP}(k)$ and available energy demands, the DAP algorithm attempts to satisfy the demands from Steps 8 to 23. First, in Steps 8 to 14, the DAP algorithm checks the availability of non-defferable demands. If $\mathcal{B}_{DAP}(k)$ is sufficient to satisfy these demands, then it supplies them. Otherwise, $\mathcal{B}_{DAP}(k)$ has to be increased and the process starts from Step 4 again. If enough energy is remaining in $\mathcal{B}_{DAP}(k)$, deferrable demands can be supplied in Steps 15 to 21. If not, their service will be delayed to another time interval. The algorithm keeps performing these iterations until there is no unsatisfied demand left in the buffer[1]. It will finally converge to a $\mathcal{B}_{DAP} \leq \mathcal{B}_{max}^{ini}$ such that it can supply all the non-defferable demands immediately, and all the deferrable demands within their maximum tolerable delays.

Note that the algorithm can easily be modified to account for violating the maximum permissible delay of a small portion ϵ of deferrable demands. In other word, we can consider the following constraint

$$Pr\big(\tau(i) \geq \tau^{*}(i)\big) \leq \epsilon, \quad \forall i \in \mathcal{D} \tag{5.22}$$

which indicates that for each demand $i \in \mathcal{D}$, the probability of having a delay greater than $\tau^{*}(i)$ is less than ϵ. This can be done in the DAP algorithm by recording the delays and adding another loop to iterate over the delays and to satisfy constraint (5.22).

5.7 Simulation Results

In this section, simulation results are presented. Provisioning costs for a single solar powered node for the DAP algorithm are compared to the SNP algorithm and the provisioning bound. In addition, the performance of DAP is evaluated in terms of maximum tolerable delay and delay distribution. Finally, the proposed delay aware provisioning algorithm is

[1] We assume that the node is equipped with a buffer in which the deferrable demands can be placed, to be served in a first-come, first-served order.

Parameter	Value
Load Power $E(i)\ \forall i \in \mathcal{D}$	1000 W
$\mathcal{L}^*$	1000 Wh
$\mathcal{B}_{outage}$	0
γ_B	3.4
γ_P	6.7
Location	Toronto, Canada (6 weeks, 01/01/80)

Table 5.1: Default Parameters for the Examples

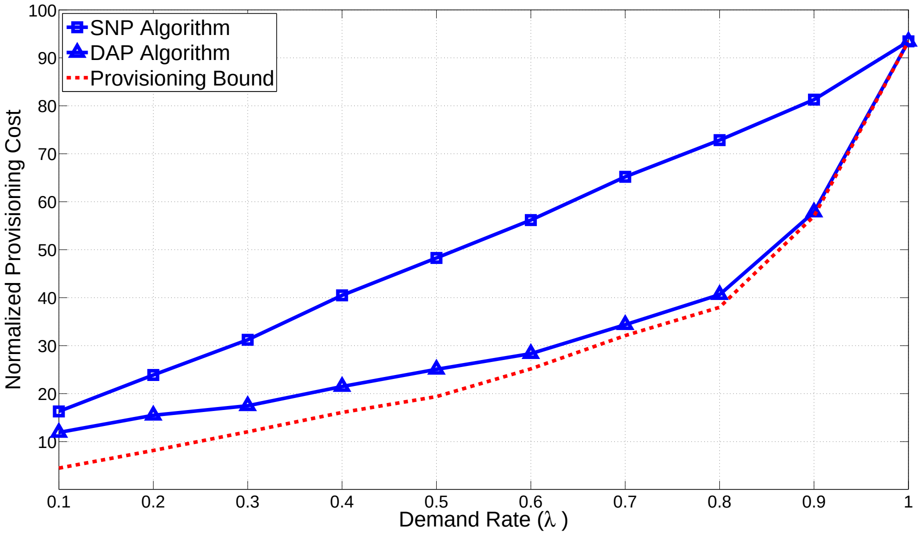


Figure 5.3: Comparison of Normalized Provisioning Cost for DAP and SNP.

investigated for different geographic locations.

Simulation parameters are shown in Table 5.1, where we have assumed that all energy demands are the same. Experiments are done using a profile of solar insolation traces for Toronto, Canada. We assume that the demand arrival rate is a Poisson process with an average of λ demands per time interval, with $0 \leq \lambda \leq 1$. The solar node deployment time is set to be 1100 hours ($\approx$ one and a half months) starting from the beginning of January[1].

[1] As mentioned in Chapter 3, node provisionings that are outage-free spanning worse-case winter months result in year-round outage-free operation.

5.7.1 Provisioning Cost Comparison

In the first experiment, we have assumed that all the demands are delay tolerant, and we have no constraint on the maximum delay i.e., $\tau^*(i) = \infty, \forall i \in \mathcal{D}$. Both SNP and the proposed DAP algorithms are used to do provisioning for 100 sample runs. In each round of simulations the provisioning cost is recorded and the average value of the results are plotted in Figure 5.3. The graph also contains the provisioning bound which is obtained from the mixed integer linear optimization LP–DAP formulated and explained in Section 5.5. It can be seen from Figure 5.3 that comparing with the SNP provisioning algorithm, DAP reduces the provisioning cost by about 50% over a wide range of demand rates. When the arrival rate increases to the maximum values shown, there is no room to support shifted deferrable demands. As a result, the performance of the DAP algorithm is limited. On the other hand, when the demand arrival rate is light, there is more room for improvement compared to when the solar node is heavily loaded.

5.7.2 DAP Average Delay

The average satisfaction delay for the aforementioned experiment when the DAP algorithm is used is shown in Figure 5.4. Comparing this figure with Figure 5.3, it is evident that the higher the improvement in provisioning cost, the more satisfaction delay that the demands have to tolerate. In this case, since there is no restriction on the maximum delay, DAP may schedule some demand satisfaction times to time intervals that may be 20 to 30 hours from the arrival time of the demand. Note that although the maximum delay is set to infinity, from Figure 5.4 it can be seen that, on average, the maximum delay is about 28 hours for a demand rate of $\lambda = 0.6$.

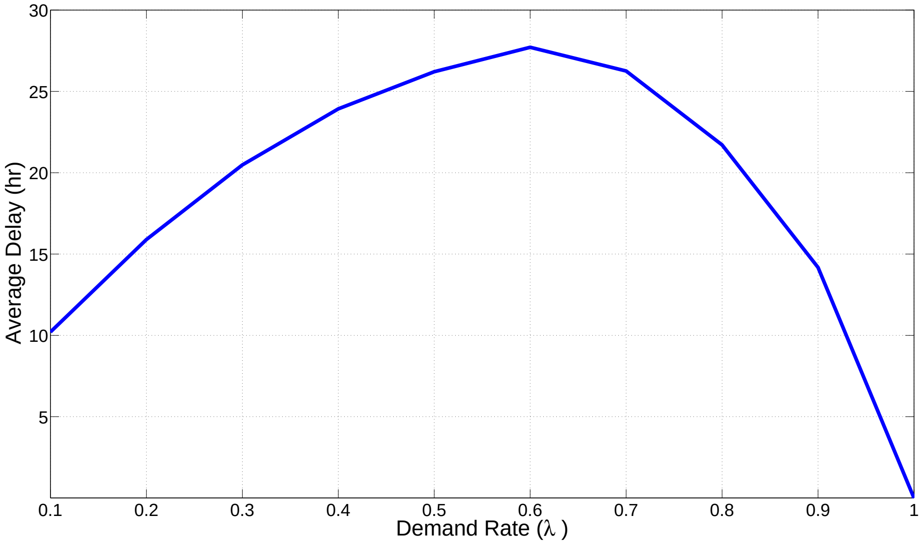


Figure 5.4: Average Demand Satisfaction Delay.

5.7.3 DAP Delay Distribution

Under the same conditions, the DAP provisioning algorithm is used for 500 different random inputs and the delay distribution for $\lambda = 0.2, 0.4, 0.6$ and 0.8 is shown in Figure 5.5. It can be seen that when the load on the solar node is light, the algorithm has more flexibility to do demand scheduling than the case when the node is highly loaded. In fact, when the demand rate is low, there are more time intervals in the near future that the algorithm can use to satisfy demands. This results in a lower delay when we compare it with a highly loaded node.

5.7.4 The Effect of ϵ and the Maximum Delay

In this experiment, we put a limitation on the maximum tolerable delay, i.e., $\tau^*(i) < \infty, \forall i \in \mathcal{D}$. In addition, we assume that all the demands have the same value for their

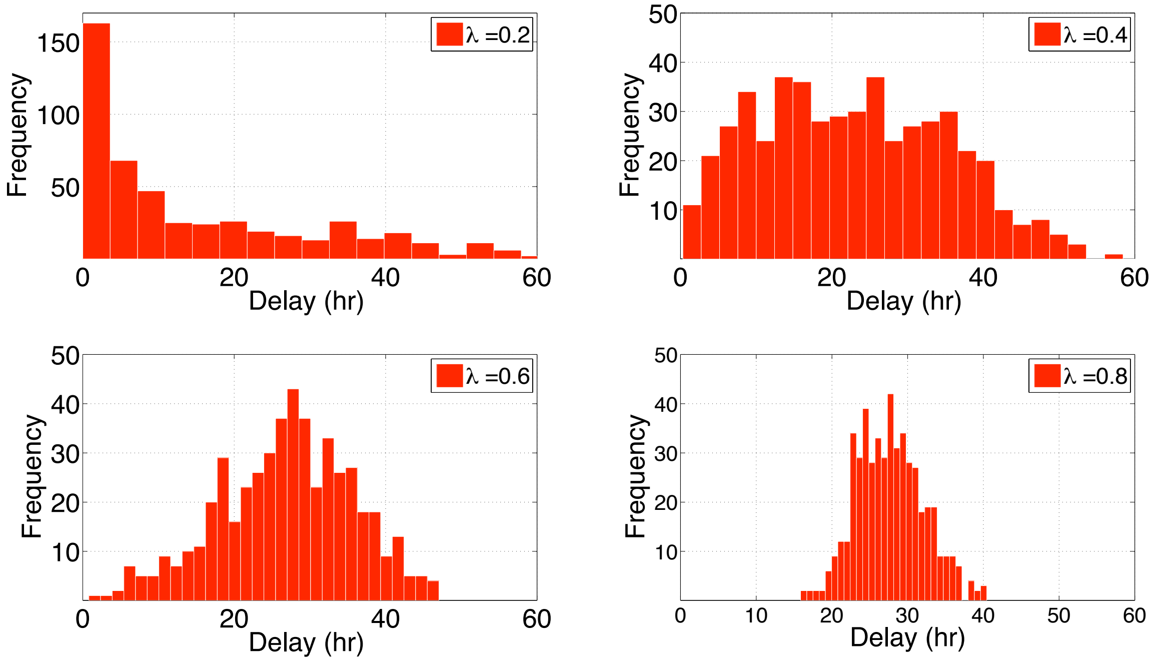


Figure 5.5: DAP Delay Distribution for $\lambda = 0.2$, 0.4, 0.6 and 0.8.

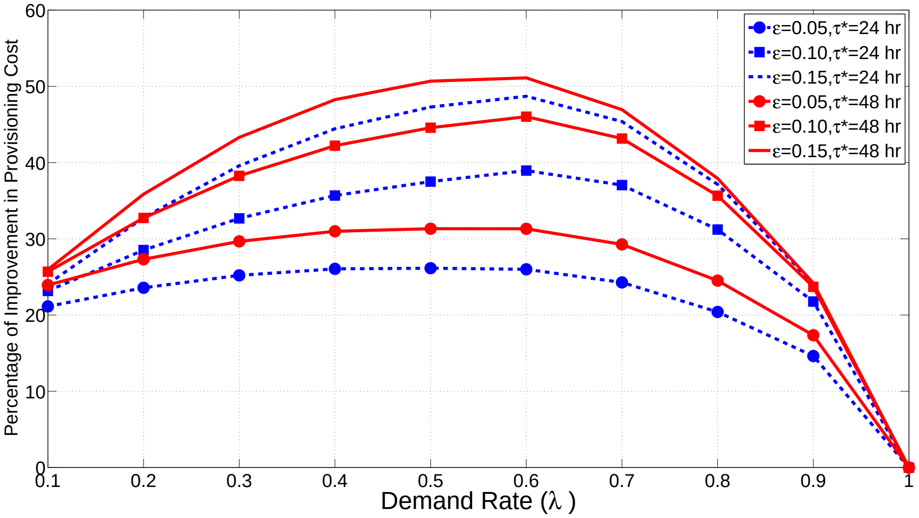


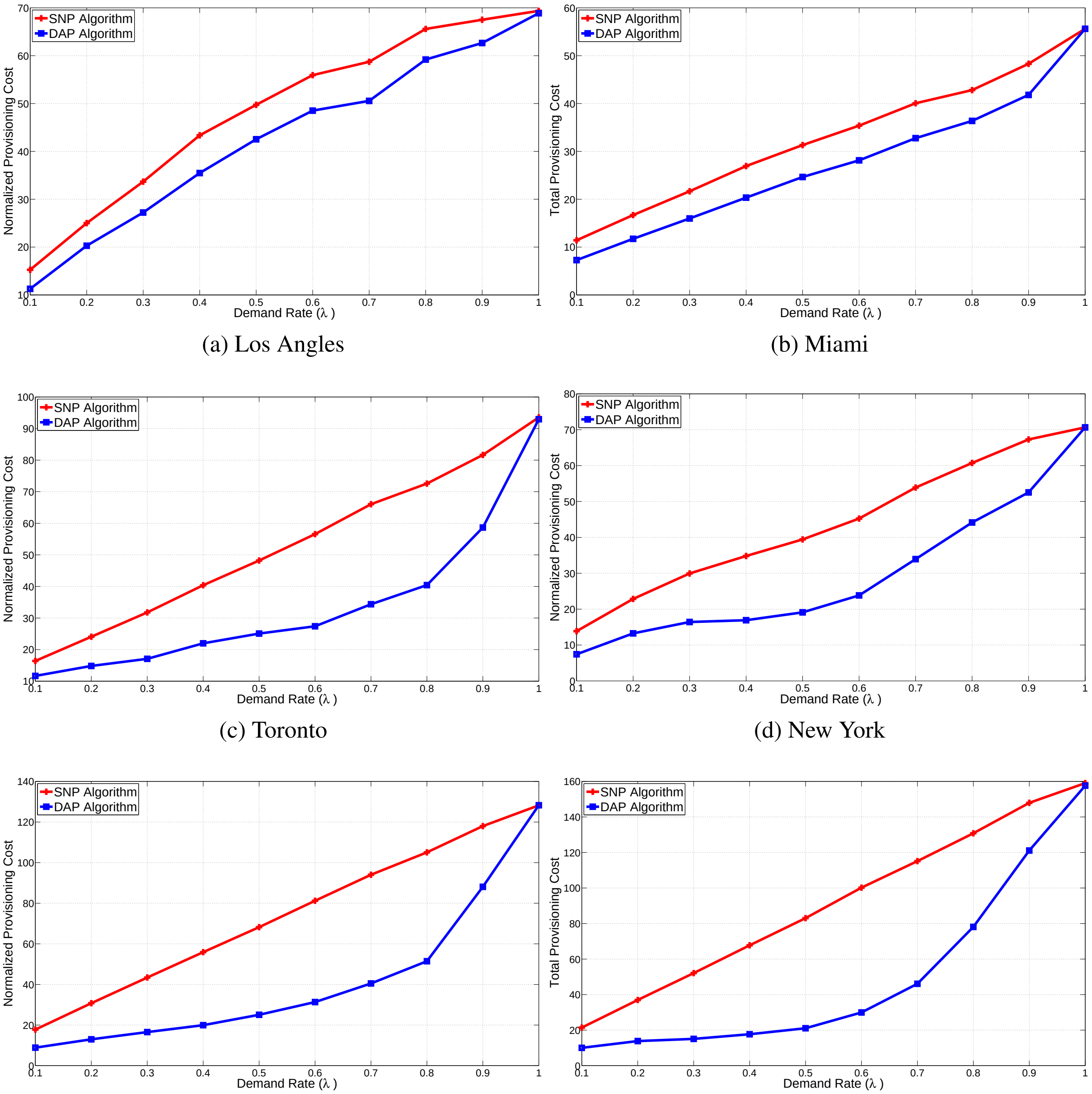
Figure 5.6: Average Percentage Improvement in Total Provisioning Cost vs. Demand Rate.

maximum delay, i.e., $\tau^*(i) = \tau^*, \forall i \in \mathcal{D}$. We also add Constraint (5.22) to the DAP algorithm such that the probability of violating the maximum delay is limited to ϵ, where ϵ is set to 5%, 10% and 15%. The performance of the DAP algorithm in comparison with the SNP algorithm is evaluated for each value of ϵ and two maximum delays, 24 and 48 hours. Both the DAP and SNP algorithms are used for solar node provisioning.

The average percentage improvement of the DAP algorithm in the total provisioning cost for 300 scenarios is achieved and the results are shown in Figure 5.6. It can be seen that in the most limited case where ϵ is 5% and τ^*=24, about 25% improvement in the total provisioning cost over a wide range of demand rates can be achieved using the DAP algorithm. When the permissible ratio ϵ is increased, a better result is obtained, e.g., when ϵ is 15% and τ^* is 24 hours, an improvement of up to 48% is observed. If the maximum permissible delay τ^* is increased to 48 hours, an even better improvement using the same ϵ can be obtained. Saturation in the improvement is observed when ϵ or τ^* is increased, at which point any further increment in ϵ or τ^* will not have any effect, resulting in about a 50% improvement.

5.7.5 Performance of DAP for Different Locations

In the next set of experiments, the performance of the DAP algorithm is evaluated for different geographical regions with varying climates. Six cities, consisting of Toronto, Yellow Knife, New York, Los Angles, Miami and Seattle are used. The deployment time is set to 45 days and the solar insolation data corresponding to each city from the beginning of January 1980 is used. As before, the input load is a Poisson arrival process with parameter λ. Both the SNP and DAP provisioning algorithms are used, and it is assumed that there is no restriction on the maximum delay. The provisioning cost versus demand rate for the



(a) Los Angles

(b) Miami

(c) Toronto

(d) New York

(e) Seattle

(f) Yellow Knife

Figure 5.7: Normalized Provisioning Cost for DAP and SNP Algorithms vs. Demand Rate for Different Locations.

DAP and SNP algorithms is shown in Figure 5.7.

It can be seen that, while the improvement in the provisioning cost is 15% and 20% for Los Angles and Miami, respectively, the observed improvement is much higher for the other cities, which are mainly located in more temperate regions. Simulation results show that the improvement in the provisioning cost for the cities of Toronto and New York, which have almost the same climatic conditions, is up to 51%. The DAP algorithm results in 63% and 74% improvements for Seattle and Yellow Knife, respectively. The reason that the improvement in the non-tropical regions is higher than the tropical regions is that, while the required optimal panel for an outage-free operation of a node in a tropical region is a major part of the total provisioning cost, for the non-tropical regions it is the battery that has a higher contribution to the provisioning cost. Therefore, when the DAP algorithm is used, which fixes the panel and decreases the battery size, its effect for provisioning in non-tropical regions is higher than in the tropical regions in terms of the total provisioning cost.

5.8 Conclusions

In this chapter, the problem of energy provisioning was considered for stand-alone PV systems where part of their energy demand is deferrable. The objective is to exploit the flexibility of deferrable energy demands to decrease the total provisioning cost. The problem was formulated as a mixed integer linear optimization which gives a lower bound on the provisioning cost. Since the solution of the optimization problem is non-casual, a casual algorithm which is called Delay Aware Provisioning (DAP) was proposed. The performance of DAP was compared to the provisioning bound and the conventional Stand-alone Node Provisioning (SNP) algorithm through several simulation experiments. The results show that the maximum tolerable delay (τ^*) has a significant effect on the total provisioning cost. For the city of Toronto, for example, when the DAP algorithm with $\tau^* = 24$ hours is used, about 25% improvement in the total provisioning cost was achieved. Under the same conditions, this value for $\tau^* = 48$ was 31%. In addition, simulation results show that the improvement in the total provisioning cost depends on the geographic location of the solar node, e.g., while the percentage of cost reduction for Miami is about 20%, this value for Yellowknife is 74%.

Chapter 6

Conclusions and Future Work

In solar powered node installations, the process of sizing the batteries and panels is referred to as *energy provisioning*. The focus of this thesis was on energy provisioning in nodes that are either networked with other solar powered nodes, e.g., wireless solar mesh networks, operate in conjunction with the power grid, or are used independently.

In Chapter 2, a brief introduction to photovoltaic systems was presented, including a discussion of topics relating to energy provisioning in solar powered nodes.

In Chapter 3, we focused on the energy provisioning problem in wireless solar powered mesh networks. To decrease the total provisioning costs, the positioning of the solar nodes was considered and a Position Aware Provisioning (PAP) algorithm was proposed. A linear optimization was developed which gives a lower bound for the PAP algorithm. The Shortest Path Provisioning (SPP), Link Aware Provisioning (LAP) and Iterative Energy Aware Provisioning (IEAP) algorithms were introduced, and their performance were compared with the PAP algorithm. Simulation results indicated a significant improvement in the total provisioning cost when the proposed algorithm is used.

In Chapter 4, we considered a single wireless communication node that is operated from a combination of power grid and solar components, where the solar part is a node add-on. The CAPital EXpenditure (CAPEX) costs and OPErating EXpenditure (OPEX) costs of the node were introduced. Different node add-on configurations were considered including, solar panel/battery/grid (PBG), solar panel/grid (PG), and battery/grid (BG) scenarios. A linear optimization was derived whose objective is to minimize the total CAPEX and OPEX costs. Three energy scheduling algorithms were then introduced to optimize online OPEX costs, namely, Grid Purchase Last (GPL), Solar Load Optimization (SLO) and Solar Load Simulation (SLS) algorithms. The case where revenue can be derived from returning unused energy to the power grid was also considered. Extensive simulation experiments indicate the value of the proposed algorithms in reducing the total CAPEX and OPEX costs.

In Chapter 5, solar powered systems that are independent of the power grid and have deferrable energy demands were considered. The objective was to reduce the total provisioning costs. A Mixed Integer Linear Program (MILP) was developed which gives a lower bound on provisioning costs. The Delay Aware Provisioning (DAP) algorithm was then introduced that uses results from the conventional Stand-alone Node Provisioning (SNP) algorithm and decreases the required battery size by scheduling the deferrable demands. Simulation results showed that when the energy demands can tolerate several hours of delay in their service time, good savings in the provisioning cost is achievable.

The work in this thesis can be extended in the future, based on the following ideas.

1. The proposed algorithms in Chapters 4 and 5 were for a single node that can be extended to a network of similar nodes.

2. In Chapter 4, it was assumed that the grid energy price is known beforehand. Instead of a predetermined value of energy price, the solar node can negotiate with the power

grid to minimize its operating cost. Game theory methods can be used in this case.

3. In Chapter 4, the optimization problem is developed under the assumption that the energy price is known for the entire deployment time. As a more realistic assumption, a random variable can be added to the energy price to model pricing uncertainty. The problem can then be solved by using risk analysis methods which should result in more reliable provisioning and scheduling algorithms.

Appendix A

Solar Irradiation and Loading Prediction Algorithm

To schedule grid energy purchases, the SLO and SLS Algorithms defined in Section 4.7 use simple estimates of future solar insolation values and loading for the next w time intervals. The prediction algorithm that is used is based on that first introduced in Ali *et al.* (2010). The prediction uses a linear combination of its current (i.e., known) value and its average over the last D days. Assuming that $p(k+1)$, is the parameter to be predicted, then

$$\widetilde{p}(k+1) = \pi\, p(k) + (1-\pi)\, \mu_D(k+1)\, \Phi_H \tag{A.1}$$

where $\widetilde{p}(k+1)$ is the predicted value and $p(k)$ is the actual parameter for the current time period, k. In Equation (A.1), $\pi \in [0,1]$ is a weighting parameter and $\mu_D(j)$ is the parameter average at time interval j over the last D days, i.e.,

$$\mu_D(j) = \frac{\sum_{i=1}^{D} p(j - i\,M)}{D} \tag{A.2}$$

where M is defined to be the number of Δt time periods in 1 day, i.e., $M\,\Delta t = 24$. Φ_H is defined as a conditioning factor that depends on the last H time intervals and the history of the parameter over the last D days. It is determined by weighted values of $h(j)$ for $j \in \{1, 2, \ldots, H\}$, defined as the parameter values in last H time intervals and normalized by the corresponding μ_D, i.e.,

$$h(j) = \frac{p(k - H + j)}{\mu_D(k - H + j)} \quad \forall j \in \{1, 2, \ldots, H\} \tag{A.3}$$

Since time intervals that are closer to $k + 1$ are more correlated with the current time, the weighting factors of the $h(j)$'s decrease from 1 to $1/H$ starting at k, and can be defined as follows.

$$\theta(j) = \frac{j}{H} \quad \forall j \in \{1, 2, \ldots, H\} \tag{A.4}$$

The conditioning factor, Φ_H, is then determined as follows.

$$\Phi_H = \frac{\sum_{j=1}^{H} h(j)\theta(j)}{\sum_{j=1}^{H} \theta(j)} \tag{A.5}$$

Note that the values of D, H and π can be found such that the estimation error is minimized. Having the predicted value $\widetilde{p}(k+1)$, the same algorithm can be used to find predictions for time interval $k + 2$, and can be continued until time interval $k + w$.

Bibliography

Ab-Hamid, K., Tan, C. E., and Lau, S. P. (2011). Self-Sustainable Energy Efficient Long Range WiFi Network for Rural Communities. In *IEEEGLOBECOM Workshops (GC Wkshps)*, pages 1050–1055.

Ali, M., Al-Hashimi, B., Recas, J., and Atienza, D. (2010). Evaluation and Design Exploration of Solar Harvested-Energy Prediction Algorithm. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 142–147.

Alizadeh, M., Scaglione, A., and Thomas, R. J. (2012). From packet to power switching: Digital direct load scheduling. *IEEE Journal on Selected Areas in Communications*, **30**(6), 1027–1036.

Arnold, O., Richter, F., Fettweis, G., and Blume, O. (2010). Power Consumption Modeling of Different Base Station Types in Heterogeneous Cellular Networks. In *Future Network and Mobile Summit*.

Badawy, G. H., Sayegh, A. A., and Todd, T. D. (2010). Energy Provisioning in Solar-Powered Wireless Mesh Networks. *IEEE Transactions on Vehicular Technology*, **59**(8).

Balouktsis, A., Karapantsios, T. D., Antoniadis, A., Paschaloudis, D., Bezergiannidou, A., and Bilalis, N. (2006). Sizing Stand-Alone Photovoltaic Systems. *International Journal of Photoenergy*, pages 1–8.

Borowy, B. and Salameh, Z. (1996). Methodology for the Optimally Sizing the Combination of a Battery Bank and PV Array in a Wind/PV Hybrid System. *IEEE Transaction on Energy Conversion*, **11**(2), 367–375.

Bu, S., Yu, F., Cai, Y., and Liu, X. (2012). When the Smart Grid Meets Energy-Efficient Communications: Green Wireless Cellular Networks Powered by the Smart Grid. *IEEE Transactions on Wireless Communications*, **11**(8), 3014–3024.

Camacho, E. F., Berenguel, M., Rubio, F. R., and Martínez, D. (2012). *Control of Solar Energy Systems*. Springer.

Chedid, R. and Rahman, S. (1997). Unit Sizing and Control of Wind-solar Power Systems. *IEEE Transactions on Energy Conversion*, **12**(1), 79–85.

Chowdhury, S. and Aziz, S. (2012). Solar-Diesel Hybrid Energy Model for Base Transceiver Station (BTS) of Mobile Phone Operators. In *2nd International Conference on the Developments in Renewable Energy Technology (ICDRET)*.

Conejo, A., Morales, J., and Baringo, L. (2010). Real-Time Demand Response Model. *IEEE Transactions on Smart Grid*, **1**(3), 236–242.

Correia, L., Zeller, D., Blume, O., Ferling, D., Jading, Y., Godor, I., Auer, G., and Van Der Perre, L. (2010). Challenges and Enabling Technologies for Energy Aware Mobile Radio Networks. *IEEE Communications Magazine*, **48**(11), 66–72.

Dimroth, F., Baur, C., Bett, A., Meusel, M., and Strobl, G. (2005). 3-6 Junction Photovoltaic Cells for Space and Terrestrial Concentrator Applications. In *31st IEEE Photovoltaic Specialists Conference*, pages 525–529.

ELBA and Class S Projects (2007). http://www.pt-it.pt-dlr.de/de/1760.php.

Farbod, A. and Todd, T. (2006). Resource Allocation and Outage Control for Solar-Powered WLAN Mesh Networks. *IEEE Transactions on Mobile Computing*, **6**(8), 960–970.

Frenger, P., Moberg, P., Malmodin, J., Jading, Y., and Gódor, I. (2011). Reducing Energy Consumption in LTE with Cell DTX. In *73rd IEEE Vehicular Technology Conference (VTC Spring)*.

Gan, L., Wierman, A., Topcu, U., Chen, N., and Low, S. H. (2013). Real-Time Deferrable Load Control: Handling the Uncertainties of Renewable Generation. In *fourth International Conference on Future Energy Systems (ACM e-Energy), Berkeley, CA*.

Goetzberger, A. and Hoffmann, V. U. (2005). *Photovoltaic Solar Energy Generation*, volume 112. Springer.

Green Radio (2007). http://www.mobilevce.com/green-radio.

Green Touch (2010). http://www.greentouch.org.

Green WiFi (2011). http://www.green-wifi.org/.

Gruber, M., Blume, O., Ferling, D., Zeller, D., Imran, M., and Strinati, E. (2009). EARTH: Energy Aware Radio and Network Technologies. In *20th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*.

GSM Association (1995). http://www.gsma.com/.

Han, C., Harrold, T., Armour, S., Krikidis, I., Videv, S., Grant, P., Haas, H., Thompson, J., Ku, I., Wang, C., *et al.* (2011). Green Radio: Radio Techniques to Enable Energy-Efficient Wireless Networks. *IEEE Communications Magazine*, **49**(6), 46–54.

Han, T. and Ansari, N. (2012). ICE: Intelligent Cell BrEathing to Optimize the Utilization of Green Energy. *IEEE Communications Letters*, **16**(6), 866–869.

Hansen, A. D., Sørensen, P. E., Hansen, L. H., and Bindner, H. W. (2001). *Models for a Stand-alone PV System*. Riso National Laboratory , Roskilde, Denmark.

Hashmi, M., Hanninen, S., and Maki, K. (2011). Survey of Smart Grid Concepts, Architectures, and Technological Demonstrations Worldwide. In *IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)*.

Ipakchi, A. and Albuyeh, F. (2009). Grid of the future. *IEEE Power and Energy Magazine*, **7**(2), 52–62.

Joe-Wong, C., Sen, S., Ha, S., and Chiang, M. (2012). Optimized Day-Ahead Pricing for Smart Grids with Device-Specific Scheduling Flexibility. *IEEE Journal on Selected Areas in Communications*, **30**(6), 1075–1085.

Kalogirou, S. A. (2009). *Solar Energy Engineering: Processes and Systems*. Academic Press.

Khatib, T., A.Mohamed, K.Sopian, and Mahmoud, M. (2012). A New Approach for Optimal Sizing of Standalone Photovoltaic Systems. *International Journal of Photoenergy*, **2012**.

Lin, L., Shroff, N., and Srikant, R. (2005). Asymptotically Optimal Power-Aware Routing for Multihop Wireless Networks with Renewable Energy Sources. *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, **2**, 1262–1272.

Lin, L., Shroff, N., and Srikant, R. (2007). Asymptotically Optimal Power-Aware Routing for Multihop Wireless Networks with Renewable Energy Sources. *IEEE/ACM Transactions on Networking*, **15**, 1021–1034.

Lopez, R. D. and Agustin, J. L. B. (2005). Design And Control Strategies Of PV-Diesel Systems Using Genetic Algorithms. *Solar Energy*, pages 33–46.

Lorenzo, E. and Navarte, L. (2000). On the Usefulness of Stand-alone PV Sizing Methods. *Progress in Photovoltaics: Research and Applications*, **8**(4), 391–409.

Louhi, J. (October 2007). Energy Efficiency of Modern Cellular Base Stations. In *29th International Telecommunications Energy Conference (INTELEC' 2007)*, pages 475–476.

Ma, C. and Yang, Y. (2011). A Battery-Aware Scheme for Routing in Wireless Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, **60**(8), 3919–3932.

Maghraby, H. A. M., Shwehdi, M. H., and Al-Bassam, G. (2002). Probabilistic Assessment of Photovoltaic (PV) Generation System. *IEEE Trans. Power Systems*, **17**(1), 205–208.

Mancuso, V. and Alouf, S. (2011). Reducing Costs and Pollution in Cellular Networks. *IEEE Communications Magazine*, **49**(8), 63–71.

Narvarte, L. and Lorenzo, E. (2000). On the Usefulness of Stand-Alone PV Sizing Methods. *Progress in Photovoltaics: Research and Applications*, **8**, 391–409.

Nema, P., Rangnekar, S., and Nema, R. (2010). Pre-feasibility Study Of PV-Solar/Wind Hybrid Energy System for GSM Type Mobile Telephony Base Station in Central India. In *The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, volume 5, pages 152–156.

Oh, E., Krishnamachari, B., Liu, X., and Niu, Z. (2011). Toward Dynamic Energy-Efficient Operation of Cellular Network Infrastructure. *IEEE Communications Magazine*, **49**(6), 56–61.

Papavasiliou, A. and Oren, S. S. (2010). Supplying renewable energy to deferrable loads: Algorithms and economic analysis. In *IEEE Power and Energy Society General Meeting*.

Paulescu, M., Paulescu, E., Gravila, P., and Badescu, V. (2013). *Weather Modeling and Forecasting of PV Systems Operation*. Springer.

Pires, C. (2007). Park WiFi. *Natoa Journal*, **15**(3).

Rappaport, T. (1996). *Wireless Communications : Principles and Practice*. Upper Saddle River, N.J. : Prentice Hall PTR.

Ru, Y., Kleissl, J., and Martinez, S. (2012). Storage Size Determination for Grid-connected Photovoltaic Systems. *IEEE Transations on Sustainable Energy*.

Saengthong, S. and Premrudeepreechacharn, S. (2000). A Simple Method in Sizing Related to the Reliability Supply of Small Stand-alone Photovoltaic Systems. *Conference Record of the Twenty-Eighth IEEE Photovoltaic Specialists Conference*, pages 1630–1633.

Safie, F. (1989). Probabilistic Modeling of Solar Power Systems. *Proceedings of the Reliability and Maintainability Symposium*, pages 425–430.

Sayegh, A. A. (2008). *Resource Allocation in Energy Sustainable Wireless Mesh Networks*. Ph.D. thesis, McMaster University, Hamilton, Ontario, Canada.

Sayegh, A. A., Todd, T. D., and Smadi, M. N. (2008). Resource Allocation and Cost in Hybrid Solar/Wind Powered WLAN Mesh Nodes. *Wireless Mesh Networks: Architectures and Protocols, Springer Science and Business Media Inc.*

Shinwari, M., Youssef, A., and Hamouda, W. (2012). A water-filling based scheduling algorithm for the smart grid. *IEEE Transactions on Smart Grid*, **3**(2), 710–719.

Shockley, W. and Queisser, H. J. (1961). Detailed Balance Limit of Efficiency of p-n Junction Solar Cells. *Journal of applied physics*, **32**(3), 510–519.

Shrestha, G. B. and Goel, L. (1998). A Study on Optimal Sizing of Stand-Alone Photovoltaic Stations. *IEEE Transactions on Energy Conversion*, **13**(4).

Sobu, A. and Wu, G. (2012). Dynamic Optimal Schedule Management Method for Microgrid System Considering Forecast Errors of Renewable Power Generations. In *IEEE International Conference on Power System Technology (POWERCON)*.

Steffel, S., Caroselli, P., Dinkel, A., Liu, J., Sackey, R., and Vadhar, N. (2012). Integrating solar generation on the electric distribution grid. *IEEE Transactions on Smart Grid*, **3**(2), 878–886.

Stine, W. B. and Geyer, M. (2001). *Power from the Sun*. Power from the sun. net.

The Meteorological Service of Canada (2012). National Climate Data and Information Archive.

UNN (2010). http://unn.edu.ng.

U.S. Department of Energy (2012). National Solar Radiation Database (NSRDB). Renewable Resource Data Center (RReDC), National Renewable Energy Laboratory (NREL).

Vukojevic, A., Parwal, N., and Yasko, C. (2012). Distribution transformer loading with deferred loads. In *IEEE Innovative Smart Grid Technologies (ISGT)*.

Wieselthier, J. E., Nguyen, G. D., and Ephremides, A. (2002). Energy-limited Wireless Networking with Directional Antennas: The Case of Session-based Multicasting. *Proceedings of IEEE 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'2002)*.

Wu, D., Gupta, D., and Mohapatra, P. (2011). QuRiNet: A Wide-Area Wireless Mesh Testbed for Research and Experimental Evaluations. *Ad Hoc Networks*, **9**(7), 1221–1237.

Xu, D., Kang, L., Chang, L., and Cao, B. (2005). Optimal Sizing of Standalone Hybrid Wind/PV Power Systems using Genetic Algorithms. *Canadian Conference on Electrical and Computer Engineering*, pages 1722–1725.

Yang, P., Tang, G., and Nehorai, A. (2012). A Game-Theoretic Approach for Optimal Time-of-Use Electricity Pricing. *To be Published in IEEE Transactions on Power Systems*.

Yu, W. and Qian, X. (2009). Design of 3kW Wind and Solar Hybrid Independent Power Supply System for 3G Base Station. In *Second International Symposium on Knowledge Acquisition and Modeling, KAM'09*, volume 3, pages 289–292.

Zefreh, M. and Todd, T. D. (2012). Energy Provisioning in Green Mesh Networks Using Positional Awareness. *To be Appeared in IEEE Transactions on Vehicular Technology*.

Zefreh, M., Todd, T. D., and Karakostas, G. (2013). Energy Provisioning and Operating Costs in Hybrid Solar Powered Infrastructure. *To be Appeared in IEEE Transactions on Sustainable Energy*.

Zefreh, M. S., Badawy, G. H., and Todd, T. D. (2010). Position Aware Node Provisioning for Solar Powered Wireless Mesh Networks. *IEEE Global Communications Conference (IEEE GLOBECOM'2010), Miami, Fla.*

Zeng, K., Ren, K., and Lou, W. (2006). Energy-Aware Geographic Routing In Lossy Wireless Sensor Networks With Environmental Energy Supply. *Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/wireless Networks*.

Zheng, Z., He, S., Cai, L., and Shen, X. (2012). Constrained Green Base Station Deployment with Resource Allocation in Wireless Networks. *Handbook on Green Information and Communication Systems, John Wiley & Sons, Inc.*