Local energy management through mathematical modeling and optimization

by

David Craft

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of
[O. R.]
Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2004

Author .....
Sloan School of Management
August 10, 2004

Certified by .....
Richard C. Larson
Professor of Civil and Environmental Engineering, MIT
Thesis Supervisor

Accepted by .....
James B. Orlin
Chairman, Co-director of Operations Research Center, MIT

Local energy management through mathematical modeling and optimization

by

David Craft

Submitted to the Department of Operations Research
on August 10, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

We develop an extensive yet tractable framework for analyzing and optimally controlling local energy networks. A local energy network is any set of generation, storage, and end-use devices existing to provide energy fulfillment to a building, a group of jointly operated buildings, or a village power system. The software developed is called TOTEM for Total Energy Management, and provides hourly (or sub-hourly) control over the flows in such energy networks. TOTEM manages multiple energy flows such as electricity, chilled water, heat, and steam together, since such energies are often coupled, particularly for networks containing cogeneration turbines (which produce electricity and steam) and absorption chillers (which use steam for driving refrigeration turbines). Due to the large number of interconnected devices in such networks, the model is kept as a linear mixed integer program, able to be solved rapidly with off-the-shelf mathematical optimization packages. Certain nonlinearities, for example input-output relationships for generators, are handled in this linear framework with piecewise linear approximations.

Modeling flexibility is achieved by taking a node-centric approach. Each device in the network is represented as a node, and depending on each node's set membership, proper constraint and objective equations are written. Given the network, TOTEM uses hourly electricity and fuel pricing, weather, and demand projections to determine the optimal operating and scheduling strategy for the day, in both deterministic and stochastic settings. MIT's cogeneration plant is used as a case study, with other examples throughout the thesis demonstrate the use of TOTEM for assessing and controlling renewable resources, storage options, and demands.

Extensions to the core TOTEM model include a demand charge model, used for making daily optimal control decisions when the electric bill includes a charge based on the monthly maximum power draw. The problem of heating, ventilation, and air conditioning (HVAC) control is treated separately since it strongly violates TOTEM's linearity assumptions. Nonetheless, we describe a solution approach to the HVAC problem which operates in conjunction with TOTEM. We also provide an analysis of storage suitability in stochastic supply and demand networks.

The node-based approach lends itself well to a software system that uses a drag-and-drop graphical network creation tool. We present a graphical user interface, the XML data representation, and the communication links to and from optimization software.

Thesis Supervisor: Richard Larson
Title: Professor of Electrical Engineering, MIT

Acknowledgments

Thanks to my mother for raising me to be diligent, and a relatively pleasant person to be around. Thanks to my first advisor at MIT, Larry Wein, who guided me for three years and sharpened my mathematical modeling skills greatly. Thanks to my current advisor Dick Larson, who let me run freely with an idea we discussed back in August. His targeted guidance got me over many humps in my research, and made for mostly smooth sailing the entire process. One particular incident sticks out. For weeks I had been grappling with how to include random events into the mathematical model, and as a result I was exploring many paths, but not really making any forward progress. Dick's advice, leave the randomness for later and just progress on the deterministic model, proved immensely useful. I appreciate immensely Dick's desire to see operations research become a valuable decision making tool in today's world, and his faith in my judgment as a researcher.

For financial support, I am indebted to the Dave Marks and the Martin Fellowship foundation, which funds students engaged in research related to improving the global environment. It is encouraging that at least some money exists for these endeavors, and that there exists awareness outside the offices of operations researchers that mathematical reasoning can lead to improved operational strategies. Many thanks also to Rob Freund, for generous financial support, and equally importantly, for his kind personal support as I stepped somewhat non-traditionally through MIT's OR program. His advice to me, make sure your life consists of doing what you love to do, while simple, could be heeded by many more people. Especially those trained in the art of optimization!

Three floors above the Operations Research Center sits the Laboratory for Energy and the Environment. As time goes by, and carbon enters the atmosphere at an increasing rate, I hope there will be more collaborations between the ORC and LFEE. Dave Miller, a fellow graduate student working on energy control, was a great partner in this project, with every meeting and conversation always leading to renewed vigor and vision. His advisor, Steve Connors, who introduced us in the first place, always

made time to assist and steer and encourage me. His upbeat style, optimism, and faith in my work, were very inspiring. Thanks to Marija Ilic who helped me understand how my work fits into the general field of electrical network control.

Across campus from the ORC sits the Building Technology Program of MIT's Department of Architecture. Many thanks to a recent graduate, Helen Xing, for sharing with me her thoughts on building energy control. Helen spent about one year getting a validated EnergyPlus building simulation model running for her optimal control investigations. She graciously shared all her computer code with me, allowing me to focus on the HVAC problem rather than software issues. I cannot overstate my appreciation of this type of knowledge sharing, which is so important for making progress in such complicated areas.

I believe we learn best by targeted guidance, with lots of room for self exploration. As my advisor Dick provided for me, so I tried to provide that to my undergraduate assistant Tao Yue. With a small amount of guidance from myself and Dave Miller, Tao developed a graphical user interface for this project single-handedly. He worked with an independence and an eye for design that will serve him well in his future. Tau co-authored Appendix C.

To my housemates Eliza and Anna and Kooky, thanks for providing a stable, fun, and supportive atmosphere, and letting me eat your food. Thanks to Ping for being a reliable lunch partner, when a break was always needed. Thanks to Lincoln, Elodie, and Melvyn for their desire to see operations research used to help level the playing field, instead of making it worse. Thanks to Lorena for being a cool friend, for the picture of the Hull wind turbine in Chapter 1, and to her parent company, Pioneer Investments, for supplying me with my notebooks. To Paul, thanks for the Brooklyn Brewery coaster advertising their use of wind power. I could not have written the second to last sentence on page 34 without it. Finally, thanks to the genius who put the couch in the ORC lunch room, and the other genius who came up with the Au Bon Pain 50% off bake sale from 4-6 PM every day.

Sometimes a scream is better than a thesis. – Ralph Waldo Emerson

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis presents a mathematical model and software framework for analysis and optimal control of local energy networks. A local energy network might be as simple as the heating and electricity configuration of a single residence, or as extensive as all of the steam, cooling, and electricity supply and demand of a connected set of buildings such as a college campus, complete with local power and steam generation and custom pricing from utilities.

Such systems are controlled in real time. When there is a certain objective to such control, such as minimize the cost to the user while keeping the state of the system (e.g. indoor temperatures, air quality) within acceptable ranges, the control choices are not obvious. Here is a partial list of features of energy networks which make optimal control especially difficult:

- Electricity prices that are different at different hours of the day.
- An electric bill that includes a demand charge, which is a charge based on the maximum power drawn at any time over a billing cycle.
- Self generation power turbines that generate electricity and steam simultaneously.
- Storage capabilities, which can be used to smooth peaks and valleys of electricity usage, but are expensive and often inefficient.

- The option to purchase renewable power sources such as wind turbines and solar panels.

When these features exist in a user's energy network all at once, the resulting optimization problem is very difficult. This thesis is devoted to 1.) Defining what the appropriate objective is in such a control problem, 2.) Establishing the mathematical optimization formulation to solve such a problem, 3.) Implementation in a user-friendly software system, and 4.) Surveying the field of energy control broadly to emphasize the promising areas for continued research.

1.1 TOTEM: Total energy management

This focus of this thesis is the development of a mathematical program for the optimal control of energy networks and the implementation of that mathematical program into a user-friendly software system called TOTEM (Total Energy Management).

The scope of TOTEM is large. First, it is a system which allows a user to define a network using a graphical drag-and-drop interface. Possible components that might exist in the user's network are arranged on a palette, and these components are connected by the user to indicate flow paths of the network. Flows between components can either model actual flows, such as electricity or chilled water, or information which is passed from one node to another. For example, when heat is captured from a gas turbine and used to create steam, the steam output might correlate well with the electricity generated, in which case the information passed to the heat recovery system is the electric output of the generator.

The network definition is the first step in applying TOTEM, see Figure 1-1. Figure 1-2 shows a high level view of the rest of the information architecture for TOTEM. An example of a physical TOTEM network is shown in Figure 1-3 which contains the main types of nodes: supply, storage, and demand.

The standard use of TOTEM is for planning and control of a network's energy usage over the short term, such as a single day, an eight hour work shift. In the early morning, TOTEM would receive all of the information it needs to optimize the

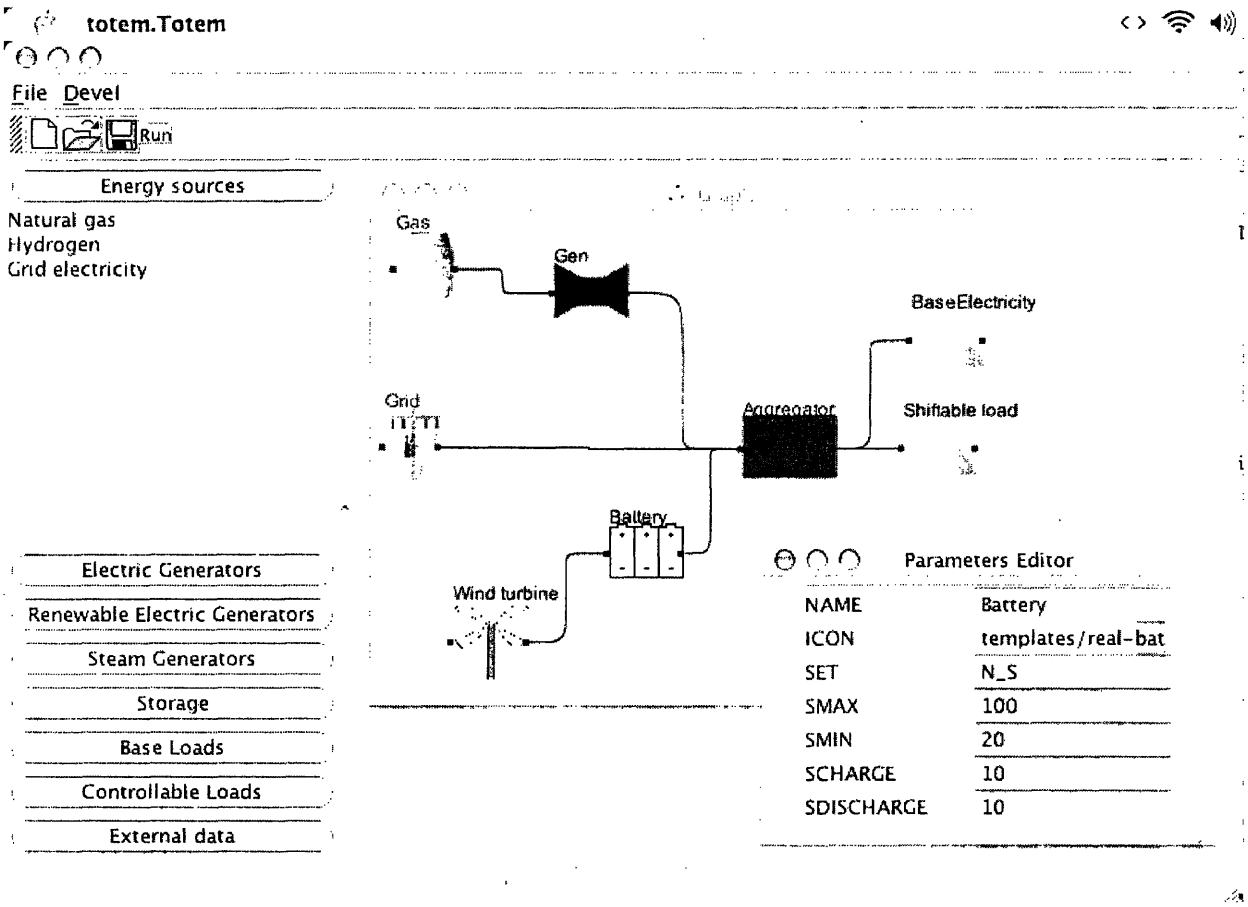


Figure 1-1: The TOTEM network definition GUI.

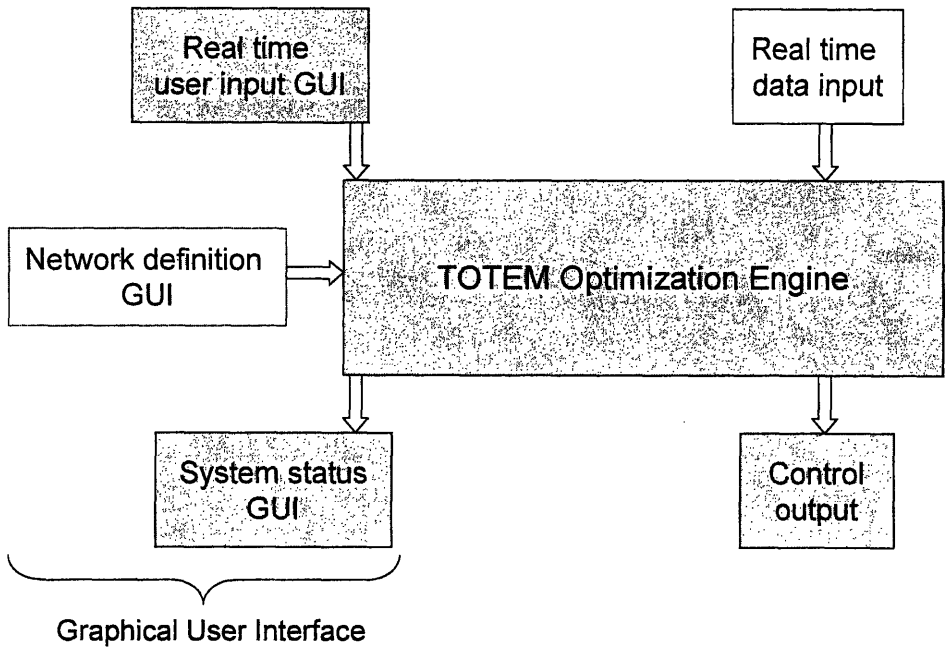


Figure 1-2: The core TOTEM solver (see Chapter 3) requires input from the sources shown in order to produce optimal operating rules. The network definition GUI (Appendix C, and Figure 1-1) is a one-time interface used at the beginning of a new TOTEM project. The other GUIs are continuously active and used for human interaction with the system.

day's energy uses. This includes weather, pricing, expected occupancy patterns and demand patterns, pricing information from the electric company, system maintenance requirements, and schedulable jobs. With that information, TOTEM computes the optimal running strategy for every controllable device in the network. TOTEM is designed to handle deterministic or stochastic environments. Many networks, while they exhibit stochastic behavior in terms of small scale energy demand fluctuations, are well predictable in an average sense. In these cases, TOTEM is run deterministically to determine device scheduling (for example, when individual generators are turned on and off) and the fluctuations are handled automatically by device control systems meant for exactly that (such as governors).

TOTEM is also suitable for longer term applications such as seasonal hydro-

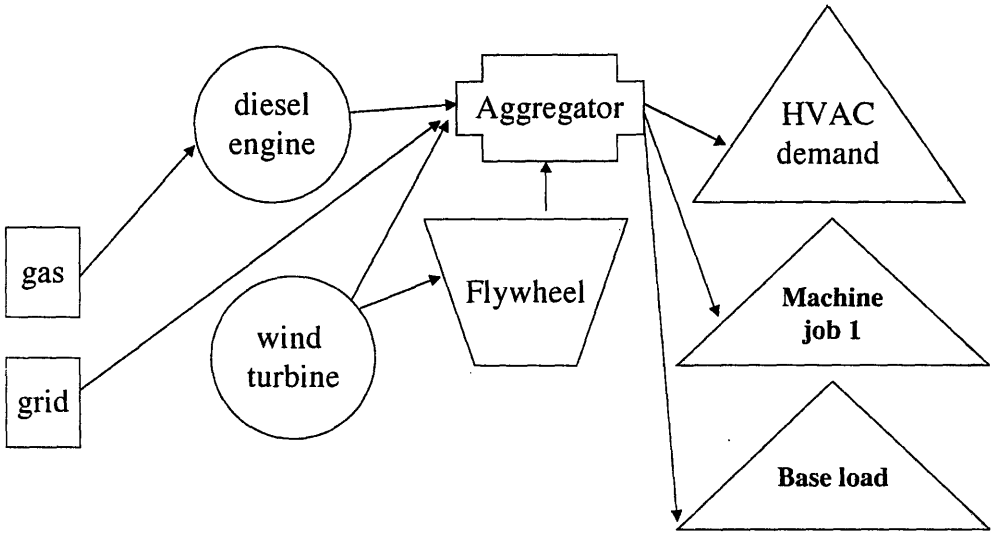


Figure 1-3: A example electricity network showing energy sources and sinks as nodes. TOTEM uses this underlying structure along with external price and weather information to decide optimal supply and demand control for a given planning horizon (for example, one day).

reservoir control (storage and release planning) and week long planning horizon for job scheduling. The horizon length over which TOTEM optimizes is primarily limited by the quality of forecasts. Depending on the complexity of the user's network, computation time could limit the horizon also, but there is an additional feature of most energy networks which reduces this concern: the natural cycling of networks shutting down at night. Boilers and generators are set to their low running levels, lights are turned off, etc., thus creating a certain independence between successive days. This is obviously system dependent but it holds to some extent in most networks which TOTEM addresses, especially those without storage. However, even for networks with storage, we will discuss how natural cycling can be handled.

1.1.1 Thesis outline

The remainder of this thesis is structured as follows. Chapter 1 continues its overview of the field of energy management by describing typical applications of TOTEM. We also overview distributed generation technologies and today's electricity pricing structures, which together create the setting for TOTEM. Chapter 2 is a literature survey. Most written works and commercial products are geared to either supply control or demand control (TOTEM handles them simultaneously) and thus we break up our literature review in this way. Chapter 3 presents the mathematical formulation which is the core of this thesis. The ending essay of Chapter 3 discusses the synergy of optimizing supply and demand at the same time. Appendix C provides the details of the software implementation of TOTEM. Chapter 4 provides some extensions of TOTEM, most importantly a stochastic version. The stochastic version models uncertainties by probabilistic scenarios, a framework both easily understood by facilities managers, and relatively simple to implement. An approach to HVAC optimization is also presented in Chapter 4. Chapter 5 introduces a method of dealing with a monthly demand charge. Since TOTEM for businesses and residential users makes control decisions over the span of a day or two, but a demand charge is levied based on the highest grid draw over a one month period, we require some logic to make daily optimal decisions given the uncertain future power draws. This chapter provides a dynamic programming model of this and shows how the model works together with the core TOTEM model of Chapter 3. Three case studies are presented in Chapter 6. The first is a remote hybrid power application (wind and diesel), and the second is a model of MIT's cogeneration plant. To further demonstrate features of TOTEM, the base MIT configuration of today is augmented by potential future hardware, and the savings are estimated. The third application is a building simulation HVAC control study. Chapter 7 is a stand alone chapter which addresses theoretically the potential of storage in energy networks. The final chapter describes the pieces needed in a real TOTEM implementation. It reviews the state of the art in short term weather prediction, the difficult task of HVAC control, and modules needed to complete the

picture. Appendices contain software code of TOTEM implementation.

1.2 Different types of energy users

Below we show some specific application areas for TOTEM. While TOTEM is most useful when several of these domains exist simultaneously, it is useful to break out different application areas to understand the coverage of TOTEM.

- **Temperature control**: Indoor space heating, ventilation, and air cooling represent significant costs to many users. Indoor environmental conditioning is good candidate for intelligent energy management since there is some inherent flexibility in how and when rooms are cooled or heated. TOTEM can use weather forecasts, occupancy information, and real-time energy prices to find optimal heating and cooling supply schedules. When a user can control valves, thermostats, flow temperatures, and flow speeds, the problem becomes sufficiently nonlinear (and the system dynamics virtually impossible to write down) and we handle it with a separate HVAC (heating, ventilation, air conditioning) module.

- **Job scheduling**: Chooses the optimal start times for a set of jobs with known energy profiles. This is done based on real-time pricing from the grid and onsite generator characteristics.

- **Distributed generation**: With increasingly cheap DG units becoming available, and the potential to harness the efficiencies of cogeneration (using the excess heat from a gas turbine for steam generation), DG is growing in market share. TOTEM handles arbitrary networks of DG devices, grid connected or isolated from the grid, deriving the optimal flows of electricy, heat, and chilled water to satisfy user demands.

- **Hybrid networks**: A popular choice for remote power applications is a hybrid network consisting of renewable sources such as wind turbines and solar panels

coupled with more standard DG such as a diesel engine. Storage is common in such off-grid applications as well, and can almost entirely eliminate the need for the backup fossil generation units. TOTEM handles the optimal operation of such a network, especially useful when capital constraints limit the amount of storage available.

TOTEM is useful for optimal real time control of existing energy networks, but, importantly, is also key as an evaluation tool for new network components and new networks. When considering new generation devices, for example cogeneration for a college campus, coarse calculations can be used initially for average loads throughout the year to determine the initial feasibility of particular products. Ultimately though, the real economic feasibility will only be known when the real operating costs are known [48], and TOTEM allows the optimal operating costs to be determined. The network definition user interface, Figure 1-1, allows a user to quickly swap trial components in and out and run the resulting TOTEM models.

1.3 Overview of distributed generation

New technologies in the last 20 years or so are making the local production of electricity a viable option for electricity users. Traditionally, due to the high cost of electricity generation and the economies of scale associated with large-scale production of electricity, most electricity has been produced by large centralized fossil-fuel or nuclear based power generators. In the 1980's, highly efficient gas turbines, modified from jet engines, became available as local electricity generators. Since then, we have seen technological advances in renewable energy sources, microturbines, fuel cells, and storage devices, all which continue to change the paradigm in electricity distribution.

As we discuss the types of equipment available, it will be helpful to compare the relative sizes in terms of electric output. Table 1.1 gives a sense of magnitude regarding kilowatts (power) and kilowatt-hours (energy) of some electricity devices and plants.

Item	Power draw
Incandescent lightbulb	100 W
Electric blanket	200 W
Iron	1 kW
Air conditioner (in window)	1 kW
Electric basboard heater, 8 ft	2 kW
Clothes dryer	4.5 kW
Water heater	5 kW
Electric furnace	15 kW
Solar panels on Harvard Fitness Center	36 kW
Hull, MA wind turbine	660 kW
VAMC Hospital, Jamaica Plain	4.5 MW (3x1.5 MW cogens)
MIT (approximate daily peak)	20 MW
Proposed Cape Wind project	170 MW (average)

Table 1.1: Representative power usage by common appliances and power draw for some nearby locations.

1.3.1 Fossil fuel based generation

Fossil fuel based electricity generation typically involves rotating an electric generator to produce an electric current. The various ways to produce turbine rotation differentiate electricity generation devices. In this section, we discuss how fossil fuel is used as the means to rotate an electric generator.

Internal combustion engines

The oldest and still the most common form of distributed generation is the internal combustion engine. Gasoline or diesel fuel are the typical inputs although just about any liquid that will burn can be used. This is a highly reliable form of DG power, since the internal combustion engine has been refined over so many years of use in the automobiles and generators. Internal combustion engines can also be started rapidly for quick emergency response. The main drawbacks are noise, maintenance, and NOx emissions (which can be decreased with catalytic converters, but with a hit on efficiency) [24].

Dealers such as Caterpillar (www.cat.com) offer gas or diesel fired engines anywhere from 7 kW to 16 MW.

Combustion turbines

Combustion turbines, related to and derived from aircraft jet engines, are a medium to large scale power generation source. MIT operates a combustion turbine which can run on oil or natural gas, and generates 21 MW of power. General Electric (www.gepower.com) offers aeroderivative power turbines in the range of 13-100 MW. These devices typically serve base power needs and are not turned on and off except for maintenance or system failures.

Microturbines

Derivatives of the combustion turbine, these relatively new power sources run at extremely high rotational speeds on air bearings, thus eliminating the need for lubricants and dramatically reducing the downtime for system maintenance. Microturbines are very efficient in their conversion of fuel to electricity, but their high installed cost has thus far kept them from capturing much of the DG market [24]. A single microturbine provides a small amount of power, in the range of 25-100 kW, but they can be placed together to create larger power banks.

Microturbines can run on a broad range of fuels, making them attractive for remote locations where fuel availability can be sporadic.

A recent application of an array of Capstone (www.capstoneturbine.com) microturbines is located at a large landfill outside of Los Angeles. This installation uses 50 microturbines to convert the landfill fumes ("biogas") into 1.3 MW of electricity. Without the turbines burning the fuel, the fuel would otherwise be flared off, a harmful source of pollutants to the environment.

Cogeneration

While centralized fossil fuel based power stations offer the best efficiency in terms of conversion of fuel to electricity, onsite power generation has two distinctions which make it comparable, and in certain cases superior, regarding energy conversion. The first is that onsite power generation does not experience the transmission losses that

a central plant does. From power station to the end residential user, once can expect a 7% energy loss. This loss is tolerated due to the increased generation efficiency and economies of scale of large centralized generation facilities. But, as DG becomes comparable in efficiency, these transmission and distribution (T&D) losses become more visible.

The second big efficiency advantage of local generation involves capturing the waste heat from the turbines. Central plants cannot do this effectively, because even if they were to recover the heat, shipping it to customers is not possible. Onsite generators though can be configured to use the exhaust heat for steam production. This capture and use of heat from turbines used in the generation of electricity is called cogeneration. Cogeneration typically doubles the energy conversion efficiency of a power plant.

MIT runs a cogeneration plant. As of the year 2004, there is one combustion turbine which provides up to 21 MW of electric power to the MIT campus, almost all of its electricity demand. The waste heat is used to generate steam in a heat recovery steam generator (HRSG), and the total system efficiency is on the order of 85%. Additional steam needed by the campus comes from three traditional boilers.

1.3.2 Renewable generation

Renewable generation refers to electricity or heat production from sources which are renewable. Fossil fuels are not renewable because they are a finite source which will eventually run out. In comparison, sunlight and wind are renewable resources since they will be available as long as the sun is shining. Other renewable sources include hydro-power and tidal power, the latter being difficult to harness but a potentially large source of energy. Agricultural crops are termed renewable if the farming processes used to create them to do not damage the earth and the environment, such that they can be grown year after year. Crops grown as fuels for power generation are called biomass, and if grown in a sustainable manner constitute another renewable resource. Most developed countries, with the exceptions of the United States and the United Kingdom, satisfy about 10% of their energy needs with biomass. Developing

countries average about about 36% [8].

This section focuses on the prime candidates for renewable generation, wind and sun. Wind turbines, similar to the fossil fuel based generators discussed in the previous section, produce electricity by rotating an electrical generator. The sun can be used for direct solar heating of air or water, for super heating of a liquid in order to drive an electric generator, or, most commonly, with solar panels to produce electricity directly.

The renewable techniques described below have the distinct advantage of not requiring fuel, and thus, aside from fixed costs and maintenance, are essentially free sources of power. The problem is that those fixed costs are often so high that these technologies are uneconomical. Another disadvantage is that renewable sources provide power only when the sun is shining or wind is blowing. Renewable power sources have great potential in reducing the dependence of developed countries on the grid, and they have a great potential for developing countries where getting on the grid is too expensive or just infeasible. The primary barrier to the use of renewable energy is the fixed cost. An energy management system like the one presented in this thesis can help overcome such barriers, especially for technologies like photovoltaics and wind power, which are becoming increasingly economically attractive, for mainstream usage and in certain niche developing country applications [61].

We also include a discussion of fuel cells in this section on renewable energies. Fuel cells rely on hydrogen and their exhaust is water, which makes many people consider them renewable resources. For this classification to be valid, hydrogen needs to be obtained renewably, and not as it is commonly done today, by refining fossil fuels.

Wind power

Wind turbines are becoming increasingly popular as a means to generate electricity. The building to replace the World Trade Center, The Freedom Tower, will have wind turbines along its upper spire which will provide about 1MW, or 20% of the building's electric load. The Brooklyn Brewery buys all of its power from a wind farm in upstate New York. In Hull Massachusetts, a single wind turbine (see Figure 1-4 taken from

Boston Harbor) on the shore generates 660 kW of electric power, which cancels the town's bill for street and traffic lights and more.

Several interesting designs for wind turbines are available, but the most common are a sleek upgrade to the old windmills of The Netherlands used to pump water out of the low, flat, wet region. A large windy area is needed, and since power available is proportional to the cube of the diameter, it is best for the rotor to be large also.

The main drawbacks of wind turbines involve environmental concerns (birds flying into them, disruption of fishing access), and that the power output depends on the instantaneous wind level.

Figure 1-4: Vesta wind turbine (660 kW) in Hull Massachusetts, as seen from Boston Harbor.

Solar thermal power system

A solar thermal power system uses sunlight to heat a liquid to temperatures high enough that the liquid can be used to create high pressure steam to run a steam turbine. The liquid used to gather the sun's energy flows through a pipe heated by reflectors which focus sunlight. As the liquid travels through the pipe, its temperature increases until it is hot enough for the steam turbine generator. In order to provide a steady source of power, the pipe network has two liquid storage vessels of heavy

insulation, one for the super hot liquid, and one for the liquid after it is used by the steam generator.

Photovoltaics

While the solar thermal power system described above is a logical way to use sunlight to drive an electric generator, there turns out to be a more practical method involving the direct conversion of sunlight into electricity. This is the solar panel, consisting of semi-conductors which produce an electric current when exposed to sunlight. This is the first electricity generation device we have discussed that involves no moving parts. It is silent and requires no maintenance, and as such it is suitable for rooftop residential and business installation. As of the writing of this thesis, the cost of solar installation is about $4.50 per watt.

Solar panels produce direct current and so typically need to be converted into alternating current for standard usage, such as within a home.

Solar potential in New England, averaged across a year is approximately 4 kWh per day per square meter of installed panel [47], [62]. This amounts to 167 Watts if spread evenly over the day (obviously, only possible with storage). Referring to Table 1.1, we see that this amount could either run a couple of light bulbs continuously, or provide the energy for a one hour clothes drying cycle. Fortunately, rooftop area greatly exceeds 1 square meter, allowing solar installations to supply much if not all of a house's energy needs.

Direct solar power

Sunlight can be used to heat water also, rather than produce electricity. This is done similar to the solar thermal power system described above: water is piped through a length of pipe along which it gathers energy from the sun. This water can then be used as a house's source of hot water, or as input to a boiler or electric heater. Parabolic mirrors can be used to focus the sunlight, but since flat black panels are so much cheaper to fabricate and install, they are more common.

Often, rather than installing direct solar power systems or solar panels, houses

are designed to take advantage of passive solar heat [41]. While designs like this are important environmentally, we do not discuss them here since they offer no control options.

Fuel cells

A fuel cell is a direct current electricity source with no moving parts. The fuel is hydrogen and the exhaust is water. These devices are undergoing intense research and development efforts around the world since they are such promising sources of clean energy, both in DG applications and as automobile power sources. They produce electricity in a process which is the reverse of electrolysis. In electrolysis, electricity is sent through water causing it to decompose into hydrogen and oxygen. In a fuel cell, hydrogen (stored in high pressure fuel tanks) and oxygen (from the air) are allowed to combine, and the fuel cell is able to extract electricity from this process.

Classifying this electricity device as a renewable energy source is a bit misleading. Hydrogen is expensive and difficult to produce, and one of the cheapest ways to obtain it today is by refining fossil fuels (stripping off the carbons from hydrocarbon chains of gasoline for example). However, a clean vision for a hydrogen-based energy economy [50] involves using other renewable sources, such as wind, to produce hydrogen from water via electrolysis.

Fuel cells are used to supply base steady power load, with a technology like microturbines or traditional gas turbines used to react to power fluctuations.

1.3.3 Storage

Storage of energy is used for three main reasons. The most common is to ensure reliability and power stability. From the perspective of the average homeowner, the grid is a reliable and stable power source, and a few outages per year are tolerated easily. For sensitive equipment and processes, outages can be very costly, and a source of power which does not fluctuate from the needed levels of voltage and amperage is critical. As such, storage is often employed, as well as backup generation.

The second main use of energy storage is to smooth the power availability from variable power sources such as wind and sun. We are still currently in a time of predominantly fossil-based and nuclear power. For example, in Massachusetts over 90% of electric power comes from fossil and nuclear and less than 1% comes from wind and solar [19]. Fuel based units are dispatchable, meaning they can be fired up any time to respond quickly to changes in demand. As such, storage to smooth out such power supply is not common. Things are different in networks that rely heavily on wind turbines and solar panels for electricity, networks which benefit greatly from storage. In fact, with enough supply and storage capacity, these renewable power source/storage units become essentially "dispatchable", increasing user attractiveness considerably.

The third main use, closely related to the second, is storing energy from the grid or DG units based on hourly grid pricing. While storage might not be needed in a network, it could save money if a user stores electricity when it is cheap for use later when it is more expensive. TOTEM handles the second two of these benefits of storage when it derives the optimal flow strategy for networks which include storage.

Next we discuss and compare key storage technologies. A survey of electrical storage is given in [36]. Figure 1-5, from the same document, displays the appropriate storage technologies based on the power output desired and the duration of such power output.

Batteries

Batteries are the most common form of electricity storage. For power applications, the most common battery is the type used in an automobile, the lead-acid battery. This battery is popular due to its ability to be deep cycled (charged and discharged) many times.

Flywheels

Flywheels are a purely mechanical form of energy storage. A flywheel is a free spinning wheel that stores energy by rotational velocity. The common application of a

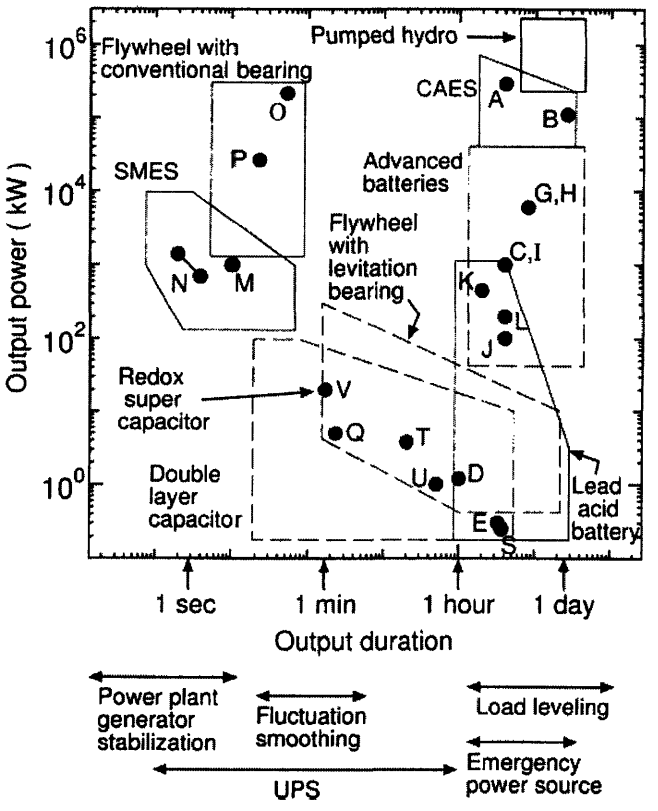


Figure 1-5: Storage technologies grouped in terms of their typical power output and power output duration. Dotted lines indicate technologies still under development. Reprinted from [36] with permission from corresponding author and Elsevier.

traditional flywheel is to stabilize the power output in a piston-based internal combustion engine. In storage applications, energy to be stored is put into speeding up the wheel and energy is extracting by connecting the spinning wheel to an electric generator, thus slowing it down. Flywheels as storage devices can be used in buses and cars to capture the energy that normally is wasted as friction in braking, storing it for the next acceleration. For urban buses, this can result in dramatic energy savings.

Flywheels require high rotational velocities for sufficient energy storage, which in turn demands high strength materials to withstand the tensile stresses in fast spinning wheels. Recent advances in composite high strength materials are making flywheels more attractive, although they are still a future storage strategy not competitive with

traditional batteries.

Compressed air

Another mechanical storage technique is compressed air. By running a pump driven by electricity or a wind turbine, one can compress air in a chamber, which can be later expelled through a turbine to generate electricity. However, significant space is required for using compressed air as energy storage, with empty mines being one possibility. As such, this is not a widely used technology, but could have potential in niche applications.

Capacitors

Capacitors store a large amount of energy which is released as a burst rather than more slowly as in batteries. This makes them suitable for specialized applications only, including high energy X-rays, sub-atomic particle research, metal forming, and hybrid cars, and are not important for the typical user of electricity. Ultracapacitors are related devices that offer advantages of capacitors combined with the steadier supply characteristics of batteries.

Superconducting magnetic energy storage (SMES)

These devices are used in power quality applications where any sudden drop in power supply is not acceptable. A SMES system can provide a sudden MW burst of electricity, which is stored in the magnetic field produced by direct current flowing in a cryogenically cooled superconducting coil. Tennessee Valley Authority utility has installed them in their distribution grid to ensure power quality.

Like capacitors, these devices are used for very short time scale power quality management, and are not considered in TOTEM networks. While they may exist in networks handled by TOTEM, they are controlled by separate control systems.

Thermal storage

Except for batteries, the storage devices listed above are not commonly used in local power networks. Thermal storage is quite common however and can dramatically increase the energy efficiency of a building, especially when used in conjunction with wind or solar energy [41]. Thermal storage does not supply electricity, but either cooling or heating power. Thermal storage may take the form of ice or hot water (or some other liquid or solid, such as rock, brick or sand heat storage). Space requirements make thermal storage most applicable in larger buildings. Thermal storage is usually used under time-of-use pricing structures (see Section 1.5) since heating and cooling demands often coincide with peak price times.

1.4 Overview of deregulation and electricity pricing

In this section, we review the history of electricity regulations in the United States. We do this to understand how and why the electric power environment has changed, and hence understand the new playing field for optimal energy control.

Early in the history of electricity usage in the United States, electricity was seen as a good candidate for a single ownership due to economies of scale in power generation and distribution. However, as monopolies are illegal in the United States, the government decided to regulate electric utilities, at the state level. As utilities grew in size in the first few decades of the 20th century, they began to serve more than one state, and as such could not be controlled by state regulations. The government therefore passed the Public Utility Holding Company Act of 1935 (PUHCA). This act provided regulation of the utilities through the Securities and Exchange Commission. One important consequence of this act is that it allowed the creation of huge multi-state utilities. Therefore, for almost the entire 20th century, it was a handful of huge utilities that provided the electrification of the US, and there was little to no room for private companies entering the field, whether in generation or transmission.

This changed with the passage of the Public Utility Regulatory Policies Act of 1978 (PURPA). This act, passed after a series of blackouts and nuclear plant difficulties in the previous decades, encouraged competition in the electricity generation sector. The Energy Policy Act of 1992 (EPACT) further encouraged private entry into the wholesale electricity generation market, but it was not until 1996 when the Federal Energy Regulatory Commission (FERC) passed Order Number 888, which required utilities to open their transmission lines to private wholesale companies generating electricity [40]. Still, while these laws all indicate the restructuring of the electric supply side, how is this affecting the end electricity consumer?

In 1998, the Clinton Administration proposed a top down national approach to electricity deregulation which would result in consumers having a choice of their electricity providers [17]. This is the key step needed in order to see competitive and varied pricing plans presented to the consumer. However, uncertainty at the federal level in the implementation of these new policies has resulted in states being more or less on their own in terms of the timeline regarding the proposals. Thus, as of the writing of this work, some states have made progress, others have not. Figure 1-6 shows the state of deregulation efforts throughout the United States. Note however that even in the states that do not allow customers a choice of their supplier, large customers (anything bigger than residential) are still usually given a choice of their pricing plan.

With deregulation, several different pricing structures are becoming available to the end electricity consumer. At the simple end of the spectrum, these tariffs involve prices that depend on the time of day and at the complicated end they involve demand charges, interruptibility plans, and plans individually tailored to specific customers. One can also find several plans where the billing is done differently for different electricity usage, for example street lights billed differently than water pumps. As deregulation continues, and more players enter the electricity market, we can expect still more pricing mechanisms to emerge.

Deregulation and technology improvements have also created a structure where electricity customers can choose to provide some or all of their own power by adding

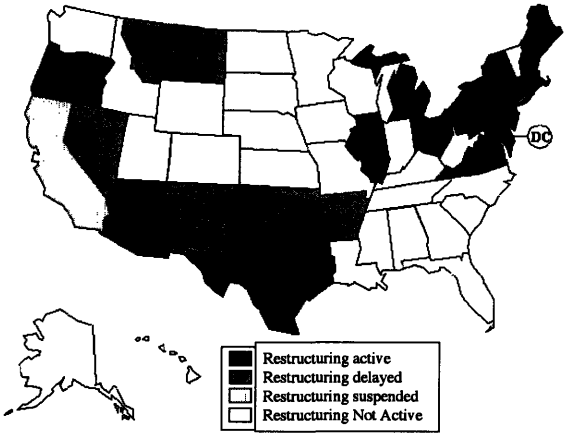


Figure 1-6: Electricity deregulation efforts by state in February 2003, reprinted from www.eia.doe.gov/cneaf/electricity/chg_str/regmap.html

distributed generation to their grid connected networks [24]. Some of these customers may have the ability to sell power to the grid if they can generate a surplus at certain times in the day.

1.5 Electricity pricing tariffs

In this section, we describe the common types of pricing plans available to consumers in the United States. The simplest and most common for residential service is the flat rate, which charges a fixed amount (around 7 cents) per kilowatt-hour (kWh) electricity used. The rates below are more complicated and offer potential for dynamic optimization.

Time-of use (TOU). This is the simplest of the dynamic pricing tariffs. The price per kWh of electricity is different for different times of the day. Typically there are just two different rates, on-peak and off-peak, but some companies, Puget Sound Energy for example, have more than two rates.

Real time pricing (RTP) and the customer baseline load (CBL). The customer baseline load (CBL) is by far the most common method of billing for real time

electricity prices, but to understand it we first explain real time pricing.

By RTP, most utilities mean hourly electricity prices notified to the customer at the end of the previous day. For example, PJM (the power company servicing Pennsylvania, New Jersey, and Maryland) and Cincinnati Gas and Electric Company post the next day's prices around 5:00 PM and those prices are fixed for the next day. Some companies provide a forecast for the next day's hourly prices, and then an hour before they send the final updated price.

The CBL is a historically based hourly load curve for the entire year specific to the customer, agreed upon by the utility and the customer. For example, this curve could be generated by averaging the customer's hourly usage over the past 5 years. The RTP bill is designed to be price neutral with respect to the CBL. This is accomplished by the following bill formula:

$$\text{BILL} = B_{\text{CBL}} + \sum_t (E_t - B_t)c_t \tag{1.1}$$

where B_{CBL} is the bill amount under the original pricing plan of the customer assuming he followed the load curve exactly as specified by the CBL, E_t is the electricity usage in hour t, B_t is the baseline amount given by the CBL curve, and c_t is the hourly real time price. Notice if your actual usage E_t equals your expected usage B_t, then your RTP bill is exactly the same as your bill as calculated by your previous pricing plan, B_{CBL}.

Fixed rates. Here, the rates of electricity change depending on how much electricity you consume over the month. For example, Duke Power offers a plan where the first 350 kWh is 6.65 cents/kWh, and anything over that is charged at 7.5 cents/kWh. This is the exact opposite of a volume discount. It creates an incentive to keep usage below 350 kWh. NSTAR – the local Boston utility, on the other hand, offers a fixed rate plan for commercial/industrial customers giving a volume discount. For June-September (winter months, same structure but cheaper, as Boston is a summer peaking area), it is 12.489 cents per kWh for the first 2000 kWh, then for the next 150 kWh it is 5.261 cents per kWh, then for anything over that it is 1.85 cents per

kWh. A demand charge is included (see demand charge section below) at the rate of 12.34 dollars per max kW power draw above 10 kW. We discuss optimization in the presence of a fixed rates tariff in Chapter 4.

Max power billing. This is an uncommon, but interesting one. FirstEnergy (a utility servicing Ohio, Pennsylvania, and New Jersey) offers it to customers who use more than 1000 kWh per month. Let x be the maximum power usage over the course of the month (in kW). The customer is charged at a rate r_1 for the first $125x$ kWh of energy use, and the remaining electricity used is billed at a lower rate r_2. The idea here is to keep your maximum power draw down as low as possible, therefore creating a more even demand pattern. Demand charges are the more common approach to this.

Demand charges. Both TOU and fixed rates tariffs are often modified by adding a demand charge, which encourages customers to keep maximum power draw low. In its simplest form, a demand charge is formed by multiplying your maximum power draw over a month (power is averaged over 15 minute intervals so that very short duration spikes do not unfairly inflate your bill) by a rate, say \$10 per kW. But often this charge is not included if your max power draw is low enough, or there is a different rate depending on the value of the max power draw. In Chapter 5 we provide a complete analysis of optimal strategies for dealing with the demand charge.

The host of pricing plans offered today, and the increased opportunities for grid connected DG, exist due to the very recent trend of deregulation. Currently it appears that small and medium commercial and industrial customers are willing to pay substantial premiums to avoid any non-constant price structure, and this is probably linked to the lack of decision tools available [22]. It is the misperception that variable pricing means higher costs which has helped keep time dependent pricing from being widely accepted. New technology is required to manage the complicated decision processes facing electricity consumers, a technology which this thesis attempts to provide.

Chapter 2

A survey of approaches to energy control

The number of options available to an energy user grows everyday. The positive side of this is that a user can purchase specialized components for his energy needs; the downside is that in a complicated local energy network, intelligent real-time decisions are very difficult. If electricity and fuel prices vary over time, and end use demands can be controlled in reaction to this, the optimal control strategy becomes complicated. Add to this devices that provide multiple energy needs (most notably, a cogeneration turbine, which provides electricity and heat), storage devices, and stochastic power sources like wind and sun, and we have an extremely challenging control problem. Along with a complicated problem comes a variety of approaches to solving it.

Because it is only recently that many customers in the United States are being given the option of time dependent or total usage dependent prices, there has been relatively little modeling effort in the area of optimal electric usage under today's price schemes. Some efforts have focused on the economic impact of demand side management, but few have provided actual algorithms necessary for smart demand side control. In addition, there are very few companies offering products which approach the problem in the way that we present in this thesis.

The commercial products which exist today, as summarized in [42], can be classified as either energy information systems or limited energy decision tools. Energy

information systems provide electricity usage to the consumer for further processing, but ultimately require manual decisions made by the user. Tools that actually make energy decisions, such as building management systems, are typically hard-coded with strict rule sets (e.g. turn on the space heater when temperature falls below 64 degrees F), and as such are not optimizing, they are merely controlling. Furthermore, due to the relatively recent appearance of distributed generation (DG) systems, there are no products offering intelligent centralized control for such networks. In fact, most products claiming to control energy networks are based on single device control, without considering the global system (other devices, external pricing, upcoming demand patterns). Without system level logic, one cannot expect optimal solutions from these control systems.

2.1 Literature review organizational approach

Because the modeling in this thesis is applicable to such a wide range of consumer energy flow problems, the following survey is necessarily broad. While the number of papers is quite large, the present work is the first to the author's knowledge which attempts to address the general problem of the optimal control of multiple energy flows in a local, customer level network containing elements of supply, storage, and demand. The work that is reviewed below are those that fall under the umbrella of what TOTEM is capable of handling.

There are two logical ways one might organize such a literature review. One is by mathematical technique employed, the other is by application domain of the model. We have chosen to organize the review by application domain since the goal of this thesis is not the introduction of new mathematics, but the introduction of an useful piece of operations research. We break up the review into three broad categories: demand management, supply management, and storage management. Demand management refers to controlling end uses, most notably heating, ventilation, and air conditioning. Other examples of managing the demand side include load curtailment and job scheduling. Supply management refers to controlling generators, boilers, and

other devices which supply energy to a network. The main literature for this area comes from power systems management, for they have historically been the ones providing all of the electricity supply. The final section covers storage. In each section, we note the types of mathematics employed in the research and discuss the rationale of such methods. This sets the stage for an understanding of the choice of the core TOTEM mathematical framework.

Any commercial products in the various subheadings below will be reviewed first, to give the reader an immediate sense of the state of technology in these areas. Following the product review, we review the scientific literature.

2.2 Demand management

By the phrase demand management, we mean the type of control exercised by a program like TOTEM: customer decisions made for the local energy uses such as HVAC, laundry, hot water, etc. This term is distinguished from the term demand side management (DSM), which refers to a utility controlling a customer's energy use for smoothing aggregate electric demand. This is typically implemented for HVAC control, since utilities are often stressed due to air conditioning load on hot summer days. The vast majority of literature on demand control deals with this type of control from the utility's side, and is geared towards designing effective and acceptable load curtailment programs. Since the goal of such efforts, the smoothing of aggregate customer demand, is different than a TOTEM user's goal, which is to minimize cost, this literature is largely irrelevant for our purposes. Rather than review it, we turn to the smaller body of products and literature which focus on the customer's side of the problem.

2.2.1 Commercial product review

WebGen Systems [42], out of Cambridge MA, offers an extensive optimization system for demand control based on the technology of neural networks. Neural networks are used first in a non-linear system identification phase, and then for optimal control.

The flagship product of WebGen is called 'Intelligent Use of Energy'. It focuses on HVAC control and does not include DG control. While the specific implementation used by WebGen is proprietary, a feature of neural network approaches is that they are flexible and they can learn a network, but they do not typically capitalize on known system attributes, and in general will give sub-optimal results. However, as we will see later in this thesis, HVAC optimal control is extraordinarily difficult, and neural network or similar approaches might be the only way of sensibly attacking the problem.

WebGen Systems is not typical of the demand management systems currently available, in that it actually uses an underlying mathematical model to make intelligent decisions. Most other companies in the area of demand management are nothing more than reporting systems with limited control capability. For example, Honeywell-Cannon (www.honeywellcannon.com) produces an energy management product called Yukon. Examine the following, taken from their website:

> Yukon is a complete set of software for metering and managing demand on the distribution system. It can be used by an energy service company to collect meter data and control peaks; by a distribution company to control power factor and monitor substations; or by corporate customers and meter data companies to aggregate and control load across multiple facilities.

It appears that we have a real demand management tool on hand here. However, further reading reveals that

> Yukon combines the real-time functions of SCADA [Supervisory Control And Data Acquisition] with revenue metering features and direct control.

Thus we see that there is no energy flow optimization happening within Yukon. Further reading of their product specifications reveals that the control that Yukon offers is limited to actions like starting and stopping generators, curtailing loads, etc, based on fixed rules.

Allen-Bradley Corporation (www.ab.com) provides a product similar to Honeywell-Canon's Yukon, called IAS. Their product also helps organize and visualize a company's electricity usage, but does not perform optimization, does not include distributed generation capabilities, and is an electricity controller, not a general energy controller. An important feature of IAS appears to be its ability to handle demand shifting and load shedding, as well as manage a priority list of demands to shed in the event of a load curtailment request. But most of these products are relatively new and the technical information regarding what they actually do is not made available.

Johnson Controls (www.jci.com) appears to offer customized solutions and consulting services along with targeted software and hardware solutions. They claim 118 years of experience in developing customized energy solutions for customers, negotiating difficult price contracts, all leading to a lower and more consistent energy bills. Their flagship energy product is Metasys, a web-front-end system for viewing your energy network, and application specific control solutions: HVAC, fire, lighting, and security. There is no mention of DG control, and no mention of optimizing energy flows.

These are the top products in the field, and even for them it is very difficult to understand from the product descriptions what they actually do. What is clear is that none of them are systems which take into account electricity tariff structure, demand forecasts, DG output dependent efficiencies, and the possible existence of renewable energies and storage in the network to make optimal control decisions. We turn to the literature for a better understanding of techniques for optimal demand control.

2.2.2 Literature review

Work by Constantopoulos and Larson [14, 15] addresses the general demand management problem under spot pricing. It is this work which provided the springboard for my own thesis, and thus I review it carefully here and point out the key differences.

For the consumer, spot pricing (which refers to electricity purchased from the spot market) amounts to hourly prices from the utility that are random, that is,

unannounced until some set time before they take affect. Therefore, by modeling spot pricing, Constantopoulos states, you handle "arbitrary electricity tariffs". More correctly, you handle any electricity tariff which is comprised of time dependent known or unknown prices. What you do not cover are tariffs such as volume discounts and demand charges, covered later in this thesis.

Constantopoulos presents a dynamic programming formulation for providing optimal demand control in the face of stochastic hourly prices and temperatures. He does not consider distributed generation at the customer's site. For these two reasons (no demand charge, and no DG), the loads decouple allowing one to solve each as its own subproblem. This decoupling, which is a central piece of their work, is not possible in networks with DG or when a demand charge exists.

A demand charge, which charges a customer based on peak usage over a month, means that for optimization, a customer must consider all of his loads simultaneously. If the optimization routine does not do this, it may choose to run all of the loads at the time of cheapest electricity prices, but then pay a huge demand charge due to the high peak load.

A user with onsite electricity generation needs to consider the coupled problem as well, because the generator's efficiency, which depends on usage level, and operating range, require the combined load profile for proper optimization.

For demands, Constantopoulos considers HVAC control and shiftable jobs. For each type of demand, he assumes an objective function which incorporates the cost of electricity as well as the utility of demand satisfaction. This is used for example to model the situation where high electricity prices mean that the user is willing to suffer a decrease in indoor comfort to save money. We preserve this multiple objective approach in the development of TOTEM. He directs considerable attention to reviewing how one might assess the economic/comfort tradeoff. This issue will continue to be a challenge until it is rigorously addressed with building occupancy experiments under RTP.

Constantopoulos presents a dynamic programming (DP) [6] formulation for the problem. He presents approximate DP solutions for real-time computational usage

of his model. DP is chosen as the modeling and solution approach due to stochastic pricing and stochastic weather. He shows that the forecasting horizon for price and weather is on the order of a few hours. That is, information about the weather and prices after that horizon are irrelevant for current decisions. In today's electricity market, customer's are not given random prices (there are some exceptions to this, but by and large, consumers object to random prices and as such the utilities take the burden of stochasticity upon themselves). The closest tariff to the spot pricing dealt with by Constantopoulos is fixed hourly prices which are revealed to the customer at 5 PM of the preceding day. Weather prediction, while ever elusive for days in advance, is very good for the near term of 1 day or less (see Chapter 8). Thus, as stochastic prices are rare in today's electricity market and near term weather forecasting is very accurate, the dynamic programming formulation is not necessary. Indeed, Constantopoulos' model becomes a linear program (with the appropriate modeling of the shiftable loads, which is contained within the present thesis) if the stochastics are removed. These considerations led me to adopt first a deterministic framework, which is later relax it to include stochastics. But rather than switch to a dynamic programming formulation, I add stochastics via stochastic programming (scenario-based), thus keeping the model as a linear mixed integer program. This implicitly takes advantage of the dramatic increase in computational power over the last 20 years, the time between when the Constantopoulos thesis was written and now, and also reflects a more palatable approach for facility managers, who plan based on scenarios rather than underlying stochastic processes. There is more discussion of the choice of scenario-based stochastic modeling in Section 4.2.

The final large difference between the work by Constantopoulos and my work is that I provide a model for multiple energy types rather than just electricity. This decision was made due to the growing presence of cogenerators, which generate electricity and usable heat simultaneously, and devices such as absorption chillers, which use steam as input to drive compressors for cooling demand. It also is important when certain demands are able to be satisfied by more than one energy type, for example heating may be done with electricity or with steam.

To summarize, the present work extends Constantopoulos' work by focusing on today's energy tariffs, including supply in addition to demand and storage, and finally, considering multiple energies coexisting in local energy networks. While I share the same philosophy as the Constantopoulos and Larson work – approach the energy usage problem with precise mathematical optimization – we differ in many of the model features and overall structure. My work generalizes their work to handle today's energy networks.

As HVAC controls offer the most promise for saving money in a non-invasive manner, most of the remaining work on local demand control deals with inside temperature and to some extent humidity control. A review of the approaches and requirements for customer adoption of these is given in [11]. Models for HVAC control fall into two broad categories, macro level control and micro level control. Macro control addresses only large scale decisions like turning a boiler or air conditioner on and off. When these works consider comfort level, they do so with simple temperature update equations and utility functions [15, 28]. Due to their modeling coarseness, they are relevant only as theoretical demonstrations, and not for useable control software.

HVAC control is extremely varied and complicated, with a large number of control options throughout an air handling distribution network [57]. Micro control refers to the control of all of these items and is usually handled by feedback control loops or simple thermostatic rules. However, it is possible to implement direct digital control (DDC) on each valve, heat exchanger, and pump in the system by a central processor [13]. Optimization at this level is desirable but extremely difficult and site specific due to the nonlinearities (in the mappings from controls to room conditions, and from controls to energy costs) of the underlying dynamics. If any intelligent control is attempted it is almost always in a local loop without global optimization considerations. System modeling at this level is too cumbersome, and system identification requires sensors and data collection throughout the network, neither of which usually exist. Systems that do propose solutions for this problem take either a simulation approach or a nonlinear systems identification approach. In each case, optimal control is achieved through direct search or some randomized search algorithm [63]. Build-

ing simulation is done with such packages as ESP-r, developed at the University of Strathclyde in Scotland, or EnergyPlus, developed by the Department of Energy of the United States. Both of these systems are extremely detailed building simulation packages, available at no charge.

A theoretical investigation comparing six types of neural networks used for HVAC control is found in [43]. Rather than search for a true optimal control, researches often reduce the HVAC problem to determining the optimal time to reset the thermostats in early morning hours in order that by the time workers arrive to work, the building is re-cooled to the target level, as is done therein.

2.3 Supply management

Supply management refers to controlling a local user's energy supply devices, such as boilers and generators. Certain devices, such as electric chillers, may be viewed as a supply and a demand problem, since they are supplied by electricity, but they ultimately fulfill an non-electric energy demand. We view chillers as a supply decision.

There are a few reasons why local supply control has not been addressed and implemented systematically:

- Heating oil is inexpensive and therefore there has not been incentive in developed countries to conserve.
- Fuel costs are fixed per gallon and therefore there is no TOU or RTP pricing incentive to optimize against.
- One needs software and local area information networks to enable centralized control, and given the above bullet points, there is not financial incentive to install these.

For newly installed boilers, efficiency gains are in the form of boiler design. Control algorithms are nothing more than set-points: when the water level or temperature in the hot water tank drops below a certain level, fire the burners. In much of Europe,

since residents experience a time-of-use pricing plan, it is common to heat water at night when it is cheaper. Real-time pricing and increased usage of local generation will certainly increase the demand for some control sophistication on the supply side.

The notable exception to the lack of intelligent supply side control is of course the supply side industry, the utilities. This is where we turn our attention for product and literature review. The utilities, every day, need to solve the problem of choosing which of their generators to have on for the day (termed the unit commitment problem), and what level to run each one that is on (chosen based on efficiency, power quality, and voltage stability).

2.3.1 Commercial product review

As described above, due to the infancy of the field of local generation and due to the lack of hourly pricing, high level supply control software (i.e. not the microlevel control of frequency and power matching that every generator comes with) is nonexistent. Power companies do however solve the supply problem, and so our product review just mentions two companies offering a commercial product for the unit commitment and economic dispatch problems. Most utilities use standard techniques to solve these, see for example [33]. Some companies offer these standard solutions implemented as software, for example Power Optimisation Ltd, out of England and EnWorkz, out of Texas.

2.3.2 Literature review

The problem of optimally controlling a large number of supply-side generators has been tackled for many years in the power generation literature under the title 'the unit commitment problem'. Unit commitment refers to committing a single unit, or generator, for a period of power production. When the number of generators onsite is large, for example, a utility may own hundreds of these, even the deterministic problem may be difficult. The stochastic unit commitment problem is treated by many authors, with much effort devoted to the efficient computation of the solution. Takriti

et al. [54] provide a thorough approach as well as many references to related works. They use a Langrange multipliers to decouple the generators, and solve each decoupled Langrangian with DP. Langrange multipliers are updated using cutting plane approximations to the dual function. Bos *et al.* [9] provide another Langrangian relaxation approach for a unit commitment model which includes heat sources (boilers) and heat storage (hot water tanks) in addition to the electrical generators. Allen and Ilić [1] consider the unit commitment problem for profit maximization in the context of stochastic external pricing, and use a DP approach as well.

Other supply management literature consists of papers dealing with specific installations rather than general networks. A deterministic model for daily control of a hospital's power plant (chilled water, turbine, absorption chillers, boilers) is presented in [58]. The time step is on the order of half a day, with deterministic yearly data used to calculate approximate yearly costs under optimal operations. They use a brute force search approach to find a good solution, and then a local quadratic programming approach from that solution, implemented in Matlab. The demand curves are taken as fixed.

Bakos [4] presents a continuous time optimal control model for controlling the auxiliary heater of a solar-heated residence in Northern Greece. This model is written as a set of linear differential equations which capture the solar heat intake and heat exchanges in a simplified two-dimensional model of the house. The author applies the Hamiltonian approach to deterministic continuous time control to determine the optimal control for the single control parameter of the auxiliary heating unit. This paper stands out for its clean mathematics, but its practical extension to complex networks is unlikely.

Genetic algorithms are a popular choice for optimizing energy networks since they handle arbitrary functional forms, including non-differentiable and discontinuous functions. Manolas *et al.* [44] provide an investigation using a genetic algorithm to find good operating strategies for an industrial cogeneration system. Their objective function is to maximize electrical output while meeting the known steam loads for the day.

Two important works attempting to tackle the supply control problem for general networks are DENNIS [49] and DER-CAM [37]. DENNIS, an acronym for "Distributed Energy Neural Network Integration System", is involved with control of both fossil and renewable DG sources. This system is currently being tested at a few sites in the northeast of the United States. DER-CAM, an acronym for "Distributed Energy Resources Customer Adoption Model", is produced by the Lawrence Berkeley National Laboratory. This tool is made for displaying to customers how their energy network would look with DG added to it, based on their historical load profiles and weather patterns. They call their local electricity network a micro-grid, and advocate a small amount of consumer load aggregation to justify the purchase of DG. These two projects present similar approaches to optimal control as TOTEM, but they contain DG control only (no load control).

2.4 Storage management

While batteries might be the most important candidate for DG energy networks, most of the literature on storage control deals with more commonly used storage configurations such as chilled storage for space cooling and hydro-storage for large scale power generation. Nevertheless, because all storage is the same from a mathematical perspective, the applications are irrelevant from the point of view of mathematical techniques employed. What we do have to concern ourselves with beyond the modeling are the physical characteristics of the storage devices used (charging and discharging rates, cycling considerations, depth of discharge considerations). [34] gives an overview of those practical considerations for the batteries, in particular the lead-acid battery. An overview for selecting the type of electrical storage system for various applications, based on storage amount and response time, is given in [36].

2.4.1 Literature review

The literature on controlling devices for large scale energy storage is almost exclusively limited to very large energy storage: hydro-storage for power generation. Fortunately,

this type of application is similar in feel to energy storage in a distributed generation local network. Due to the large number of hydro-reservoirs in typical hydro applications, Lagrangian decomposition approaches like those discussed in Section 2.3.2 are common. For example, [31] presents a stochastic scenario based model which, upon Langrangian relaxation, decomposes into sub hydro-storage models that can be solved using a network flow model. Water pump scheduling, which is considered storage control since elevated water is a form of energy storage, is addressed in [10]. They present a genetic algorithm to solve their non-linear model.

A fuzzy control approach is provided in [38] for controlling a flywheel electric storage system used in a diesel generator and wind turbine network. The authors choose a fuzzy framework to break up the real time wind power and flywheel speed into three levels: small, medium, and large. The objective is specific to the application: reduce power fluctuations on the diesel generator. This represents a specialized off-grid application where wind and storage exist only for diesel output power smoothing.

Henze [32] uses dynamic programming to provide optimal control for chilled storage under uncertainty in real-time electricity pricing. He also includes uncertainty in terms of cooling loads, cooling-electrical loads, and weather. He compares his DP result against two commonly used cool storage strategies, and shows that even with relatively inaccurate forecasts, the DP approach does significantly better. The DP approach employed uses forecasts of future loads and prices, rather than more detailed probabilistic information. Cooling loads and other base loads are predicted using the simple bin method of ASHRAE [35], which is a manual method of calculating energy loads in buildings.

Another example of dealing with storage under real-time pricing is given by Daryanian *et al.* [16], in which they derive a simple linear programming formulation for electricity storage under known prices and known demands. Their simple model reduces to a capacitated min-cost flow problem, which is solved very fast.

2.5 Combined supply and demand management

2.5.1 Literature review

A very broad overview of a system similar to TOTEM in scope and application is presented in [27]. They focus mainly on high level system description and how the modules of forecasting and control work together. This paper is cited since it shows that other groups are thinking about the total energy management problem, but their report is at too high a level for any further comparison.

Entchev [20] provides an overview of a stochastic differential equation based technique for controlling a residential fuel cell combined heat and power (CHP) system. The optimal control includes such features as minimizing fuel cell cycling for long term efficiency, and HVAC management including air handler blower control for warm air heating, a common feature in newly built North American homes. Neural networks are used for system identification, and control is implemented with a fuzzy-based control scheme.

Chapter 3

Mathematical model for total energy management

In this chapter, we propose a tractable mathematical framework for controlling an entire local energy network. The model incorporates any possible combination of energy types (maybe for example, steam, cold water, and electricity), and models energy supply (distributed generation and grid purchases), energy demand units, and energy storage units. Each unit is called a node, linked by (possibly flow constrained) arcs

We adopt a mathematical programming approach to solving the problem of customer power generation, grid and fuel purchases, load control, and storage strategies. In such a comprehensive model, it is necessary to make assumptions about the nature of such networks to keep the model solvable. Later, we consider relaxations of many of the restrictions. The base model is a deterministic, discrete time, linear system, which results in a linear mixed integer program. The model is written with great generality, not assuming any particular components in the user's setup or any fixed arrangement of nodes. This modeling approach is similar to the 'object oriented linear programming' approach put forth by Berman *et al.* [5] for a network problem involving scheduling workflow in high volume factories.

AMPL, a modeling language for mathematical programming [23], is highly suited for such an object oriented approach, in that the abstract model appears in one file,

and the specific instance for each user is created as a separate file. This way, the end user only needs to supply data, and is shielded from the inner workings of the math program. While the abstract model, which is presented in this chapter, is all-inclusive, a typical user's instantiation, in the form of an AMPL model file, will not be, but this is handled smoothly by AMPL, which automatically detects this and only generates the necessary equations.

We provide a general abstract model up front, and the user can define his network using a drag-and-drop interface. After being defined, that model is converted to a data file to be read by AMPL. Additionally, AMPL allows users to use custom solvers, important if for example certain non-linearities are present which cannot be handled by traditional solvers. CPLEX, by Ilog, Inc is the industry standard for mathematical optimization, but we note that LPSOLVE, a free solver that works with AMPL, can handle linear MIPs and thus is suitable for TOTEM as well. More detail on the user interface and the use of AMPL is provided in Appendix C.

3.1 A comment on real versus reactive power control

Regarding the control of electric generators, TOTEM handles only the control of real power, as opposed to reactive power. A lower level of generator control, covered in books such as Power Systems Analysis by Saadat [52], deals with issues arising in AC networks serving real and reactive (i.e. inductive and capacitive) power including voltage stability, power quality and frequency regulation. TOTEM provides the next level up of operational control: macroscopic decisions regarding which generators are on and at what levels. With these decisions made, control power quality is assumed to be taken care of by existing systems.

3.2 Model description and definitions

The sets, variables and parameters of the model are described in Tables 3.1–3.2. The node is the basic entity in the model. By enforcing balance equations at each node, we enforce global network consistency. The node-based model is the key to to the flexibility of TOTEM.

Set	Description	Example
N_V	General energy conversion nodes	Fuel-based generators
N_{IO}	General input-output nodes	aggregators
N_U	All nodes with outgoing arcs having a u (on/off) variables	
N_R	Renewable energy sources	Wind turbine
N_F	Fuel source nodes	Oil
N_D	All demand nodes	
N_D^{sl}	Time shiftable loads	Laundry
N_D^A	Analog controllable nodes	Air-conditioning
N_D^B	Base (uncontrolled) load	Lighting
N_S	Storage nodes	Flywheel
N_{SINK}	Sink nodes	Sell to grid
N	All nodes	
$I(i)$	All nodes incident to node i	
$O(i)$	All nodes connected to node i's ouputs	
STATES	All the tracked states	Hot water level
S_P	States with penalties	
S_C	States with constraints	
UGROUP_MIN_ NODES	Groups of nodes with restriction on min number on	Boilers
UGROUP_MAX_ NODES	Groups of nodes with restriction on max number on	Grid buy/sell

Table 3.1: Sets for the base model.

Variable	Description
x_{ijt}	Flow (appropriate units) from i to j during time interval t
y_{it}	Storage level for storage node i at time t
d_{it}	Demand at node i at time t
z^a_{it}	Control for analog node i at time t
z^{sl}_{ij}	Binary variable for starting shiftable load i at time j
S_{kt}	Component k of state vector at time t
δ_{kt}	Deviation of state k from its target value
u_{ijt}	Binary variable indicating flow from node i to j
Parameter	**Description**
T	Time horizon
c_{it}	Unit cost from fuel source i at time t
G_{in}	Multiplier for node i output with input from node n
V_{inj}	Conversion slope for node i output to node j with input from node n
B_{inj}	Conversion intercept for node i output to node j with input from node n
r_{it}	Level of energy time t provided by renewable source i
b_{it}	Base load profile for demand node i at time t
a_i	Multiplier converting analog control z^a_{it} to flow demanded
$M(i)$	Maximum allowable start time for shiftable demand node i
l_{ijt}	Energy drawn by shiftable load i at time t if started at time j
lower_{ij}	Lower bound for i to j flow
upper_{ij}	Upper bound for i to j flow
MinUpTime_i	Minimum up time for node i (technically, for the outgoing arcs)
MinDownTime_i	Minimum down time for node i
UGmin_i	Minimum number of devices operating in UGROUP i
UGmax_i	Maximum number of devices operating in UGROUP i
I_k	Initial state, component k
T_{kt}	Target state level, component k, time t
γ_{kt}	Cost parameter for state deviation of component k, time t
$E^{\min}_{kt}$	Lower state bound, time t
$E^{\max}_{kt}$	Upper state bound, time t
$S^{\min}_i$	Minimum storage level at storage node i (similar for max)
S^{charge}_i	Maximum charge rate storage node i (similar for discharge)
W_{it}	"Weather" vector (exogenous time dependent data), ith component
L_W	Length of the weather vector
H_{kj}	Linear state update matrix: state element k updated by state element j

Table 3.2: Variables and parameters for the base model.

The most basic entities in the model are the nodes, which represent any energy device in the system, whether it be a production node, storage node, aggregation or splitting node, or end use node. Nodes are connected by directed arcs, and the arc flows are the main variables in the model.

We consider a discrete time system constant through each time interval, the system only changing at the discrete steps. The flow from node i to node j for time $\tau \in [t, t+1)$ is given by x_{ijt}. We restrict the flows to be non-negative. We also restrict the number of arcs between each pair of nodes to be at most 1. This is done without loss of generality: a circular flow between i and j can be modeled by splitting either node i or j, and forming a three node cycle. It is not necessary to specify the type of quantity that is flowing over the arcs because for each network instance the balance equations parameters will be input by the user who knows what is flowing on the arcs. For example, when a user models a generator, he selects the units of fuel flow (gallons per minute) and then specifies the appropriate conversion equation to go from fuel flow to electricity output. Note, we make no distinction between the concepts of power (energy per unit time) and energy in this section, and in fact use them interchangeably (since we have a fixed time step for a model, power and energy are in a fixed proportion). When a model is created, this consistency must be ensured.

3.3 Balance equations

Each node i in the model is associated with balance equations, which relate the output to the input. In the next sections, we describe the balance equations for each type of node in the network.

3.3.1 Conversion nodes

We assume an affine input output relationship (i.e. $y = mx + b$) for fuel-based generators and other general input-output nodes (this assumption is relaxed in Chapter 4). The affine relationship captures generators which have increasing or decreasing efficiency with running level. Fuel-based generators include electricity generators such

as reciprocating engines, combustion turbines, and fuel cells, heating units such as steam generators and boilers, and cooling units such as chillers. We assume that a generator can be powered by multiple fuels, and each fuel has its own affine input-output relationship. We also allow for the generator to have multiple outputs, such as an engine which produces electricity and heat simultaneously. Flow out is given by:

$$x_{ijt} = \sum_{n \in I(i)} \left(V_{inj} x_{nit} + B_{inj} u_{nit} \right). \tag{3.1}$$

In practice, generators only use one fuel at a time, and so this equation is combined with a restriction equation, see Section 3.5 below on UGROUPS.

3.3.2 Input-output nodes

In general, there will be nodes in the network which are not classified as fuel-based generators, but have similar behavior: they take some input flow and convert it to an output flow. The standard uses of IO nodes is aggregation and splitting.

The differences between these nodes and conversion nodes of the previous section are that these nodes handle single energy types (energy type in = energy type out), there is no intercept term, and there is only one equation for each such node, as opposed to one equation for each outgoing arc.

The loss factor at node i from input source n is given by the parameter G_{in}, leading to the following expression:

$$\sum_{n \in I(i)} G_{in} x_{nit} = \sum_{j \in O(i)} x_{ijt}. \tag{3.2}$$

3.3.3 Renewable generation nodes

Renewable energy sources are modeled as nodes which provide a known power load over the course of the optimization horizon. As such, the balance equation for them

simply equates the total renewable power to the sum of the outgoing power flows:

$$r_{it} = \sum_{j \in O(i)} x_{ijt}. \tag{3.3}$$

Here, r_{it} is a parameter specifying the power available from renewable source i at time t.

3.3.4 Storage nodes

The balance equations for storage nodes are similar to Equation 3.2, but since the node can be storing or emptying energy, it is no longer true that at any given time t input equals output. We introduce a new variable y_{it} for the storage level of storage node i at time t. The balance equation becomes:

$$\sum_{j \in I(i)} G_{ijk} x_{jit} = \sum_{j \in O(i)} x_{ijt} + y_{i(t+1)} - y_{it}, \tag{3.4}$$

which states that the input equals the output plus the change in storage level. Although we have not yet described the modeling of states, at this point it is worth mentioning that the storage level could have also been modeled as a state, but due to the simplicity of the storage level, it is more transparent this way.

A subtle point arises when optimizing networks which have storage over a daily horizon. If the final battery level is unconstrained, it will be driven to its lower bound to maximize "free" energy use. In general, this is probably not the optimal policy for two reasons. One is that deep cycling of batteries, as laptop users know well, reduces the performance of a battery greatly over time. Car batteries last so long because they are never deep discharged. It is difficult to quantify a daily penalty for depth of discharging so this was left out of the objective function and only included as a constraint. The second reason that emptying a battery at the end of a time horizon is probably not optimal is because it is an merely an artificial remnant of the finite horizon of the TOTEM model. There are a couple ways to alleviate this. The simplest is to constrain the final storage level to be greater than or equal to the initial

storage level. The more correct way to handle the final storage level is to penalize it via a linear term in the cost function, such as $P_i \cdot (S_i^{\max} - y_{iT})$ where P_i is the cost penalty per unit of energy storage you are away from the maximum storage capacity $S^{\max}$.

3.3.5 Demand nodes

For each demand node i in the model we represent the amount of power required at time t by d_{it}. For base load nodes, meaning nodes that use power and cannot be controlled, this amount is set as a parameter. For other nodes the demand level may depend on control decisions.

Base demand nodes

These are nodes which draw power of some type but are not controllable in the model. At MIT for example, if an energy management system such as TOTEM was installed today, it could only control the HVAC in approximately half of the buildings. The other half of the buildings are self controlled, and thus to the central facilities department at MIT, they are base uncontrollable loads.

The amount of power required by base demand node i at time t is given by the parameter b_{it}. The balance equation is then

$$d_{it} = b_{it}. \tag{3.5}$$

Analog control nodes

This type of demand node is controllable in the sense that at each time t, the levels can be adjusted which in turn alter the amount of power required by the node. The term analog is used to help picture this idea of setting the level, but the control could be discrete 0-1 control as well. The demand is assumed to be a linear function of the control z_{it}^a:

$$d_{it} = z_{it}^a a_i. \tag{3.6}$$

Shiftable demand nodes

This type of demand node is fundamentally different from the analog control demand nodes. Rather than adjusting the level of the load throughout the day, for this type of load the only decision is which hour to start the load. This is the type of node used in job scheduling problems. If we let the control variable be the time at which the load started, the resulting model would be nonlinear. The easiest way to see this is to plot the demand at a certain time versus this decision variable. Even for simple load profiles, the resulting curve will be nonlinear (and include jumps discontinuities).

To avoid a nonlinear model, we introduce a binary variable z_{ij}^{sl} for shiftable load i and possible start time j. We also introduce the load profile curve l_{ijt}. This parameter specifies the load drawn by node i at time t if started at time j.

Given that we are modeling finite horizon problems, depending on the length of the load cycle, each shiftable load will also have a maximum allowable start time given by $M(i)$. The demand drawn by shiftable load i at time t is given by:

$$d_{it} = \sum_{j=1}^{M(i)} l_{ijt} z_{ij}^{\text{sl}}. \tag{3.7}$$

To enforce that each load gets started once and only once, we have the following coupling constraint for the z variables:

$$\sum_{j=1}^{M(i)} z_{ij}^{\text{sl}} = 1 \tag{3.8}$$

Supply the demand

Now that each demand node has the consistent variable d_{it} assigned to it, we can use the same supply-demand equation for all of the demand nodes (in certain applications, equality might be used):

$$\sum_{n \in I(i)} x_{nit} \geq d_{it}. \tag{3.9}$$

3.4 Allowable operating range equations

This section presents the inequalities used to restrict storage and generation devices to operate within their allowable ranges.

3.4.1 Storage ranges

A storage device will have maximum and minimum storage levels, enforced by the following:

$$S_i^{\min} \leq y_{it} \leq S_i^{\max} \tag{3.10}$$

A storage device will also have maximum charge and discharge rates, enforced by Equations 3.11 and 3.12 respectively:

$$\sum_{j \in I(i)} x_{jit} \leq S_i^{\text{charge}} \tag{3.11}$$

$$\sum_{j \in O(i)} x_{ijt} \leq S_i^{\text{discharge}} \tag{3.12}$$

3.4.2 Fuel based generator and fuel flow ranges

Generators need to be dealt with a little more carefully than the storage devices, regarding operating ranges. Specifically, a generator can either be off or be on within a certain range away from 0. This non-convex space can be handled by introducing binary variables u_{ijt} indicating the state of the flow on i to j. We then restrict the operating range with:

$$\text{lower}_{ij} u_{ijt} \leq x_{ijt} \leq \text{upper}_{ij} u_{ijt}. \tag{3.13}$$

These binary variables u are also helpful in modeling startup and shutdown costs for generators as well as cycling constraints (see Section 3.6).

Equation 3.13 can also be used to restrict fuel flows. This modeling convenience is used for example to cap the electricity flow from the grid (modeled as a fuel node) in an effort to reduce the demand charge (see Chapter 5).

3.4.3 Analog control range

We represent the allowable ranges for analog control nodes with the following:

$$z_{it}^{a} \in Z_i \tag{3.14}$$

This is meant to represent range restricted to an interval of real values, a binary choice, or a discrete set of possibilities (which can be handled in a linear fashion by including supplementary binary variables).

3.5 UGROUPS

In order to handle constraints on the number of devices operating simultaneously, we introduce the UGROUPS. The name refers to the usage of the on-off variable u_{ijt} associated with flow from i to j. These groups are used, for example, if a user has a reliability constraint that forces him to have at least 2 of his boilers running at all times. This is an example of a group called UGROUPS_MIN. A typical use of the set UGROUP_MAX is grid buying and selling. Utilities who allow distributed generation owners to sell back to the grid generally allow either selling to the grid or buying from the grid, but not both simultaneously. Another use of a UGROUP_MAX group is to restrict the number of fuels a generator can use simultaneously to 1.

Let UGROUP_MIN_NODES$_n$ be a user defined group of nodes. We index these groups to allow a user to define multiple UGROUPS. Let UGmin$_n$ be the minimum number of nodes from UGROUP_MIN_NODES$_n$ on at any one time. To enforce this, we add the following inequality to the model:

$$\sum_{(i,\ j)\in \text{UGROUP\_MIN\_NODES}_n} u_{ijt} \geq \text{UGmin}_n \tag{3.15}$$

With similar parameters and sets introduced for UGROUP_MAX_NODES$_n$, we have:

$$\sum_{(i,\ j)\in \text{UGROUP\_MAX\_NODES}_n} u_{ijt} \leq \text{UGmax}_n \tag{3.16}$$

3.6 Generator cycling constraints

A set of constraints that makes the optimal generator operating schedule especially difficult is that generators often have minimum up and down times, reflecting the desire to keep cycling down. Using the binary on-off variables already introduced, we can write the following linear constraints to handle a minimum up time requirement:

$$u_{ijt} - u_{ij(t-1)} \leq u_{ij\tau} \quad \begin{array}{l} \tau = (t+1) \ldots \min\{t + \text{MinUpTime}(i) - 1, T\} \\ t = 2 \ldots T \end{array} \tag{3.17}$$

These constraints work in the following way. For every time t, the left hand side of Equation 3.17 is 1 only if $u_{ijt} = 1$ and $u_{ij(t-1)} = 0$, which indicates that the generator was just turned on. Therefore, the generator needs to be on for the next $\text{MinUpTime}(i) - 1$ time units, which is enforced by $u_{ij\tau}$ equalling 1 as a result of the inequality.

Similarly, we handle a minimum down time requirement with:

$$u_{ij(t-1)} - u_{it} \leq 1 - u_{ij\tau} \quad \begin{array}{l} \tau = (t+1) \ldots \min\{t + \text{MinDownTime}(i) - 1, T\} \\ t = 2 \ldots T \end{array} \tag{3.18}$$

If the generators are in a particular state at the beginning of the optimization period, it is simple to append these constraints with additional restrictions on the initial u_{ijt} variables.

3.7 State equations

At this point we have described a framework to handle flows in an energy network in a fairly general way. We have abstracted the concept of energy to handle networks which simultaneously include electricity, steam, etc.

One purpose of such networks is to alter the environment around them. This includes such entities as the lighting conditions, the indoor humidity, air quality, and temperature. These demands are represented as analog control nodes. In this

section we develop a general modeling approach for keeping track of the state of the environment that we are controlling. Once the state is successfully tracked, we can constrain it to fall between user-defined bounds, or add a penalty cost to the objective function for deviations from the target state.

As a motivating example for the general state model, let us consider a standard application of state control, the indoor temperature of a room, controlled by an air conditioning unit in a summer month. Here, let S_t, the state of the room at time t, be temperature. The temperature at the next time step S_{t+1} will depend on the current temperature S_t, the external temperature W_t, and the control applied to air conditioning unit, z_t. In keeping with our linear model assumption, we assume the following temperature update equation:

$$S_{t+1} = \epsilon S_t + (1 - \epsilon)(W_t - \alpha z_t). \tag{3.19}$$

To generalize this, we begin by vectorizing the state variable S_t. We also vectorize the external influence vector (the outside temperature in the motivating example), W_t.

Let H be the data which specifies the linear update relationship. Loosely, we have

$$S_{t+1} = H \times (S_t,\ W_t,\ z_t) \tag{3.20}$$

We decompose H into three submatrices, for state, weather, and nodal control: $H = (H^s,\ H^w,\ H^z)$. Then, in full detail, we have

$$\begin{aligned} &S_{k(t+1)} = \textstyle\sum_{j=1} H^s_{kj} S_{jt} + \\ &\textstyle\sum_j H^w_{kj} W_{jt} + \\ &\textstyle\sum_j H^a_{kj} z^a_{jt} \qquad\qquad \forall k \in \text{STATES}, \end{aligned} \tag{3.21}$$

where the sum over j is done for state components in the first summation, weather components in the second, and nodal controls in the third.

3.7.1 State bounds and state deviations

The TOTEM model offers two ways to force the states to fall within acceptable bounds. The easiest implementation is to simply provide bounds for the state to fall in. This is done in the following (note we let the state bounds vary with time, to handle situations such as time varying building occupancy):

$$E_{kt}^{\min} \leq S_{kt} \leq E_{kt}^{\max} \tag{3.22}$$

This approach is simple to implement, but it fails to capture the economic trade-off between comfort level and cost. Assuming we are able to economically assess deviations from a target state, we need to track those deviations. We provide a simple model for this, based on the absolute value of the deviation. Let T_{kt} be the user-defined target level for state component k at time t. Let δ_{kt} be the deviation:

$$\delta_{kt} \equiv |T_{kt} - S_{kt}|. \tag{3.23}$$

The following inequalities, coupled with the objective function, Equation 3.27, will enforce relationship 3.23, by driving down δ_{kt} such that one of the inequalities will be tight:

$$\begin{aligned} \delta_{kt} &\geq S_{kt} - T_{kt} \\ \delta_{kt} &\geq T_{kt} - S_{kt} \end{aligned} \tag{3.24}$$

Generalizations of the absolute value function (i.e. any piecewise linear convex function) can be handled also (see [7] for example).

3.8 Storage and state initializations

Initial storage level for storage node i is given by I_i:

$$y_{i1} = I_i. \tag{3.25}$$

Likewise, initial value for state component k is given by I_k, and we have:

$$S_{k1} = I_k \tag{3.26}$$

3.9 External prices

We assume known prices on arcs emanating from fuel source nodes N_F. Therefore the grid is modeled as a fuel source node. The unit fuel cost from fuel source i at time t is given by c_{it}. This covers real time pricing (RTP) and time-of-use pricing (TOU). Other rate structures not covered by this form are discussed later in this thesis.

3.10 Objective function

Our objective is to minimize the cost of procuring energies to fulfill our daily demands, while keeping the system state within the range tolerances and penalizing for state deviations from target levels. In this objective function, we implicitly assume that the user is able to quantify the tradeoff between comfort and money spent, a tradeoff represented by the parameters γ below.

Let c_{it} be the per unit flow cost from fuel source node i at time t. Let γ_{kt} be the cost multiplier for the deviation of state variable S_{kt} from its target level T_{kt}. We then have the following cost to minimize:

$$\sum_{i \in N_F} \sum_{j \in O(i)} \sum_{t=1}^{T} c_{it} x_{ijt} + \sum_{k \in S_P} \sum_{t=1}^{T} \gamma_{kt} \delta_{kt} \tag{3.27}$$

where S_P are the state components in the user-specified state penalty set.

If one desires to include generator startup costs in the objective function, we define a binary variable s_{jt} which equals 1 if generator is started at time t and 0 otherwise, enforced by the inequality constraint $s_{jt} \geq u_{ijt} - u_{ij(t-1)}$ (i is a fuel source input node). This term with a startup cost multiplier is then added to the objective function.

Shiftable loads can be costed as well by associating different costs with different

start times. If this is desired then a term like $\sum_{j=1}^{M(i)} c_{ij}^{s} z_{ij}^{\mathrm{sl}}$ where c_{ij}^{s} is the cost of starting load i at time j.

3.11 Full mathematical model

The formulations described above give rise to the following linear mixed integer math program, listed on the next two pages.

minimize	$\sum_{i \in N_F} \sum_{j \in O(i)} \sum_{t=1}^{T} c_{it} x_{ijt} \quad +$	$\sum_{k \in S_P} \sum_{t=1}^{T} \gamma_{kt} \delta_{kt}$
subject to		
(conversion)	$x_{ijt} = \sum_{n \in I(i)} (V_{inj} x_{nit} + B_{inj} u_{nit})$	$\forall i \in N_V,\ j \in O(i),\ t$
(io)	$\sum_{n \in I(i)} G_{in} x_{nit} = \sum_{j \in O(i)} x_{ijt}$	$\forall i \in N_{IO},\ t$
(renewable_equality)	$r_{it} = \sum_{j \in O(i)} x_{ijt}$	$\forall i \in N_R,\ t$
(storage_equality)	$\sum_{n \in I(i)} G_{in} x_{nit} = \sum_{j \in O(i)} x_{ijt}$	$\forall i \in N_S,\ t = 1 \ldots$
	$+ y_{i(t+1)} - y_{it}$	$T - 1$
(base_demand)	$d_{it} = b_{it}$	$\forall i \in N_D^B,\ t$
(analog_demand)	$d_{it} = z_{it}^a a_i$	$\forall i \in N_D^A,\ t$
(analog_range)	$z_{it}^a \in Z_i$	$\forall i \in N_D^A,\ t$
(shiftable_demand)	$d_{it} = \sum_{j=1}^{M(i)} l_{ijt} z_{ij}^{\mathrm{sl}}$	$\forall i \in N_D^{\mathrm{sl}},\ t$
(start_each_sl)	$\sum_{j=1}^{M(i)} z_{ij}^{\mathrm{sl}} = 1$	$\forall i \in N_D^{\mathrm{sl}}$
(supply)	$\sum_{n \in I(i)} x_{nit} \geq d_{it}$	$\forall i \in N_D,\ t$
(range_constraint)	$\mathrm{lower}_{ij} u_{ijt} \leq x_{ijt} \leq \mathrm{upper}_{ij} u_{ijt}$	$\forall i \in N_U,\ j \in O(i),\ t$
(min_up_time)	$u_{ijt} - u_{ij(t-1)} \leq u_{ij\tau}$	$\forall i \in N_U,\ j \in O(i),$ $t = 2 \ldots T,\ \tau = (t+$ $1) \ldots \min\{t + \mathrm{Min}$ $\mathrm{UpTime}(i) - 1, T\}$
(min_down_time)	$u_{ij(t-1)} - u_{it} \leq 1 - u_{ij\tau}$	$\forall i \in N_U,\ j \in O(i),$ $t = 2 \ldots T,\ \tau = (t+$ $1) \ldots \min\{t + \mathrm{Min}$ $\mathrm{DownTime}(i) - 1, T\}$
(ugroup_min)	$\sum_{(i,\ j) \in \mathrm{UG\_MIN\_NODES}_n} u_{ijt} \geq \mathrm{UGmin}_n$	$\forall n \in 1..\mathrm{nUGMin},\ t$
(ugroup_max)	$\sum_{(i,\ j) \in \mathrm{UG\_MAX\_NODES}_n} u_{ijt} \leq \mathrm{UGmax}_n$	$\forall n \in 1..\mathrm{nUGMax},\ t$

(3.28)

$$
\begin{array}{lll}
\text{(state\_initialize)} & S_{k1} = I_k & \forall k \in \text{STATES} \\
\text{(storage\_initialize)} & y_{i1} = I_i & \forall i \in N_S \\
\text{(state\_update)} & S_{k(t+1)} = \sum_{j=1} H^s_{kj} S_{jt} + & \\
& \sum_j H^w_{kj} W_{jt} + & \\
& \sum_j H^a_{kj} z^a_{jt} & \forall k \in \text{STATES},\ t \\
\text{(state\_deviation1)} & \delta_{kt} \geq S_{kt} - T_{kt} & \forall k \in S_P,\ t \\
\text{(state\_deviation2)} & \delta_{kt} \geq T_{kt} - S_{kt} & \forall k \in S_P,\ t \\
\text{(state\_range)} & E^{\min}_{kt} \leq S_{kt} \leq E^{\max}_{kt} & \forall i \in S_C,\ t \\
\text{(storage\_range)} & S^{\min}_i \leq y_{it} \leq S^{\max}_i & \forall i \in N_S,\ t \\
\text{(storage\_charge)} & \sum_{n \in I(i)} x_{nit} \leq S^{\text{charge}}_i & \forall i \in N_S,\ t \\
\text{(storage\_discharge)} & \sum_{j \in O(i)} x_{ijt} \leq S^{\text{discharge}}_i & \forall i \in N_S,\ t \\
\text{(non-negativity)} & x,\ y,\ z,\ d \geq 0 & \\
\text{(analog\_range)} & z^a_{it} \in Z_i & \\
\text{(binary)} & u,\ z^{\text{sl}} \text{ binary} &
\end{array}
\tag{3.29}
$$

3.12 Opportunities for TOTEM

For TOTEM to be a worthwhile addition to an energy network, the network needs a few key features. We sketch by example configurations suitable or not suitable for TOTEM.

Consider a grid connected network without storage and without dispatchable generation. Money-saving opportunities are restricted to load control, notably HVAC control and job scheduling. However, without a pricing plan which rewards load shifting, such as TOU, RTP, or any tariff including a demand charge, load control is not necessary. Even if renewable energy exists suppling a small fraction of the total load, such as in the proposed design for the Freedom Tower, it does not change the

above story. One needs some ability to control loads together with time-dependent grid prices will make TOTEM useful.

Supply-side management without load control is interesting when devices have minimum up and down constraints and when there are multiple generators of similar characteristics, making the optimal blend of them not clear. But in settings where the generation devices are much different from one another and it is obvious which ones to use given the various loads, TOTEM does not offer much.

Adding storage either in the case of supply-side generation with operationally limited generation devices or with time-dependent grid prices also makes TOTEM useful. Remote village settings often have renewable sources, storage, and backup dispatchable generation such as diesel, making them strong candidates for TOTEM optimization. However, when storage is oversized TOTEM is less useful.

The networks which make the best use of TOTEM optimization are those which include as many of the features discussed above: inflexible supply-side generation, storage, real-time pricing, and load control. The synergistic effects of being able to simultaneously exercise supply side and demand side control are discussed next.

3.13 A note on synergy

TOTEM is designed to handle systems where a user has control over supply and demand, although in most applications to date, usually only one of the two is controlled, and in cases where both supply and demand are controlled, this happens by independent systems that do not communicate with each other. We would like to know in what types of energy configurations will there be a noticeable advantage to controlling demand and supply with a single integrated control system such as TOTEM.

Demand side control involves essentially two techniques: load shifting and load dumping. Load shifting encompasses the obvious case of rescheduling loads, done in order to take advantage of TOU rates, and the subtler case of HVAC control, for example. In intelligent HVAC control, a building cooler might be turned on prematurely in order to take advantage of a cheaper rate, even though the building

actually requires cooling at some time in the near future instead. Thus, this is a version of load shifting.

Load dumping is opting not to perform a particular load operation at all, or shifting it far enough into the future to make it invisible to local optimal control.

Symmetrically, we can organize supply side controls into two types: storage and self generation. To see the symmetry, notice that storage essentially moves the supply in time (so it is analogous to load shifting), while generation provides power otherwise not available (which is the same result as load dumping). The synergy question is, under what conditions is it useful to be able to pull all of these levers in coordination? We would like to determine if there are instances where the amount of savings realized if both DSM and SSM are implemented together is far more than if either are implemented alone.

We will start with some extreme examples where without both DSM and SSM there are no savings.

- Self generation available, but not economically viable to run the generator on a given day (without shifting, load is not otherwise high enough or consolidated enough).

- Load shifting available, but flat rate and no demand charge, so no benefit from load shifting. Requires self generation to take advantage of load shifting.

- Consumer faces a substantial demand charge. Self generation can reduce it as can load shifting, but only with these together is it possible to avoid it completely.

In each of these examples, large savings are possible only with coordination between supply and demand controls. We will call such cases **super-additive** in the sense that the sum total savings when both DSM and SSM are employed is larger than adding the savings from each technology implemented alone.

Sub-additive cases are situations where DSM and SSM are taking advantage of the same thing and when they are used together, the total savings is on the order

of the savings achieved if either one alone was used. For example, a generator is brought on if grid prices are high enough. Load shifting may take advantage of the same thing (time varying prices), and once one of these technologies is implemented, there is nothing left for the other to capitalize on. Another version of sub-additivity is a customer trying to avoid a demand charge. Let us say that the customer purchases a battery to store energy to be used to avoid a peak charge, and he is successful in doing so. In that case, there is no need for him to shift or dump loads additionally, and thus we have a strongly sub-additive situation.

Of course, there is a natural continuum between the super-additive cases and the sub-additive cases, and the distinguishing characteristic which parametrize this continuum is *how load shifting and curtailing makes self generation more attractive.* In the super-additive cases, load control adds to the attractiveness of self generation by clumping loads, or by dropping enough load so that in addition to the generator's production level, a demand charge can be avoided. In contrast, in the sub-additive cases load control removes any benefit of self generation, and vice versa.

It is the flexibility of generators (and storage) which dictates how load shifting adds to the usefulness of self-generation. In particular, generators have a limited range with which they can operate within, and if the load is below that range it is not feasible to run the generator. Also, generators may have minimum up and down times (referred to as cycling constraints), which means that, for example, even if a high load for one hour can be supplied by a generator, if that generator has a minimum up time of 5 hours, it may not make sense to fire it up for that brief high load period.

If a user's generators are flexible either individually or collectively regarding both ranges and cycling constraints, then shiftability is less important. Shiftability becomes important when a user's self generation is inflexible. Then, it is load shifting which allows the user to bring his operation into a range where the self-generation is economical.

Chapter 4

Extensions to the core model

In this chapter we present a few non-trivial extensions to the model developed in Chapter 3. The first is a piecewise linear approach to representing nonlinear input output relationships. This may be important for certain multi-generator supply applications, where selecting the right generators based on the load profile is difficult. The second extension covers stochastic networks. Demands and renewable power supplies (wind, sun) may be better modeled as random data, and this extension allows TOTEM to handle these situations.

The next two extensions presented, priority-based load curtailment and modeling the fixed rates tariff, are much simpler, but important for handling these common policies used in electricity management. The final section on HVAC control is quite separate from the TOTEM model in its approach. Optimal HVAC control cannot be handled with a linear model, and the nonlinear model presented in Section 4.1 is only for nonlinear input-output curves, not for the types of nonlinearities present in HVAC models. Therefore, we depart from the MILP structure of TOTEM and discuss an approach to HVAC optimization that would be used in conjunction of the TOTEM model of Chapter 3.

4.1 Nonlinear input-output relationships

Up until this point, we have only considered linear power convertors, where power output is a sum of linear functions of the fuel inputs, plus an offset term:

$$x_{ijt} = \sum_{n \in I(i)} \left(V_{inj} x_{nit} + B_{inj} u_{nit} \right). \tag{4.1}$$

In this section, we show how to model piecewise linear input-output relationships. The technique we use is a direct extension of the work of Trecate *et al.* [55], which derives a linear mixed integer program for the minimization of an arbitrary piecewise linear cost function (not necessarily continuous) subject to linear constraints. We extend this work to allow piecewise linear approximations for constraint equations.

4.1.1 Single variable approach to piecewise linear relationships

We will begin by showing the technique for a single input-output relationship. Let x denote the input variable and y denote the output. We seek a method of replacing the linear relationship $y = mx$ with an arbitrary piecewise linear relationship, such as the one shown in Figure 4-1. Let N be the number of piecewise segments (N=3 in the Figure 4-1). Let the piecewise function be defined by the x-axis positions of the $N-1$ breakpoints, which we denote by u_s, and the linear function $a_s x + b_s$ in each section s.

We will introduce a binary variable δ_s and a real-valued variable z_s for each breakpoint $s \in \{1 \ldots N-1\}$.

We will use linear constraints to enforce the following relationship between the δ's and the value of x:

$$\delta_s = 1 \iff x \geq u_s. \tag{4.2}$$

We will also use linear constraints to assure that $\sum_{s=1}^{N-1} z_s = f(x)$.

First, the constraints required to ensure consistency of the δ's. Let M be a large

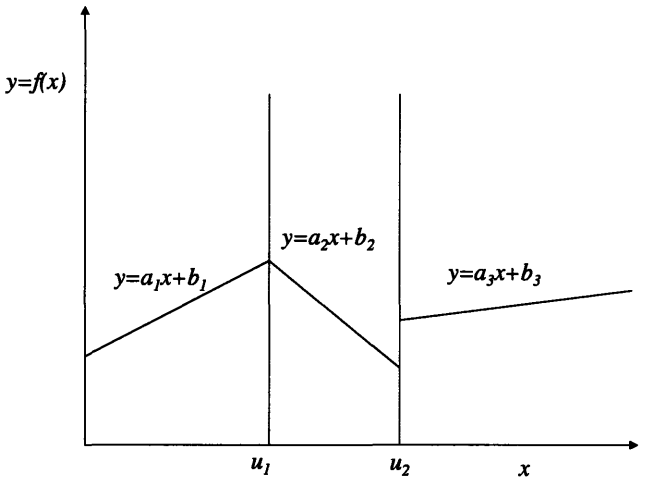


Figure 4-1: Example of a piecewise linear input-output function with $N = 3$.

number, and ϵ be a small tolerance. We have

$$M\delta_s \leq -(u_s - x) + M \tag{4.3}$$

$$(-M - \epsilon)\delta_s \leq (u_s - x) - \epsilon \qquad \forall s \in \{1 \ldots N-1\} \tag{4.4}$$

and

$$\delta_s \leq \delta_t, \quad \forall t < s, \quad s = 2 \ldots N-1 \tag{4.5}$$

To see how these enforce 4.2, assume that $\delta_s = 1$ and $\delta_{s+1} = 0$, meaning x should lie between u_s and u_{s+1}. For $\delta_s = 1$, inequality 4.3 becomes $x \geq u_s$, which is correct, and inequality 4.4 enforces nothing (except that $u_s - x$ is larger than some small negative number, which it always will be). For $\delta_{s+1} = 0$, inequality 4.3 enforces nothing and inequality 4.4 becomes $x < u_{s+1}$. Similar arguments show that 4.3-4.5 enforce the 'if and only if' condition of 4.2.

The next set of constraints will ensure the following relationships for the z variables. Note that the sum of z_s up to a particular value of s will be equal to $a_{s+1}x + b_{s+1}$.

$$z_1 = \begin{cases} a_2 x + b_2, \text{if } \delta_1 = 1 \\ a_1 x + b_1, \text{else} \end{cases} \tag{4.6}$$

$$z_s = \begin{cases} a_{s+1} x + b_{s+1} - (a_s x + b_s), \text{if } \delta_s = 1 \\ 0, \text{else, for } s = 2 \ldots N-1 \end{cases} \tag{4.7}$$

To achieve these, we enforce the following constraints.

$$M\delta_1 - z_1 \leq -a_2 x - b_2 + M \tag{4.8}$$

$$M\delta_1 + z_1 \leq a_2 x + b_2 + M \tag{4.9}$$

$$-M\delta_1 - z_1 \leq -a_1 x - b_1 \tag{4.10}$$

$$-M\delta_1 + z_1 \leq a_1 x + b_1 \tag{4.11}$$

$$\vdots$$

$$-M\delta_s + z_s \leq 0 \tag{4.12}$$

$$-M\delta_s - z_s \leq 0 \tag{4.13}$$

$$M\delta_s + z_s \leq (a_{s+1} x + b_{s+1} - (a_s x + b_s)) + M \tag{4.14}$$

$$M\delta_s - z_s \leq -(a_{s+1} x + b_{s+1} - (a_s x + b_s)) + M, \qquad \text{for } s = 2 \ldots N-1 \tag{4.15}$$

With these constraints on z, it is straightforward to check that if t is the index such that $\delta_s = 1$ for all $s \leq t$ but $\delta_s = 0$ for $s = t+1, \ldots, N-1$, then the sum of all z_s will equal $a_{t+1} x + b_{t+1}$. This completes the modeling for a single piecewise linear input-output relationship.

4.1.2 Extended notation for the TOTEM model

In this section we extend the single input-output relationship to the TOTEM model. Consider a generator, node n, in an energy network. We classify the generators into linear and nonlinear regarding their conversion of inputs to outputs. We will allow each separate input to a nonlinear generator to have its own nonlinear curve, but

assume that this curve is fixed throughout the optimization horizon. Recall for the linear case we have the following relationship ouput of generator i:

$$x_{ijt} = \sum_{n \in I(i)} \left(V_{inj} x_{nit} + B_{inj} u_{nit} \right). \tag{4.16}$$

For the nonlinear generators, we will replace this with the following:

$$x_{ijt} = \sum_{n \in I(i)} \left(\sum_{s=1}^{NB_{i,n}-1} z_{inst} \right). \tag{4.17}$$

So, for each nonlinear generator i, each of its inputs $n \in I(i)$, we need a full set of piecewise linear function descriptors, which includes:

- $NB_{i,n}$: Number of break points for the curve of generator i, input source n.
- a_{ins} and b_{ins}: slope and intercept (respectively) for piecewise linear section s.
- v_{ins}: location of breakpoint s, in units of the input variable x_{nit}.

Since at each time step we have different values for the inputs, we need to include a time index t on the variables that we introduce, but not the above parameters. These variables are:

- δ_{inst}: Binary variable which indicates if $x_{nit} \geq v_{ins}$.
- z_{inst}: Auxiliary variables whose sum over s will yield the contribution of input n to the output of node i.

4.2 Stochastic TOTEM

The TOTEM model as presented in Chapter 3 does not allow for any of the quantities to be stochastic. If there are only small fluctuations, the core model is probably preferable, but if certain time series, such as base demand profiles or renewable power availability are very difficult to predict, we require a stochastic version of TOTEM.

Depending on the underlying system dynamics and availability of information regarding the stochastic processes, a modeler chooses between a dynamic programming based approach and a scenario based approach. In terms of the solution procedure, the scenario based approach – the approach we adopt – is simpler. As will be shown, if the underlying deterministic model is a linear program, then the scenario based stochastic model is also a linear program. But this is not enough to warrant using it over the less restrictive DP approach. In order to understand why the scenario method is chosen, we need to understand the type of randomness in the system that it is important for TOTEM to consider.

Power fluctuations of small time scales, anywhere from seconds to minutes, are handled intrinsically by either the onsite generators or by the grid. A system planner in a DG network needs to be certain that such power fluctuations can be absorbed, but in terms of real time control decisions, these fluctuations are automatically controlled by micro system controllers. A good example of such a controller is a governor in a constant speed engine: if resistive torque is increased, then as the engine slows down, the governor automatically lets in more fuel and air to compensate and quickly speed the engine back up. In a DG example, one may have several engines supplying power. The role of TOTEM is to decide which engines are on which hours of the day, while the governors control the small time scale power dynamics. If it was necessary to model these small scale fluctuations, a scenario-based approach would be a very bad choice. But since it is average energy levels over the time scale of hours that dictate generator on/off decisions, it is possible that a scenario-based approach is the best choice.

Before making this conclusion though, we need to understand the unpredictability of hourly energy levels and how these hourly energy levels evolve in time. We begin by noting that base electric profiles for small networks such as residential or rural village networks and large networks such as MIT are highly predictable over the short term, primarily being a function of the weather and the type of day (workday or non-workday). In cases where it is unpredictable, it is likely one of the following two cases. The first is the smooth type of unpredictability that preserves the overall

shape of the base load profile, but moves it up or down (i.e. the day was hotter or colder than the mean predicted level). The second is that the base load is affected by the random arrival of a large job to the system (for example, in a residential network, this could be a laundry cycle). In each of these cases, a scenario-based approach is completely sufficient for modeling the uncertainty. In the first approach, one assigns a high probability value on the expected profile, and smaller probabilities on the enveloping curves. In the job arrival case, the forecasting system will assign a certain probability to the arrival of a load that significantly alter the base profile, and automatically create the scenarios to include in the model.

It is very important to note that the scenario approach being defended above is not a static approach not responding to real time forecast updates and information gathering. Inherent in the scenario based approach is that the system is making real time hourly decisions based on where on the scenario tree it finds itself. It is not making all of its decisions at the beginning of the day like in the deterministic model. Therefore, it is almost identical to the dynamic programming approach, with the difference being the explicit modeling of potential system states in the scenario based approach, and the more 'physics based' random process state updates of the DP approach. But this difference is lost when one considers that even in the DP approach, unless one can determine closed form optimal policies, which would certainly not be the case if TOTEM was modeled as a DP, then one needs to make some simplifications, including state discretization, to implement the model anyhow. In this case, we are back to bucketing the possible future system states, just as we do in the scenario based method. The main advantage of the DP approach is if the DP recursion can be approximated without resorting to state discretization. In this case, a continuous system which may hit any of its infinite states can be controlled in real-time based on where it is, not where it is approximately. Given the above discussion on the nature of the stochastics of energy networks, this advantage is not very relevant.

A final note in the justification of the use of scenario based stochastic programming is the absence of a random process for grid hourly prices. If hourly prices move as a jumpy random process (much variation around the mean), a DP approach like the one

used by Constantopoulos [14] is the correct choice. But such price processes are not given to the end customer. Furthermore, even if the customer was buying electricity from the spot market, it is not clear that the spot market follows a process jumpy enough to necessitate the DP approach (see Figure 6-17).

Therefore, the choice between a DP approach and a scenario approach becomes one of modeling and computational simplicity. The scenario approach (also referred to as stochastic programming) allows us to use a slightly modified version of the TOTEM model presented in Chapter 3, without switching to an entirely new code base if we had adopted a dynamic programming model. In a stochastic programming formulation, we minimize the expected costs, while assuring that whatever scenario plays out, all of the model constraints will be satisfied. This scenario based approach is described thoroughly in [51] and applied to the unit commitment problem in many papers, see for example [12] and [53].

4.2.1 Scenario-based stochastic modeling for TOTEM

A scenario tree is depicted in Figure 4-2. This data is stored by identifying the parent nodes and child nodes for each branch point in the scenario tree. In order to decrease the storage requirements for stochastic problems, we use the following observation. If two scenarios, call them A and B, are the same until some time τ, then these scenarios are bundled together as one scenario for $t \leq \tau$. Thus, while we may have many scenarios "active" at the end of a problem, at any time point before the end there will be fewer scenarios active.

This scenario grouping intrinsically takes care of the non anticipativity constraints inherent in stochastic programming. These are the constraints which need to be imposed so that future information is not used by the solution, i.e. the outcomes cannot be anticipated. The implementation is described below, where we will see how these constraints are intrinsically enforced.

For each node, we need to store the nearest parent, one step back in time. For the child nodes though, we typically require the children (nodes forward in time, or to the right in Figure 4-2) more than just a single step in time. This is needed for any

constraint which couples present decisions to decisions in the future. Battery storage update equations requires only one step ahead in the future, but generator minimum up and down times require several steps. We define MaxDepth as the maximum number of steps ahead for which we need to store child node information.

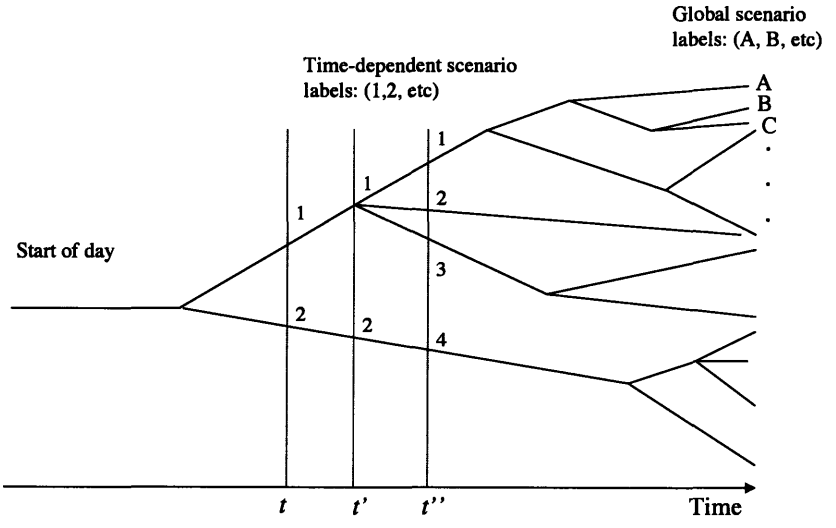


Figure 4-2: Example of a scenario tree for the stochastic TOTEM formulation.

As an example, let us say MaxDepth is 2. Referring to Figure 4-2, we would have the following information coded into the AMPL data file:

$$CS(t, 1, t') = \{1\} \tag{4.18}$$

$$CS(t, 1, t'') = \{1,\ 2,\ 3\} \tag{4.19}$$

$$CS(t, 2, t') = \{2\} \tag{4.20}$$

$$CS(t, 2, t'') = \{4\} \tag{4.21}$$

where $CS(t, s, \tau)$ denotes the child scenarios of scenario defined by time t and time scenario label s at time in the future τ. Similarly, PS denotes the single parent

scenario and for time t'' as our example we have

$$PS(t'', 1) = \{1\} \tag{4.22}$$

$$PS(t'', 2) = \{1\} \tag{4.23}$$

$$PS(t'', 3) = \{1\} \tag{4.24}$$

$$PS(t'', 4) = \{2\} \tag{4.25}$$

Notice that since we are aggregating scenarios until the point they split, just a number for the scenario is insufficient (scenario "2" at time t is different than scenario "2" at time t''). Instead, we use a (time, scenario) index to identify the scenarios. Probabilities are specified this way as well: p_{ts} is the unconditional probability that we ever find ourselves at time-scenario pair (t, s). For example we might have:

$$p_{t'1} = .8 \tag{4.26}$$

$$p_{t',2} = .2 \tag{4.27}$$

For convenience in looping through the scenarios, we also define LS as the last scenario (numerically) at any given time step. In Figure 4-2 for example, we have $LS(t) = LS(t') = 2$ and $LS(t'') = 4$.

While the above fully specifies the stochastic structure of the problem, it is convenient to define the following for problems with shiftable loads. Let GS denote the set of all global scenarios A, B, etc. For each global scenario we define the full set of time-based scenario numbers that define the scenario. For example, in Figure 4-2, the uppermost scenario, A, will have the scenario number list

$$SN(\cdot,\ A) = 1\ 1\ 1\ 1\ 1\ 1\ 1\ \ldots$$

whereas the lowermost scenario would have something like

$$SN(\cdot,\ L) = 1\ 1\ 1\ 2\ 2\ 4\ 4\ 7\ldots$$

The purpose of defining these scenario number lists will be clear when the modeling of the shiftable loads is discussed below.

Having specified the structure and storage of the stochastics of the model, the final information is the actual values of the parameters which are random. We allow prices, loads, exogenous data (the "weather vector"), and renewable energy sources to be stochastic. We assume, without loss of generality, that each stochastic quantity in the model has the same underlying branching structure (if they do not, then a combined tree can always be made). Data, values and associated probabilities, are indexed by (time, scenario) pairs. All decision variables are also indexed by (time, scenario) pairs. This includes the primary decision variables, the flows x, on-off values u, and demand node variables z, as well as secondary decision variables power levels demanded d, state variables S, and state deviations δ. The timing of the underlying model is that the decision maker, having observed his current state, is able to use that information and statistics about the future for making his current decision.

Next, we present the stochastic model equations for various parts of the network. The four subsections below are in order of increasing difficulty.

Stochastic balance equations

The simplest equations to implement in the stochastic framework are the conversion and the input-output equations, since they are written for single points in time. Therefore, we just add the scenario index to the variables. The conversion equation, Equation 3.1, becomes

$$x_{ijts} = \sum_{n \in I(i)} \left(V_{inj} x_{nits} + B_{inj} u_{nits} \right), \tag{4.28}$$

where, for each t, $s = 1 \ldots LS(t)$. The input-output relationship, Equation 3.2, is similar.

Stochastic storage and state updates

Storage quantity and state updates relate the next time step to the current time step, and we use the notion of child scenarios to implement these equations. The storage level y at time $t+1$ is equal to the current storage level plus (flow in - flow out). Rearranging these terms, and noting that we need the child scenario(s) for time $t+1$, we have:

$$\sum_{j \in I(i)} G_{ijk} x_{jits} = \sum_{j \in O(i)} x_{ijts} + y_{i(t+1)c} - y_{its}, \tag{4.29}$$

where one equation is written for each $c \in CS(t, s, t+1)$. Similarly, for the state update equations we have

$$\begin{aligned} &S_{k(t+1)c} = \textstyle\sum_{j=1} H^s_{kj} S_{jts} + \\ &\textstyle\sum_j H^w_{kj} W_{jts} + \\ &\textstyle\sum_j H^a_{kj} z^a_{jts} \qquad\qquad \forall k \in \text{STATES}. \end{aligned} \tag{4.30}$$

Stochastic generator cycling constraints

The following constraints require the use of the parent scenario p and the child scenarios c.

$$u_{ijts} - u_{ij(t-1)p} \leq u_{ij\tau c} \tag{4.31}$$

This inequality, as before in Chapter 3, is written for each $\tau = (t+1) \ldots \min\{t + \text{MinUpTime}(i) - 1, T\}$. But at each future time τ we must write the equation for each child scenarios, and thus, for each τ, we write 4.31 for every $c \in CS(t, s, \tau)$. The same is true for the minimum down time constraint:

$$u_{ij(t-1)p} - u_{its} \leq 1 - u_{ij\tau c}. \tag{4.32}$$

In Equations 4.31 and 4.32, the parent node is used at time $t-1$.

Stochastic shiftable loads

The decision variable associated with shiftable loads is a set of binary variables summing to one specifying which time step starts the load. The summation equal to one specifies that the load gets started once and only once. Under the probabilistic scenario framework, we need to enforce that for each possible scenario the load gets started exactly once. Thus we associate the starting 0-1 variables with the global scenario labels A, B, etc. We replace the variable z^{sl}_{it} with z^{sl}_{ij} with $z^{\mathrm{sl}}_{ij\tilde{s}}$, where $\tilde{s}$ refers to a particular global scenario. For each global scenario $\tilde{s} \in GS$ we must enforce that the load gets started exactly once:

$$\sum_{j=1}^{M(i)} z^{\mathrm{sl}}_{ij\tilde{s}} = 1 \tag{4.33}$$

Recall, $M(i)$ is the last allowable start time for node i.

The amount of demand drawn by shiftable demand node i is given by:

$$d_{itSN(\tilde{s},\ t)} = \sum_{j=1}^{M(i)} l_{ijt} z^{\mathrm{sl}}_{ij\tilde{s}}. \tag{4.34}$$

The $SN()$ function is required to map the global scenario label $\tilde{s}$ back to the local time dependent scenario numbering.

Note that in this modeling, we have not included the possibility of a shiftable load stochastically entering the system, although this is certainly possible. Instead, we assume the shiftable loads are well defined and the stochastic modeling takes care of choosing their optimal start times given the uncertain future.

4.2.2 Full stochastic TOTEM model

minimize	$\sum_{i\in N_F}\sum_{j\in O(i)}\sum_{t=1}^{T}\sum_{s=1}^{LS(t)} p_{ts}c_{its}x_{ijts} \quad + \quad \sum_{k\in S_P}\sum_{t=1}^{T} p_{ts}\gamma_{kt}\delta_{kts}$	
subject to		
(conversion)	$x_{ijts} = \sum_{n\in I(i)} (V_{inj}x_{nits} + B_{inj}u_{nits})$	$\forall i \in N_V,\ j \in O(i),\ t,$ $s = 1 \ldots LS(t)$
(io)	$\sum_{n\in I(i)} G_{in}x_{nits} = \sum_{j\in O(i)} x_{ijts}$	$\forall i \in N_{IO},\ t,\ s = 1 \ldots LS(t)$
(renewable_equality)	$r_{its} = \sum_{j\in O(i)} x_{ijts}$	$\forall i \in N_R,\ t,\ s = 1 \ldots LS(t)$
(storage_equality)	$\sum_{n\in I(i)} G_{in}x_{nits} = \sum_{j\in O(i)} x_{ijts}$ $+y_{i(t+1)c} - y_{its}$	$\forall i \in N_S,\ t = 1..T-1$ $s = 1..LS(t),\ c \in CS(t,s,t+1)$
(base_demand)	$d_{its} = b_{its}$	$\forall i \in N_D^B,\ t,\ s = 1 \ldots LS(t)$
(analog_demand)	$d_{its} = z^a_{its}a_i$	$\forall i \in N_D^A,\ t,\ s = 1 \ldots LS(t)$
(analog_range)	$z^a_{its} \in Z_i$	$\forall i \in N_D^A,\ t,\ s = 1 \ldots LS(t)$
(shiftable_demand)	$d_{itSN(\tilde{s},\ t)} = \sum_{j=1}^{M(i)} l_{ijt}z^{\mathrm{sl}}_{ij\tilde{s}}$	$\forall i \in N_D^{\mathrm{sl}},\ t, \tilde{s} \in GS$
(start_each_sl)	$\sum_{j=1}^{M(i)} z^{\mathrm{sl}}_{ij\tilde{s}} = 1$	$\forall i \in N_D^{\mathrm{sl}},\ , \tilde{s} \in GS$
(supply)	$\sum_{n\in I(i)} x_{nits} \geq d_{its}$	$\forall i \in N_D,\ t,\ s = 1 \ldots LS(t)$
(range_constraint)	$\mathrm{lower}_{ij}u_{ijts} \leq x_{ijts} \leq \mathrm{upper}_{ij}u_{ijts}$	$\forall i \in N_U,\ j \in O(i),\ t,$ $s = 1 \ldots LS(t)$
(min_up_time)	$u_{ijts} - u_{ij(t-1)p} \leq u_{ij\tau c}$	$\forall i \in N_U,\ j \in O(i),$ $t = 2 \ldots T,\ \tau = (t+1) \ldots \min$ $\{t + \mathrm{MinUpTime}(i) - 1, T\},$ $s = 1 \ldots LS(t),\ c \in CS(t,s,\tau)$
(min_down_time)	$u_{ij(t-1)p} - u_{its} \leq 1 - u_{ij\tau c}$	$\forall i \in N_U,\ j \in O(i),$ $t = 2 \ldots T,\ \tau = (t+1) \ldots \min$ $\{t + \mathrm{MinDownTime}(i) - 1, T\},$ $s = 1 \ldots LS(t),\ c \in CS(t,s,\tau)$

(4.35)

(ugroup_min)	$\sum_{(i,\ j)\in \text{UG\_MIN\_NODES}_n} u_{ijts} \geq \text{UGmin}_n$	$\forall n \in 1..\text{numUGroupsMin},\ t,$ $s = 1 \ldots LS(t)$
(ugroup_max)	$\sum_{(i,\ j)\in \text{UG\_MAX\_NODES}_n} u_{ijts} \leq \text{UGmax}_n$	$\forall n \in 1..\text{numUGroupsMax},\ t,$ $s = 1 \ldots LS(t)$
(state_initialize)	$S_{k1s} = I_k$	$\forall k \in \text{STATES},\ s = 1 \ldots LS(t)$
(storage_initialize)	$y_{i1s} = I_i$	$\forall i \in N_S,\ s = 1 \ldots LS(t)$
(state_update)	$S_{k(t+1)c} = \sum_{j=1} H^s_{kj} S_{jts} +$ $\sum_j H^w_{kj} W_{jts} +$ $\sum_j H^a_{kj} z^a_{jts}$	$\forall k \in \text{STATES},\ t,$ $s = 1 \ldots LS(t),$ $c \in CS(t, s, t+1)$
(state_deviation1)	$\delta_{kts} \geq S_{kts} - T_{kt}$	$\forall k \in S_P,\ t,\ s = 1 \ldots LS(t)$
(state_deviation2)	$\delta_{kts} \geq T_{kt} - S_{kts}$	$\forall k \in S_P,\ t,\ s = 1 \ldots LS(t)$
(state_range)	$E^{\min}_{kt} \leq S_{kts} \leq E^{\max}_{kt}$	$\forall i \in S_C,\ t,\ s = 1 \ldots LS(t)$
(storage_range)	$S^{\min}_i \leq y_{its} \leq S^{\max}_i$	$\forall i \in N_S,\ t,\ s = 1 \ldots LS(t)$
(storage_charge)	$\sum_{n\in I(i)} x_{nits} \leq S^{\text{charge}}_i$	$\forall i \in N_S,\ t,\ s = 1 \ldots LS(t)$
(storage_discharge)	$\sum_{j\in O(i)} x_{ijts} \leq S^{\text{discharge}}_i$	$\forall i \in N_S,\ t,\ s = 1 \ldots LS(t)$
(non-negativity)	$x,\ y,\ z,\ d \geq 0$	
(binary)	$u,\ z^{\text{sl}}$ binary	

(4.36)

4.3 Priority-based load curtailment

In the event that demand exceeds available supply in an electrical network, a system shutdown (a blackout) will occur unless loads are curtailed to reduce demand. In the setting modeled here, we imagine that the central network controller has the ability to selectively disconnect base loads from the system. Such action might be necessary in DG applications if some of the generators, either fossil-based or renewable, go down,

and the load is too large for the rest of the generators to handle.

This is something that TOTEM will not plan ahead for (given that load curtailment should be a rare event, it should not be modeled in the stochastic framework), but instead will re-plan for if it occurs, in near real-time. In order that a blackout is avoided, we assume that the system has some short term energy it can draw on such as storage until the problem is identified and TOTEM performs its re-planning step, which identifies the loads to shed.

Base load i ($i \in N_D^B$) is assigned a priority M_i, which is a large number (larger than the total cost of the running the system over the planning horizon). M_i represents a cost penalty that will be incurred for shedding load i, so the larger the value is, the more important the load is. Let q_i be a binary decision to shed load i ($q_i = 1$) or not ($q_i = 0$). We modify the cost function by adding the term second below:

$$\text{minimize} \quad \text{cost} + \sum_{i \in N_D^B} q_i M_i \tag{4.37}$$

We also modify the base load demand, Equation 3.5, to the following:

$$d_{it} = b_{it}(1 - q_i). \tag{4.38}$$

Note that the model stays linear since in each modification, q_i multiplies known parameters and not other variables. A more refined implementation could choose the time at which to curtail load i, thus using the variable q_{it} instead of q_i, along with inequality constraints to avoid cycling of curtailing and not curtailing (which would certainly affect the underlying base load, in addition to being an undesirable policy).

This curtailment model is demonstrated in the hybrid rural village case study presented in Section 6.1.2.

4.4 TOTEM implementation of fixed rates tariffs

We introduced the fixed rates tariff in Section 1.5. This type of tariff charges one rate for electricity up to a set amount, and then charges a different rate for kWh bought

above that set amount. The solid lines in Figures 4-3 (A) and (B) show the fixed rate tariff as a volume discount and volume penalty, respectively.

Here we address what to use for an electricity price in a TOTEM optimization under a fixed rates structure. In both load control settings and distributed generation settings, TOTEM uses the hourly price of electricity to make control decisions. Consider a volume discount tariff with one transition, such as that shown in Figure 4-3A. If one optimized each day using the current rate of electricity, midway through the month the rate would decrease and so the optimal strategy would change. Each month would have this transition after energy usage exceeded e_1, but in reality this rate should not make users shift their consumption pattern midway through each month. Rather, the user should estimate his total electricity usage for the month E_T *a priori* and use that value and the fixed rates curve to compute the average unit price of electricity.

To indicate the suboptimality of a nearsighted approach, consider a user with a linear generator, the dashed line, and volume discount rate, in Figure 4-3. Consider a two-period month, where the first day his total usage is right at the break point e_1. If he just compares generator cost to the cost of the first e_1 kWh of electricity from the grid, he will choose to generate for the first period. The second period is then just like the first: he does not get the volume discount so he again chooses to self generate, making his total bill higher than if he had considered his total electricity usage E_T initially. Had he made that consideration, his choice would have been not to generate at all.

This example can be generalized. Consider a user with a volume discount rate (electric bill is concave in electricity purchased) and a generator with efficiency increasing with total usage. Let h be the fraction of monthly electricity supplied by the generator. The cost curves are depicted in Figure 4-3C. The total cost is the sum of these two concave functions, and thus is concave. The minimization of a concave function occurs at one of its endpoints, which reveals that in these types of situations, the optimal user strategy is "all-or-none": either use the generator to its full capacity or do not use it at all. This optimal strategy can be understood by considering the

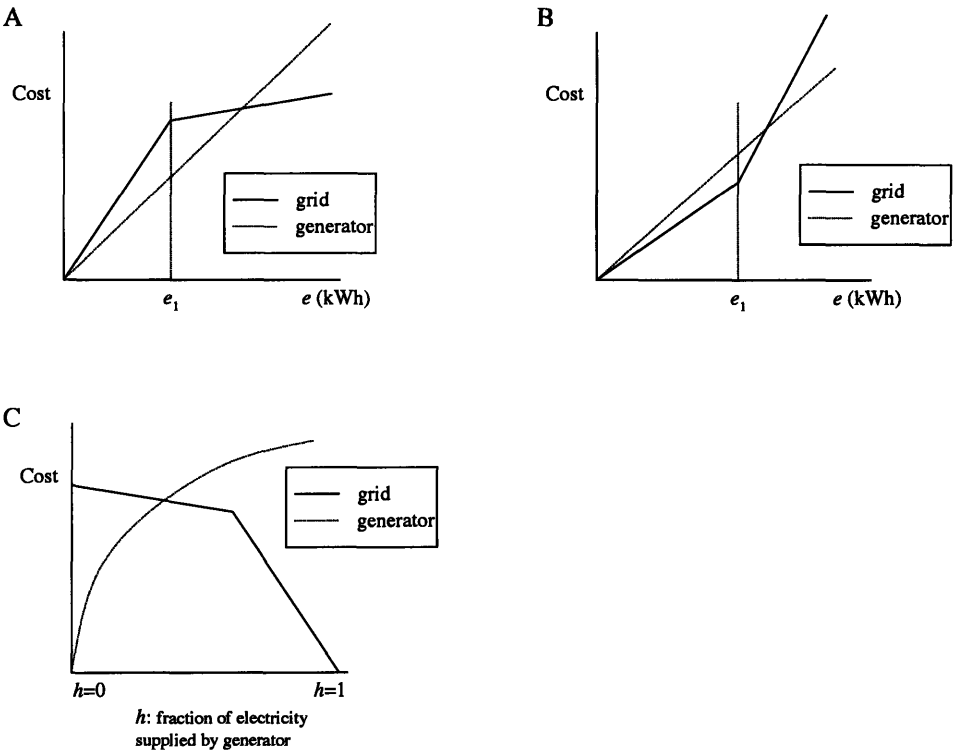


Figure 4-3: A) Volume discount fixed rates plan shown with a linear self generation cost curve. Initially, while the generator seems like the best choice, the volume discount effect takes over eventually leading to an optimal policy of purchasing all power from the grid. B) A fixed rate tariff which charges a higher rate for electricity purchased beyond the quantity e_1. In this case, it would be optimal to use the generator if the total usage is on the far right of the graph. C) Costs for electric purchases and self generation shown as a function of fraction of total electricity usage provided by self generation. Since both functions are concave, the sum is concave, and the minimizing h will be an endpoint.

following. Take a user under a volume discount fixed rates plan who is not currently self generating. He considers generating an increment of his electricity, and finds that he saves money. His generator performs more efficiently if he uses it more, and with each increment not purchased from the grid, his average grid price will rise, and thus if he found it initially beneficial to self generate an increment of his electricity usage, he will continue to find this. A similar argument can be used to show that if the user is fully self generating and finds it economical to buy an increment from the grid and

reduce his self generation incrementally, then he will find it optimal to switch to the grid completely.

In general, the generator cost versus usage curve will not be concave, or convex for that matter. In such a case, an "all-or-nothing" strategy may not be optimal. This means that there is some optimal generator strategy that's not at one of the end points of its possible total usage amounts, and then one may find the following procedure helpful. This procedure finds the unit price of electricity, p_e, to use in daily TOTEM optimization run.

Let p_e equal the final unit price of electricity from the grid, and let $c(e)$ denote the fixed rates tariff for purchasing e kWh of grid electricity. Let $h(p_e,\ E_T)$ be the optimal fraction of electricity to be supplied by the generator. h will be increasing in p_e. The function h can be computed by TOTEM in a series of runs varying p_e and E_T. By definition, p_e is the total cost of electricity purchased in one billing cycle, divided by the amount of electricity purchased in one billing cycle. Thus, p_e is given implicitly by the following:

$$p_e = \frac{c\left((1 - h(p_e, E_T))E_T\right)}{(1 - h(p_e, E_T))E_T} \tag{4.39}$$

This is the correct value to use for the electricity price from the grid. If a higher value is used, then the user will tend to generate too much of his own power, and if a lower value is used then he will generate too little.

To summarize, if a user has a volume discounting tariff and a concave generating cost versus generation amount curve, then his optimal strategy will be an endpoint strategy: either use the generator to full capacity, or do not use it at all. In other situations, having an estimate of his total monthly electricity usage, and historical or simulated data which can be used with TOTEM to calculate the optimal generator fraction curve $h(p_e,\ E_T)$, he can find the electricity price to use within daily optimizations from Equation 4.39.

When a fixed rates tariff includes a sizeable demand charge, we refer the reader to Chapter 5.

4.5 HVAC Control

Heating, ventilation, and air conditioning (HVAC) energy usage represents a major portion of a building's energy bill [63]. Therefore the optimal control of HVAC equipment is an important field, and a very challenging field as well, with much room for growth. This section provides a broad overview of some of the key issues in HVAC control, and describes a numerical approach to HVAC optimization. Specifically, we cover the following points:

- The difficulty of the HVAC control problem.
- Standard HVAC controls versus optimal controls.
- The difference between the HVAC control described in Chapter 3 and the realistic HVAC problem.
- A simulation based optimization scheme for HVAC optimization, implemented by tying together various free software packages.

Many of my ideas on this subject came from a careful read of the recent thesis by Helen Xing at MIT [63], which focuses on the last bullet point above. I am greatly indebted to Helen for sharing her work with me on this topic. I extend her work by optimizing over more controls and using a more current optimization algorithm better suited for the HVAC control problem. Numerical results are given in the HVAC case study of Chapter 6.

4.5.1 The difficulty of the HVAC control problem

The simple HVAC model presented in Chapter 3 is used by many researchers for the temperature control problem in various pricing/DG settings. It is extremely idealized in its assumption that temperature change is linearly related to a single control input. With much construction going on at MIT these days, we often see buildings at the skeletal stage, displaying the inner maze of ventilation ducts. This gives the correct impression to the observant passerby: the number of flows and pipes and controls in

such a building is enormous. The temperature and humidity in one room is related to the nearby rooms, the radiant sunlight entering the room, its occupancy level, and each of the controls in such a system. The controls found in most buildings include thermostatic set points, fan speeds, chiller usage, temperature of the air being supplied to the room, and mixing level of the economizer (a device which combines recycled indoor air with fresh outdoor air). Most of these controls can be adjusted in real time, and separately for different zones of the building.

Exactly how these controls collectively influence the objective function depends of course on the objective function itself. The objective function may be as simple as total energy supply cost. Whether this cost is simply proportional to total energy usage, or results from an underlying TOU or RTP structure, and possibly a demand charge, it will be a nonlinear function of the controls [63]. Much more nonlinearity enters the picture when the cost function includes a comfort term, as shown in the example multi-attribute objective function in Figure 4-4.

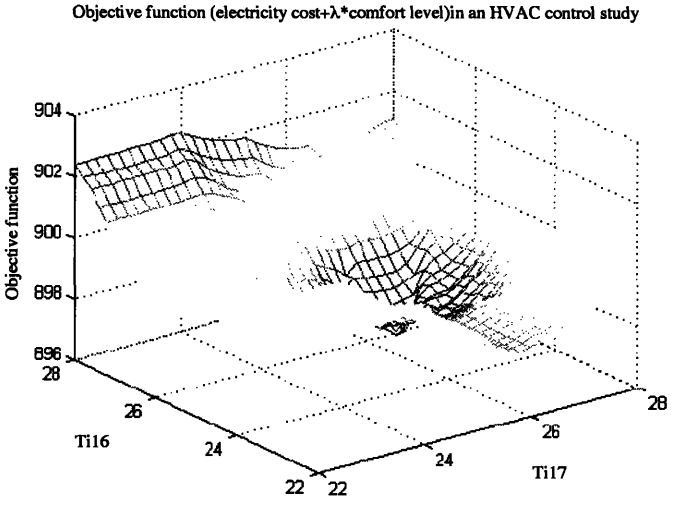


Figure 4-4: Nonlinear example of an objective function made up of the weighted sum of energy costs and a measure of comfort level. The two horizontal axes, Ti16 and Ti17, are thermostatic set points (in degrees Celcius) at hours 16 and 17 of a simulation of a summer day in Austin Texas. Figure reprinted from Helen Xing's MIT doctoral thesis, Building Load Control and Optimization (2004) [63], used by permission of the author.

HVAC control literature is largely made up of studying the stability and response time of control implementations. This is because, before the advent of full digital control, mechanical control systems were used (and are still the main HVAC controls found today). This type of control handles local devices and does not make decisions based on outside price signals or future projections. The issue of stability occurs in feedback control systems, but not in systems where it is assumed that the system dynamics are known, and thus the control problem becomes one of choosing from a set of stable controls the one that minimize some objective function.

The HVAC modeling discussed in Chapter 3 is based on a simple linear state update for temperature based on the control applied. It is assumed that the same control which is linearly related to energy use is linearly related to the state update. This is a major simplification used by many authors to study HVAC control, but not of any practical use. Neither the control/cost nor the control/state update relationships are in practice linear. Also, thermal comfort is a difficult topic and is influenced by many factors besides just temperature [21]. Due to the shear number of controls and the underlying nonlinear model, we require numerical search procedures to solve real HVAC control problems.

There are two main steps to practical HVAC control. The first is the identification of the system dynamics, which can be done by either data gathering and learning the system using neural networks for example, or by a detailed modeling of the building and subsequent use of that model in a building energy simulation tool such as Energy-Plus (U.S. Department of Energy product, free) or ESP-r (University of Strathclyde in Scotland product, also free). The first method, nonlinear system identification, is more suitable for practical implementations since it avoids the work of detailed building modeling and gives faster results than simulation. Simulation is more suitable for academic studies since obtaining real building data and testing various control schemes is not feasible. This is the approach we take below.

The second step is the optimization of building control. Whether we have our system represented as a closed form nonlinear system, or as a simulation, we require intelligent ways to search the space of feasible controls for the optimization portion of

the problem. For a review of optimization algorithms for simulation based objective functions, see [63]. Below, I focus on an algorithm provided in the GenOpt package (Lawrence Berkeley National Laboratory, free) [60] which uses initially a Particle Swarm (PS) approach and finishes with the more local Hooke-Jeeves (HJ) general pattern search algorithm. This hybrid algorithm attempts to avoid local minima that result from using just the HJ algorithm. The simulation-based optimization approach is depicted in Figure 4-5. An example using this approach is given in Chapter 6.

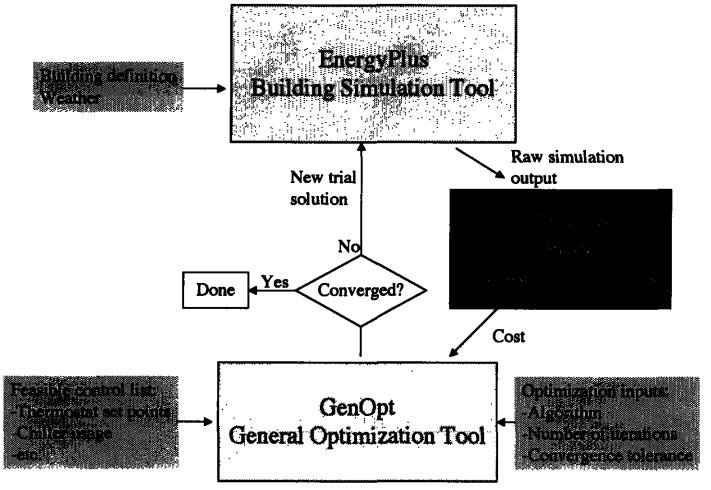


Figure 4-5: Data flow for simulation based HVAC control optimization. The GenOpt package handles updating the trial solutions and rewriting the input decks for any text input simulation program, EnergyPlus in our case.

Chapter 5

Making daily decisions under a monthly demand charge

One way that electric power distributers encourage their customers to keep maximum loads down is by including a charge based on the maximum power power drawn in the entire billing cycle. The reasoning behind this charge is that the transmission and distribution equipment must be sized to meet the customer's maximum power draw. The demand charge presents the customer with an interesting optimization problem: what do I do today regarding my peak power draw when my future peak power draws are uncertain? For example, if early in the month the customer has a high peak load, he is less concerned later in the month, having likely already incurred the demand charge. Conversely, if no demand charge has occurred and it is towards the end of the month, the customer is inclined to do more job shifting then previously in order to keep peak loads down. This is all complicated by the fact that future electricity demands are unknown.

A customer has several strategies to reduce his daily peak load, thereby avoiding a high demand charge:

- Shift loads within the day
- Shed loads entirely, or reschedule them for another day
- Decrease or reschedule power consumption for analog control nodes (HVAC,

etc.)

- Bring on local generation to cut down on grid purchases
- Employ storage to reduce grid peak power draws

Implicit in all of these is that, without the demand charge, it is best for the user to draw power in such a way that ignores the peak load used. In order to formulate a decision strategy for how much you desire to decrease the peak load on a given day, it is necessary to put an economic value on the various magnitudes of peak load reduction, for which TOTEM is the appropriate tool.

For a fixed network and load set, one can use TOTEM to compute a curve like the one shown in Figure 5-1. One obtains this by running TOTEM for the day's profile several times, each run with a different value of maximum allowable grid draw.

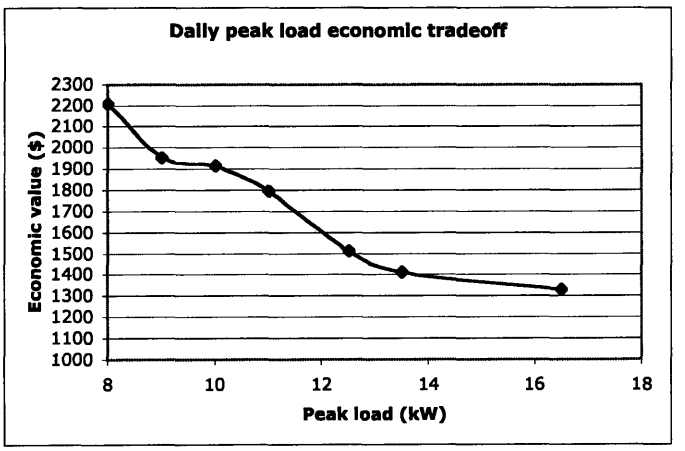


Figure 5-1: Curve showing a hypothetical tradeoff for peak load reduction. Curve shows the economic incentive required to make attractive curtailing the peak load to a given level. In the curve shown, capping the daily peak to 10kW would result in a daily cost of $1900, either in direct distributed generation costs, or estimated economic costs of rescheduling/curtailing loads.

5.1 Mathematical model

While it is not always simple to put an economic value on peak reductions (although it is easy if a TOTEM model of the plant exists), we assume that such an economic tradeoff curve is available. Let this curve be represented by $g(u)$ where u is the peak level chosen for the day, and $g(u)$ is the daily economic value of that choice (Figure 5-1 is example of such a curve). Note that this is not necessarily the total electric peak draw, but only the peak obtained from the utility which charges the demand charge, i.e. the grid. It is important to stress here that $g()$ measures the *daily* economic value, and is not related to the demand charge levied at the end of the month on the maximum peak daily level which occurred. We assume that $g(u)$ is non-increasing in u. Note that if there are certain times of the day or week for which demand charges apply, then the peak level chosen u implicitly means the peak level during those demand charge applicable times. A simple extension to this model, not presented here, handles the case of distinct demand charges for different periods of the day, for example off-peak and an on-peak demand charges.

If each day of the month provides the same tradeoff curve g, then the optimal demand charge strategy is easily solved as follows. Clearly, whatever peak level u the user chooses should be the same for all days.[1] If on any one day the user used a smaller value, he can decrease his overall bill by increasing to the maximum level used on the other days, since g will decrease and the demand charge will stay the same. Therefore, to find the optimal u to be used every day, he minimizes his bill:

$$\text{Bill}(u) = \text{NumDaysInPeriod} \times g(u) + D(u) \tag{5.1}$$

[1]Technically, the strategy need not be same every day, if the user finds himself off of the optimal strategy at some point. For example, early on in a billing period, even if you are in a low peak state, meaning you have not experienced a high peak value, it may be advantageous to use a high u anyhow since you have enough days left to benefit from that high usage. But later on towards the end of the billing period, you will become more conservative in the same low state. This requires you to 'go off of the optimal path' though, since if you were using the aggressive high u strategy early on, you will not find yourself in a low peak state. This can be numerically demonstrated using the application found at http://orc-pumba.mit.edu/dcraft/THESIS/DemandChargeApp/, using the default interpolated demand charge, removing stochastics, and using daily curve #5.

where $D(u)$ is the (non-decreasing) demand charge as a function of the peak load.

The problem is equally simple if different days have different cost tradeoff curves, but this information is known in advance. With the same argument as above, we can see that there will be one peak value that he should use every say. Let $g_k(u)$ be the curve for day k and we have the following function to minimize to find that optimal peak value:

$$\text{Bill}(u) = \sum_{k=1}^{\text{NumDaysInPeriod}} g_k(u) + D(u) \tag{5.2}$$

The problem becomes more realistic when the tradeoff curve for each day is unknown. We assume the following setting: while we do not know the tradeoff curve in advance for day k, we do know that it comes from a family of possible tradeoff curves, indexed by i, and we have a probability p_i of it being curve i. Let w_k^1 be an integer valued random variable giving the daily curve to use (in the discrete case). At the beginning of the day, when the daily strategy is to be chosen, we assume we know which type of tradeoff curve we have for the day (for example, the weather forecasts are in, and the loads to be scheduled are known, etc.) The decision maker's problem is to choose the daily peak level u_k.

A second disturbance w_k^2, not known at the time of the decision, causes a modification in both the daily cost incurred and the daily peak load. For daily cost curve of type i, the daily cost is given by $g^i(u_k,\ w_k^2)$, and the peak load is given by $p^i(u_k,\ w_k^2)$. In the discrete case, we will assume the value $w_k^2 = i$ occurs with probability q_i.

To summarize, the timing for this planner is as follows. At the beginning of the day, the first random variable w_k^1, the parameter which sets the daily cost function $g()$ and the daily peak load function $p()$, becomes known. With that the planner chooses his schedule for the day, which amounts to choosing his desired peak load level u_k. As the day unfolds, the random disturbance w_k^2 augments the planned daily cost and peak level, and the daily cost is incurred (which is a function of u_k, w_k^1 and w_k^2).

The relevant state for this DP is $x_k = \{m_k, d_k\}$ where m_k is the current maximum power drawn on all days prior to day k, and d_k is an integer identifier of the type of

cost curve applicable on day k. The state update equations are:

$$m_{k+1} = \max\ \left(p^{d_k}(u_k,\ w_k^2),\ m_k\right) \tag{5.3}$$

$$d_{k+1} = w_k^1 \tag{5.4}$$

The DP formulation for this is given below. On the final day, day N, we pay the demand charge, while on every other day we incur a daily cost $g()$. The function J_k represents the cost to go at day k.

$$J_N(m_N) = D(m_N) \tag{5.5}$$

$$J_k(m_k,\ d_k) = \min_{u_k \in U} \mathrm{E}_w \left(g^{d_k}(u_k,\ w_k^2) + J_{k+1}(\max(p^{d_k}(u_k,\ w_k^2), m_k),\ w_k^1)\right) \tag{5.6}$$

for $k = 1 \ldots N-1$, where expectation over w refers to the two random variables w_k^1 and w_k^2.

A discretized version of Equation 5.6 is given by the following:

$$J_N(m_N) = D(m_N) \tag{5.7}$$

$$J_k(m_k,\ d_k) = \min_{u_k \in U} \left(\sum_j q_j g^{d_k}(u_k,\ j) + \sum_i \sum_j p_i q_j J_{k+1}(\max(p^{d_k}(u_k,\ j), m_k),\ i) \right) \tag{5.8}$$

for $k = 1 \ldots N-1$.

5.2 Numerical implementation of discrete model

We computationally solve the discrete version of the DP within a web browser using the scripted language Javascript, which is usually used for checking form input and manipulating web images, and not considered a smart choice for scientific computing. There are a few reasons that I choose Javascript. One is to demonstrate that with today's computer speeds, even a notoriously difficult computational task, solving a DP, can be done quickly using a notoriously slow language, Javascript. From a more

practical standpoint, coding the DP in Javascript allows users to run the application with their own data on their own computer without installing any software, provided they have a web browser. Additionally, since the Javascript is client-side, the user also does not need to be connected to the web to run the application. The application is available at http://orc-pumba.mit.edu/dcraft/THESIS/DemandChargeApp/.

5.3 Numerical examples of discrete demand charge model

Here we consider a specific instance of Equation 5.8. We assume that the peak control disturbance w_k^2 adds to the users chosen peak level u_k. Therefore, the peak function, which we assume is identical each day, is given by $p^{d_k}(u_k,\ w_k^2) = u_k + w_k^2$. We assume there are five potential daily cost curves, each one successively steeper. The curves are shown in Figure 5-2. We use a linear demand charge with slope 3. We have four possible peak disturbances, 0, 1, 2, and 3, with probabilities q_i for each of 0.3, 0.2, 0.1, 0.4. A screen dump of the application interface shows this information, see Figure 5-3.

We begin by observing the difference in the control strategy for two users. Let user A be faced with cost curve 1, and user B cost curve 5, the steepest one. Thus user 5 has more influence over his daily cost. Figure 5-4, which displays the optimal solution to the DP for both users, shows that since User B has more influence over his daily cost, he finds it optimal to in the beginning of the billing period to use that influence and lower his cost at the expense of a large demand charge. Towards the end of the billing period of ten days, the two users' strategies converge since the demand charge portion of the bill dominates over the remaining daily costs, and the demand charge is the same for the two users.

A similar strategy difference exists between User A and User C, who has a smaller demand charge (slope of 1.5) than User A, but all else is the same. The result resembles that shown in Figure 5-4, so is not shown separately. The point is that

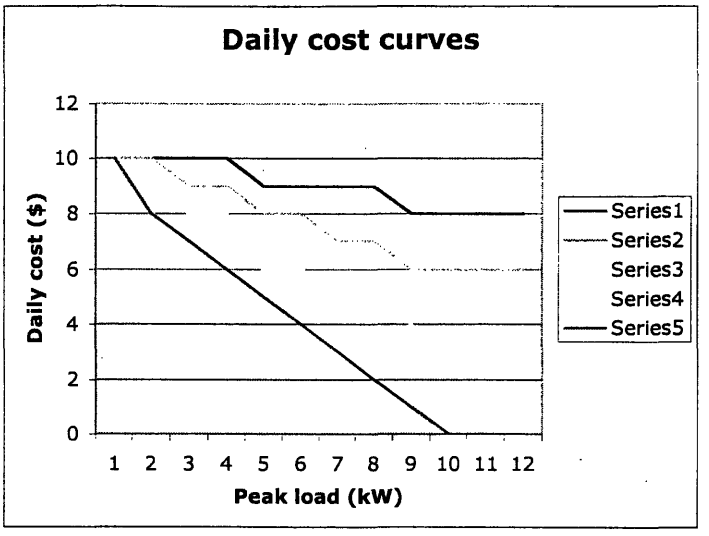


Figure 5-2: Five peak/daily cost tradeoff curves used in demand charge exploration.

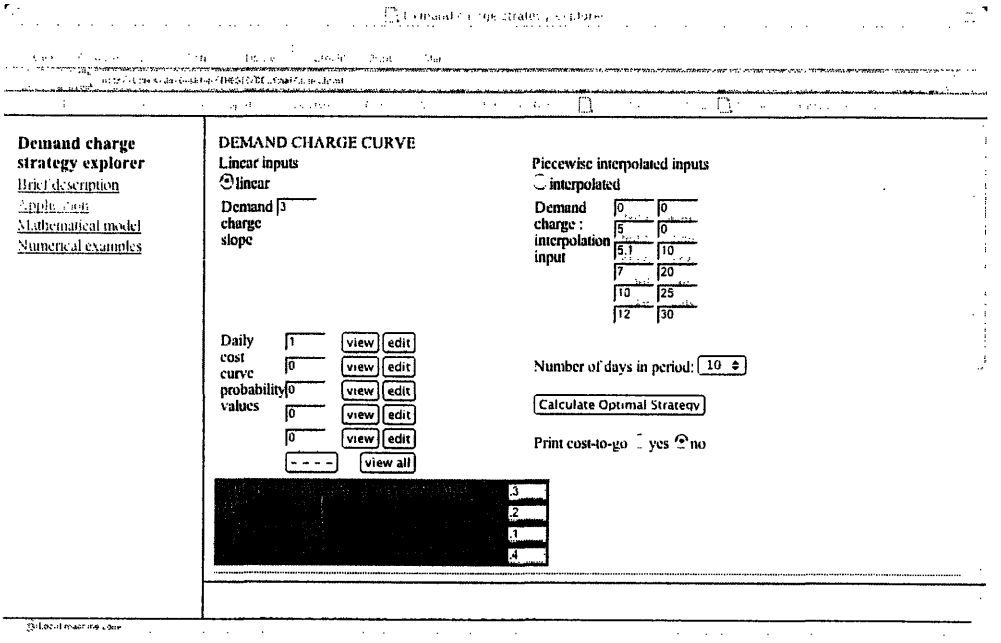


Figure 5-3: Application interface for web-based demand charge explorer available at http://orc-pumba.mit.edu/dcraft/THESIS/DemandChargeApp/ .

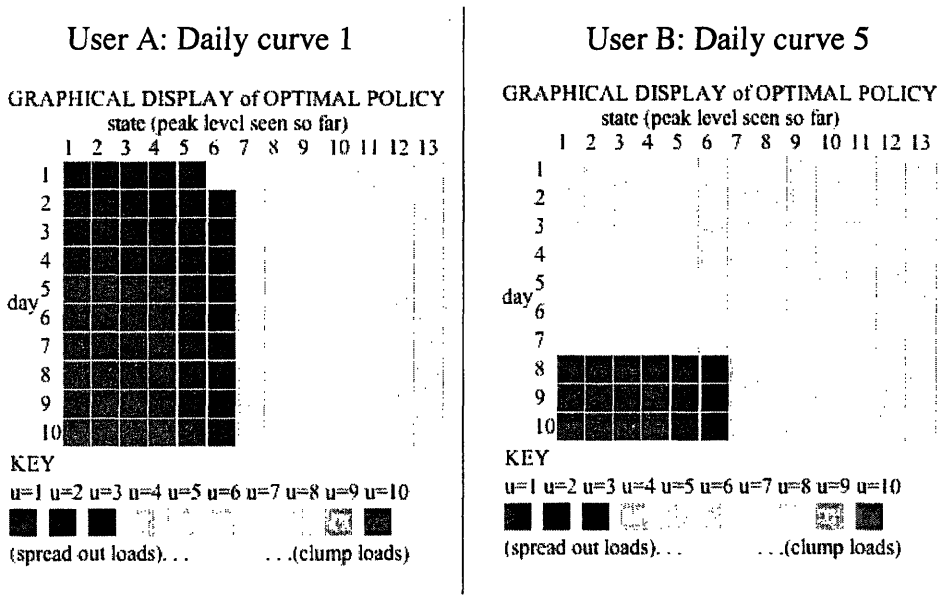


Figure 5-4: Visual comparison of strategies for User A and User B.

when daily costs are a larger relative component of the bill, the users, if following an optimal strategy, will tend to use high peak strategies early on in the billing cycle, provided the probability that they would have reached that peak stochastically anyhow is high enough.

If the probability of spiking later in the month is lowered, we see a user adopt a more conservative strategy in the early days of the billing period. This is shown in 5-5, which presents the results for User D, whose spike probabilities q_i are altered from 0.3, 0.2, 0.1, 0.4. to 0.5, 0.48, 0.01, 0.01. Interestingly, while initially User D adopts a more conservative (low u) strategy, since he might very well avoid a high demand charge, towards the end his strategy becomes slightly more aggressive than User A's since User A is taking into account the fact that there is a good chance even on the last day that he will see a spike which will cause a high demand charge.

In all of the cases presented thus far, we have used a linear demand charge func-

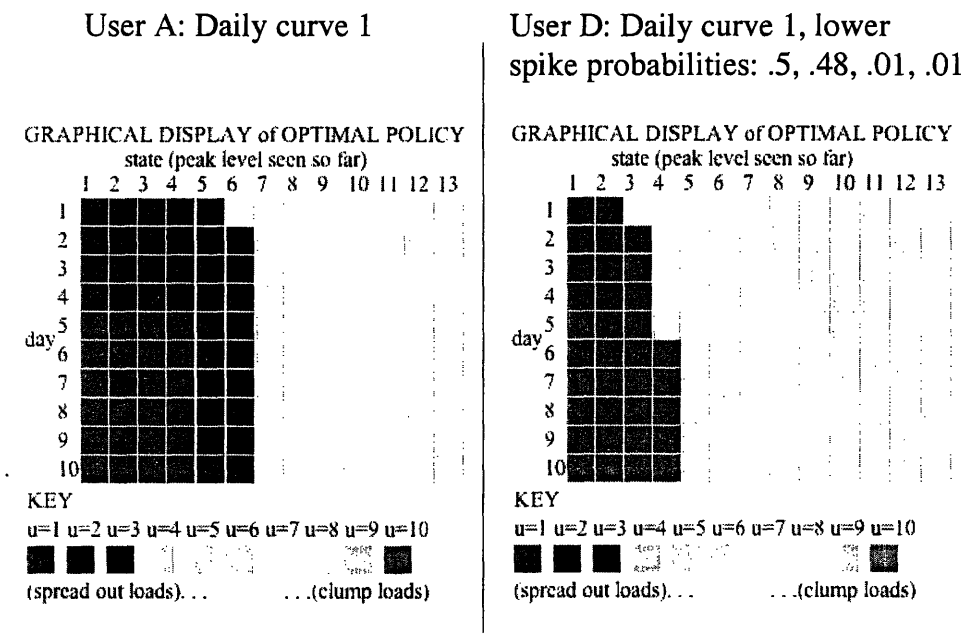


Figure 5-5: Visual comparison of strategies for User A and User D, who has reduced chance of experiencing a high peak load unintentionally.

tion. If the user faces a demand charge function which has discrete jumps, which is sometimes the case, then the resulting strategy can be jumpy also. For example, let us consider a demand charge where as long as the user's peak is less than 7 kW, there is no demand charge, but after 7 kW there is a demand charge which grows at a linear rate of \$3 per kW, starting at the value of \$21. Thus, the demand charge curves for User A and User E are both linear with slope 3, but if User E does not exceed 7 kW, then his demand charge is 0. The difference between the two users' optimal strategies is depicted in Figure 5-6. The jump behavior for User E is clearly evident: if he is below the peak level of 7, then he adopts the conservative strategy $u = 4$, allowing for the high likelihood that the peak addition spike disturbance $w_k^2 = 3$, which happens with probability $q_3 = 0.4$. By choosing $u = 4$, even if on some day k, $w_k^2 = 3$, he still avoids the demand charge. If he ever exceeds the level of $m_k = 7$, then his strategy is identical to User A's.

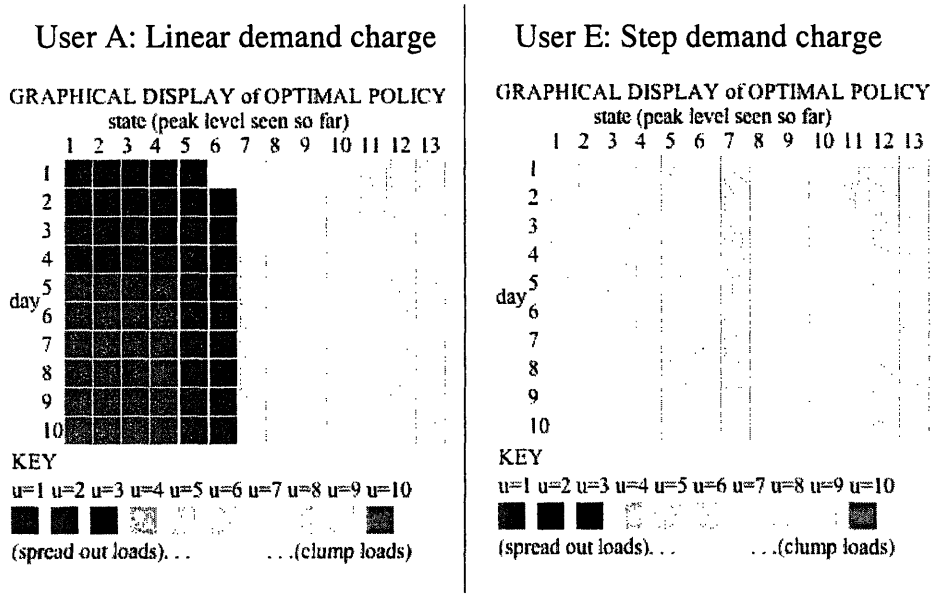


Figure 5-6: Visual comparison of strategies for User A and User E, who has a step function behavior demand curve at the value of peak=7 kW.

5.4 Combining TOTEM with the demand charge model

TOTEM is a daily planning tool which involves local generation decisions as well as demand side controls. In contrast, the demand charge model presented in this chapter works over a time horizon of a complete billing period, typically one month. Implicit in the demand charge decision tradeoff is that to save money daily, you clump your loads and perform them at low kWh prices, but to save money monthly you want to avoid clumping, thereby avoiding a high demand charge.

The way to use TOTEM and the demand charge model together is as follows. The demand charge model assumes you have historical data giving the types of cost/peak tradeoffs you can expect. Given this, you can run the demand charge model, which will yield for your current day the peak load you should strive to achieve without

exceeding. This maximum level can then be added to TOTEM as a flow constraint. TOTEM is then run for the current day with this constraint in place. We give an example of this below.

5.5 Demand charge and TOTEM: an example

There are two types of strategies one adopts to avoid a demand charge: load curtailment (or the less extreme load shifting, which included HVAC control) and the use of self-generation. A user who owns a generator, even if it is an inefficient and therefore expensive to run, may find that on a day of particularly high load, it is worthwhile to serve some of that load with the onsite generator to avoid the demand charge. In this study we consider such a user, who also has load shifting capabilities. We investigate his optimal behavior as given by Equations 5.8 and compare his expected monthly bill to the bill he would see if he adopted some sub-optimal policies.

Figure 5-7 shows the TOU grid pricing and historical data for this user. We assume that, for each of the five jobs j there is a probability p_j that that job will arrive to the system on any given day, independently of other job arrivals and independently of other days. At the beginning of each day, the controller is aware which jobs, if any, have arrived, and can schedule them to start at any time, provided they are done before the day is over. We also assume that which outside temperature profile and which base load occurs are independent across days and the other same day random variables. The probabilities for the data are given in Table 5.1. Given that there are 3 base loads, 3 temperature profiles, and 5 jobs, we have total of $3 \cdot 3 \cdot 2^5 = 288$ possible day types.

Gas price is chosen so that the cost of generating a MWh of electricity is higher than the largest TOU cost of \$80. Therefore, the user will only use his generator to avoid a demand charge. Depending on the magnitude and structure of the demand charge, the user will use one or both of load scheduling and self generation.

Each hour of air conditioning time requires 5 MWh of electricity. Temperature

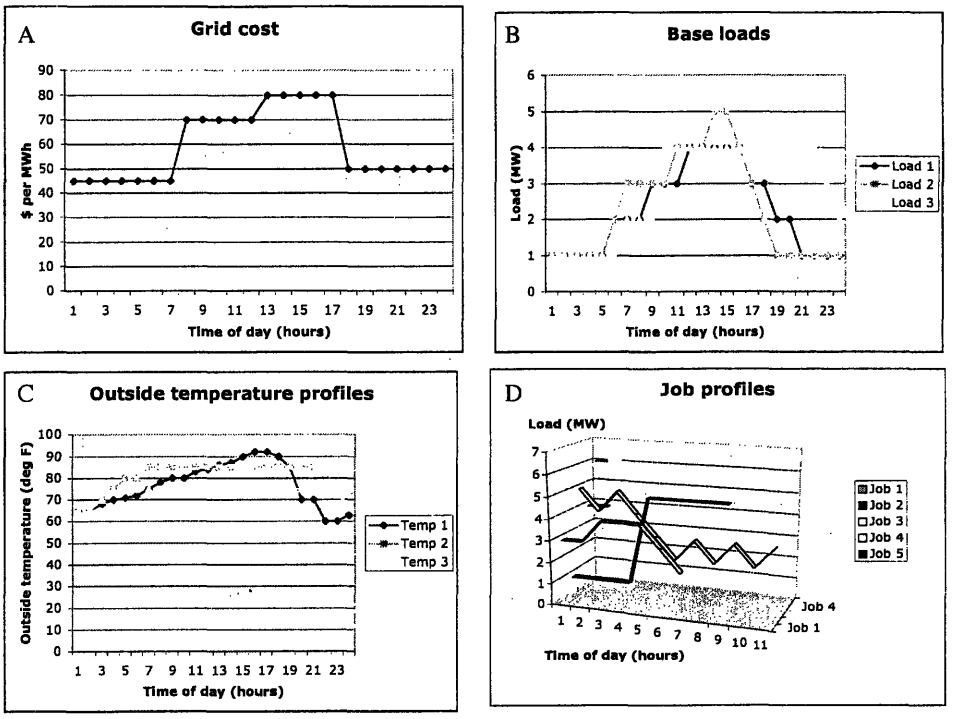


Figure 5-7: Data for a customer optimizing against a demand charge as well as daily time-varying costs. A) Time-of-use cost structure. B) Three representative base loads (uncontrollable). C) Three representative outside temperature profiles. D) Five job profiles. These jobs may be shifted, performed any time during the day.

(S for state, as in Chapter 3) update is given by the equation:

$$S_{t+1} = H_s S_t + H_w W_t + H_z z_t = 0.7 S_t + 0.3 W_t - 6.0 z_t, \tag{5.9}$$

where W_t is the outside temperature (the weather vector) at time t, and z_t is the binary control of the air conditioning units at time t. Temperature is required to fall within the time dependent range as shown in Figure 5-8.

In order to generate the daily cost curves, we run TOTEM for each of the 288 day types, and for each day we run TOTEM 10 times, once for each upper bound in the set $1, 2 \ldots 10$ on the maximum electricity drawn from the grid. This creates 288 daily cost curves, shown in Figure 5-9. We bucket these data into 5 representative averaged curves for use in the demand charge dynamic program.

Item	Probability
Load 1	0.2
Load 2	0.5
Load 3	0.3
Temp 1	0.2
Temp 2	0.5
Temp 3	0.3
Job 1	0.1
Job 2	0.2
Job 3	0.05
Job 4	0.05
Job 5	0.02

Table 5.1: Probabilities for daily decisions under a monthly demand charge example. Each day, exactly one base load is chosen and one temperature profile, and thus those values sum to 1. Each job however has an independent probability of being part of each day's load, and thus those probabilities do not sum to 1.

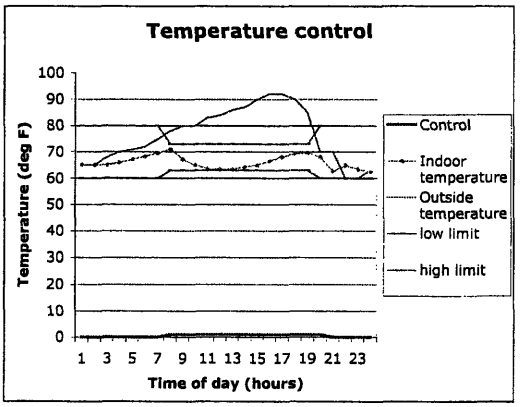


Figure 5-8: Temperature control attempts to find the on/off air conditioning strategy which minimizes cost while keeping the temperature within the given time dependent range. The control shown above is a simple thermostat heuristic which turns on the AC when the temperature rises above the high limit, and is turned off at the end of the working day.

5.5.1 Optimal static policy

Demand charges are typically either linear in maximum power draw (averaged over each fifteen minute interval) or linear in maximum power draw minus some minimum power level below which you are not charged a demand charge. Before investigating the optimal solution using dynamic programming, let us first understand how one

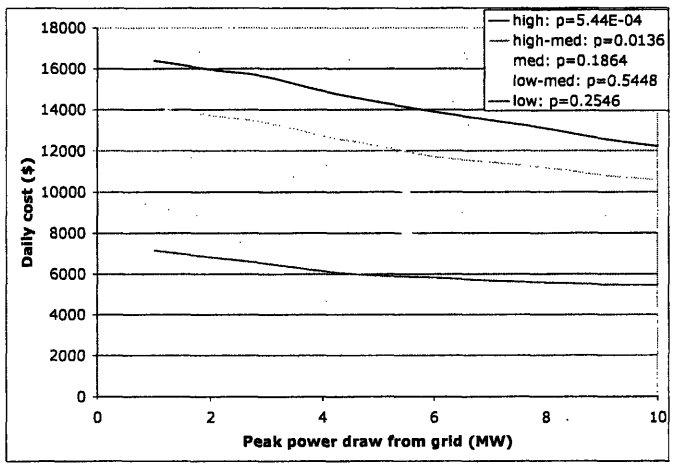


Figure 5-9: Cost curves for a user with load shifting and self generation capabilities. These curves show the tradeoff of daily cost and maximum power draw from the grid. The tradeoff is due to the fact that under a TOU plan, a customer would like to schedule all his jobs at the cheapest times, but when the maximum power draw is restricted, he cannot do this and therefore his cost goes up.

calculates the optimal static policy. Let u be static policy target demand level for each day. Let $g_i()$ denote cost curve i, which occurs with probability p_i. The expected monthly bill (the probabilistic version of Equation 5.2) is

$$\text{Bill}(u) = \text{NumDaysInPeriod} \cdot \sum_{i \in \text{Day types}} p_i g_i(u) + D(u) \tag{5.10}$$

The first observation is that for linear cost curves g_i (see Figure 5-9, which shows nearly linear functions) and a linear demand charge, the bill is linear in u and thus the optimal static policy is at one extreme point of the u range.

If the demand charge is zero before a certain cutoff value v and linear thereafter, then, depending on the slope of the demand charge function after v, the optimal static policy is either an endpoint of the u range or at $u = v$.

In both cases, the minimizer will in general depend on the number of days in the period. This is important in the dynamic programming solution as well. The decision to incur a large demand early on may be justified if there are enough days in

the upcoming period to capitalize on the fact that you have incurred a large demand charge already. In the section below, we compare the optimal static policy to the optimal dynamic programming strategy.

5.5.2 Optimal dynamic programming strategy

We solve the dynamic programming problem exactly using the numerical solver http://orc-pumba.mit.edu/dcraft/THESIS/DemandChargeApp/ . The expected electric bill under the optimal policy and the optimal static policy differ very little in this case due to the relatively static nature of the optimal policy, see Table 5.2. This is true regardless of the functional form of the demand charge because in this example, the slope of the daily cost curves is of similar magnitude for each of the five possible days. A static policy would be far from optimal if for example one of the daily cost curves was very high for all $u < 10$ but then suddenly dropped off at $u = 10$. If all of the other curves were nearly flat, and optimal policy would be to use the lowest possible u every day to avoid the demand charge, unless the unusual day type arrived, in which case you would switch to $u = 10$ from that day forward.

While in the above discussion, we see that a static policy is not optimal, we can show that the optimal policy is always a "climbing" policy in the case where that there is no peak augmenting random variable w_k^2. By a climbing policy, we mean that the optimal control (peak level chosen) at time $k+1$ is not less than the optimal control at k, and that as long as the daily curve type on day $k+1$ is the same, we guarantee that $u_{k+1}^* = u_k^*$. To prove this, consider the two alternatives. The first, that your policy gets more conservative, is incorrect since without random peak augmentation, you would never choose a smaller peak level then you have already experienced since you will save more money on that day and incur no larger a demand charge by choosing the current peak draw as your control. For the other case, we use the fact that general, regardless of the presence of a random peak augmenting term, we have $u_{k+1}^*(s) \leq u_k^*(s)$ since as the days progress the relative impact of the demand charge grows and thus you become more conservative regarding peaking.

Demand charge slope	5000	7000	10,000
Optimal static policy peak level	$u = 7$	$u = 6$	$u = 5$
Expected bill	\$237k	\$240k	\$ 240k

Table 5.2: Expected bill and optimal static policies shown. Optimal dynamic policy coincided with optimal static policy, so separate results not included. Notice that as the demand charge increases, the static policy lowers the daily peak until the demand charge is so high that the user chooses $u = 5$, the point after which a demand charge is levied.

5.5.3 Adding a peak augmenting term

In this section, we modify the above model by adding a random term which augments the chosen peak level u_k for each day k. We assume that the user chooses a level u_k but the actual peak turns out to be $u_k + w_k^2$ where $w_k^2 = 3$ MW with small probability q and $w_k^2 = 0$ with probability $1 - q$. Obviously the number of days left in the month is important now, for if there are many days left and q is not too small, there is a good chance of seeing that peak spike occur, but if the month is almost over, one can be more certain there will not be any spikes.

If q is very small, there is almost no chance of seeing a spike and one expects a static policy to be close to optimal. Also, if q is quite large, then the user will almost certainly see the spike and so will plan for it as if it was a certainty. In this case, a static policy will also be optimal. For values of q in between these two extremes, one expects to see the dynamic optimal policy dominate the static policy more noticeably. Table 5.3 shows this behavior by showing the percentage saved with the optimal policy versus the best static policy.

The savings reported in Table 5.3 are not large. This is due to the underlying cost curves for this user. It is the slope of a cost curve which sets the policy, not the absolute levels. This is because there is nothing one can do about the levels, but the u control can change the daily cost a lot or a little depending on the slope of the curve. The slope of the highest curve in Figure 5-9 is about 2.5 times the slope of the lowest curve. To see large savings using the dynamic programming approach presented herein, one needs to have possibilities of a daily curve with a very large slope, or even a sudden drop-off at a large u value. But even then, if the chance of

q, spike probability	.005	.01	.015	.02	.05
for DC slope=7000	0.36%	0.53%	0.57%	0.55%	0.27%
q, spike probability	.05	.07	.09	.1	.2
for DC slope=10,000	0.32%	0.55%	0.67%	0.66%	0.28%

Table 5.3: Percentage savings from using the optimal dynamic programming strategy versus the optimal static policy, as a function of the spiking probability. Notice that mid ranges of u show the largest savings, as discussed in the text. For larger demand charges (slope $\geq 20,000$), the optimal policy is to avoid the demand charge and hence use low u values. Since the spike level is only 3 MW in this example, the expected cost becomes independent of q since for $u \leq 2$, the demand charge is 0 even with spikes.

that day type occurring is high enough, the static policy chosen will reflect that and choose a high level of u, and if the chance is low, then the optimal static policy of a low u will typically do well also. So it is for moderate values of the probability of a 'difficult' day occuring which we expect the dynamic programming approach to be most useful.

Chapter 6

Case studies

In this section, we show the use of TOTEM for two different energy networks, and for the third case study, we apply the HVAC simulation and optimization routine described in Chapter 4. The first example is based on an operational hybrid network, which demonstrates TOTEM's capabilities of supply side control with storage, renewables, and fossil-based generation. Hybrid networks (referring to the use of both renewables and fossil-based generation) are common in rural off-grid village powering applications. Our example is modeled after a small scuba and fishing resort in Mexico.

The second case study is the campus power system of MIT, which runs a cogeneration plant providing both heat and electricity via a large turbine in a combined heat and power configuration. After initial analyses, showing how daily boiler and chiller decisions are made by TOTEM based on the daily demand curves and device characteristics, we extend the base MIT configuration to explore a future MIT network containing renewables and demand side control.

The third case study demonstrates the problem of optimal HVAC control. As described in Chapter 4, realistic HVAC optimal control instances are highly nonlinear and require special general optimization procedures not currently embedded in the core TOTEM linear software. As this case study demonstrates, simulation- and neural network-based optimization are hopeful candidates in the field of HVAC optimal control.

6.1 Hybrid networks: Costa de Cocos, Mexico

Costa de Cocos is a small eco-tourism center on the coast of Mexico, on the Yucatan Penisula. The system, as of 1999, consisted of a diesel 25 kW diesel generator, a 7.5 kW wind turbine, and a battery pack with storage capacity of 50 kWh. The network is shown in Figure 6-1. Load curves and other data are based on information within [18], which describes the performance of the system from 1997-1999. Based on the load curves shown, which peak around 3 kW, I have chosen to model the system with a smaller battery to demonstrate optimal storage control. The battery chosen is sized at 20 kWh. It may seem odd that such a large, 50 kWh battery is chosen for a system with the load curve shown in Figure 6-2. However, as reported in [18], the average wind power available is 1.8 kWh per hour[1] and the average load is 1.4 kWh per hour. Since these average levels are close, we expect that a large battery array may be useful, as shown in Chapter 7. A large battery is also a conservative choice, useful if there is no intelligent control system and if deep cycling is an issue.

We provide two case studies of Costa de Cocos. The first is an analysis based on today's reported load levels, specifically designed to demonstrate some of the non-obvious control strategies of TOTEM. The second analysis projects into the future, when demand levels rise in response to the availability of electric power. In this setting, the system size is much less conservative, and if the wind turbine fails, it might be that the system load is too high for the generator and battery to supply. If this occurs, the priority load curtailment model of Section 4.3 is appropriate, which we demonstrate.

6.1.1 Base Costa de Cocos model

Our initial modeling is meant to demonstrate TOTEM's ability to control the generator and storage in ways that are different than controls often applied in such systems. A typical heuristic for systems involving renewables is to charge the batteries when

[1]The unit kWh per hour is the same as kW, but I use kWh per hour in following conventional electricity parlance.

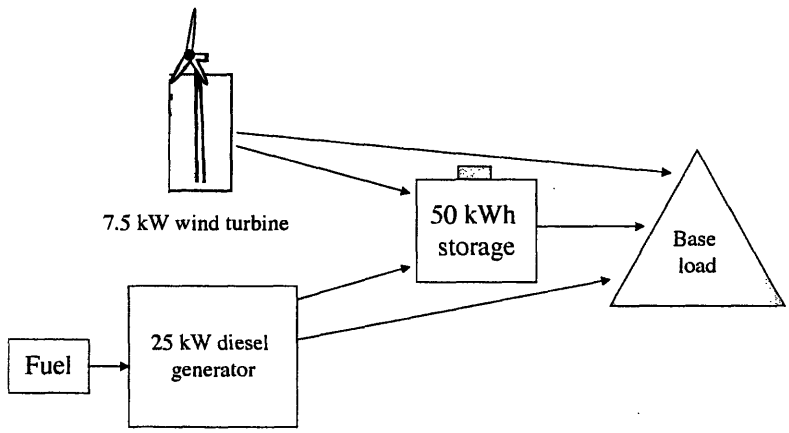


Figure 6-1: Hybrid power supply network at Costa de Cocos eco-tourism resort, Mexico.

there is surplus wind energy available, and drain the batteries when demand exceeds supply. The diesel is used as a back up generator in the event where there is not enough wind power and storage to satisfy the demand. Here are two scenarios where such control is suboptimal.

- You have full storage, and no wind in the future, but you hold on to your storage and generate now since the load is high now and will not be later, and thus you use the diesel generator more efficiently.

- You have high load now, lots of wind, and nothing in the battery. If in a few hours you expect low load but no wind, you will want to fire up the generator now to charge the battery so you do not have to supply the low load with the diesel generator.

In both of these, the suboptimality of the base control strategy stems from the inflexibility of the generator, and that efficiency is higher at full load than partial load. Doing better than the base strategy requires fairly accurate predictions of the

near term future. Base electric load is fairly consistent day to day. Wind on the other hand is more difficult, but it is not unreasonable to expect fairly accurate (90%) near term wind prediction with sophisticated prediction algorithms (see Chapter 8 for more information on weather prediction).

Generator capacity	25 kW
Generator lower operating limit	6.6 kW
Generator minimum up time	2 hours
Generator maintenance period	First 9 hours
Diesel fuel cost	\$5 per gallon
Fuel (F) to energy (E) conversion equation	$E = 14.14F - 0.52$
Battery capacity	20 kWh (50 kWh used in Section 6.1.2)
Battery minimum charge level	10 kWh
Battery initial charge level	15 kWh
Battery maximum charge and discharge rate	5kW
Battery storage efficiency	0.75
Battery storage restrictions:	final charge=initial charge=15 kWh

Table 6.1: Base operating parameters for the hybrid network of Costa de Cocos. Fuel to energy equation is a line fit from data within [18].

We study a 72 hour period with varying wind conditions and examine the non-obvious results of optimizing the energy flows. A minimum up time of 2 hours is imposed on the generator, and a maximum battery charging and draining rate of 5 kWh per hour. The AMPL file for this model is given in Appendix B.

The optimal strategy is shown along with the load curves in Figure 6-2. The turquoise and yellow curves, which show diesel generator activity, indicate that the generator is turned on for four brief periods throughout the three days. The first usages, at $t = 9$ and $t = 40$ hours shows that the generator waits until the last moment possible before recharging the battery (a lower bound of 10 kWh is imposed for the battery storage). For the 9th hour generator usage (and similar for the 40th hour usage), this is to take advantage of the fact that this is the largest base load level in the first 9 hours: the generator is more efficient the closer it is to its full operating capacity. This one generator attribute is behind all of the generator decisions in this example, and is a key strategic principle which TOTEM uncovers.

The third generator usage shows something interesting: the battery is not charged to its full capacity of 20 kWh, but only to a little more than 16 kWh. Even if it were recharged fully, it would still not make it through the rest of the time without another recharge, and this partial recharge allows the generator to recharge it at a time when the load is high, at $t = 57$, which is more efficient for the generator. The key lesson from this example is that when networks involve storage and diesel generators, coupling across time periods results in a complicated optimal plan which is not achievable through simple heuristic strategies. Note however that in this particular example, the heuristic policy mentioned above provides a same cost solution, $20 dollars for 3 days, due to the relatively constant generator efficiency, the over sizing of the equipment, and the fact that I did not alter the data to display the true suboptimality of the base policy.

TOTEM is also useful for system configuration and sizing studies. For example, we can investigate the daily cost savings associated with a larger battery. In the base curves shown, the total load is 25% larger than the total wind energy available over the 3 days simulated. Table 6.2 shows that, as expected, a larger battery offers a larger percentage savings if the average wind power available is closer to the average load.

	20 kWh Battery	50 kWh Battery	Percentage savings
Base wind power	$20	$18	10%
25% wind power increase	$12.5	$ 10	20%

Table 6.2: Costs for different battery size and wind levels

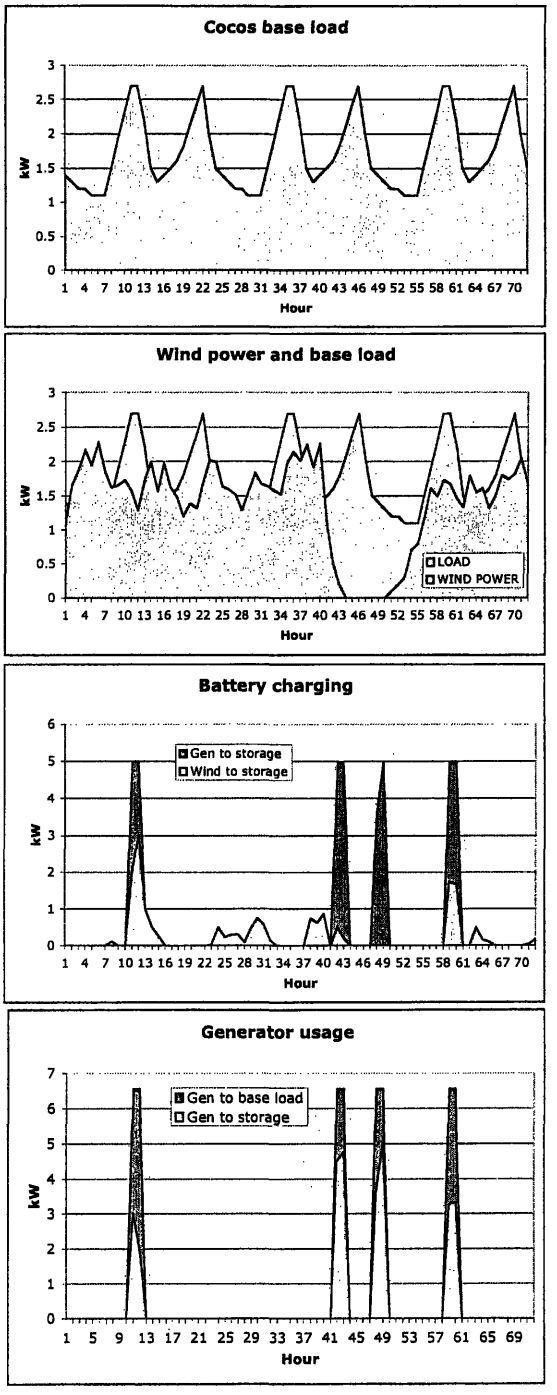


Figure 6-2: A three-day hourly simulation of the hybrid network at Costa de Cocos is studied, with the load and wind levels shown. The generator is turned on 4 times throughout the three days, due to insufficient energy from the wind turbine. The demand curve is taken from real site data, while wind curve is generated based on reported wind levels.

6.1.2 Cocos control with unexpected wind failure and subsequent load curtailment

The typical pattern seen in rural electrification efforts is the quick rise in demand once electricity becomes available. Accordingly, we project into the future to a time when the 7.5 kW (average) wind / 25 kW diesel / 50 kWh battery system at Costa de Cocos is just enough to supply the load. We assume that the base load is made up of 6 component base loads (maybe separated by region, or maybe by application) and each component load is individually known. Furthermore, we assume that the central controller has prioritized the loads in the event that load curtailment needs to be done. Figure 6-3 shows the wind pattern over the coming 72 hours, and also that at hour 25 the wind turbine fails unexpectedly, and remains down for the remaining period.

At the time of failure, TOTEM solves the priority load curtailment problem developed in Section 4.3, which is similar to a knapsack problem: choose from weighted loads the ones to curtail so that demand $\leq$ supply in the most cost effective way. With the prioritization coefficients given in Table 6.3, and the relatively small contribution of the wind turbine, TOTEM is able to drop just load 6 and replan as shown in Figure 6-4.

Load	Priority (M_i)	Average power level
Load 1	30,000	1.7 kW
Load 2	40,000	5.2 kW
Load 3	90,000	5.5 kW
Load 4	100,000	3.8 kW
Load 5	50,000	5.8 kW
Load 6	40,000	6.7 kW

Table 6.3: Prioritization coefficients and average power draws for 6 component base loads at Costa de Cocos.

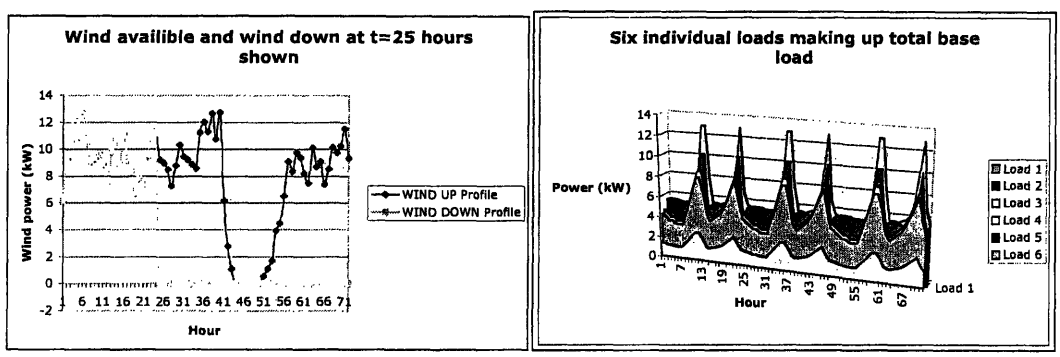


Figure 6-3: A three-day hourly simulation of the hybrid network at Costa de Cocos is studied. At hour 25, the wind turbine fails unexpectedly. The wind levels and 6 component loads making up the total system load are shown.

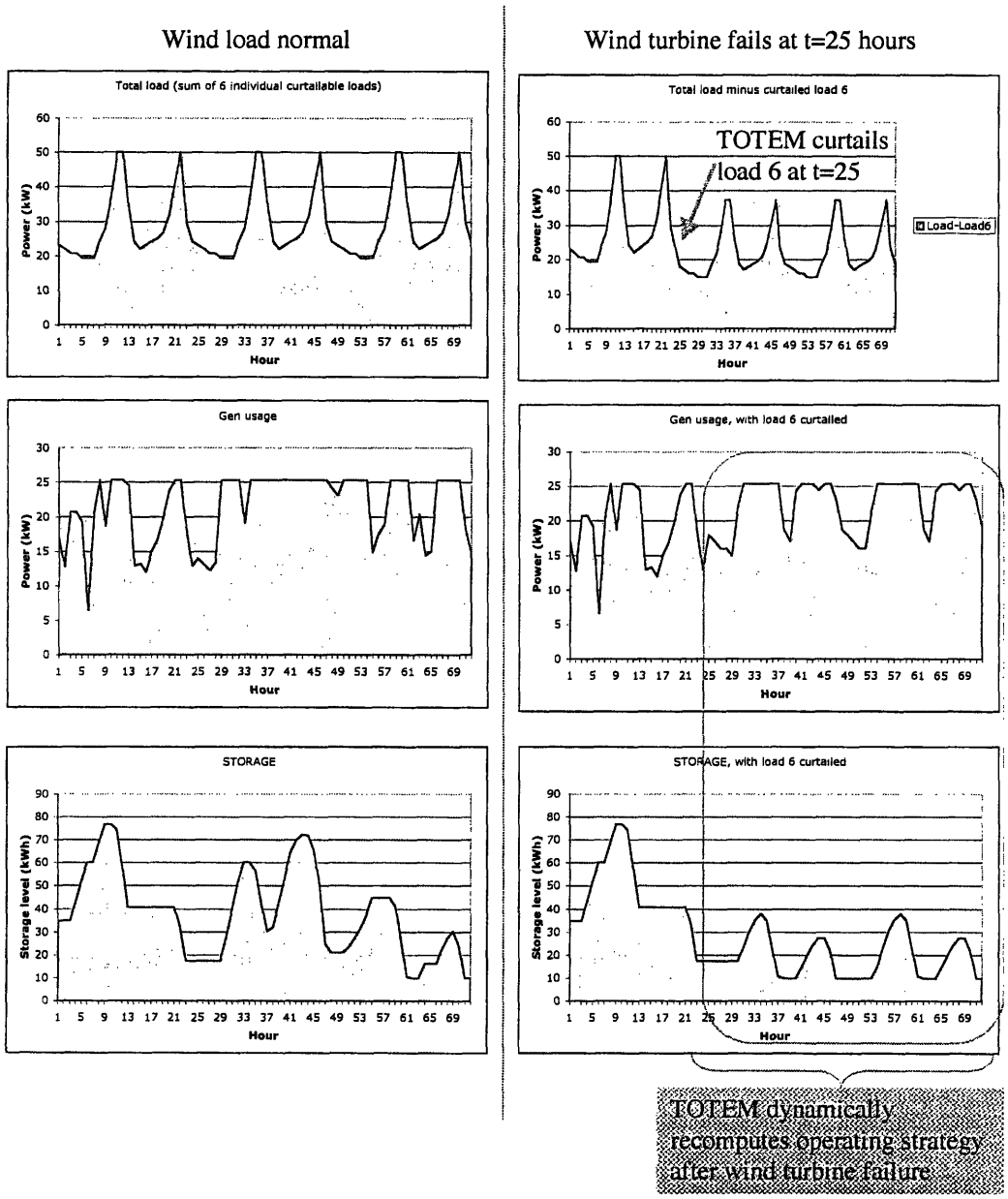


Figure 6-4: A three-day hourly simulation of the hybrid network at Costa de Cocos is studied. At hour 25, the wind turbine fails unexpectedly. TOTEM responds by shedding load 6 and dynamically recomputing optimal storage and generator strategies given the curtailment and the lack of a wind turbine.

6.2 MIT campus power system: cogeneration and chiller plant

MIT's power generation network offers a rich example of a supply side DG application. In this section we model MIT's full power generation plant and use real data for device definition, network connectivity, and hourly loads. Section 6.2.1 presents MIT's network, followed by Section 6.2.2 which presents the input-output relationships for the individual devices. Once the model is thus defined, we use TOTEM to derive optimal daily strategies for reprentative days of each season in Section 6.2.3. Stochastic planning for MIT is shown in Section 6.2.4, and the impact of adding cycling and reliability constraints are shown in Section 6.2.5. The final section on MIT, Section 6.2.6, is a look into the future, where MIT becomes a large player in the renewable energy field, and implements demand side management. This section demonstrates the full power of TOTEM and provides a concrete example of controlling supply and demand simultaneously (see the note on synergy, Section 3.13).

6.2.1 MIT energy supply network

MIT generates almost all of its peak electric load, which is about 22 MW. The exhaust heat from the 20 MW combustion turbine is captured by a heat recovery steam generator (HRSG) and used to produce steam for the campus. Three boilers are also available for additional steam generation. In the winter, the steam is used to heat the buildings on campus. In the summer, as is typical in efficient cogeneration networks, the steam from the HRSG is used to drive steam turbines which in turn provide energy to run 6 large chillers. The chiller capacities range from 1500 tons to 5000 tons.[2] East campus has three additional 1000 ton chillers run by electricity. Figure 6-5 shows the network details of MIT's power system.

[2]Tons is a power rating used for large chillers. 1 Ton equals 12,000 BTU/hr.

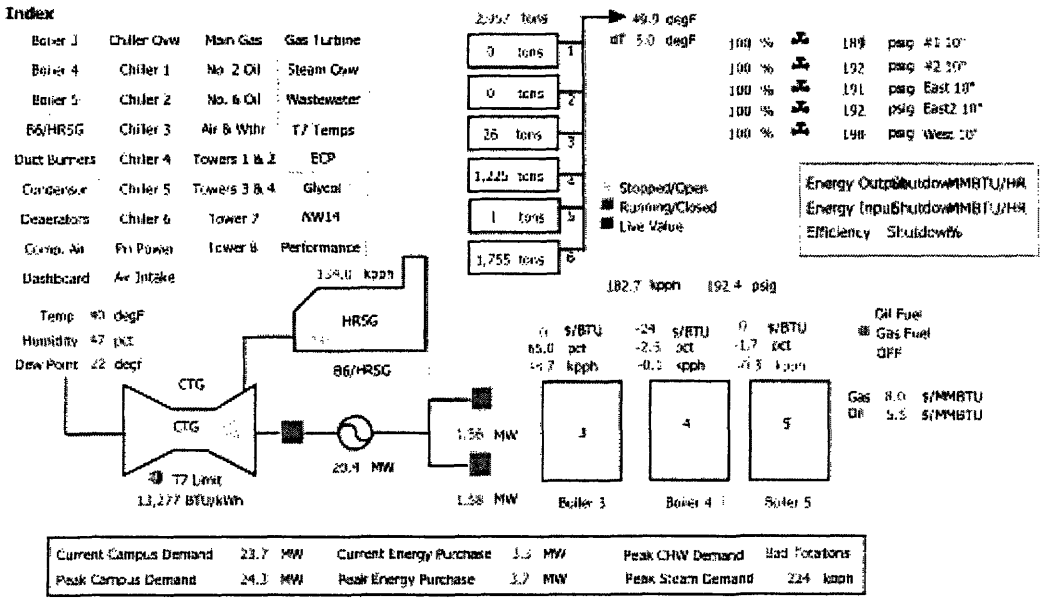


Figure 6-5: Energy flow diagram for MIT campus. CTG is the combustion turbine generator, which provides nearly all of MIT's electricity.

6.2.2 Input output relations for MIT network components

MIT's department of facilities has installed software called PI (Plant Information), which stores real time data every few seconds on every major flow in the network. This data is extremely useful for constructing io curves, since it reflects real operations rather than manufacturer specifications, which are highly idealized and do not reflect actual operating conditions (due to weather dependence and machine degradation over time). The PI software system, specifically an Excel add-in called DataLink, has been used to obtain data for the io curves of every power generation device shown in Figure 6-5. Curves from each power component, along with the linear regressions, are shown in Figures 6-6 to 6-12.

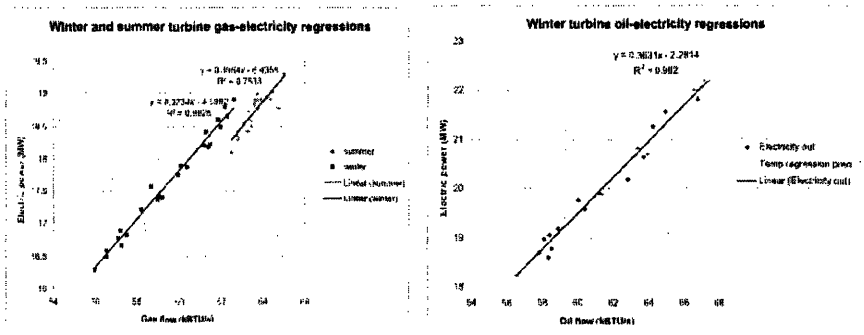


Figure 6-6: Gas turbine electricity output as a function of fuel inputs for winter and summer months. Higher output in winter is due to increased airflow mass rate from colder, more dense air.

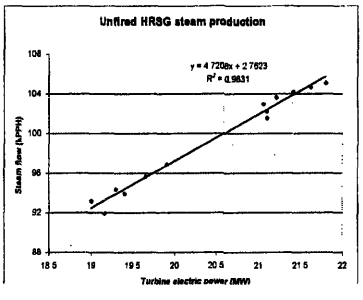
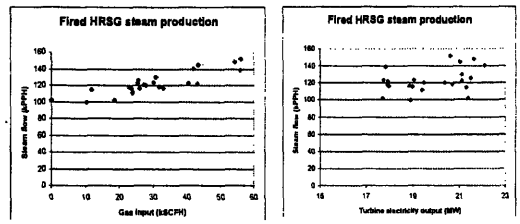


Figure 6-7: Steam output from the HRSG when it is not additionally fired. In this mode, the heat from the turbine exhaust is used directly to produce steam.



	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	11.19148997	7.881443968	1.456951	0.160651986	-4.831713922	27.21469387
Gas input	0.908493784	0.039586305	22.9497	7.69563E-16	0.825918237	0.99106833
Electricity generated	4.244589173	0.377799868	11.23502	4.31441E-10	3.456512824	5.032665522

Figure 6-8: Steam output from the HRSG when it is fired. Since steam output depends on both the electric rate of the turbine and the firing level, we perform a multiple linear regression and report the results below the two curves, which clearly show that a single linear relationship would be insufficient. It is more convenient to relate HRSG fired steam output to electric output from the turbine generator then to turbine exhaust heat supplied, since this quantity is not readily available.

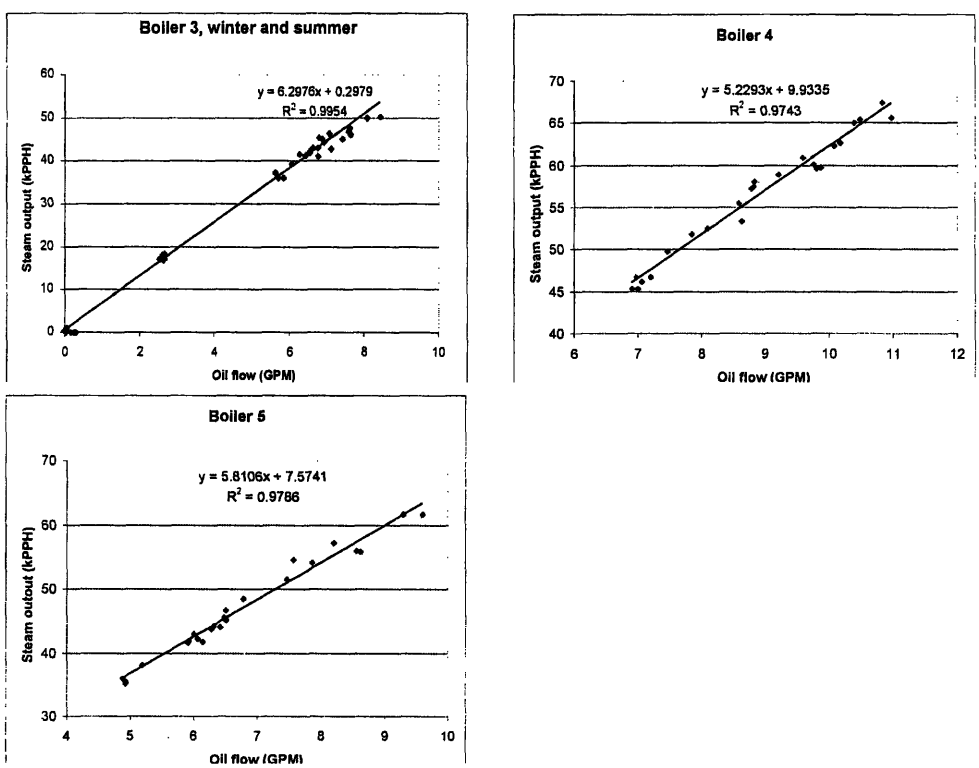


Figure 6-9: Steam production by MIT's three standalone boilers.

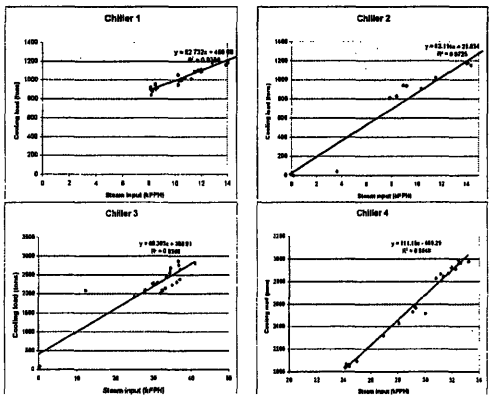


Figure 6-10: Chilled water production measured in chiller tons for steam driven chillers 1-4. Chillers 1 and 2 are sized at 1500 tons each, chiller 3, 3500 tons, and chiller 4, 4000 tons.

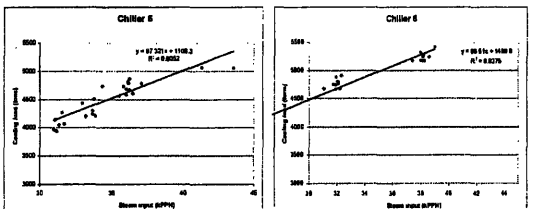


Figure 6-11: Chilled water production measured in chiller tons for steam driven chillers 5-6, MIT's largest chillers, 5000 tons each.

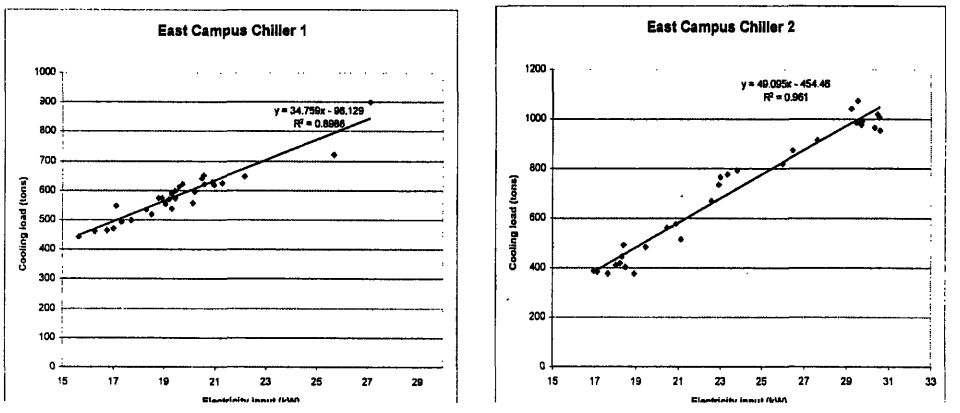


Figure 6-12: East campus 1000 ton electric chillers, located in the basement of building E40. Based on historical data, it appears that east campus chiller 3 no longer gets used regularly. Chilling power in tons was not available directly from the PI data system, and so it was calculated using the volume water flow rate and the temperature differential between supply and return water.

Description	Regression formula
	E = Electric power, G = gas flow, O = oil flow, S = steam flow, C = cooling load
Turbine gas to electricity (winter)	$E = 0.3734G - 4.5992$
Turbine gas to electricity (summer)	$E = 0.3954G - 6.4358$
Turbine oil #2 to electricity (winter)	$E = 0.3631O - 2.2814$
Unfired HRSG steam production	$S = 4.7208E + 2.7623$
Fired HRSG steam production	$S = 3.17G + 4.2446E + 11.1915$
Boiler 3 (oil6) steam production	$S = 2.520O + 0.30$
Boiler 4 (oil6) steam production	$S = 2.092O + 9.93$
Boiler 5 (oil6) steam production	$S = 2.324O + 7.57$
Boiler 3 (gas) steam production	$S = 2.274G + 4.164$
Boiler 4 (gas) steam production	$S = 2.809G + 1.366$
Boiler 5 (gas) steam production	$S = 3.748G + -5.332$
Chiller 1 cooling output	$C = 52.732S + 466.89$
Chiller 2 cooling output	$C = 83.114S + 21.834$
Chiller 3 cooling output	$C = 60.393S + 380.91$
Chiller 4 cooling output	$C = 111.15S - 659.29$
Chiller 5 cooling output	$C = 97.321S + 1108.3$
Chiller 6 cooling output	$C = 99.51S + 1489.9$
East campus electric chiller 1 output	$C = 34.759S - 98.129$
East campus electric chiller 2 output	$C = 49.095S - 454.46$

Table 6.4: Table of linear input output regressions of MIT power devices for use in TOTEM optimization. Oil and gas flows are put in the consistent units of KBTU per second.

6.2.3 Comparison of optimal seasonal strategies

Using average base loads from the winter and summer, we view the different strategies for the boilers and the chillers. The demands are shown in Figure 6-13. The large steam load in the summer is because steam is used to drive the steam turbine chillers.

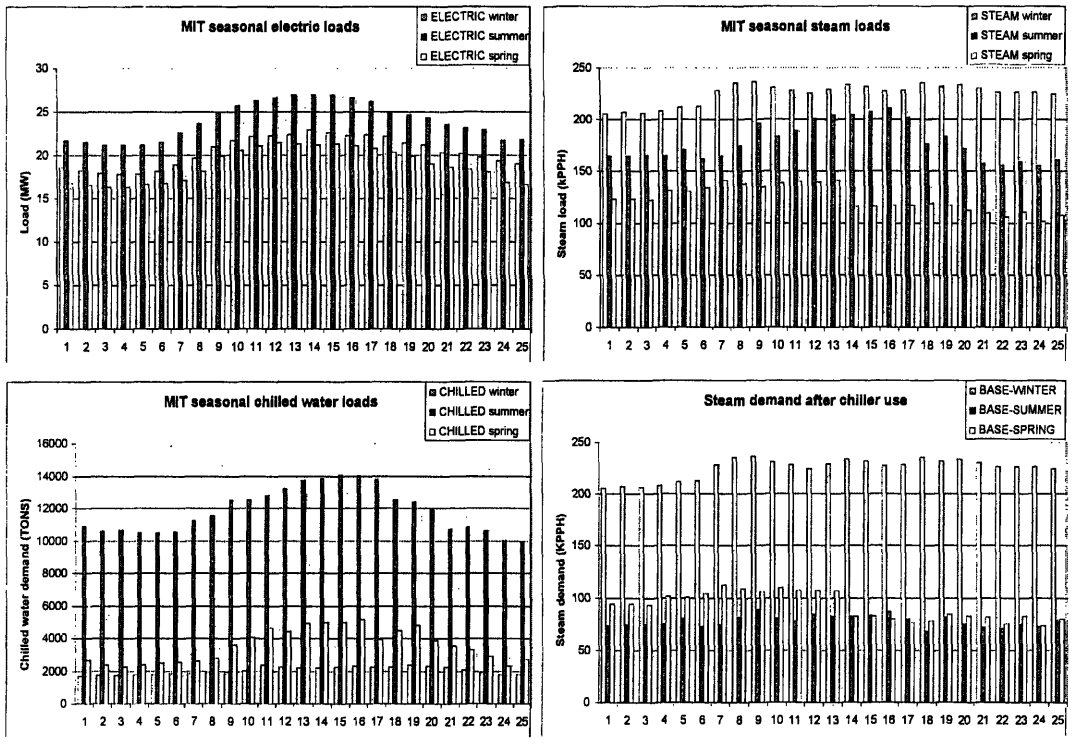


Figure 6-13: Typical demand patterns for a winter workday (1-7-04), summer workday (8-6-03), and spring (or fall) workday (5-22-02) at MIT. MIT is a summer peaking facility regarding electricity. It is also evident that whereas electric demand is smooth, chilled water demand is less so, and steam demand is even less so. Two graphs of steam load are shown. The upper right graph shows total steam load, including steam used to drive the chillers. The lower right graph shows the amount of steam used for application apart from the chillers, and hence shows much lower values on the warmer days.

On the winter day, the turbine is on full power the entire time, even though the electricity produced is more than required. This is due to the heat recovered in

the HRSG: it is cheap to generate steam via the HRSG, so a surplus of electricity is generated. If this surplus electricity cannot be sold to the grid, one will use an equality instead of an inequality for demand satisfaction of electricity. The results for the boilers and the steam driven chillers (east campus chillers are not used in the optimal solution) are shown in Table 6.5. Notice that boiler 5 is always used at its highest level and boilers 3 and 4 at their lowest. This is because boiler 5 exhibits an increasing efficiency with output level and boilers 3 and 4 decreasing efficiencies, as apparent from the io curves given in Table 6.4. Additional steam is supplied by Cambridge Steam, a subsidiary of NSTAR.

Cycling constraint for Chiller 4

In the optimal solution reported in Table 6.5, during the summer day Chiller 4 was on and off hourly throughout the day. Since this strategy could be considered unacceptable, we employ a minimum up time constraint, Equation 3.17, for the chillers of 6 hours. This causes the daily cost to increase to $36,407, a 7% increase from the base case.

Device	Winter day	Summer day
Boiler 3	On lowest	On lowest
Boiler 4	On lowest	On lowest
Boiler 5	On highest	On highest
Chiller 1	On, fluctuating near low	Off
Chiller 2	On, fluctuating full range	Off
Chiller 3	Off	Off
Chiller 4	Off	Sporadically on and off
Chiller 5	Off	Fluctuating in full range
Chiller 6	Off	Fluctuating in full range
Daily Cost	$26,207	$ 33,979

Table 6.5: Base case optimal control for a typical winter and a typical summer day, with HRSG unfired. Electricity and steam are purchased when there is insufficient local production. For example, with the numbers as is, it is cheaper to buy steam from Cambridge Steam than turn up boilers 3 and 4 any higher than their lowest set point.

6.2.4 Stochastic planning

Here we apply the stochastic TOTEM model presented in Section 4.2. One key challenge in stochastic planning stems from the fact that there is inertia regarding the states of the generation devices: they cannot be turned on and off in real time to respond to real time variations. We capture this inertia with minimum up and down time constraints. For example, a minimum down time constraint of 2 hours may reflect that it takes 2 hours of person work to bring a chiller online. These constraints make the stochastic problem difficult, for it may be desirable to bring a chiller online if the demand suddenly rises, but this cannot be done without 2 hours of preparation time.

Due to the large size of the MIT network model, solving to optimality even a simple 3 scenario stochastic problem takes too long to be practically useful. Fortunately, we can take advantage of the behavior of the intelligent branch-and-bound implementations, such as the one in CPLEX, in which good solutions are found quickly and then most of the optimal search is spent on reducing the small gap between the optimal solution and the highest lower bound found thus far. For example, in the study below, to find the optimal solution took over 2 hours, whereas a solution within .5% of optimality is found in about 5 seconds.

We consider a stochastic cooling load since the boiler strategy appears to be the same for summer and winter conditions, see Table 6.5. For ease of exposition, we will assume 3 scenarios are possible for the cooling loads of a spring day, see Figure 6-14. If the stochastic scenario curves are all in the same ballpark, then the stochastic problem is less challenging since, for a given overall level of cooling load, the chiller(s) that best match the load are chosen. In Figure 6-14 though, we can see that Chiller 6 is probably appropriate for the med and high curves, but Chillers 1 and 2 are better for the low curve.

Minimum up times of 6 hours are applied to all boilers and all chillers. Additionally, two extra constraints are added to Chiller 6: a minimum down time of 6 hours, and a constraint forcing it to be on for the first 2 hours. What these constraints do

together is force Chiller 6 to stay off until hour 9 if it is shut off at hour 3. This will become the key operational decision in this problem.

Optimal control is summarized for the individual scenarios, each run deterministically, in Table 6.6. Table 6.7 shows how the optimal stochastic strategy differs depending on the probabilities of each scenario. The key differences in the stochastic solution are that 1) Chiller 6 is shut down at hour 3, due to its minimum down time of 6 hours, and 2) Chiller 4 is fired up to compensate for Chiller 6 being turned off from 3-8. The expected cost of the stochastic solution shows that we do very well by discovering this optimal strategy that, while suboptimal for the individual runs, provides a good balance for unexpected future demands. An obvious choice for a heuristic strategy would be to run Chiller 6 always in anticipation of the high loads, which would yeild a cost of approximately $500 (per day) above the optimal solution.

	Low	Medium	High
Cost	$29,778	$30,364	$30,382
Strategy	Ch6 on till $t = 7$, then Ch1-2 on. B3-4 off.	Ch6 on always. Ch2 on $t = 8..13$. B3-4 off.	CH6 off at end, Ch1-2 on at end. B4 off, B3 on peak.

Table 6.6: Summary of individual optimal control for three scenarios run deterministically.

	Low	Medium	High
Probability	0.6	0.3	0.1
Cost	$29,905		
Strategy	Ch6 off at $t = 3$ all scenarios.	Ch6 on at 9 for med and high.	Ch4 used, on at 3, all B3 at end for med, high.

Table 6.7: Summary of stochastic optimal control for three scenarios.

6.2.5 Comparing the costs of cycling and reliability

For the spring, winter, and summer days shown in Figure 6-13, the base costs are $10,700, $26,400, and $20,700, respectively. Notice that the spring day, where the outside temperature is close to the desired indoor temperatures, is much cheaper than the other days.

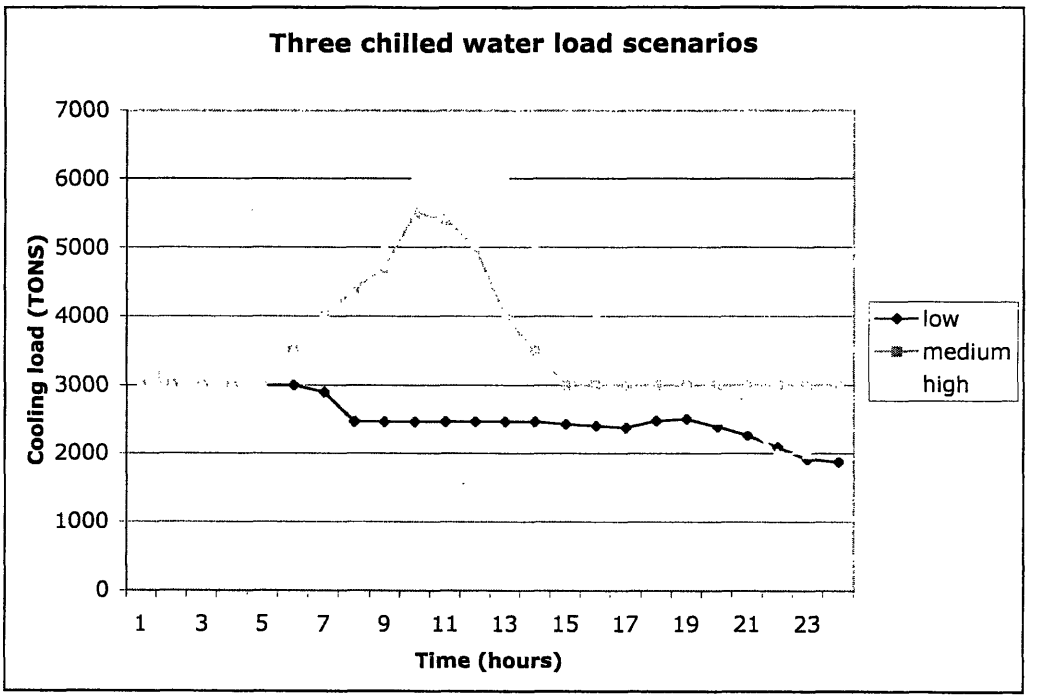


Figure 6-14: Three scenarios for chilled water load. The high and the medium scenarios branch off at hour 5, and they branch from each other at hour 7.

Cycling constraints were added for the boilers and the chillers, constraining the minimum up and down times to 6 hours. The winter base optimal control strategy had very little cycling, and in fact chillers 1 and 2 were both on the entire day, each with a brief one hour rest period. In contrast, the base optimal strategies for the spring and summer days had chillers and boilers being turned on and off hourly, responding to the demand fluctuations. Surprisingly though, TOTEM found solutions even with the cycling constraints which were within dollars of the base solutions in all cases. In the summer and the spring days, the solutions which satisfied the cycling constraints were quite different from the base solutions even though the resulting costs were very close.

For reliability, conversations with MIT plant managers revealed that reliability for

them meant keeping two boilers on at medium levels rather than having one supplying the full load. In the winter and summer cases, at least two boilers from boilers 3-5 are on all the time anyhow. For the spring day though, where one boiler is all that is required, enforcing the reliability constraint raises the daily cost to $14,300, a 40% increase. (Boilers 4 and 5 in particular have relatively large intercepts, which means two at partial loads can be more efficient than one at full load.)

There are two reasons why we do not see much penalty in the daily cost when we add cycling and reliability constraints in the MIT network. One is that MIT is not under an RTP plan, and so it is not the case that they might find it cheaper to buy then self generate. Additionally, the cogeneration heat capture makes the turbine very attractive to operate, and so even under RTP, one would need very low electricity prices to make it worthwhile to buy more from the grid. The second reason is that the time scale of large changes in demand levels for chilled water and steam are larger compared to the cycling constraints of six hours imposed above. Once the boilers and chillers are selected given the approximate ranges for the day, they are often the best ones to use throughout the day. In more volatile demand applications, cycling constraints will be more costly.

6.2.6 MIT: a look towards the future

While the current MIT network provides a vehicle for understanding supply-side management, it does not excercise many of the interesting capabilities of TOTEM: load control, renewables, and optimization under time-of-use pricing. In this section, we imagine a near-future MIT network that includes the following components, none of which MIT currently has:

- Centralized HVAC control in a large number of its buildings.
- Job scheduling (for example, lab experiments that draw a sizable fraction of the instantaneous power).
- Renewable power supplies: wind turbines and solar panels.

- A grid pricing tariff that attempts to better match supply and demand, such as real time pricing or time-of-use pricing.
- Additional self generation in the form of small dispatchable units such as microturbines.

	Daily total cost	Electricity used	% from grid	Fuel costs	
				Gas	Oil #6
Base	$10,371	578 MWh	15%	$3355	$1850
+HVAC control	$8,716	543 MWh	11%	$3326	$1860
+Job scheduling	$8,309	537 MWh	10%	$3335	$1856
+renewables	$6,445	538 MWh	4%	$3286	$1869
Base+renews	$8,002	578 MWh	8%	$3286	$1869
TOU pricing					
TOU	$11,002	578 MWh	15%	$3355	$1850
+HVAC control	$9,092	543 MWh	11%	$3317	$1864
+Job scheduling	$8,367	538 MWh	10%	$3332	$1856
+renewables	$6,361	545 MWh	4%	$3310	$1859
TOU+renews	$8,364	578 MWh	8%	$3331	$1860

Table 6.8: Comparisons for various futuristic MIT settings. We present results for a flat rate plan and a TOU plan, shown graphically in Figure 6-15A. The average of the TOU rate is equal to the flat rate. Under each rate, we successively add in load control and renewables. Thus, the listing "+renewables" includes HVAC control, job scheduling control, and renewables. The final entry in each section has no load control, just renewables, and is present to show that your incremental gain from having just renewables is significantly less than if you have renewables and load control.

Table 6.8 shows the results of successively adding components into the base MIT network. Job scheduling, as expected, makes more of an impact under the TOU tariff. The savings from job scheduling under the flat tariff is due to shifting loads together for better usage of the generator. The last two entries in each section, Base+renewables and TOU+renewables, are included to emphasize the importance of having load control when renewables are added to a network (without storage), since load control allows you to shift work into the periods where the renewables are producing energy.

In this example, the TOU rate is not desirable until you add in both types of load control and renewables. This is because in order to take advantage of the varying

hourly price, MIT must be able to shift away demand from the peak times. Imposing the TOU tariff without load control hurts MIT since its demand peak coincides with high TOU prices.

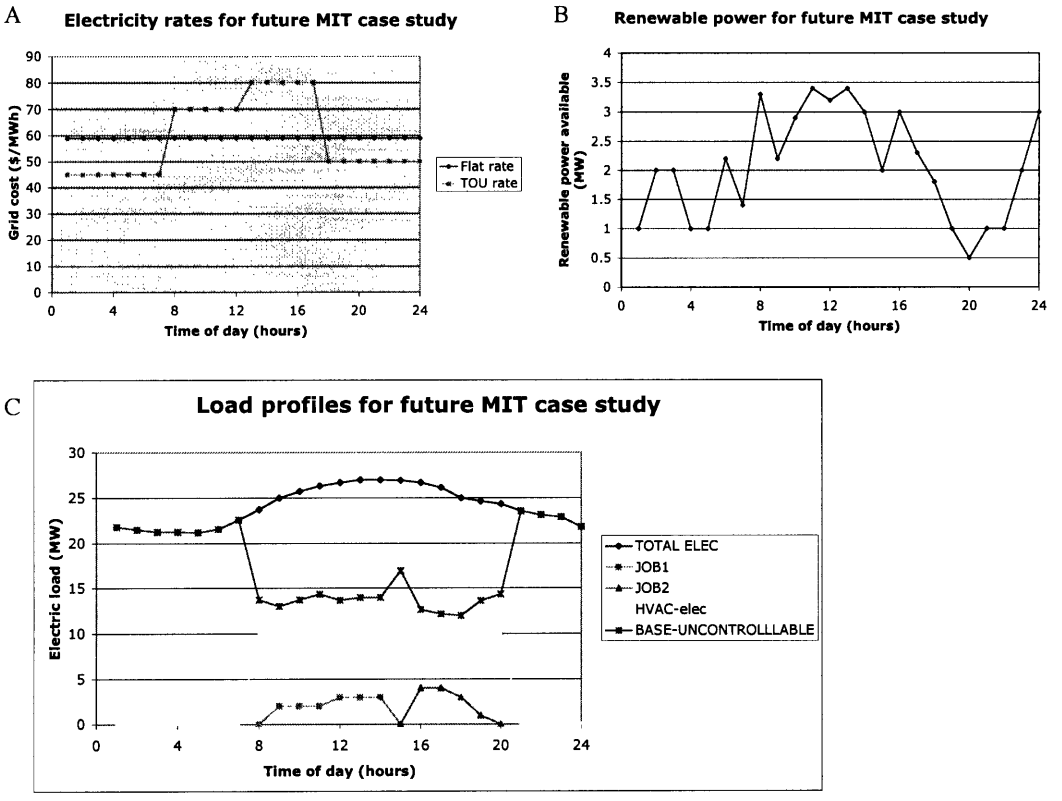


Figure 6-15: A) Flat rate and TOU rates, where average of TOU rate is equal to the flat rate. B) Aggregate power available from a network of renewables, possibly consisting of some combination of wind turbines and solar panels. C) We use an actual aggregate summer electric load curve from the MIT PI system database. We fictionally break this curve up into component pieces, including default HVAC control, two schedulable jobs, and the remaining uncontrollable base load.

Adding microturbines

In the above examples, a large portion of the bill comes from grid purchases of electricity. For example, in the TOU pricing with renewables run, $3,172 out of a total of $8,364 comes from grid purchases. MIT is currently considering an additional turbine engine to their DG network. In this section, we consider the TOU+renewables

network and add to it three different sized microturbines, for increased flexibility and control options. The specifications for the machines, along with the existing MIT gas turbine for comparison, are given in Table 6.9. Since each of those devices is the most efficient one at its peak operating range (i.e. if the base load was 1 MW, you would do best to choose the 1 MW microturbine), TOTEM will swap between them in real time to keep the system running at optimal efficiency. This is of course complicated in the presence of minimum up and down times, but in the analysis results shown in Figure 6-16 we do not impose such constraints.

Device	IO equation	low output, efficiency	high output, efficiency
500 kW turbine	$P = 0.4F - 0.1$	204 kW, 0.268	508 kW, 0.334
1 MW turbine	$P = 0.42F - 0.2$	0.44 MW, 0.288	1.1 MW, 0.354
2 MW turbine	$P = 0.43F - 0.4$	0.91 MW, 0.298	2.2 MW, 0.364
20 MW turbine	$P = 0.37F - 4.6$	16.3 MW, 0.291	20.4 MW, 0.305

Table 6.9: Operating ranges and efficiencies for 3 microturbines and existing MIT gas turbine. The microturbines show increased efficiency for higher operating ranges, and that each of them is the best turbine for satisfying load at its upper range. P denotes power output (MW), F denotes fuel input (gallons)

With the three added microturbines, the grid purchases are almost completely eliminated, and the resulting daily cost drops from \$8364 to \$5544. Figure 6-16 displays a change in chiller strategy in the microturbine network, but as described above, the MIT network is robust in the sense of having multiple solutions close to optimal. Since the boiler strategy does not change perceptibly, and the boilers feed the chillers, we would not expect a change in chiller strategy, thus we are merely seeing an artifact of the optimization (the solutions presented are solved by branch and bound within 0.2% of optimality).

Analyses like the ones presented in this section, done for many more days of the year, can assist planners in assessing hardware options for handling demand growth.

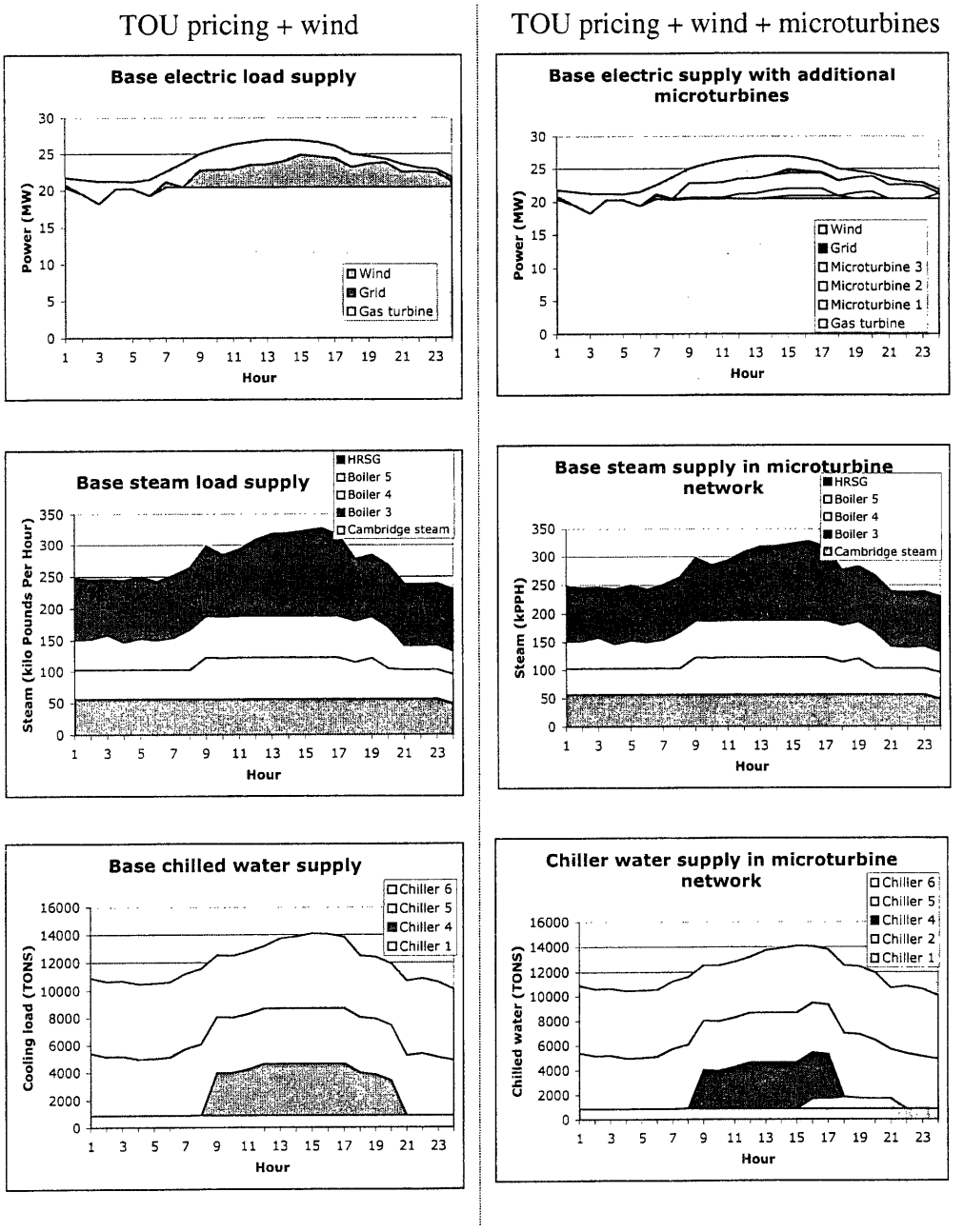


Figure 6-16: A comparison of optimal strategies for a summer day. The top right figure shows the optimal microturbine strategy used during the afternoon peak. The cost savings, roughly \$2,800, is large in this new network since expensive grid purchases are almost entirely eliminated.

6.3 HVAC optimal control study

In this section we present an application of the HVAC optimal control algorithm described in Section 4.5. The goal is to select the hourly controls (thermostat levels, flow temperatures, fan speeds, etc.) to minimize a function which combines the real time costs of energy and thermal comfort levels. Due to the nonlinearity of HVAC control, the algorithm involves computationally expensive building simulation and a generic optimization scheme.

The base model used in the study is the model from Helen Xing's recent MIT doctoral thesis [63]. The physical building being modeled is a three room top floor of a commercial building with a gross area of 126 square meters. The HVAC system is a variable air volume (VAV) system, which contrasts with the older constant volume (CV) systems. A VAV system, where each room has its own VAV box fan and thermostat settings, offers more control and hence more efficiency potential than a CV system. A simplified diagram of the HVAC control system and the three rooms is shown in Figure 6-18. We model a summer day in San Francisco using weather data provided with the EnergyPlus software.

We extend Helen's work in three ways:

1. Additional control options explored. We simultaneously allow the following to vary in the optimization: hourly thermostat set points, hourly chiller and fan on/off, hourly chilled water temperatures, hourly cold air supply temperatures, and fan capacity rating (a hardware optimization choice).

2. More robust global search/local search hybrid optimization algorithm used. The algorithm chosen, which begins with a particle swarm search and finishes with a Hooke Jeeves search, called GPSPSOCCHJ in GenOpt 2.0 [60], was not available at the time of Helen's thesis writing, but it is a very good choice for this type of optimization.

3. More realistic cost model used, which includes a comfort term, a demand charge, and real time pricing. Since intelligent HVAC control is a more interesting and

a more important problem under dynamic pricing tariffs, we use such a setting here. The result is a highly non-linear, non-convex cost function strewn with local minima just waiting to trip up the best optimization packages.

6.3.1 Cost function

We assume the customer faces known real time hourly prices c_t, which we obtain from historical market clearing prices from the New England Independent Systems Operators (ISO), averaged over the month of August in 2002 (courtesy of Steve Connors, MIT). This data is shown in Figure 6-17. The customer's daily electric charge is then given by $\sum_t c_t e_t$, where e_t is the electric energy used in hour t.

When a demand charge is also added, we use a linear demand charge with slope of $s = 150$ for every unit of peak draw above $p = 120$ (units scaled from Joules/sec, the power unit used by EnergyPlus).

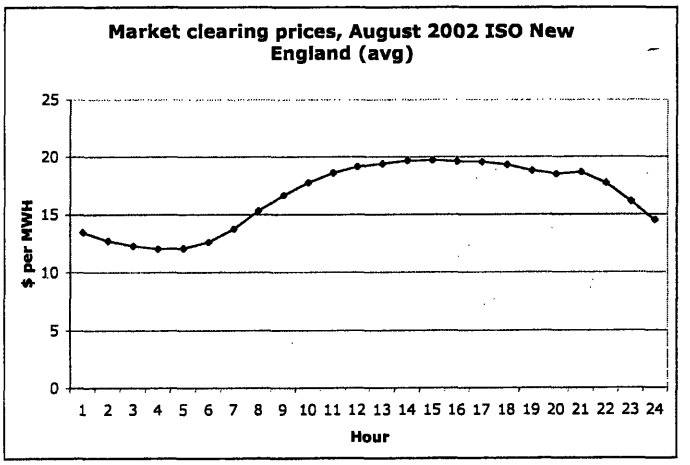


Figure 6-17: New England ISO market clearing prices for grid electricity. Hourly data is averaged across all 31 days, but a typical day also looks like this (i.e. there is not much variance to the hourly prices).

The other component of the cost function is the comfort term. Without embarking on a thesis discussing possible ways one might tradeoff comfort and cost, we simply

propose the following comfort term for our cost function. It combines a daily average comfort level and also penalizes the daily maximum discomfort level. We use Fanger's [21] PPD (predicted percentage dissatisfied) measure, derived from PMV (predicted mean vote), which is available each hour in each room in an EnergyPlus simulation. The PMV is averaged across the working hours and the rooms, and converted to PPD using Fanger's equation:

$$\text{PPD} = 100 - 95e^{-(0.03353(\text{PMV})^4 + 0.2179(\text{PMV})^2)} \tag{6.1}$$

Maximum PPD is obtained from the hourly room averages, maximum taken over the working hours of the day.

The complete objective function is then constructed as follows:

$$\text{Objective function} = \underbrace{\sum c_t e_t}_{\text{Energy cost}} + \underbrace{s * (M - p)^+}_{\text{Demand charge}} + \underbrace{\lambda_1 \text{PPDaverage} + \lambda_2 \text{PPDmax}^2}_{\text{Comfort term}} \tag{6.2}$$

where M is the maximum power drawn over the day, and λ_1 and λ_2 are weighting factors. Note that while demand charge is typically levied over the entire month, the approach we take here, having a demand charge term in the daily cost function, is equivalent to constraining the maximum power draw below some value, the procedure recommended in Chapter 5, via a penalty in the cost function.

6.3.2 Controls

The optimization routine used, GPSPSOCCHJ, allows continuous and discrete optimization variables. Thermostatic set points, chilled water temperatures, and supply air temperatures are modeled as continuous variables, and chiller and fan usage are modeled as binary variables. Ranges are given in Table 6.10. These controls are displayed in bold in the HVAC system diagram, Figure 6-18.

Variable	Description	Lower	Upper
Ti1-Ti10	Thermostat set points for 10 working hours	22 deg C	28 deg C
ChWT	Chilled water temperature	6.67 deg C	12 deg C
SAT	Supply air temperature	6.67 deg C	12 deg C
Fi1-Fi8	Fan on/off for 24 hours, in 3 hour bundles	off	on
Ci1-Ci8	Chiller on/off for 24 hours, in 3 hour bundles	off	on
FanPressure	Used as surrogate for fan capacity (discrete)	450 Pascals	600 Pascals

Table 6.10: Control variables in HVAC simulation/optimization study.

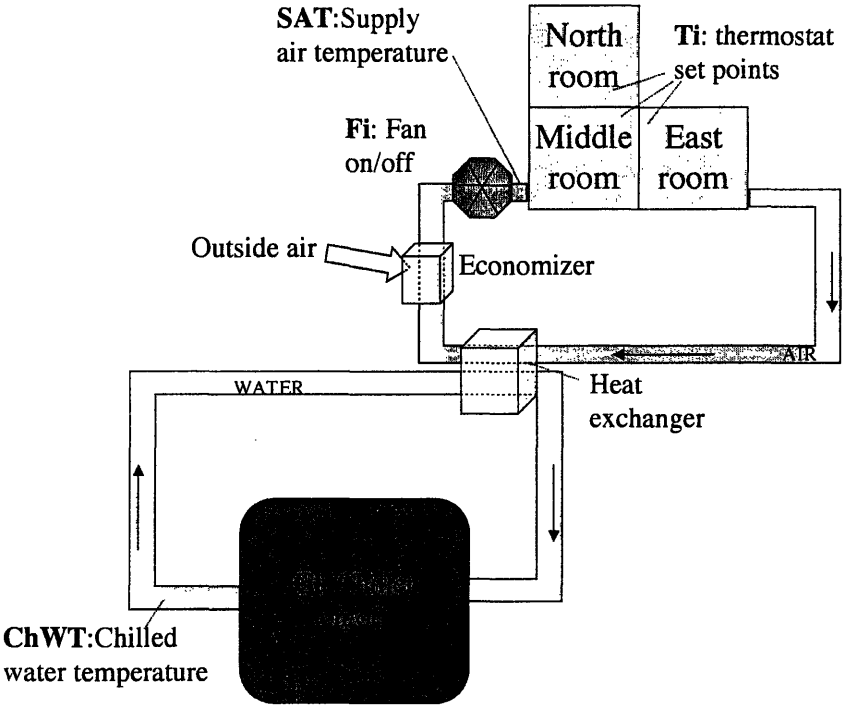


Figure 6-18: HVAC VAV system for simulation studies, with control variables shown in bold. Subscript i refers to hourly indexed control variables. Description of variables is given in Table 6.10.

6.3.3 Optimization results

The EnergyPlus building simulation program is used to compute the energy costs associated with each possible control vector. Optimization is done by the GenOpt

major improvement is made when all possible controls are used in the optimization. In fact, the savings is less than 1% if just thermostat levels are adjustable (cost goes from $2,547 to $2,533). When chiller usage, fan usage, and flow temperatures are optimized over, the cost drops to $1,911, a savings of 25%.

In the original model, all three rooms are on the same thermostatic set point schedule. While this might be an effective strategy from a total energy usage standpoint, if a demand charge is included it might be important to stagger the individual room's demands to keep the maximum power draw down. In the final result of the table, we allow the three rooms to each have their own thermostat schedule. This increases the number of variable significantly. In Particle Swarm and other population-based optimization algorithms, one uses more trial solution vectors (called "particles" in PS algorithms) when the number of variable is greater. This decreases the chances of converging to a local optimum. For example, when run ALL-DC of Table 6.11 was run with 10 particles instead of 40, the optimizer converged to a value of $2,443 rather than the reported $2,419. We used 40 particles for the optimization with 3 different thermostat set point schedules. As expected, 3 room separate thermostat control is able to reduce the demand charge and hence reduce the total cost from $2,419 to $2,283.

Run name	Brief Description	Objective
	NO DEMAND CHARGE	
Base	Static default controls applied	$2,547
Ti-HJ	Just vary thermostats, Hooke-Jeeves algorithm	$2,540
Ti-GPSPSOCCHJ	Just vary thermostats, hybrid algorithm	$2,533
ALL	All controls available, hybrid algorithm	$1,911
	WITH DEMAND CHARGE	
Base-DC	Static default controls applied	$3,431
ALL-DC	All controls	$2,419
ALL-3RoomTi-DC	Each room with separate thermostat	$2,283

Table 6.11: Results from HVAC optimization runs.

Chapter 7

Theoretical investigation of storage in a stochastic environment

In this chapter we provide a model to assess the suitability of storage in stochastic supply and demand networks. The configuration we study is one of stochastic power sources such as wind or sun, with auxiliary power available from the grid. Demands may also be stochastic, and we ask the question, under what conditions is a battery suitable for the user? Such work is helpful in highlighting key aspects of power configurations which make them likely candidates for adding storage capacity. This storage may be on the large scale, such as a wind power network with hydro-storage, or a smaller scale DG application. The modeling in this chapter complements what TOTEM provides: whereas TOTEM is giving detailed tactical planning, this chapter provides a simple model which yields higher level observations regarding the suitability of storage.

For the model presented herein, we assume that additional power is supplied at a constant (as opposed to time-of-use) rate, thus the optimal strategy is to use the battery to supply demand when possible. In this analysis, we ignore battery cycling and depth of discharge issues, and instead focus on the general characteristics of supply and demand which result in favorable battery opportunities. We assume a discrete time model where energy supplies and demands are constant (lumped) quantities for each time interval.

In one extreme, supply always matches demand and therefore there will be no opportunity for battery storage. With stochastic environments, this will not be the case, but it is an important end case to keep in mind. Batteries are also not helpful if either supply always exceeds demand (in which case there is always enough power for direct usage) or demand always exceeds supply (in which case there is never surplus power to store). So, the situations where batteries offer improvement are ones where sometimes supply is higher than demand (battery gets charged) and sometimes demand is higher than supply (battery gets drained). In the analysis and modeling that follows, we attempt to refine this observation.

We consider the battery storage level as the state variable for a discrete time Markov chain. When the battery plus the stochastic source cannot satisfy the demand, power is assumed to be obtained from the grid at a set cost per kWh. Grid power is never used to charge the battery since it is always available directly and therefore without the loss associated with battery storage. The optimal strategy is to use your stored electricity right when you need it, and the battery storage level can then be defined as the Markov chain depicted in Figure 7-1c. In this model, we assume that the power supplied and demanded at each time interval are independent of each other and past levels, and that they are discrete levels whose difference (supply−demand) is a positive integer multiple of the smallest possible non-zero jump of size Δ. The battery can hold a maximum of $B = M \cdot \Delta$ kWh of energy.

It is straightforward to write the Markov chain transition probabilities given supply and demand probability mass functions such as those in Figure 7-1a and b (see Appendix D). Let the state k denote the battery storage level $k\Delta$. Let D_t and S_t be the amount demanded and supplied at time t, respectively. We assume that both of these random variables are drawn from independent and time independent distributions D and S with means $\bar{D}$ and $\bar{S}$. The purchase level PL is the average amount of electricity which needs to be purchased at each time step. The general result for no battery storage available is

$$\mathrm{PL}_{B=0} = E[(D - S)^+] \tag{7.1}$$

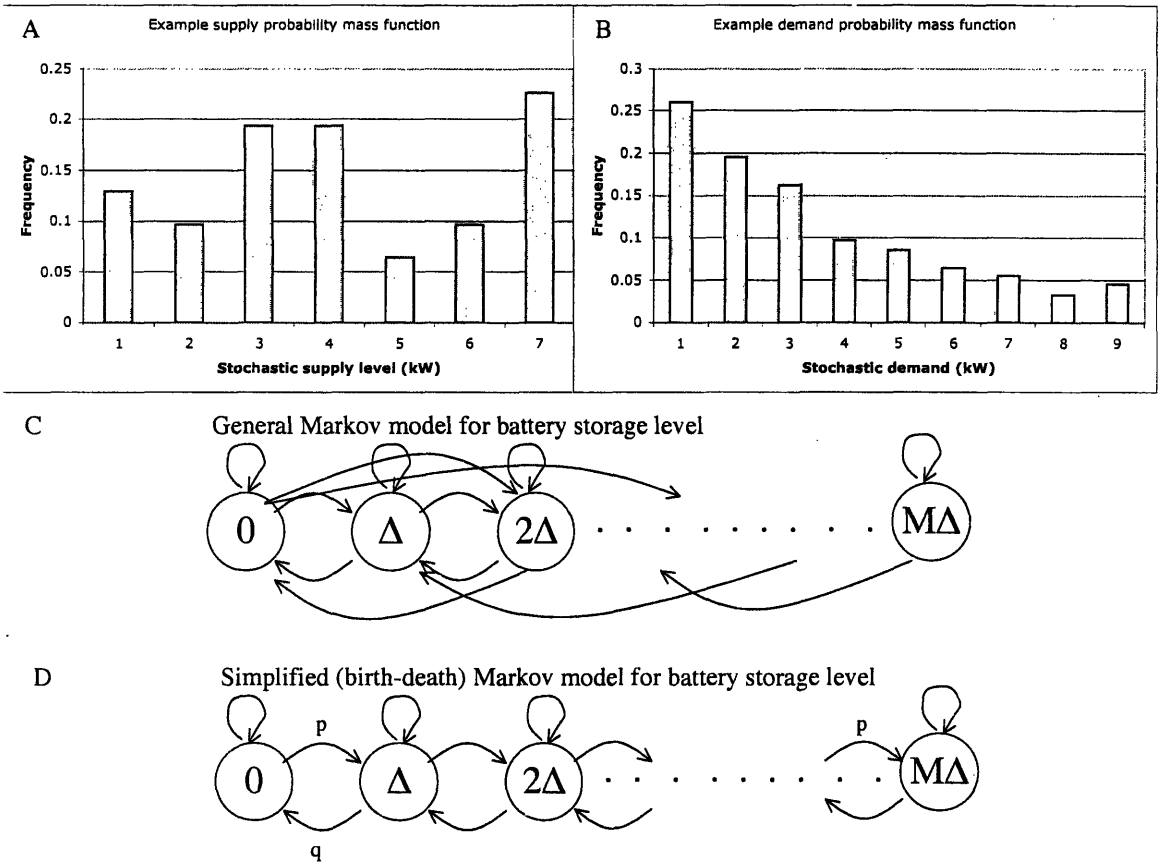


Figure 7-1: A) Example probability mass function (PMF) for a stochastic supply source, such as a wind turbine. B) Example PMF for a stochastic demand. C) General Markov chain representation where the Markov state is the amount of storage in the battery at time t. Arbitrary jumps represent the different possibilities for (supply − demand) at each time interval. The size B of the battery determines the maximum size of the chain, $M\Delta$. D) Simplified chain for the restricted case where supply and demand differ by $\pm\Delta$ or 0.

because it is only when D_t exceeds S_t that a purchase needs to be made, and in that case, since there is no storage, the entire quantity must be purchased. On the other end of the spectrum, we have infinite battery storage capacity, in which case

$$\mathrm{PL}_{B=\infty} = (E[D] - E[S])^+ = (\bar{D} - \bar{S})^+. \tag{7.2}$$

This result can be obtained by considering any sample path of the Markov chain. If

$\bar{S} > \bar{D}$, then the storage level will approach infinity over time, and hence the purchase level will be zero. On the other hand, if $\bar{D} > \bar{S}$, then the battery allows us to store any excess at each time, and thus all we have to purchase in the long run is the average difference $\bar{D} - \bar{S}$. We expect storage to be most beneficial when there is a large difference between PL for zero batteries and PL for an infinite size battery. In the following section, we show analytically that this is true for a simplified model.

7.1 Analysis of the birth-death storage Markov chain

To gain some more insight into the problem, we will analyze in detail the simpler chain shown in Figure 7-1d. For this process, supply and demand are either matched or they differ by $\pm\Delta$, and thus we have a birth-death Markov chain, for which we can easily write down the steady state probabilities π_i. Let $p = \text{Prob}(S - D = \Delta)$ and $q = \text{Prob}(D - S = \Delta)$. The self loop transition probability is therefore $1 - p - q$. The steady state purchase level is related to the steady state probability π_0 for the simple birth death chain by the following equation:

$$\text{PL} = \pi_0 q \Delta. \tag{7.3}$$

The equation has the steady-state probability π_0, the fraction of time that the state of the chain is in state 0, multiplied by the fraction of transitions where demand is larger than supply, q, and then scaled by the size of the unmet demand, Δ. For the finite birth-death chain of Figure 7-1d, we have

$$\pi_0 = \frac{p - q}{p(p/q)^M - q}. \tag{7.4}$$

This leads the following expression for steady state purchase level:

$$\text{PL} = \frac{q(p-q)}{p(p/q)^M - q}\Delta = \frac{q(p-q)}{p(p/q)^{B/\Delta} - q}\Delta. \tag{7.5}$$

For the special case where $p = q$, the limit of Equation 7.4 is $1/(1+M)$ which gives:

$$\text{PL})_{p=q} = \frac{q}{1+M}\Delta. \tag{7.6}$$

For the following sensitivity results, we will assume that $q = 1 - p$ (i.e. no self loops) in order to reduce the number of independent parameters. We are interested in how the purchase level compares for various values of p, probability that supply exceeds demand, B the battery size, and Δ the jump size. For $q = 1 - p$ we have

$$\text{PL} = \Delta\frac{1+(2p-3)p}{1-p-p(p/(1-p))^{B/\Delta}}. \tag{7.7}$$

which is increasing in Δ (note, this is not completely obvious with a quick glance, but for $p > 1/2$ the numerator and denominator are both negative, for $p < 1/2$ both positive, and thus the $(p/(1-p))^{B/\Delta}$ term also makes PL increase with Δ). This can visualized from the fact that for an increase in Δ, the chain shortens, thus forcing the state to be at the 0 end longer, where purchases are made.

Since purchase level increases with increasing jump size, we investigate how battery size can compensate for this. Figure 7-2 displays this tradeoff. We observe that for small battery sizes, the impact of Δ is greater than for larger battery sizes.

Next we investigate how PL varies as a function of battery size, for various jump sizes. Clearly as B increases PL decreases, but we find that this relationship depends strongly on the value of p. For small values of p, we know that even with an infinite capacity battery, there where still be a non-zero value of PL, as indicated by Equation 7.2. In this regime, the difference between the no battery limit and the infinite battery limit is relatively small, and thus batteries cannot help too much. In contrast, for $p \geq 0.5$, batteries can reduce the average purchase level by a large amount. Notice that this effect is not monotonic in p, and in fact for $p = 0.5$ (in general, when $\bar{S} \approx \bar{D}$) the amount that batteries can save you is maximized. This is observable in Figure 7-4 which displays that the largest differences between the no battery limit and the infinite battery limit are for p values close to 0.5. In fact, for p close to 0 or 1, batteries are not helpful, as discussed earlier in this chapter. To show

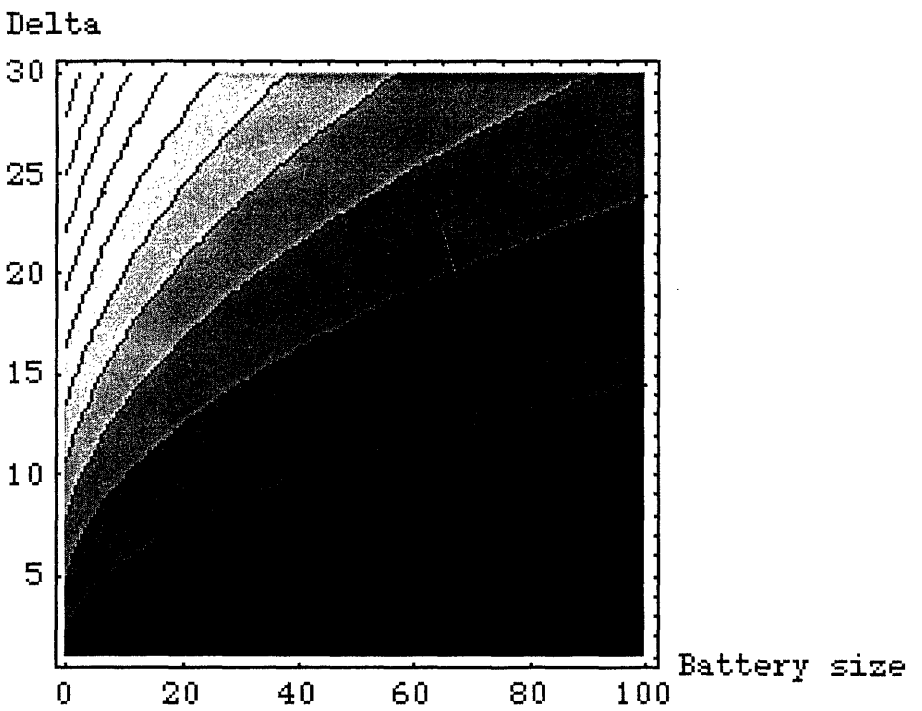


Figure 7-2: Isocurves of purchase level as a function of jump size Δ and battery size B.

that batteries are maximally marginally effective at $p = 1/2$, let us consider a Taylor series expansion of Equation 7.7 around $B = 0$. We get:

$$\mathrm{PL}(B) \approx \Delta(1-p) + \frac{(1-p)p\ln\left(\frac{p}{1-p}\right)}{2p-1}B \tag{7.8}$$

The term multiplying B is 0 at $p = 0$ and $p = 1$, and is minimized at $p = 1/2$, at a value of $-1/2$. Thus we see that the marginal contribution to purchase level savings is achieved when $p = 1/2$. This is in addition to the fact that the total achievable

improvement by batteries (defined as the difference between the zero battery PL limit and the infinite battery PL limit) is maximum for $p = 1/2$.

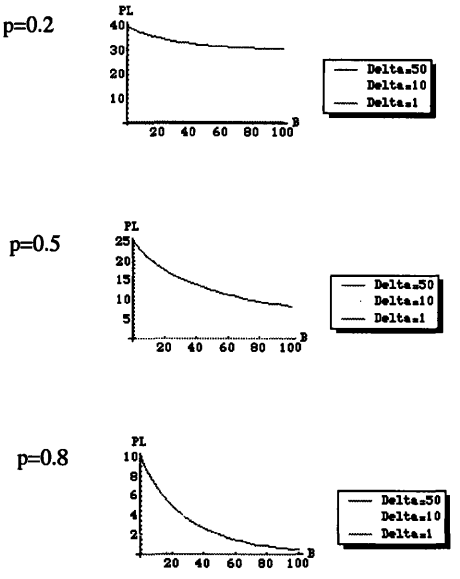


Figure 7-3: PL as a function of battery sizes for 3 different values of probability p.

As a final way to emphasize these findings, we consider the question of what battery size will get us to within x percent of the infinite battery purchase level. Define $\beta(x,\ p,\ \Delta)$ as this target purchase level:

$$\beta(x,\ p,\ \Delta) = \text{PL(infinite batteries)}+x/100(\text{PL(0 batteries)} - \text{PL(infinite batteries)}) \tag{7.9}$$

Then, to solve for the correct battery size B_x, we solve (equating Equation 7.7 to Equation 7.9)

$$\Delta\frac{1+(2p-3)p}{1-p-p(p/(1-p))^{B_x/\Delta}} = \beta(x,\ p,\ \Delta), \tag{7.10}$$

which yields

$$B_x = \frac{\Delta \ln\left(\frac{(1-p)(\Delta(2p-1)+\beta(x,\ p,\ \Delta))}{p\beta(x,\ p,\ \Delta)}\right)}{\ln\left(\frac{p}{1-p}\right)}. \tag{7.11}$$

This function is maximized at $p = 1/2$, and in fact taking the limit as p tends to $1/2$, we get the expression for the peak level:

$$B_x(p = 1/2) = \Delta\left(\frac{100}{x} - 1\right) \tag{7.12}$$

For fixed $\Delta = 20$ we plot Equation 7.11 versus p, which shows the peaking behavior and the peak level predicted by Equation 7.12 of $20(100/10 - 1) = 20(9) = 180$.

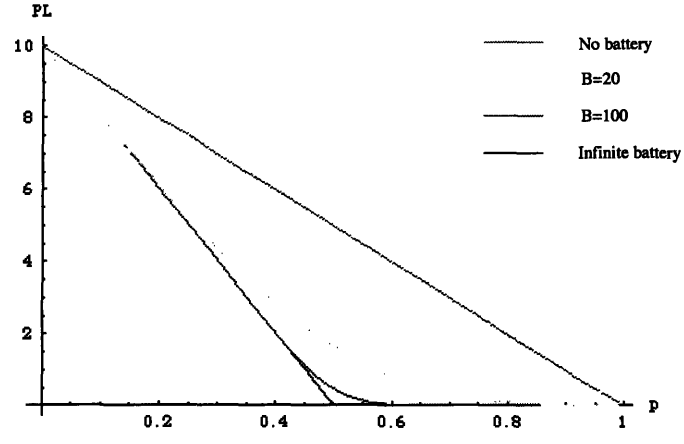


Figure 7-4: Enveloped below by the infinite battery PL and above by the no battery PL, we see how PL varies as a function of p for different battery sizes.

We have shown that batteries are most helpful at $p = 1/2$, where average supply exactly equals average demand. Not only are batteries marginally most effective at $p = 1/2$, but also, the total amount of improvement that can be achieved is maximum for $p = 1/2$. However, as available from Figures 7-3-7-5, it is at $p = 1/2$ where one requires a large capacity battery to come close to the infinite battery PL limit. In the next section, we see how these results extend to the general case of the chain shown in Figure 7-1c.

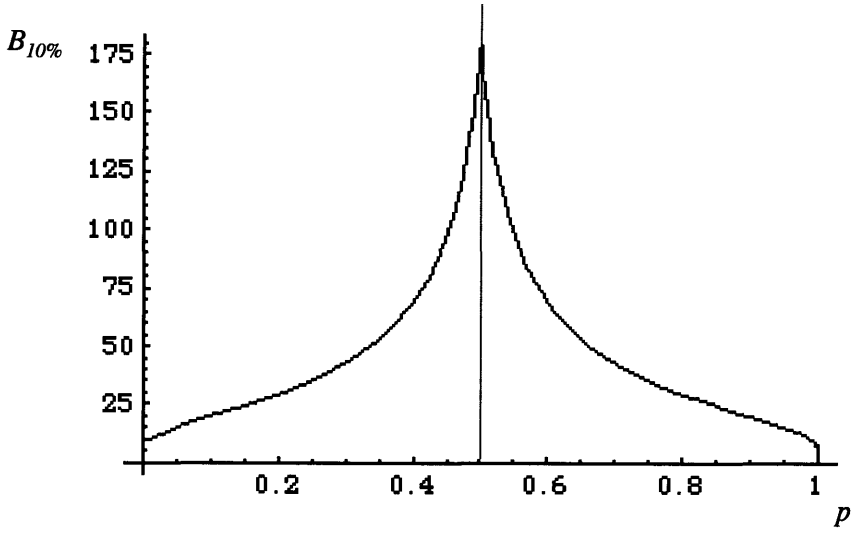


Figure 7-5: Battery size required to get within x percent of the infinite battery PL limit, shown for $x = 10$ and $\Delta = 20$. The curve peaks for $p = 1/2$ at 180, given by Equation 7.12.

7.2 Results for general model

For general distributions of supply and demand such as those shown in Figures 7-1a and b, we can determine what distributions have the largest gap between zero battery purchase level and infinite battery purchase level by solving the following nonlinear math program.

$$\begin{array}{lll} \max & \sum_{i=0}^{S_{\max}} \sum_{d=s+1}^{D_{\max}} p_s p_d (d-s) & -(\sum p_d d - \sum p_s s)^+ \\ \text{s. t.} & \sum p_d = \sum p_s = 1 & \\ & p_d,\ p_s \geq 0 & \forall d,\ s \end{array} \tag{7.13}$$

The objective function maximizes PL(0)–PL(∞) while the constraints ensure the solution satisfies the rules for probability distributions. We can decompose this problem into the following two subproblems and then take the maximum result from them to

get the overall maximum.

$$\begin{array}{lll} \max & \sum_{i=0}^{S_{\max}} \sum_{d=s+1}^{D_{\max}} p_s p_d (d-s) & \\ \text{s. t.} & \sum p_d = \sum p_s = 1 & \\ & pd,\ ps \geq 0 & \forall d,\ s \\ & \sum p_s s \geq \sum p_d d & \end{array} \tag{7.14}$$

and

$$\begin{array}{lll} \max & \sum_{i=0}^{S_{\max}} \sum_{d=s+1}^{D_{\max}} p_s p_d (d-s) & -(\sum p_d d - \sum p_s s) \\ \text{s. t.} & \sum p_d = \sum p_s = 1 & \\ & pd,\ ps \geq 0 & \forall d,\ s \\ & \sum p_s s \leq \sum p_d d & \end{array} \tag{7.15}$$

For both problems 7.14 and 7.15, the inequality constraint will be tight, which makes the two problems equivalent. The proof is by contradiction. For problem 7.14, consider an optimal solution for which the inequality is not tight. One can then construct a strictly better solution by decreasing $\bar{S}$ by the amount of the gap. For problem 7.15, a similar argument works. Here, consider a feasible solution where $\bar{D} > \bar{S}$. Consider shifting $\bar{D}$ down by an amount ϵ (take the highest non zero probability mass, p_k, and reduce it to $p_k - \epsilon$ while increasing p_{k-1} to $p_{k-1} + \epsilon$). This has causes the right term of the objective function to increase by ϵ while the left term decreases by the sum of the nonzero terms $p_s \epsilon$. Since at most this sum is equal to ϵ, our objective function has at least gone up while decreasing the gap by ϵ. We can continue this argument until the gap is 0.

Therefore, problems 7.14 and 7.15 are actually the same (the right hand term of the objective function of 7.15 is zero at optimality), and are equivalent to the following simpler problem:

$$\begin{array}{lll} \max & \sum_{i=0}^{S_{\max}} \sum_{d=s+1}^{D_{\max}} p_s p_d (d-s) & \\ \text{s. t.} & \sum p_d = \sum p_s = 1 & \\ & pd,\ ps \geq 0 & \forall d,\ s \\ & \sum p_s s = \sum p_d d & \end{array} \tag{7.16}$$

Using similar arguments as above, we can show that we can restrict our search

for optimal solutions to "extreme solutions" in the following sense: we only need to consider positive probability mass for the 4 end points of the two probability distributions: 0, $D_{\max}$, 0, and $S_{\max}$. This yields a problem we can solve in closed form. Breaking out the cases for various relationships between $D_{\max}$ and $S_{\max}$, Table 7.1 shows the optimal solutions and objective values. We see that for a fixed value of $D_{\max}$, the objective value is bounded above by the same $D_{\max}$, and is achieved by letting $S_{\max}$ go to infinity. This is because for no battery, most days you end up with no supply and demand of $D_{\max}$, and the occasional very high value of $S_{\max}$ cannot be stored so is essentially worthless (its only worth is that it satisfies the entire demand on that day). Similarly, for a fixed value of $S_{\max}$, the objective value is capped by $S_{\max}$, achieved as $D_{\max}$ goes to infinity. In this case, on most time periods supply exceeds demand and the daily purchase level is 0, but the surplus is not stored, so when on occasion you get a high demand day, you must pay for it all.

The solution in Table 7.1 also shows that large batteries are required if *both* $S_{\max}$ and $D_{\max}$ are large, but if either are small, then that small one is the one that dictates how much improvement storage can offer.

Region	$P_s^{\max}$, $P_d^{\max}$	Objective value
$S_{\max} \leq D_{\max}/2$	$p_d^{\max} = \frac{S_{\max}}{D_{\max}}$, $p_s^{\max} = 1$	$S_{\max}\left(1 - \frac{S_{\max}}{D_{\max}}\right)$
$D_{\max}/2 \leq S_{\max} \leq D_{\max}$	$p_d^{\max} = 1/2$, $p_s^{\max} = \frac{D_{\max}}{2S_{\max}}$	$D_{\max}/4$
$D_{\max} \leq S_{\max} \leq 2D_{\max}$	$p_d^{\max} = \frac{S_{\max}}{2D_{\max}}$, $p_s^{\max} = 1/2$	$S_{\max}/4$
$2D_{\max} \leq S_{\max}$	$p_d^{\max} = 1$, $p_s^{\max} = \frac{D_{\max}}{S_{\max}}$	$D_{\max}\left(1 - \frac{D_{\max}}{S_{\max}}\right)$

Table 7.1: Optimal solution for problem 7.13. This problem finds the distributions which maximize the difference between the steady state purchase level (PL) for the 0 battery case and the infinite battery case. The means of the supply and demand distributions are equal for this quantity to be maximized, so this table can be used for upper bounds on savings available when $\bar{S} = \bar{D}$. This is the case where batteries offer the largest potential for savings.

The final highlight of this analysis is that our original simplified chain is actually quite useful in that it intrinsically models the scenarios that maximize the PL difference between zero and infinite batteries. In particular, when $D_{\max} = S_{\max} \equiv \Delta$, then we either have a jump of plus or minus Δ, which reduces to the simple chain with

$p = q = 1/4$ and self loop probability of $1/2$.

7.3 Summary

We find throughout the above analyses that the best time to use storage is when the average supply equals the average demand but there is a lot of "spread" between the two distributions, meaning there are times when supply exceeds demand (you can store energy) and demand exceeds supply (you can use the stored energy). If the means are equal, then Table 7.1 gives an upper bound on the storage savings available from storage by assuming an infinite battery array (leading to a PL of zero) and the distribution which makes the zero battery PL the highest.

Figure 7-4 on the other hand assumes $D_{\max} = S_{\max}$, but, by varying p, allows the means of the distributions to be different. This figure implies that if the means are different, batteries are not as helpful, especially if the means are very different (p near 0 or 1). Although the figure shows that there is a larger gap (gap = potential savings) for p close to 1 rather than 0, this does not imply that customers with average supply lower than average demand should not consider storage. The analysis does emphasize however that in such as case, one needs to have a good "spread" to make it worthwhile, and it will take a relatively small amount of storage to get you close to the infinite battery storage limit. For the general power users whose supply and demand distributions do not fit into the special cases above, they should consider the solution to the general Markov chain, which is shown implemented in Mathematica in Appendix D.

Chapter 8

Towards a complete implementation

This thesis has focused primarily on the mathematical model involved with optimal energy control in buildings and other energy networks. To implement this approach, one needs a wealth information that is not trivial to obtain. Temperature forecasts should be straightforward in most regions of developed countries. For example, in the United States, the National Digital Forecast Database (www.nws.noaa.gov/ndfd) provides hourly temperature, wind, and many other data pieces for the upcoming hours (www.wunderground.com provides a human readable interface to this data). Spatial resolution depends on the weather data pieces provided, for example wind needs a finer scale than temperature.

Wind and solar radiation prediction are required for optimal control of renewable energy networks. Temperature, wind, and solar radiation also impact indoor temperatures, and thus their prediction is important for any user using TOTEM with HVAC control. These data are used to map how HVAC controls impact the indoor air conditions important in human comfort, notably air temperature, radiant temperature, and relative humidity [21]. Obtaining good state update equations for these data is a non-trivial task, as discussed in Section 6.3.

Similarly, one needs to understand user comfort level and determine a logical and quantitative tradeoff between comfort level and cost of energy consumption. These

technical issues are reviewed in the sections below. We finish with a brief review of the information technology and hardware needed for the communication system of a TOTEM implementation.

8.1 Weather prediction modules

Given an optimization horizon on the order of one or two days, we are interested in short-term weather forecasts. Short-term forecasts refer to forecasts over the next 72 hours. Forecasts over the next six hours or less are sometimes referred to as nowcasts.

Techniques of weather prediction can be broadly classified into two categories: physical model based prediction, and statistical learning type prediction. Prediction based on physical models are quite complicated to develop given the complexities of weather, and even if developed are computationally expensive to implement. This is important to keep in mind for rural settings. Statistical learning methods, including linear regression, moving averages, Kalman filters, and neural networks are much simpler and less location specific, but they tend to do well only in the very short term (depending on what is being predicted), being outperformed by physical models for longer term forecasts. Forecasting systems that employ a mix of both types are common.

8.1.1 Wind

Wind prediction has advanced markedly over the past decade. This us due to power companies with installed wind generation desiring to get accurate wind power output predictions for the coming day to solve their morning fossil unit commitment problems: a good prediction of the wind power available for the coming day is necessary to solve the unit commitment problem at the start of each day (that is, decide what fossil based generators will be brought on to satisfy the demand). Research is quite active in this field, especially in windy countries in Europe such as Holland, Denmark, Germany, and Spain. As of 1998, Denmark generated about 10% of its electric power from wind (www.wind-works.org/articles/Overview.html). As of 2001, Germany had

an installed wind capacity of 8000 MW compared to the United State's capacity of 4250 MW (www.pollutionengineering.com).

As with all turbine technologies, micro fluctuations (on the scale of seconds and milliseconds) are taken care of by either sophisticated electric controls or simply the inertia of the rotors themselves, which slow down or speed up to accommodate these load fluctuations. Thus, we are interested in larger time scale wind level fluctuations. It is customary in the field to report hourly wind speed averages. Provided there are not many gusts and lulls this is sufficient information for energy planners. Since wind farms are usually built in places where gusts and lulls are infrequent this is generally sufficient.

The ANEMOS project (anemos.cma.fr) is a four year research and development project for short term wind power forecasting. The first deliverable from that project, a report titled "State of the art on short-term wind prediction" [25] provides a solid review of the field. Some issues specific to wind speed prediction included the down-scaling of wind data, which refers to projecting reported wind speeds to the correct height of the wind turbine, and also upscaling, which refers to projecting results for a few turbine locations to an entire wind farm.

The common validation measure for wind speed predictions is root mean square error (RMSE), expressed as a percentage of the rated output capacity of the turbine. [25] shows that current forecasting techniques have an RMSE of about 8-10% and that this value is suprising steady over the 24 hours. That is, the predictive capability does not significantly diminish with the length of the forecast horizon. An important issue for the use of such results in energy optimization is whether or not the errors are correlated over time or not. In [30], which uses neural networks for short term wind power prediction, we find that the errors are not correlated. For example, over prediction at hour k does not imply an over prediction at hour $k+1$. This trait of the random wind process may inspire the use of robust optimization, a topic which should be further investigated.

8.1.2 Temperature

A common technique for short term temperature prediction combines a numerical weather prediction (NWP) module with a statistical approach called model output statistics (MOS). MOS develops a statistical relationship between inputs to the model and outputs to tighten the prediction accuracy [26].

Day ahead hourly temperature prediction, like wind prediction, is site specific. For example, predicting temperatures at the Denver airport in Colorado is more difficult than at the Reagan National Airport (God rest his soul) in Washington D.C. [56], due to the dramatic wind events from the north in Denver. In [56], they report that for Denver, the mean absolute error was about 4 deg F for the first 24 hours, and only 2 deg F at Reagan. Similar numbers for temperature prediction accuracies can be found at www.etl.noaa.gov/programs/2003/nehrtp/.

8.1.3 Solar radiation

Less work has been done on predicting hourly solar radiation, the prime predictor for PV power output. Nevertheless, a short work [64] shows that one can use the atmospheric transmittance of the atmosphere early in the morning along with the geometry of the sun's movement overhead to predict hourly solar radiation levels for the day. They show that this method predicts hourly PV energy with an accuracy averaging about 14% for each hour, similar to typical values for wind power prediction. It is expected that by using satellite cloud movement data, predictions of local solar hourly power will improve greatly, especially for the near term of two hours or less [29].

Assessing the value of general weather forecast accuracy for energy planning will be a helpful future study. For example, depending on the components of an energy network (storage, generators, etc.), what is a sufficient forecast accuracy? Is it economically worth implementing a prediction a system that will decrease prediction errors from 10% to 5%? Such analyses can always be done using stochastic TOTEM

for particular configurations, but a general analysis would be useful.

8.2 HVAC module

HVAC control presents perhaps the greatest challenge and the greatest potential for optimizing energy usage. The literature and effort directed at HVAC control is vast. In control theory literature, control refers to keeping a quantity or quantities, such as a room's temperature and humidity level, within prescribed bounds. Nearly all of the literature addresses topics of control such as stability, robustness, and system identification. Specific topics include 1) how to guarantee a control system will be stable, and 2) once stability is addressed, how to tune the parameters for fast room temperature response. These investigations are based on standard hardware in HVAC systems which are local set points (fire the boiler if the water return temperature drops below 65 deg F, for example) and other local control options. For optimal control in the TOTEM sense, where every possible knob in the system is attached to a central controller and the state update equation is known, we seek *optimal control* rather than merely control, and we must adopt a different viewpoint.

Traditional HVAC work concentrates on stability of the HVAC system under localized control responding to local sensors. Proportional-Integral-Derivative control (PID control, see [57] for example) is the standard framework for these types of systems. This approach sets the current control based on the present value of the error, the integral of the error, and the derivative of the error, where the error is the difference between where the system should be and where it is. On the other hand, a global controller which knows the impacts of every control choice made could exercise control on the system in the holistic manner required for TOTEM-like control (optimal control in the face of real time prices and building distributed generation supplies). This is a far leap from today's current HVAC practice, and in this section we discuss why it is so difficult, and outline an approach for its realization.

Before a system can be controlled it needs to be identified. Even the simplest pieces of an HVAC system are nonlinear objects which are very difficult to model from

first principles. First principles for the thermodynamics of HVAC systems result in second order partial differential equations, and given the complicated geometries and materials of buildings, as well as the influence of weather, it is hardly conceivably to write down – let alone solve – these equations. If we go a level of abstraction up, and consider each device in the network as an input-output node, we still have problems with nonlinearities. For example, consider the control of a home radiator, which is a heat exchanger: hot fluid passes through it and heats up the air in the room. For optimal control, we would need control of the rate of the fluid as well as its incoming temperature. These two control variables however get multiplied together when calculating the heat transfer to the room, resulting in a nonlinear optimization problem. Convective and radiative heat transfers are not linear in the temperature differential, thus adding even more nonlinearities to this very simple HVAC problem. Scale this arrangement up to a realistic building with hundreds of rooms, air handlers, valves, windows, and floors, and the thought of a tractable optimization model even with simplified input-output nodes is baffling. HVAC control is an example of multiple input and multiple output (MIMO) nonlinear control theory, which is a subject of ongoing research [2].

Given the above issues, neural networks for system identification and a derivative-free optimizer for optimization [45] seem the most pragmatic approach for HVAC systems. We have included one approach for this problem in this thesis, but many questions need to be researched to help understand what approximations and neural networks will be best suited implementations. Examples are:

- What level of component grouping works the best?
- How does one best optimize a system consisting of collection of linear equations (the TOTEM model presented in this thesis) in addition to a highly nonlinear HVAC module?
- How does one balance issues of identification and control (adaptive control)?
- What is the best strategy for searching for an optimal solution? Genetic algorithms, global search methods such as those found in the package GenOpt

2.0 [60], gradient-based optimization, or some new strategies that utilize the structure of the HVAC optimal control problem?

A large area of future research will be the joint optimization of TOTEM-like systems, which can be linearized as shown in Chapter 3, with an HVAC optimal control module. An optimization procedure is only as fast as its slowest subroutine, which means that in HVAC control applications, the linear TOTEM model is not representative of the computational challenges of optimal control.

8.3 User preference and occupancy module

When TOTEM is used for energy affecting comfort levels, it is necessary to decide how comfort is quantified and then decide upon the tradeoff between the price of energy and comfort levels. A model for user preference regarding indoor environmental conditions is the American Society of Heating, Refrigeration, a Air Conditioning Engineers (ASHRAE, www.ashrae.org) guide for thermal comfort bands [3]. The seminal work by Fanger [21] addresses the problem of thermal comfort in depth. Due to that publication, numerous comfort indices have been applied, including effective temperature, predicted mean vote, predicted percentage dissatisfied, equivalent temperature and tropical sun index [39, 59].

Presumably the comfort/economic tradeoff values will be time dependent, or more precisely, occupancy dependent. Prediction modules based on neural networks could learn the occupancy patterns of a building, and hence be used to set tradeoff parameters for the coming hours. In a residential setting, similar modules could be used to predict base loads (televisions and computers coming on after school, hot water demand in the morning, laundry machine usage, lighting) and feed this information to a stochastic TOTEM solver [46].

Another implementation extension of TOTEM involves dealing with real time user requests, such as a laundry drying cycle. In one possible scenario, a user starts the dryer but is asked if delaying the start time by three hours is acceptable (to take advantage of reduced electricity prices). Depending on their response, the dryer is

started or not, and TOTEM reoptimizes. Optimal control under situations like this, with stochastic job entry coupled with user interaction, should exist in real TOTEM installments.

8.4 Tying the pieces together

The key to a system like TOTEM being implemented is interoperability of the devices within an energy network. Many buildings today, built long ago, are equipped with devices which cannot easily be added to an information network. When buildings began to be built with networkable components, the information engineering for these components was almost always proprietary and therefore not suitable for interconnectivity and energy management [13]. In recent years, following the open source and open standards movement of software, open standards are entering the building energy management sector.

Dozens of protocols have emerged and which ones will stabilize and gain market share is not yet known. BACnet, which stands for Building Automation and Control Network, seems a particularly promising one. Put forth by ASHRAE, this protocol specifies four implementation steps a hardware manufacturer must follow in order to be certified as BACnet compliant. One important feature of BACnet is that it allows manufacturers to preserve proprietary systems while providing a means for these systems to communicate with other systems on the network.

Echelon Corporation (www.echelon.com) is responsible for producing a physical device attached to each control node in a network, creating a hardware-based solution to interoperability. Their solution includes a microchip which incorporates communications and control, as well as a protocol (LonWorks), like BACnet, for complete network information gathering and control.

Add to these networks sensors which detect occupancy and environment conditions, and one has the ingredients to make total energy management systems a reality. Since these technologies are in an emerging growth phase, we expect intelligent control systems that manage supply and demand concurrently to follow shortly after. This

thesis has presented a mathematical system for controlling such networks, and for controlling off grid village power systems with arbitrary combinations of renewables, fossil generators, and storage. For a host of reasons (environmental concerns, erratic oil economics, the closing gap between fossil-based generation and renewables), we can expect individuals, communities, and governments to place increased focus on alternative sources of energy and energy efficiency. Distributed generation will play an increasingly large role, giving end users a larger share of control. Through detailed modeling and optimization of energy networks, this thesis proposes a unified control system overseeing energy purchases, production and usage, a system which becomes more relevant as the energy world becomes more diverse.

Appendix A

TOTEM model in AMPL

```
#  June 2004
#  David Craft
#  TOTEM: Total Energy Management.
#  Coded in AMPL language for use with a mixed integer linear
#  solver such as CPLEX or lpsolve
##
## This contains a full energy decision model for local power
## generation, local power consumption, and storage decisions.
##
#DEMAND NODES

set N_D_SL; #demand nodes:  shiftable loads
set N_D_A; #demand nodes:  analog control
set N_D_B; #demand node

set N_D:= N_D_SL union N_D_A union N_D_B;

set N_IO;  # aggregate input -> aggregate output,
# implicitly single energy type
set N_V;   #conVersion nodes (single output channel equations),
```

```
#can handle multiple output energies

set N_S;

#max and min levels for each storage device.
param SMAX{N_S};
param SMIN{N_S};

#max charging and discharging rate
param SCHARGE{N_S};
param SDISCHARGE{N_S};


#FUEL SOURCES

set N_F;

#RENEWABLE GENERATION NODES

set N_R;

#SINK NODES -- these for gridBUY node for example.
set N_SINK;

#Subset of generation and aggregation nodes which are
#described by nonlinear (piecewise linear) relationships

set N_NONLINEAR within N_V;

#Subset of io nodes which we have operational
```

```
#controls/ranges etc for

#All NODES

set N:= N_D union N_S union N_IO union
N_V union N_F union N_R union N_SINK;

# input and output nodes for each node

set I{N} within N;
set O{N} within N;

set STATE_NAMES;
set PENALTY_STATES within STATE_NAMES;
set CONSTRAINT_STATES within STATE_NAMES;

set N_U;  #nodes with outgoing arcs having u variable

#UGROUPS
param numUGroupsMin default 0;
param numUGroupsMax default 0;

set UGROUPS_MIN_NODES{1..numUGroupsMin} within {n in N_U, i in O[n]};
set UGROUPS_MAX_NODES{1..numUGroupsMax} within {n in N_U, i in O[n]};

param UGroupMinNum{1..numUGroupsMin};
param UGroupMaxNum{1..numUGroupsMax};

param numUTieGroups default 0;
set UTIED_ARC {1..numUTieGroups} within {n in N_U, i in O[n]};
```

```
set UTIED_TO {1..numUTieGroups} within {n in N_U, i in O[n]};



#set TIMES;
param T>0;

#TAKE THESE OUT!
param NumCurves;
param DemCurves {1..NumCurves,1..T};


#MinUpTime and MinDownTime for generators
param MinUpTime {i in N_U} default 1;
param MinDownTime {i in N_U} default 1;

param StayOnUntil {i in N_U} default 0;
param StayOffUntil {i in N_U} default 0;


#BALANCE EQUATIONS FOR conversion nodes
#slope
param V {i in N_V, n in I[i], j in O[i]};
#intercept
param B  {i in N_V, n in I[i], j in O[i]} default 0;

#and for io nodes multiple input output
param G {i in N_IO union N_S, n in I[i]} default 1;

#params for demand nodes->analog control
```

```
param a {i in N_D_A};

#params for demand nodes->base (uncontrollable) load
param b {i in N_D_B, 1..T};

#params for demand nodes->shiftable loads
param MUST_START_BY{N_D_SL}>=0, integer;
param l {i in N_D_SL,j in 1..MUST_START_BY[i],1..T}; #load matrix

#Renewable generation: here, like for the base load demand nodes,
#d=r in the constraint section
param r {i in N_R,  1..T};

#Fuel-based generation and grid-based.
#c: costs for fuels and grid purchases
param c {i in N_F, 1..T};

#For fuel-based, on/off variables and operating ranges
param LOWER {i in N_U, j in O[i]} default 0;
param UPPER {i in N_U, j in O[i]};

####Renewable power equalities
#param r {i in N_R, 1..T};

param INITIAL_STORAGE{n in N_S} default 0;

# states
param INITIAL_STATE{k in STATE_NAMES} default 0;

#vars and params for penalty for being del away from
```

```
#target TG state levels
param TG {k in PENALTY_STATES, 1..T};
param gamma {k in PENALTY_STATES, 1..T};

#params for states that are constrained to be within certain regions
param EMIN{k in CONSTRAINT_STATES, 1..T} default -Infinity;
param EMAX{k in CONSTRAINT_STATES, 1..T} default Infinity;

#WEATHER VECTOR: holds things like weather that influence state
#but are not changed themselves.
set WEATHER_NAMES;
param W {i in WEATHER_NAMES,1..T};


#params for state update equations.
param Hs {k in STATE_NAMES,j in STATE_NAMES} default 0;
param Hw {k in STATE_NAMES,j in WEATHER_NAMES} default 0;
param Hz {k in STATE_NAMES,j in N_D_A} default 0;


#### N-L ##### params for nonlinear i.o.
#note: NBP, number of break points, is the same as N-1, where
#N is the number of linear segments
#N is the param used in the math writeup.
param NBP {i in N_NONLINEAR, j in I[i]};

# aaux and baux define the piecewise
# linear curves, and uaux are the breakpoints.
# uaux: one for each breakpoint (these ARE
# the breakpoints!), a and b, one less than numBreakPoints
```

```
param uaux {i in N_NONLINEAR, j in I[i], s in 1..NBP[i,j]};
param aaux {i in N_NONLINEAR, j in I[i], s in 1..NBP[i,j]+1};
param baux {i in N_NONLINEAR, j in I[i], s in 1..NBP[i,j]+1};

param M;  #LARGENUM
param ep; #SMALLTOLERANCE

##### vars
var x {i in N, j in O[i],1..T} >=0; #all flows

var y {i in N_S, 1..T}; #storage levels

var d {i in N_D,1..T}>=0; #power demanded

var za {N_D_A,1..T}>=0;    #end use, analog controls

#end use, shiftable loads
var zsl {i in N_D_SL, 1..MUST_START_BY[i]} binary;
var STATE{k in STATE_NAMES, 1..T};

var del {k in PENALTY_STATES, 1..T};

var u {i in N_U, j in O[i], 1..T} binary;

#### N-L #### vars for nonlinear i.o.
# removed for now #


############################
#### OPTIMIZATION MODEL ####
```

```
############################


#objective function models cost and absolute deviation for states
#in the "penalty" set, with multiplier gamma
minimize objective_function:
sum {i in N_F, j in O[i], t in 1..T} c[i,t]*x[i,j,t] +
sum {k in PENALTY_STATES, t in 1..T} gamma[k,t]*del[k,t];


#note, these do not apply to the fuel source nodes and
#renewable source, which have output but no input
#IMPORTANT: O[i] should consist of one node unless
#we are dealing with cogen or some other multi energy out!
subject to conversion_out {i in N_V diff N_NONLINEAR,
j in O[i], t in 1..T}:
x[i,j,t] = sum {n in I[i]} (V[i,n,j]*x[n,i,t] + B[i,n,j]*u[n,i,t]);

subject to io_equality {i in N_IO, t in 1..T}:
sum {n in I[i]} G[i,n]*x[n,i,t] = sum {j in O[i]} x[i,j,t];

####NONLINEAR REMOVED #####

subject to storage_equality {i in N_S, t in 1..T-1}:
sum {n in I[i]} G[i,n]*x[n,i,t] = sum {j in O[i]} x[i,j,t]
+ y[i,t+1] - y[i,t];

#constraint so the final storage level is not thrown down to 0:
#subject to storage_last {i in N_S}:
# sum {n in I[i]} G[i,n]*x[n,i,T] - sum {j in O[i]} x[i,j,T] +
```

```
y[i,T] >= SMIN[i];

#set final storage level equal to initial one.
subject to storage_last {i in N_S}:
sum {n in I[i]} G[i,n]*x[n,i,T] - sum {j in O[i]} x[i,j,T] + y[i,T] >=
INITIAL_STORAGE[i];

subject to renewable_equality {i in N_R,t in 1..T}:
sum{j in O[i]} x[i,j,t]=r[i,t];

#
#setting the demand variables
#

#### for base load

subject to set_demand_base {i in N_D_B, t in 1..T}:
d[i,t] = b[i,t];

# for analog control, depends on analog control decision z

subject to set_demand_analog {i in N_D_A,t in 1..T}:
d[i,t] = za[i,t]*a[i];

# for shiftable load control .  .    ....
# params for demand nodes->shiftable loads
#
subject to set_demand_sl {i in N_D_SL,t in 1..T}:
d[i,t] = sum{j in 1..MUST_START_BY[i]} l[i,j,t]*zsl[i,j];
```

```
subject to start_each_sl {i in N_D_SL}:
sum{j in 1..MUST_START_BY[i]} zsl[i,j]=1;


##then, satisfy each demand with enough power

subject to supply {i in N_D, t in 1..T}:
sum{j in I[i]} x[j,i,t] >= d[i,t];


#### FUEL_BASED GENERATORS ON/OFF and
#### OPERATING RANGES constraints

subject to on_off_lower {i in N_U, j in O[i], t in 1..T}:
x[i,j,t]>=LOWER[i,j]*u[i,j,t];

subject to on_off_upper {i in N_U, j in O[i], t in 1..T}:
x[i,j,t]<=UPPER[i,j]*u[i,j,t];

#min up/down-time constraint
#param MinUpTime {i in N_U} default 1;
#param MinDownTime {i in N_U} default 1;

subject to min_up {i in N_U, j in O[i], t in 2..T,
tau in (t+1)..min(t+MinUpTime[i]-1,T)}:
u[i,j,t] - u[i,j,t-1] <= u[i,j,tau];

subject to min_down {i in N_U, j in O[i], t in 2..T,
tau in (t+1)..min(t+MinDownTime[i]-1,T)}:
u[i,j,t-1] - u[i,j,t] <= 1 - u[i,j,tau];
```

```
## if initially the device is on for less than minUpTime or off for
## less than minDownTime then
subject to stay_on {i in N_U, j in O[i], t in 1..StayOnUntil[i]}:
u[i,j,t]=1;

subject to stay_off {i in N_U, j in O[i], t in 1..StayOffUntil[i]}:
u[i,j,t]=0;


subject to storage_initialize {i in N_S}:
y[i,1]=INITIAL_STORAGE[i];


# State updates
#var STATE{n in N,i in 1..STATE_LENGTH[n],1..T};
#param W {i in 1..W_LENGTH, 1..T};

subject to state_initialize {k in STATE_NAMES}:
STATE[k,1]=INITIAL_STATE[k];

subject to state_update {k in STATE_NAMES, t in 1..T-1}:
STATE[k,t+1]=sum{j in STATE_NAMES} Hs[k,j]*STATE[j,t] +
     sum{j in WEATHER_NAMES} Hw[k,j]*W[j,t]+
             sum{j in N_D_A} Hz[k,j]*za[j,t];

#STATE CONSTRAINTS
subject to state_min {k in CONSTRAINT_STATES, t in 1..T}:
STATE[k,t]>=EMIN[k,t];
```

```
subject to state_max {k in CONSTRAINT_STATES, t in 1..T}:
STATE[k,t]<=EMAX[k,t];

#STATE DEVIATIONS (to capture absolute value in the objective function)
subject to deviation_1 {k in PENALTY_STATES, t in 1..T}:
del[k,t]>=STATE[k,t]-TG[k,t];

subject to deviation_2 {k in PENALTY_STATES, t in 1..T}:
del[k,t]>=TG[k,t]-STATE[k,t];

subject to storage_max {i in N_S, t in 1..T}:
y[i,t]<=SMAX[i];

subject to storage_min {i in N_S, t in 1..T}:
y[i,t]>=SMIN[i];

#max charging and discharging rate
#param SCHARGE{N_S};
#param SDISCHARGE{N_S};

subject to storage_max_charge {i in N_S,t in 1..T}:
sum {j in I[i]} x[j,i,t]<=SCHARGE[i];

subject to storage_max_discharge {i in N_S,t in 1..T}:
sum {j in O[i]} x[i,j,t]<=SDISCHARGE[i];


##UGROUP constraints
#param numUGroupsMin default 0;
```

```
#param numUGroupsMax default 0;

#set UGROUPS_MIN_NODES{1..numUGroupsMin} within {N,N};
#set UGROUPS_MAX_NODES{1..numUGroupsMax} within {N,N};

#param UGroupMinNum{1..numUGroupsMin};
#param UGroupMaxNum{1..numUGroupsMax};

subject to ugroup_min {n in 1..numUGroupsMin,t in 1..T}:
sum {(i,j) in UGROUPS_MIN_NODES[n]} u[i,j,t] >= UGroupMinNum[n];

subject to ugroup_max {n in 1..numUGroupsMax,t in 1..T}:
sum {(i,j) in UGROUPS_MAX_NODES[n]} u[i,j,t] <= UGroupMaxNum[n];

#param numUTieGroups default 0;
#set UTIED_ARC {1..numUTieGroups} within {n in N_U, i in O[n]};
#set UTIED_TO {1..numUTieGroups} within {n in N_U, i in O[n]};

subject to ugroup_tie {n in 1..numUTieGroups,t in 1..T}:
sum {(i,j) in UTIED_ARC[n]} u[i,j,t] =
sum {(i,j) in UTIED_TO[n]} u[i,j,t];
```

Appendix B

TOTEM input deck: Costa de Cocos village hybrid network

The AMPL input deck below is from the hybrid village network studied in Chapter 6. It is reduced to a 10 hour day to save space.

```
#May 12, 2004
#############
#  A hybrid network for a rural village, for example.
#  Wind turbine + diesel gen + storage, all fully connected for
#  maximum control possibilities.  The idea is to examine in
#  both the deterministic (base load and wind) and stochastic
#  settings the importance of smart storage control rather than
#  the "you have surplus wind, store it" nominal control.  A couple
#  of motivating examples of why the nominal control can be
# suboptimal:
#
#  1) You have full storage, and no wind in the future, but you hold
#       on to your storage and generate now since the load is high
#       now and won't be later, and thus you use the gen set more
#       efficiently.
```

```
#  2) High load now, lots of wind, nothing in the battery.  If in a few
#       hours you expect low load but no wind, you will want to fire up
#       the gen now to charge the battery so you do not have to supply
#       the low load with the diesel generator.
#
## In both these examples, it is the lower range of the diesel gen, and
## potentially higher efficiency at higher loads, that makes storage
## strategy a little complicated.  Notice also that optimal storage
## strategy depends on future loads and future wind levels.
##
##############

## equipment
## 7.5 kW Bergey wind turbine
## battery array storing 50.4 kWh
## 25 kW diesel generator
param T:=10;

set N_IO := GENAGG;
set N_V := GEN;
set N_R := WIND;

set N_U:= FUEL;

set N_D_SL := ;
set N_D_A := ;
set N_D_B := BASE_ELECTRICITY;

set N_S:= STORAGE;
```

```
set N_F:= FUEL;

set I[WIND]:= ;
set O[WIND]:=BASE_ELECTRICITY STORAGE;

set I[STORAGE]:=WIND GENAGG;
set O[STORAGE]:=BASE_ELECTRICITY;
set I[BASE_ELECTRICITY]:=STORAGE WIND GENAGG;
set O[BASE_ELECTRICITY]:= ;

param StayOffUntil :=
FUEL 10
;

param MinUpTime :=
FUEL 2
;

set I[GEN]:=FUEL;
set O[GEN]:= GENAGG;

set I[GENAGG]:=GEN;
set O[GENAGG]:= STORAGE BASE_ELECTRICITY;

set I[FUEL]:= ;
set O[FUEL]:=GEN;

#penalty and state constrained sets
set STATE_NAMES;
set CONSTRAINT_STATES:= ;
```

```
set PENALTY_STATES:= ;

set N_SINK := ;

set N_NONLINEAR := ;

param:          SMAX    SMIN    SCHARGE SDISCHARGE:=
STORAGE         20      10      5       5;

param INITIAL_STORAGE:=STORAGE 15;

param b:=[BASE_ELECTRICITY,*]
1 1
2 2
3 3
4 4
5 5
6 10
7 10
8 10
9 10
10 10;

param r:= [WIND, *]
1 3
2 3
3 3
4 0
5 15
6 15
```

```
7 12
8 8
9 4
10 2;

#Fuel-based generation and grid-based.

param c:=
 [FUEL,*]
1 10
2 10
3 10
4 10
5 10
6 25
7 25
8 25
9 10
10 10;

param G:=
STORAGE WIND 0.75
STORAGE GENAGG 0.75;

param V:=
GEN FUEL GENAGG 14.14;

param B:= GEN FUEL GENAGG -0.52;

param: LOWER UPPER :=
```

```
FUEL GEN .5 1.83;
```

Appendix C

Graphical user interface and AMPL implementation

This appendix gives an overview of the software front-end implementation of the TOTEM model. The front-end exists to simplify the process of network definition, to simplify the solution process, and to visualize the model output. We begin by describing our internal storage representation of a user network. We then describe the graphical user interface for TOTEM, and lastly the connection to commercial solvers via AMPL (A Modeling Language for Mathematical Programming) [23].

The full data flow tying these elements together is given in Figure C-1.

C.1 The use of XML

We use XML (extendible markup language) as our core data structure. XML is a standard system to represent hierarchical data sets. We develop our own data structure so that we are not coupled to the representation used by our optimization interface package AMPL, in case we later decide to use a different package. Also, AMPL's language is very well suited for describing a problem which is fully created and 'ready-to-go', but not a model that is work in process. More importantly, AMPL is not written as a language for displaying network definitions or model inputs and solution outputs. XML is used for GUI definition files, making the GUI customizable

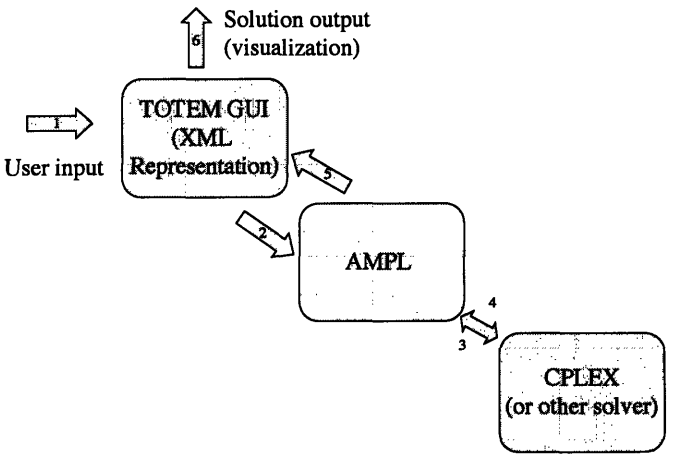


Figure C-1: A schematic for the TOTEM system data flow. Step 1 is the user input, where the network is defined along with cost information and system component information. This information is stored in XML representation, and converted into AMPL format in Step 2. Steps 3 and 4 represent AMPL passing the model and data to a math program solver, and retrieving the solution from the solver, respectively. Step 5 is TOTEM retrieving and parsing the solution information from AMPL and storing it as an xml object. Step 6 uses the solution object for user defined visualizations and post-processing.

without recoding, for template definitions of standard network components, and for storing a concrete user network instance.

C.2 TOTEM network definition GUI

Figure 1-1 shows the main window of the TOTEM GUI with a sample energy network layout open. On the left panel are component templates which can be dragged and dropped into the network. These nodes have appropriate parameters and values associated with them; for example, a battery has a maximum capacity as well as minimum and maximum rates of charge and discharge.

The middle window displays the network layout. Each component has an input port on the left side and an output port on the right. Connections between these can be created by control-clicking and dragging from an output to an input port.

Connections between components are rendered using a Manhattan layout scheme which allows only orthogonal directions for edges, with right angle turns permitted to align the head and tail at the respective endpoints. This makes it easy to see relationships between components and avoids cluttering up the layout. On the right is the parameters editor for the currently selected node.

C.2.1 Achieving customization without recoding

The network definition GUI was designed to be flexible and configurable. Changes in the AMPL optimization model are supported *without* requiring the source code of the GUI to be modified. In fact, this was necessary as the AMPL model was being worked on in parallel with the GUI.

A configuration file, `params.xml` lists the parameters that AMPL expects to find. AMPL input files give parameters individually, then subscript these parameters by each component which has that parameter. This is reversed from the graph view, where the components are at the top level, and each component has a parameter. Thus, specifying the expect parameters in a configuration file ensures that AMPL never fails to find a parameter in its input, even if it is blank and unused in the actual network.

Likewise, `sets.xml` lists the sets that AMPL expects to find components associated with. All sets are printed even if empty.

Note that, in addition to giving the names of the parameters, we also assign characteristics to parameters. `timeseries` indicates that a parameter varies over time, `inedge` indicates that a parameter is associated with the edge going into a node, and `innode` associates the parameter with the node which connects to a node.

Component templates are stored in the `templates` subdirectory. Template files simply give the parameters that the component possesses, along with values for each. The `NAME` and `SET` parameters are given special meaning, since they identify the node and indicate which sets (generator, renewable generation, nonrenewable generation, storage, etc.) it belongs to. In addition, the `ICON` parameter allows a descriptive image to be used for the component in the graphical view of the network. Components

```
<params>
  <param name=''SCHARGE''/>
  <param name=''SDISCHARGE''/>
  <param name=''SMIN''/>
  <param name=''SMAX''/>
  <param name=''INITIAL_STORAGE''/>
  <param name=''b'' char=''timeseries''/>
  <param name=''r'' char=''timeseries''/>
  <param name=''c'' char=''timeseries''/>
  <param name=''G'' char=''inedge''/>
  <param name=''LOWER'' char=''inedge''/>
  <param name=''UPPER'' char=''inedge''/>
  <param name=''MinUpTime'' char=''innode''/>
  <param name=''MinDownTime'' char=''innode''/>
</params>
```

Figure C-2: Sample `params.xml`

```
<sets>
  <set name=''N_V''/>
  <set name=''N_IO''/>
  <set name=''N_SINK''/>
  <set name=''N_NONLINEAR''/>
  <set name=''N_R''/>
  <set name=''N_F''/>
  <set name=''N_S''/>
  <set name=''N_U''/>
  <set name=''N_D_SL''/>
  <set name=''N_D_A''/>
  <set name=''N_D_B''/>
</sets>
```

Figure C-3: Sample `sets.xml`

```
<parameters>
  <param name=''NAME'' value=''Battery''/>
  <param name=''ICON'' value=''templates/real-battery.gif''/>
  <param name=''SET'' value=''N_S''/>
  <param name=''SMAX'' value=''100''/>
  <param name=''SMIN'' value=''20''/>
  <param name=''SCHARGE'' value=''10''/>
  <param name=''SDISCHARGE'' value=''10''/>
  <param name=''INITIAL_STORAGE'' value=''20''/>
</parameters>
```

Figure C-4: Sample template for a battery

```
<predefined>
  <shade name=''Electric Generators''>
    <template name=''Gas turbine'' file=''ge-gasturbine''/>
    <template name=''Microturbine'' file=''ge-microturbine''/>
    <template name=''Fuel cell'' file=''ge-fuelcell''/>
  </shade>
  <shade name=''Steam Generators''>
    <template name=''Boiler'' file=''gs-boiler''/>
    <template name=''HRSG'' file=''gs-hrsg''/>
  </shade>
</predefined>
```

Figure C-5: Portion of a `predefined.xml`

without associated icons simply render as rectangles.

Templates are specified and grouped in `predefined.xml`. The TOTEM GUI reads this configuration file to generate the component palette on the left of the main window, sorting components into groups under expandable and collapsible windowshades in the palette. This is how we are able to customize the GUI without recoding.

A basic user can use TOTEM without ever editing these configuration files or even seeing them. Simply drag and drop components, define connections and parameters, and save the network. TOTEM saves the network both as a node-centric XML file which makes it easy to parse and edit, and a parameter-centric AMPL input file which is ready to be run through the optimizer.

A more advanced user may need to define custom components. For example,

a company considering expanding its DG may have four models of generators with different capacities, and would like to make each generator model a template. This can be done by dragging-and-dropping the most similar template, changing the values of the associated parameters, and saving the one-component network. Then, strip out all the node information from the XML file except the parameters, add any parameters that are missing, add this to the list of predefined templates, and the new component will appear the next time TOTEM starts.

Note that many changes in the AMPL model, for example new parameters or changes in the set arrangement, can be accomodated by editing `params.xml` which lists which parameters to expect. Of course, any templates that refer to out-of-date parameters or sets will need to be changed, but no source code need be changed, only the configuration files.

Major changes in the AMPL model, such as parameters which associate with outgoing edges, do require source code modifications. However, except when necessary, the TOTEM GUI aims to not be tied to the underlying AMPL model, functioning more like a passthrough between the network and the AMPL data file. This allows for the configurability that exists; one certainly does not want to have to define a new Java class when a different type of component is introduced.

C.2.2 GUI implementation details

Extending DIVA, a network graphics java package

The TOTEM GUI is built on top of Diva, a Java"software infrastructure for visualizing and interacting with dynamic information spaces" [1] designed at the Center for Embedded Systems Design at Berkeley. Diva provides the means for representing and manipulating graph structures, as well as tools for creating document-based applications, including XML parsers and application classes.

Diva was designed with the goal of being applicable to almost any graph-based application. Their design philosophy states that they "prefer interfaces over abstract

[1] http://embedded.eecs.berkeley.edu/diva/

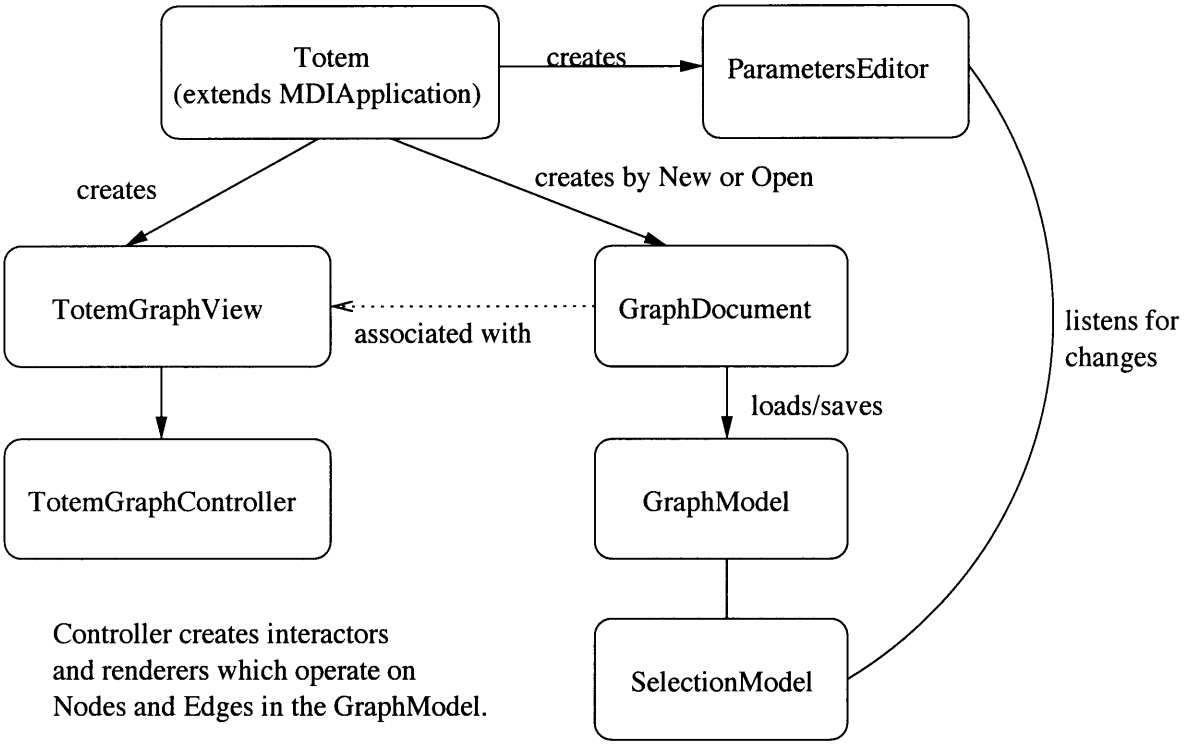


Figure C-6: Class Relationships in TOTEM GUI

classes, components over frameworks, and sub-systems over applications ... " [2] In other words, Diva is extremely modular and makes full use of object orientation, with the goal of making it easy to extend.

The Diva designers' goals are laudable and make sense when designing an infrastructure for long-term use and gradual extension. However, when creating an application based on Diva, one sometimes comes to cases where it is helpful to break the Diva guidelines somewhat to achieve additional functionality. A case in point would be the rendering of icons for different components, as described in subsection C.2.2.

Class relationships

Figure C.2.2 give the class relationships in the TOTEM GUI. At the highest level, the application itself is the class that runs when the program is launched. The `MDIApplication` class that is extended takes care of creating and displaying toolbars,

[2] http://embedded.eecs.berkeley.edu/diva/about/development/design.html

menu bars, palettes, and individual windows for each open graph document. We just need to populate these bars and palettes. `ParametersEditor` is an extension since it is not handled by the existing `MDIApplication`.

The application is connected to a factory for creating new `GraphDocument` objects (or for opening them from a file). Each `GraphDocument` object contains a `GraphModel` which defines the nodes and the edges between the nodes. When a graph document is created or opened, a `TotemGraphView` is created. A view consists of figures, lines, and other geometrical objects which represent the graph model. Note that multiple graph views on a single graph model are possible in Diva, even though we do not make use of this feature. A `TotemGraphController` object is linked to the view and handles interactions which affect the graph view – for example, creating new edges. The view and controller classes are based on the `SchematicGraphEditor` sample application included with Diva.

Parameters

The `ParametersEditor` class creates a panel with a set of labels and edit boxes for displaying and editing the parameters associated with a particular component. Parameters are stored in a `Parameters` object which is read and written to when the panel is created or changed.

When the user selects or deselects components, the `selectionChanged` method is called in listeners to the `SelectionModel`. `ParametersEditor` is added as such a listener by the main `Totem` class, and implements this method. When the selection model changes, the previous parameter values are copied from the edit boxes to the `Parameters` object, and labels and edit boxes are created for the new primary selection object's parameters.

Note that multiple selection is still permitted even though only one node's parameters are displayed. It is helpful to allow the user to select multiple nodes to delete them all at once, or to move them to another location without changing the relative locations of the group. At the same time, since the identifying name of the component is displayed in the parameters editor, little confusion should result.

It is helpful for parameters to be displayed to the user in a consistent order every time, but it is also desirable to be able to access parameters by name. Thus, a `Hashtable` is used to keep the values, but a separate hashtable assigns an integer to each parameter name so they can be accessed in order.

Drag and drop of network components

Since a template consists of a collection of parameters that pertain to a predefined component, drag-and-drop is achieved through the `Parameters` object. `TotemGraph DropTarget` creates a drop key for each template in `predefined.xml`, with the value of the drop key being a `Parameters` object created using the specified template file.

When an actual drop occurs, the `drop` method is called. We create a new node, make a deep copy of the `Parameters` object associated with this dropkey, and set this fresh copy of `Parameters` to be the semantic object for the new node.

Note that Diva does provide for properties to be associated with each node. One could use the properties to store parameters. We do not use this function since `TotemGraphDropTarget` would then have to handle adding all the properties. In the current implementation, we instead have a `Parameters` object which handles the storage of all parameters associated with a given node, encapsulating the functionality so that `TotemGraphDropTarget` need only make a deep copy of one object to accept a drop.

In addition, we end up using properties for temporary purposes throughout the program, such as for saving state between accesses to a node when it is difficult to pass such data elsewhere without further breaking the Diva object hierarchy. This is helpful when positioning the nodes in their previous places when loading a file, for example.

Reading and writing TOTEM data files

Network files are saved in an XML native format. The format is an extension of the simple node and edge tags used in the `SchematicGraphEditor` sample Diva application. As can be seen in the sample network file in Figure C.2.2, network compo-

```
<graph>
    <compositeNode id=''node_0'' x=''62.5'' y=''63.0546875''>
        <node id=''node_1'' inout=''in''/>
        <node id=''node_2'' inout=''out''/>
        <param name=''NAME'' value=''Gen''/>
        <param name=''SET'' value=''N_V''/>
        <param name=''LOWER'' value=''30''/>
        <param name=''UPPER'' value=''40''/>
        <param name=''MinUpTime'' value=''30''/>
        <param name=''MinDownTime'' value=''30''/>
    </compositeNode>
    <compositeNode id=''node_3'' x=''204.6015625'' y=''62.875''>
        <node id=''node_4'' inout=''in''/>
        <node id=''node_5'' inout=''out''/>
        <param name=''NAME'' value=''BaseElectricity''/>
        <param name=''SET'' value=''N_D_B''/>
        <param name=''b'' value=''1 2 3 4 5 6 7 8 9 10''/>
    </compositeNode>
    <edge directed=''true'' head=''node_4'' id=''edge_0''
         tail=''node_2''>
        <param name=''NAME'' value=''Edge''/>
        <param name=''Capacity'' value=''1''/>
    </edge>
</graph>
```

Figure C-7: Sample two-node saved file with a generator and base electricity demand component.

nents are represented as `compositeNodes`, with the ports being represented as non-composite `nodes`. Parameters are included as individual XML elements under each `compositeNode`. The geometrical location of each node in the layout is represented by the `x` and `y` attributes assigned to the `compositeNode`.

Saving and loading XML graph files

The loading and saving of graph files is handled within the `GraphDocument` class. The XML graph file is saved by first looping over all the nodes in a given graph model. During the loop, parameters and attributes are output, and a unique integer is associated with each node. Next, the edges are printed, looking up the appropriate node ID for each node. The use of a unique, automatically generated XML file allows

us to ensure a valid graph structure even if the user leaves a file uncompleted, such as by leaving repeated names for components.

The `save` and `saveAs` methods of the `GraphDocument` class are called by the application when the appropriate actions are performed either through the menu bar or by clicking on the icon in the toolbar. The actual work is split up into `generateNodes` and `generateEdges` methods, and contained within the `BasicGraphBuilder` class. Several classes supporting XML elements and attributes are present in Diva and are used in generating the XML file.

Because of the multiple-views capability of Diva, a graph model only contains the nodes and the edges connecting them. The locations of the nodes are a geometrical feature of the view. A view is associated with a document, but a document can be associated with multiple views. However, it is desirable for the geometrical layout of the graph to be preserved when the user saves and opens up a new file. To do this, the `GraphDocument` class associates itself with a single `TotemGraphController`, in effect associating itself with a single view of the graph model. Although this breaks the Diva hierarchy somewhat, it is necessary if layouts are to be preserved. Since we never create or use more than one view of a graph, so we do not run into any ambiguities.

Note that, at this point, we have no way of setting the geometrical location of nodes. The `TotemGraphView` is created *after* the `GraphDocument` is created. We thus save the `x` and `y` attributes as properties of the node. Then, when the `TotemGraphView` is created, we make use of these properties to move the node to the proper location. This use of properties to convey internal information within the program is made possible by the encapsulation of the `Parameters` class. Although the Diva separation of view from model forces us, in effect, to place each node twice (first in the upper-left corner, then at the saved position), the moving of nodes occurs rapidly enough to be not noticeable by the user.

When loading the file, we do the opposite. A hashtable now associates node IDs with the newly-created node, and `Edge` objects are created between `Node` objects based on this lookup table. The `generate` methods are replaced with `build` methods

when we are building a new graph model from a file.

Exporting AMPL data files

At the same time that the graph file is saved, the corresponding AMPL file is also written. The `exportAs` method of `GraphDocument` uses the `AMPLExporter` class, and is called by the `saveAs` method of `GraphDocument` once the main XML file has been written. Rather than creating a series of classes which understands AMPL in the same way that the Diva XML classes understand XML, the AMPLExporter class is a save-only class. The XML graph representation is the primary, editable representation of the energy network configuration, including layout locations and other elements which are not provided for in the AMPL model.

Furthermore, the export-only nature of the AMPL file, as well as the simultaneous saving when a relevant XML file is saved, ensures a degree of indirection between the user view and the AMPL model. For example, it may be convenient in AMPL for some properties to be associated with the node which connects to the current node, rather than to the node itself. For example, uptime and downtime constraints for generators are enforced with the supply fuel node. However, it would be confusing to the user to place the parameter on the fuel node, since these parameters are typically considered properties of a generator.

Thus, these requirements are set as XML attributes, with the XML representing the view presented to the viewer and some renaming done to create the AMPL file. Having made the decision to separate the XML from the AMPL file, we leave the door open to further extensions of the TOTEM GUI. For example, we will be able to introduce the notion of energy types (fuel, electricity, steam) to ensure a degree of reasonableness in a graph model, even though the TOTEM AMPL optimizer does not have any conception of an energy type.

Since AMPL has sets and parameters at the top level, the AMPL export code makes use of hashtables to reverse the hierarchy. First, we print the sets. We set up the hashtable by reading the `sets.xml` configuration file to determine which sets AMPL expects to find. Then, we loop over the nodes and add each node to the sets

that it belongs to, and finally print the nodes associated with each set.

Then for each node, we pick up the edge connections. Edges are connected to ports, not to the main `CompositeNode` object which represents the component. It is nontrivial to get the parent node of any noncomposite node, since the `getParent` method provided by Diva for the `Node` class gets the overall graph and there is no class designed specifically for subnodes. We handle this by going through the composite nodes and assigning a name property to each port node based on the `NAME` parameter of the composite node. At the same time, we turn around edges if they face the wrong direction, since the order of dragging determines the head or tail of a nondirected edge. Having done this, we can simply loop over all the edges going into the input port, and the edges coming out of the output port, to print the edge relationships with the correct names.

Likewise, with parameters we set up a hashtable of expected parameters by reading `params.xml`. Each parameter can have values for multiple subscripts. The subscripts generally correspond to the name of components, but if a parameter is associated with the incoming edge or incoming node, then the name is changed appropriately. Thus, the hashtable of parameters has hashtables as the values, and the inner hashtable uses the names as the key. This hashtable-of-hashtables relationship is shown in Figure C-8.

We then loop over the hashtable of parameters to print the name-value pairs to a file. Timeseries are handled at this print stage, where the string is split into an array of strings and each string in the array is printed as an individual value for a particular time index.

Rendering and interaction

The `TotemGraphController` controls interactions between the user and the graph view. Edge creation and movement are fairly straightforward and handled by the default interactors. Likewise, node movement is also handled in the default manner.

However, our need to render nodes with the name displayed requires a `TotemNode Renderer` class to be created. The `render` method is the only method of a `NodeRender`

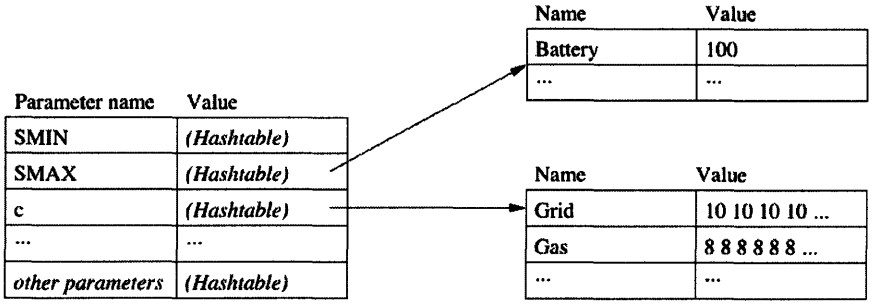


Figure C-8: Structure of `paramValues` hashtable for outputting parameters to AMPL data format.

that is required to be overridden. In this case, we create a `CompositeFigure` to contain the graphical elements used to display a node. The label is placed in this composite figure, as well as an icon (specified by the `ICON` parameter for a node). If no icon is specified, then a rectangular shape is drawn by default to represent the node.

The creation of a separate node renderer class for the TOTEM GUI permits the icon to be chosen differently for each component. Diva does provide a typed renderer functionality, where a different renderer is specified for each object type, and the typed renderer looks at the semantic object associated with each node to determine which renderer to use. However, this would require us to have a different Java class for each icon that we wish to use, which in effect requires the user to modify the source code of the TOTEM GUI each time he wishes to add a new template with a different icon. Instead, we use a separate renderer class which looks *inside* the `Parameters` object for a node continues the goal of allowing maximum configurability without

source code modification.

C.3 AMPL implementation

AMPL (A Modeling Language for Math Programming) is highly suited for the implementation of the TOTEM model. AMPL allows a user to break up the mathematical modeling two pieces.

The first is a model file containing an abstract representation of the model (see the full math model in Equations 3.28 and 3.29). This is a full description of the model, an exact copy of Equations 3.28 and 3.29 written in the AMPL language. It is not ready to be solved because the user instance data has not been specified yet. The AMPL TOTEM model is given in Appendix A.

The second file then is a data file which contains all of the user supplied, instance-specific problem information. For example, in the data file the user will describe all of his nodes, and how they are connected, the input-output relationships, hourly costs of fuel and electricity, demands, etc. By breaking up the modeling input in these two parts, model and data, we are able to use the same model description for every possible energy network. This follows the object-oriented design paradigm of separation of model and data. An example AMPL data file for TOTEM is given in Appendix B.

The popularity of AMPL stems from the fact that it is able to call many different math programming solvers with no change to the model definition step. AMPL is not an optimization solver per se, but rather provides a consistent gateway to the optimization solver of choice for the particular user. CPLEX by Ilog, Inc. is a the industry standard math program solver for integer programs, and AMPL provides a seamless interface to CPLEX. Another recommended choice for users of TOTEM is the linear mixed integer program lpsolve by Michel Berkelaar of the Technical University of The Netherlands in Eindhoven. It is free and has been compiled for the Mac OS X operating system, which has made it particularly useful for the author of this thesis.

Appendix D

Mathematica implementation of Markov Storage Solution

We show an implementation of the general Markov storage model discussed in Chapter 7 here. The coding is done in Mathematica (www.wolframresearch.com), a numerical and symbolic math software. The model calculates the steady state probabilities for a Markov chain constructed from the primitive data of hourly supply and demand levels and probabilities. An example of such data is shown in Figure 7-1 A and B. The variables s and d are the offset values for supply and demand. So, the possible demand values in the Mathematica input shown below are 10, 11, and 12.

The code first computes the transition probability matrix for the chain shown in Figure 7-1C. It the uses this to compute the steady state probabilities, and finally converts the steady state probabilities to the purchase level using an extension of Equation 7.3 which accounts for the fact that there are multiple values of potential unmet demand if storage is insufficient.

```
M = 10;
SUPPLYPROBS = {.41, 0, 0, .34, 0, 0, 0, 0, .25};
DEMANDPROBS = {.3333, .3334, .3333};
s = 5; Smax = s + Length[SUPPLYPROBS] - 1;
d = 10; Dmax = d + Length[DEMANDPROBS] - 1;

transitionProbs = Table[0, {i, -Dmax + s, M + Smax - d}, {j, -Dmax + s, M + Smax - d}];

Ps[x_] := If[x < s || x > Smax, 0, SUPPLYPROBS[[x - s + 1]]];
Pd[x_] := If[x < d || x > Dmax, 0, DEMANDPROBS[[x - d + 1]]];
For[i = -Dmax + s, i ≤ 0, i++,
  For[j = -Dmax + s, j ≤ Smax - d, j++,
    transitionProbs[[i + Dmax - s + 1, j + Dmax - s + 1]] = ∑_{k=s}^{Smax} (Ps[k] * Pd[k - j]);
   ];
 ];
For[i = 1, i ≤ M, i++,
  For[j = i - Dmax + s, j ≤ i + Smax - d, j++,
    transitionProbs[[i + Dmax - s + 1, j + Dmax - s + 1]] = ∑_{k=s}^{Smax} (Ps[k] * Pd[k - j + i]);
   ];
 ];
For[i = M + 1, i ≤ M + Smax - d, i++,
  For[j = M - Dmax + s, j ≤ M + Smax - d, j++,
    transitionProbs[[i + Dmax - s + 1, j + Dmax - s + 1]] = ∑_{k=s}^{Smax} (Ps[k] * Pd[k - j + M]);
   ];
 ];
n = Length[transitionProbs];
EE = Table[1, {n}, {n}] - IdentityMatrix[n];
ee = Table[1, {n}];
steadyStateProbs = ee.Inverse[transitionProbs + EE];

PurchaseLevel = ∑_{i=-Dmax+s}^{0} (Abs[i] * steadyStateProbs[[i + Dmax - s + 1]]);
```

Bibliography

[1] E. Allen and M. Ilić. *Price-based commitment decisions in the electricity market.* Springer, 1999.

[2] M. Anderson. *MIMO robust control for HVAC systems.* PhD thesis, Colorado State University, Fort Collins, Colorado, April 2001.

[3] ASHRAE. *ANSI/ASHRAE Standard 55-2004: Thermal Environmental Conditions for Human Occupancy.* ASHRAE, 2004.

[4] G. Bakos. Energy management method for auxiliary energy saving in a passive-solar-heated residence using low-cost off-peak electricity. *Energy and buildings*, 31:237–241, 2000.

[5] O. Berman, R. Larson, and E. Pinker. Scheduling workforce and workflow in a high volume factory. *Management Science*, 3(2):158–172, 1997.

[6] D. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 1995.

[7] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, 1997.

[8] S. Borowitz. *Farewell fossil fuels: Reviewing America's energy policy.* Plenum Trade, 1999.

[9] M. Bos and R. Beune andR. van Amerongen. On the incorporation of a heat storage device in lagrangian relaxation based algorithms for unit commitment. *Electrical Power and Energy Systems*, 18(4):207–214, 1996.

[10] P. Boulos, Z. Wu, C Orr, M. Moore, P. Hsiung, and D. Thomas. Optimal pump operation of water distribution systems using genetic algorithms. Technical report, RBF Consulting, March 2002.

[11] J. Braun. Load control using building thermal mass. *Journal of solar energy engineering*, 125:292–301, 2003.

[12] P. Carpentier, G. Cohen, J. Culioli, and A. Renaud. Stochastic optimization of unit commitment: A new decomposition framework. *IEEE Transactions on Power Systems*, 11(2):1067–1073, 1996.

[13] M. Coffin. *Direct digital control for building HVAC systems.* Kluwer Academic Publishers, 1998.

[14] P. Constantopoulos. *Computer-assisted control of electricity usage by consumers.* PhD thesis, Massachusetts Institute of Technology, Massachusetts Institute of Technology, Cambridge, MA, 02145, June 1983.

[15] P. Constantopoulos, F. Schweppe, and R. Larson. Estia: A real-time consumer control scheme for space conditioning usage under spot electricity pricing. *Computers Operations Research*, 18:751–765, 1991.

[16] B. Daryanian, R. Bohn, and R. Tabors. Optimal demand-side response to electricity spot prices for storage-type customers. *IEEE Transactions on Power Systems*, 4(3):897–903, 1989.

[17] J. Dooley. `http://energytrends.pnl.gov/usa/us004.htm`.

[18] A. Ellis, L. Estrada, C. Newcomb, and D. Corbus. Costa de cocos wind-diesel hybrid system: Results of three years of monitoring. Technical report, NREL, 2001.

[19] Mass Energy. `http://massenergy.com/Green.FAQs.html`.

[20] E. Entchev. Residential fuel cell energy systems performance optimization using soft computing techniques. *Journal of Power Sources*, 118:212–217, 2003.

[21] P. Fanger. *Thermal comfort.* McGraw-Hill, 1972.

[22] A. Faruqui and M. Mauldin. The barriers to real-time pricing: separating fact from fiction. *Public Utilities Fortnightly*, 140(14):30–40, 2002.

[23] R. Fourer, D. Gay, and B. Kernighan. *AMPL: a modeling language for mathematical programming.* Duxbury Press, 2002.

[24] P. Fraser and S. Morita. *Distributed generation in liberalised electricity markets.* International Energy Agency, 2002.

[25] G. Giebel. The state-of-the-art in short-term prediction of wind power. Technical report, Project ANEMOS, 2002.

[26] H. Glahn and D. Lowry. The use of model output statistics (mos) in objective weather forecasting. *Journal of Applied Meteorology*, 11:1203–1211, 1972.

[27] K. Hagino and H. Onishi. Control system for aies. *Pattern Recognition*, 28(10):1517–1522, 1995.

[28] R. Hamalainen, J. Saari, J. Ruusunen, and P. Pineau. Cooperative consumers in a deregulated electricity market - dynamic consumption strategies and price coordination. *Energy - The International Journal*, 25:857–875, 2000.

[29] A. Hammer, D. Heinemann, C. Hoyer, and E. Lorenz. Satellite based short-term forecasting of solar irradiance - comparison of methods and error analysis. Technical report, Department of Energy and Semiconductor Research, Faculty of Physics Carl von Ossietzky Universitat, 2000.

[30] M. Hayashi and B. Kermanshahi. Application of artificial neural network for wind speed prediction and determination of wind power generation output. Technical report, ICEE, 2001.

[31] H. Heitsch and W. Romisch. Hydro-storage subproblems in power generation: An approach with a relaxation method for network flow problems. *IEEE Bologna Power Tech Conference*, 2003.

[32] G. Henze. Impact of real-time pricing rate uncertainty on the annual performance of cool storage systems. *Energy and Buildings*, 35:313–325, 2003.

[33] B. Hobbs. *The next generation of electric power unit commitment models.* Kluwer Academic Publishers, 2001.

[34] M. Illandala and G. Venkataramanan. Battery energy storage for stand-alone micro-source distributed generation systems. *6th IASTED Intl. Conf. on Power and Energy Systems*, 2002.

[35] D. Knebel. *Simplified energy analysis using the modified bin method.* ASHRAE, 1996.

[36] J. Kondoh, I. Ishii, H. Yamaguchi, A. Murata, K. Otani, K. Sakuta, N. Higuchi, S. Sekine, and M. Kamimoto. Electrical energy storage systems for energy networks. *Energy conversion and management*, 41:1863–1874, 2000.

[37] R. Lasseter. Integration of distributed energy resources: The certs microgrid concept. Technical report, Consortium for Electric Reliability Technology Solutions, April 2002.

[38] L. Leclercq, B. Robyns, and J. Grave. Control based on fuzzy logic of a flywheel energy storage system associated with wind and diesel generators. *Mathematics and Computers in Simulation*, 63:271–280, 2003.

[39] Z. Liao and A. Dexter. The potential for energy savings in buildings through improved boiler controls. *Energy and Buildings*, 36(3):261–271, 2004.

[40] W. Liggett. `http://www.eia.doe.gov/cneaf/electricity/chg_stru_update/update2000.htm%l`.

[41] J. Littler and R. Thomas. *Design with energy: The conservation and use of energy in buildings.* Cambridge University Press, 1984.

[42] D. Mahling. Why intelligent agents can reduce energy cost better than building management systems or human experts. Technical report, WebGen Systems, April 2002.

[43] M. Mahmoud and A. Ben-Nakhi. Architecture and performance of neural networks for e?cient a/c control in buildings. *Energy Conversion and Management*, 44:3207–3226, 2003.

[44] D. Manolas, C. Frangopoulos, T. Gialamas, and D. Tsahalis. Operation optimization of an industrial cogeneration system by a genetic algorithm. *Energy Conservation and Management*, 38(15):1625–1636, 1997.

[45] K. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE transactions on Neural Networks*, 1:4–27, 1990.

[46] Eds.L. Niklasson and M. Boden. *Current trends in connectionism.* Erlbaum, 1995.

[47] U.S. Department of Energy. `http://www.eia.doe.gov/emeu/reps/rpmap/rp_new-eng.html`.

[48] J. Orlando. *Cogeneration design guide.* ASHRAE, 1996.

[49] T. Regan, H. Sinnock, and A. David. Distributed energy neural network integration system. Technical report, UMass Lowell, June 2003.

[50] P. Roberts. *The end of oil.* Houghton-Mifflin, 2004.

[51] R. Rockafellar and R. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

[52] H. Saadat. *Power Systems Analysis.* WCB/McGraw Hill, 1999.

[53] S. Takriti, J. Birge, and E. Long. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, 11(3):1497–1508, 1996.

[54] S. Takriti, B. Krasenbrink, and L. Wu. Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem. *Operations Research*, 48(2):268–280, 2000.

[55] G. Trecate, P. Letizia, and M. Spedicato. Optimization with piecewise-affine costfunctions. Technical report, Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), 2001.

[56] A. Tylor and L. Leslie. Bias and absolute error in model output statistics (mos) surface temperature forecasts for continental u.s. locations. Technical report, School of Meteorology, University of Oklahoma, 2002.

[57] C. P. Underwood. *HVAC Control Systems: Modeling, analysis, and design.* E and FN Spon, 1999.

[58] A. van Schijndel. Optimal operation of a hospital power plant. *Energy and buildings*, 34:1055–1065, 2002.

[59] A. Warren. Towards common energy standards for buildings across europe. *Energy World*, 297:8–10, 2002.

[60] Michael Wetter. `http://gundog.lbl.gov/GO/index.html`.

[61] G. Wilkins. *Technology transfer for renewable energy: Overcoming barriers in developing countries.* Earthscan Publications, Ltd., 2002.

[62] H. Willis and W. Scott. *Distributed power generation, planning and evaluation.* Marcel Dekker, Inc., 2000.

[63] H. Xing. *Building load control and optimization.* PhD thesis, Massachusetts Institute of Technology, Massachusetts Institute of Technology, Cambridge, MA, 02145, February 2004.

[64] S. Yamamoto, J-S. Park, M. Takata, K. Sasaki, and T. Hashimoto. Basic study on the prediction of solar irradiation and its application to photovoltaic-diesel

hybrid generation system. *Solar energy materials and solar cells*, 75:577–584, 2003.