Building Load Control and Optimization

By

Hai-Yun Helen Xing

M.S., Building Technology (2000)
Massachusetts Institute of Technology

Submitted to the Department of Architecture
In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Building Technology

at the
Massachusetts Institute of Technology
February 2004

© 2004 Massachusetts Institute of Technology
All rights reserved

Signature of the author…….……...

Department of Architecture
January 9, 2004

Certified by…….……
Leslie K. Norford
Professor of Building Technology
Thesis advisor

Accepted by…….……...
Stanford Anderson
Chairman, Department committee on Graduate Student
Head, Department of Architecture

THESIS COMMITTEE:

Professor Leslie Norford, Professor of Building Technology

Professor Leon Glicksman, Professor of Building Technology and Mechanical Engineering
George Macomber Professor

Professor Steven Leeb, Associate Professor of Electrical Engineering and Computer Science

Building Load Control and Optimization

By

Hai-Yun Helen Xing

Submitted to the Department of Architecture on January 9, 2004
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Building Technology

Abstract

Researchers and practitioners have proposed a variety of solutions to reduce electricity consumption and curtail peak demand. This research focuses on load control by improving the operations in existing building HVAC (Heating, Ventilating and Air-Conditioning) systems and by aggregating individual loads based on optimization studies. Emphasis is placed on electricity rates and climate data in California, where electricity costs have been of particular concern. The optimization problem in this research is multi-objective in the sense that we aim to reduce building load while maintaining an acceptable level of comfort.

The first part of this research focuses on optimizing controls in a single building. A simple three-zone VAV system model is built in EnergyPlus (E+). The cost function structures and the potential difficulties associated with simulation-based optimization are discussed. Discontinuity and nonlinearity are of major concern. Two optimization algorithms are tested and applied to a variety of problems: Direct Search (DS) and Genetic Algorithms (GA). An E+ simulation based GA optimization environment is developed in Matlab. DS is found to be efficient with small problems in this research, while GA works in almost any situation with the price of intensive computation. A few operations guidelines are proposed.

The second part of this research presents three ways of optimizing load control in an aggregation pool: Enumeration, multi-GA and model-based nonlinear optimization. Enumeration relies on expert rules to find a small set of feasible solutions through automated E+ simulations and search exhaustively for the optimal solution. Multi-GA solves the aggregation problem in the Matlab GA environment with sequential E+ simulations as the function evaluator. Because simulation-based optimization is very computationally intensive in handling multiple buildings, the model-based approach develops for each aggregation participant a time series model and several regression models to predict individual load profiles under load control. It then applies an interior-point-method-based commercial solver LOQO to these simplified building models. This system is fast and easy to scale up. Certain precision is lost due to modeling simplifications, but the results are still satisfactory for practice purposes.

Overall, load aggregation offers load diversification opportunities among participants and improves the aggregated load profile. Load shedding later individual load profiles in a way that enhances the aggregation performance.

Thesis Supervisor: Prof. Leslie Norford
Title: Professor of Building Technology

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my advisor Professor Leslie Norford for his insight, guidance and enormous support in the past five years. I wouldn't be able to finish this work without his help. He has always been patient, understanding, and open-minded. He has encouraged me to explore the unknown and to pursue my interest. I am fortunate to have him as the advisor.

I also want to thank my committee members Professors Leon Glicksman and Steve Leeb for their advice. Working with them has been a pleasant experience and their impact on me goes beyond research.

A special thank-you to Phil Haves at LBNL. His encouragement has accompanied me since the Tsinghua years. He also offered valuable advice on load shedding issues in this work.

Many thanks to the California Energy Commission for funding this project. I appreciate the financial support from the Institute in the past five years.

Many colleagues and friends helped me along the way. Michael Wetter at Laurence Berkeley National Lab advised me with great patience on GenOpt and Direct Search. Michael Witte and the EnergyPlus support team helped me debug EnergyPlus code. Colleagues at the MIT Operations Research Center helped me with both course work and my research. Discussions with Henry Spindler and Peter Armstrong have been helpful and enjoyable. Henry Spindler and Phil Sun offered good suggestions in improving the thesis presentation.

My deepest gratitude goes to my parents and my brother. They are always there for me and never have any doubt. I thank Tony for his love and support, and for going through the difficult times with me.

TABLE OF CONTENTS

Abstract 5

Acknowledgements 7

List of Tables 12

List of Figures 14

Chapter 1 Introduction 19

1.1 Introduction 19

1.2 Research overview 20

1.3 Literature research 21

1.4 Problem description 37

Chapter 2 Single Building Simulation-Based Parametric Studies on a Simple VAV System 41

2.1 Model description 41

2.2 Model test 45

2.3 Energyplus-based parametric studies – part I 46

2.4 Energyplus-based parametric studies – part II 52

2.5 Another load shedding strategy – night cooling 64

2.6 Single building load control summary 74

Chapter 3 Single Building EnergyPlus Simulation-Based Optimization 77

3.1 Direct Search algorithms and GenOpt 77

3.2 Genetic Algorithms and GAOT 81

3.3 Cost function structure 83

3.4 Single building optimization based on Direct Search and EnergyPlus 87

3.5 A GA-based optimizer for single-building study 92

3.6 Algorithm comparison 100

3.7 A hybrid algorithm – future work 100

Chapter 4 Simulation-Based Multi-Building Optimization 103

4.1 Smart Enumeration – a rule based engineering approach 103

4.2 Set thermostat set points for multiple buildings by Smart Enumeration 104

4.3 Matching results of a two-building case for night cooling based load control 121
4.4 A GA approach to the multi-building problem 129
4.5 Computation concerns of simulation-based approaches 135
4.6 Multi-building optimization – economy of scale 136

Chapter 5 A Model -Based Nonlinear Optimization Approach to the Multi-Building Problem 141
5.1 Problem formulation 141
5.2 Base load predictor – a time series model 144
5.3 Load reduction model 157
5.4 Comfort model 162
5.5 A nonlinear central optimizer 165
5.6 Comparison of three approaches to the multi-building problem 170

Chapter 6 Non-Technical Aspects of the Load Control Problem 171
6.1 Energy crisis review 171
6.2 Non-technical issues 174

Chapter 7 Conclusions 179
7.1 Conclusions 179
7.2 Future work 182

References 185

Appendix A
A.1 EnergyPlus input file of the base model 189
A.2 Core of the EnergyPlus input file 189

Appendix B
B.1 Matlab GA code for both single and multiple buildings 191
B.2 GenOpt code 202
B.3 VBA post processing 204

Appendix C
C.1 EnergyPlus key inputs for three models used in the aggregation studies 206

C.2 Matlab code for exhaustive search in Enumeration 206
C.3 VBA code for feasibility check in Enumeration 209

Appendix D
D.1 SPlus code for model identification 216
D.2 AMPL code of the nonlinear optimizer 223

LIST OF TABLES

Table	Table name and content	Page
2.1	Basic building model	44
2.2	Demand reduction vs. Load shedding strategies	59
2.3	Night cooling summary	74
3.1	Optimize peak load with different start points through Direct Search	89
3.2	Optimize different objective functions through Direct Search	90
3.3	Example rate structure	89
3.4	Optimize trade-off between energy and comfort through Direct Search	91
3.5	Performance of the Matlab-GA results	100
3.6	Comparison of GA and DS	101
4.1	Building model types used in the multi-building research	105
4.2	Summary of simple thermostat-based aggregation cases	108
4.3	Individual load control vs. aggregated load control	109
4.4	Example rate structure	110
4.5	Summary of simple aggregation cases, cost-based	111
4.6	Individual load control vs. aggregated load control, cost-based	111
4.7	Two EnergyPlus models used for night cooling	123
4.8	Night cooling schedules	123
4.9	Night cooling based load aggregation: peak load and total energy cost	123
4.10	Aggregated night cooling details: contribution by individual participants	124
4.11	Optimizing two-building thermostat set points with the Matlab GA	131

Table	Table name and content	Page
4.12	Optimizing two-building night cooling schedules with the Matlab GA - peak load as the cost function	132
4.13	Optimizing two-building night cooling schedules with Matlab GA- total cost with a $6.5/kW demand charge	132
4.14	Optimizing two-building night cooling schedules with Matlab GA- total cost with a $1.5/kW demand charge	132
4.15	EnergyPlus runs taken by multi-GA	136
4.16	Total computation time comparison between Enumeration and multi-GA	136
4.17	Aggregation performance – mix matters	138
4.18	Aggregation performance – economy of scale	138
5.1	Model Coefficients and t- values for building 1 - A first cut	148
5.2	Model coefficients and t values for building 1- with only significant variables	150
5.3	Peak load as a function of exogenous variables	153
5.4	SPlus regression results for load reduction model at different hours	159
5.5	Load reduction model prediction errors in the testing set	159
5.6	PMV model prediction errors in the testing set	165
5.7	Model-based optimizer vs. Enumeration	168
5.8	Aggregation using the model-based approach	169
5.9	Comparison between three approaches to the multi-building problem	170

LIST OF FIGURES

Figure	Figure name and content	Page
2.1	Floor plan of the three-zone VAV system	42
2.2	Plant layout of the three-zone VAV system	42
2.3	Air loop of the three-zone VAV system	43
2.4	Daytime average chiller, fan, total power, and PPD vary with thermostats	47
2.5	Total, chiller and fan power, and PPD vary with the combination of supply air temperature and chilled water temperature with the fan capacity fixed at full	49
2.6	Chiller power varies with chilled water temperature and fan power varies with supply air temperature, at full and 75% fan capacity	50
2.7	Total, chiller and fan power vary with combination of supply air temperature and chilled water temperature at full and 75% fan capacity	50
2.8	Chiller and fan power vary with economizer set points	51
2.9	Power and PPD profiles with hours 14-17 thermostats adjustment	54
2.10	Daytime total and chiller power reductions with different thermostat set points during hours 14-17	54
2.11	Average total power reductions and contributions by fan and chiller with supply fan capacity reduction and hour 14-17 thermostats adjustment	55
2.12	Power and PPD profiles with combinations of supply air temperature increases, chilled water temperature increases, and fan capacity reduction	57
2.13	Average total demand reduction and contributions by fan and pump and by chiller for combinations of supply air temperature increase, chilled water temperature increase, and fan capacity reduction	58
2.14	Daytime average power consumption and peak load with different scheduling durations and starting times under thermostat-based load control	61
2.15	Power and PPD profiles with different scheduling durations and starting times under thermostat-based load control	61

Figure	Figure name and content	Page
2.16	Average power consumption and PPD for three chiller on/off cases and three types of thermal mass	62
2.17	Power and PPD profiles vary with thermal mass under temporary chiller-off strategies and load setback recovery comparison between 1mass and 3mass	63
2.18	Load profiles and peak/average load differences of night cooling by mechanical ventilation with different discharging processes	66
2.19	Load profiles and peak/average load differences of chiller-based night cooling with different discharging processes	67
2.20	Load profiles and summary statistics of night cooling by mechanical ventilation with different fan starting times	70
2.21	Load profiles and summary statistics of chiller-based night cooling with different chiller starting times,	71
2.22	Load profiles of no night cooling, chiller-based and fan-based night cooling, LA and Austin	72
2.23	Peak load and average load comparison between three thermal mass types, three night cooling strategies and two locations of LA and Austin	73
3.1	GenOpt overall organization	80
3.2	GenOpt optimizer class	80
3.3	EnergyPlus-based GAOT scheme	82
3.4	Cost function surface of total load varies with hours 16 and 17 thermostats	85
3.5	Cost function surface of PPD varies with hours 16 and 17 thermostats	85
3.6	Cost function surface of peak load varies with hours 16 and 17 thermostats	86
3.7	Power profiles of different scenarios with hours 16 and 17 thermostats as control variables	86
3.8	Cost function surface of aggregated cost (total load + weight * PPD) varies with hr16, hr17 thermostats - a complicated cost function structure with local optima	87
3.9	Direct Search in GenOpt optimizes total load over hours 14-17 thermostats without comfort constraint	88

Figure	Figure name and content	Page
3.10	Optimal power and PPD profiles of several thermostat-based operation strategies with different cost functions	90
3.11	Direct Search finds multi-objective trade-off between power and thermal comfort by varying thermostat set points during hrs 14, 15, 16, and 17	92
3.12	GA varies hrs 14-17 thermostats to minimize the total daily load: GA best and average individual traces throughout all generations	94
3.13	GA varies hrs 14-17 thermostats to minimize the total daily load: base and optimal power profiles	94
3.14	GA varies 10 work time thermostats to minimize the total electricity cost: set point, power and PPD profiles	97
3.15	GA varies 4 early morning thermostats and fan starting time to minimize the peak demand and a weighted sum of total and peak demand: thermostat set points, power and PPD profiles of the base case and the optimal case	98
3.16	GA varies all ten work time thermostats and fan starting time to minimize the peak demand and a weighted sum of total and peak demand: thermostat set points, power and PPD profiles of the base case and the optimal case	99
4.1	Individual load profiles of buildings 1, 2 and 3	112
4.2	“Optimal” load aggregation between buildings 1 and 2	112
4.3	“Optimal” load aggregation between buildings 1 and 3	113
4.4	“Optimal” load aggregation between buildings 2 and 3	113
4.5	“Optimal” load aggregation among buildings 1, 2 and 3	114
4.6	PPD profiles corresponding to three-building optimal” load aggregation	114
4.7	Comparisons between aggregated and individual load control, buildings 1 and 2	115
4.8	Comparisons between the base case, individual load control, and aggregated load control, buildings 1 and 2	115
4.9	“Optimal” load aggregation between buildings 1and 1	116
4.10	“Optimal” load aggregation between buildings 2and 2	116

Figure	Figure name and content	Page
4.11	“Optimal” load aggregation between buildings 1 and 2, cost-based	117
4.12	“Optimal” load aggregation between buildings 1 and 3, cost-based	118
4.13	“Optimal” load aggregation between buildings 2 and 3, cost-based	119
4.14	“Optimal” load aggregation between buildings1, 2 and 3, cost-based	120
4.15	Cost comparison for two-building and three-building aggregation	121
4.16	“Optimal” load aggregation between models 1 and 2 with night cooling available, peak-load based	125
4.17	PPD plots corresponding to “Optimal” load aggregation between models 1 and 2 with night cooling available, peak-load based	125
4.18	Optimal load aggregation between models 1 and 2 with night cooling available, cost based with $6.5/kW demand charge	126
4.19	Optimal load aggregation between models 1 and 2 with night cooling available, cost based with varying demand charges below $6.5/kW	126
4.20	A sequential GA process for a two-building aggregation case	129
4.21	Traces of the Matlab GA process for a two-building aggregation case	131
4.22	Aggregated and individual power profiles for the base and GA optimal cases, two-building fan-based night cooling to minimize the aggregated peak	133
4.23	Aggregated and individual power profiles for GA and Enumeration optimal cases, two-building fan-based night cooling to minimize the aggregated peak	133
4.24	Aggregated and individual power profiles for base and the GA optimal cases, two building fan-based night cooling to minimize the total cost	134
4.25	Aggregated power profiles for the base, GA and Enumeration optimal cases, 2bldg fan-based night cooling to minimize the total cost	134
4.26	“Optimal” load aggregation between buildings 1, 1, and 3	139
4.27	“Optimal” load aggregation between buildings 1, 1, 3, and 3	139
5.1	ACF plots of differencing schemes	146

Figure	Figure name and content	Page
5.2	In-sample residual info: residual plots and ACF plots of residuals	149
5.3	Testing data residuals with the model of Eqn5.4, building type model 1	150
5.4	Relationship between peak load and maximum temperatures in summer, LA	153
5.5	Base load profiles prediction and peak correction	154
5.6	Base load profile prediction vs. simulation	155
5.7	Prediction performance for specific hours over the last 20 days in August	156
5.8	Base load profile prediction improvement based on similar day	157
5.9	Power reduction at hours 11, 12, 13 and 14 vary with base load and thermostat increases	160
5.10	Power reduction at hour 14 vary with base load and thermostat increase	160
5.11	Power reduction at hours 13-17 vary with thermostat increases	161
5.12	Hour14 power reduction prediction vs. simulation	161
5.13	PMV increases at hours 12-17 vary with thermostat change	164
5.14	PMV increases at hour 14 vary with base load under different thermostat increases	164
5.15	Relationship between daytime peak PMV values and peak load in summer, LA	165
5.16	Aggregating two identical buildings with the model-based approach vs. simulation + Enumeration	169

CHAPTER ONE
INTRODUCTION

1.1 Background

This thesis work was spurred by the recent energy issues in California, where high peak demand and lack of supply growth created electricity shortages and resulted in high cost and economic inefficiency in 2001. In summer months in California, air conditioning (AC) accounts for 29% of the peak demand with residential AC load contributing 14% and commercial AC load 15% [Ilic 2002]. A variety of solutions have been proposed to reduce the overall electricity consumption and curtail peak demand [Rabl et al. 1991, Keeney et al. 1996, Braun et al. 2001]. Our research focuses on the load control by improving the operations in the existing building HVAC (Heating, Ventilating and Air-Conditioning) systems and by aggregating the individual loads based on optimization studies. The optimization problem in this research is multi-objective in the sense that we aim to reduce building electricity consumption while maintaining an acceptable service level – a reasonably comfortable indoor environment.

Electric load aggregation is considered an effective means of maximizing savings and mitigating risks in today's emerging power markets. Load aggregation is the process by which individual energy users band together in an alliance to secure more competitive prices than they might otherwise receive working independently. Aggregation can be accomplished through a simple pooling arrangement or through the formation of clusters where individual contracts are negotiated between the suppliers and each member of the aggregate group. Load aggregation has the following benefits: 1) increased buying power lowers per unit cost for pool members; 2) load diversity among multiple facilities improves load factors, which leads to a smaller demand charge; 3) load aggregation reduces transaction costs and creates economies of scale; 4) a facility may be able to realize significant savings by acquiring a portfolio of energy products that meets its anticipated needs more efficiently than a full-requirements contract. The candidate buildings do not have to be physically close, and being on the same utility bill with demand charge applied is enough.

A natural question faced by load aggregators is which buildings should go into the aggregation pool. A load aggregator should choose a variety of individual profiles and take advantage of diversification to make aggregation effective. Our research answers this question in a proactive way by allowing load shedding in individual buildings. Based on the improved individual profiles, we explore the cooperation nature between buildings aggregated. By changing building operations temporarily, load shedding offers opportunities to reduce and/or shift peak demand. For example, one building has a much larger peak

demand than another; it might help to curtail the chiller load of the smaller-load building at the time when the larger-load building reaches its peak, so that the coincident peak is reduced. The "cushion" effect of building thermal mass on indoor thermal environment and human beings' adaptability to varying thermal environment allow load to be shifted to a different time without degrading service level.

A building electric load consists mainly of the electricity consumed by lighting, equipment, and HVAC. Cutting equipment electricity use might cause building malfunction and therefore would not be considered a viable cost-saving approach in this research. Lighting control is straightforward, as the optimal strategy in summer would always be to keep the lowest acceptable lighting level to minimize cooling load. This work focuses on controlling HVAC electricity use. We will explore several major operation changes such as increasing thermostat set points and shutting down chillers temporarily and will look at the optimal ways of determining these parameters. We choose those control variables that are easy for building operators to change and have good load shedding potential.

1.2 Research overview

In this section we give a brief introduction to what this research intends to accomplish. Related literature will be reviewed in the next section. The entire thesis is to answer one central question – on a short-term basis, e.g. a day or even several hours, how a building operator should control the operations of the target building(s) to minimize energy cost. It could be a building or a group of buildings if aggregation is available. Several key questions are as follows.

- What load control strategies can be implemented?
  A variety of load control approaches and their performance are reviewed. Load control scheduling is often a companion problem. Comparison between strategies will be made in Chapters 2-4 with load control implanted to a specific VAV model in this research.
- What optimization algorithms and/or systems are used?
  We will review optimization algorithms used in previous building optimization research, their global convergence and computational intensity, ways of handling the multi-objective aspect of the problem, and ease of integration with simulation.
- How are building dynamics represented?
  Optimization requires an objective function evaluator – a load model for a building system. It is implemented in two ways in this research: full-scale simulation using EnergyPlus and a simplified load model. A variety of simulation models, including full-scale packages, statistical approximations

and those in between, are reviewed and compared regarding accuracy, computational intensity and ease of integration with optimization.

- How is the aggregation aspect of the multi-building problem captured?
  Direct load control research in electrical engineering is reviewed as it deals with a certain type of load control with multiple participants involved. Several optimization schemes are also discussed to handle aggregation.

It is to be noted that most of the examples in this thesis minimize the peak demand. This, as we will argue late in this research, is mathematically equivalent to minimizing the energy cost in terms of optimization problem structure. Although these two may produce different optimization results, the difference is only a matter of implementation decided by the pricing vector or rate structure used, as the analysis is identical.

1.3 Literature review

This section reviews previous research addressing the key questions raised in Section 1.2: simulation, optimization, load control strategies and aggregation concerns. We try to address them separately, but most load control research projects cover more than one aspect and therefore only the most important one is emphasized.

1.3.1 Simulation

A big portion of our research relies on building simulation to handle the complex building and plant dynamics. A simulator is essentially a function evaluator in many optimization systems. Three types of simulation are common in research: full-scale simulation package, simplified models, and statistics-based simulation.

a) Full-scale simulator

EnergyPlus, DOE2 and BLAST are examples of full-scale system simulation packages. They cover a wide range of building systems and components, take detailed system description and produce a large number of energy and comfort outputs. Writing modeling script can be quite laborious if started from the beginning, but with knowledge of the software and understanding of the building system, the process does not require sophisticated physics-based modeling skills.

DOE-2 and BLAST are two building energy simulation programs widely used and supported by the US government for more than 20 years. The main difference between the programs is the load calculation method – DOE-2 uses a room weighting factor approach while BLAST uses a heat balance approach. A new energy simulation program, EnergyPlus [Crawley et al. 2001] [EnergyPlus 2003] is built on BLAST and DOE-2 but with a better modular program structure. The major improvement in EnergyPlus over previous energy simulation programs is an integrated (simultaneous loads and systems) simulation for accurate temperature and comfort prediction, rather than taking a sequential approach as in DOE2. In detail, the process in EnergyPlus is referred to as a Predictor-Corrector process. Loads calculated (by a heat balance engine) at a user-specified time step (15-minute default) are passed to the building systems simulation modules at the same time step. The building systems simulation module, with a variable time step (down to seconds if necessary), calculates heating and cooling system and plant and electrical system response. Feedback from the building systems simulation module to loads not met is reflected in the next time step of the load calculations in adjusted space temperatures and humidity if necessary. As a comparison, the sequential approach in DOE2 uses a room weighting factor and calculates the zone conditions and determines all heating/cooling loads at all time steps; this information is fed to the air handling simulation to determine system response, and that response does not affect zone conditions; similarly the system information is passed to the plant simulation without feedback. This sequential technique works well when the system response is a well-defined function of the zone temperature. However, in most cases, the system capacity also depends on outside conditions and/or other parameters of the conditioned space. EnergyPlus realizes the fully integrated simulation of loads, systems, and plant through the building systems simulation manager, which makes the simulation modular and extensible.

For the heat and mass balance simulation, the hardwired 'template' systems (VAV, Constant Volume Reheat, etc.) of DOE-2 and BLAST are replaced in EnergyPlus by user-configurable heating and cooling equipment components formerly within the template. This gives users much more flexibility in matching their simulation to the actual system configurations. EnergyPlus [Crawley et al. 2001] allows users to evaluate realistic system controls, moisture adsorption and desorption in building elements, radiant heating and cooling systems, and interzone air flow – little of which can be simulated well before.

A full-scale simulation package can be plugged in the optimization process, but the full-scale simulator would make the process time-consuming and data processing complex. Such a simulator considers many design and operation aspects, and certain parameters we are particularly interested in are likely be buried in overwhelming details. Although we can post-process the simulation results as we will do late in this

research, this approach provides no direct relation to and insight on how those parameters affect load control.

b) Statistical simulator

Statistical function approximation is a widely-used approach to represent the nonlinear building dynamics. A variety of artificial neural networks (ANNs) and time series models have been used in load prediction and control research.

- ANNs

ANNs take advantage of the highly nonlinear properties of their architecture and are able to replicate precisely a variety of dynamics given appropriate training. Large amount of experimental or simulation data are required to train ANNs. Although able to represent complex nonlinearity, ANNs give little insight into the system physics.

[Narendra and Parthasarathy 1990] introduces in detail the concepts of using ANNs to identify and control a dynamic system and demonstrates them using several examples. The paper emphasizes models for both identification and control. Static and dynamic back-propagation methods for the adjustment of parameters are discussed. Multilayer and recurrent networks are compared and shown to be closely related, so that they can be studied in a unified fashion. Based on this, the concept of generalized neural networks is presented with four system setups, so that most nonlinear dynamic systems can be generated. Eleven examples based on different plant models are presented to show how the identification and control can be done for nonlinear dynamic systems using neural networks. Of these examples, the identification and/or control results are compared with those of the reference models. The comparison shows that neural networks perform well.

The concept of a general regression neural network (GRNN) is presented in [Specht 1991] as an innovative algorithm of neural network training. GRNN is a memory-based network that provides estimates of continuous variables and converges to the underlying (linear or nonlinear) regression surface. GRNN is a one-pass learning algorithm with a highly parallel structure. Compared to the back-propagation (BP) algorithm, GRNN is more computationally efficient. In many cases, BP tends to take a large number of iterations to converge to the desired solution. A similar one-pass neural network learning algorithm is the probabilistic neural network (PNN) [Specht 1990]. It is an alternative to BP in classification problems.

A few tools for system identification and control with neural networks have been developed. If used properly, these tools can potentially make an application problem easier. Some examples of general purpose software that might be applied to system identification and, to a very limited extent, control system design are NeuralWorks Professional II/PLUS from NeuralWare Inc. [Neuralworks 2003], the Neural Network Toolbox for MATLAB from The MatchWorks Inc. [Mathworks 2003], and NeuroSolutions from NeuroDimension Inc. [Neurosolutions 2003].

In recent years, a wide range of HVAC applications have found neural networks useful. An ANN model [Anstett and Kreider 1993] is used to predict energy use in a complex institutional building without the need for a data acquisition system. The normal predications were done using a formula that was given by a previously developed energy management system using linear regression and other statistical measures. The motivations of incorporating neural networks into the system are 1) to improve the predictive performance; and 2) to provide adaptability to changes in the building's use and energy plant configuration by taking advantage of the fact that ANN can be developed to update automatically their learned knowledge over time. Ten independent variables are used as inputs, including times, schedule of operations, and air/water temperatures. Four dependent variables, neural network outputs, are usages of steam, electric, natural gas and water. BP is applied as the training algorithm. Several configurations and different parameters are studied and compared. The results show that ANNs are useful for predicting energy consumption in buildings even with no data acquisition system present.

[Curtis et al. 1993] discusses the results from a computer simulation that used ANNs for predictive control of a hot water coil used to warm an air stream. The coil model itself is a neural network that has been trained on actual data and mimics the nonlinearities of the coil well. Normal PID control of this process has not been very successful, since the controller, feedback, and auxiliary inputs vary across a wide range of values. Based on the system modeled by a well-trained ANN, the predictive control performances are compared between a conventional PID controller and two types of ANN controllers: FANN (future ANN) and IANN (integrated ANN). IANN takes the RMS error over the predicting windows and uses that in the back propagation, but FANN only looks at a single error at some point in the future. The results show that both FANN and IANN have the potential to outperform the standard PID algorithm. Overall, this research shows that neural networks can be used for adaptive and predictive control of a building systems process. The controller is adaptive in the sense that the output of the network used to model the process reflects the changing operating environment, and it is predictive because it examines the future effect of the current controller action.

As an alternative to the BP algorithm and a promising method with computational efficiency and simplicity to implement, GRNN and its applications in HVAC process identification and control have been explored. A local HVAC control example of a heating coil [Ahmed et al. 1996] is chosen to test the GRNN's effectiveness. A control topology combining feedforward and feedback algorithms is chosen to demonstrate the principle of GRNN and to discuss the role of GRNN in identifying and controlling HVAC control processes. By using this combination topology, the majority of the control signal can be generated from the feedforward block such that the feedback block only deals with a small steady-state error. As a result, the control speed is improved in tracking the set point change. The feedforward component employs a GRNN for HVAC system identification and control, while the feedback component provides a control signal to offset any steady-state error. The GRNN is used to capture the static characteristics for both valves/dampers and coils. Both simulated and experimental characteristics are used as identification as well test data for the GRNN. The GRNN captures the characteristics well and due to its simplicity exhibits promise for implementation in real controllers. The combined topology algorithm uses GRNN to identify static characteristics and then subsequently uses those in a feedforward controller to generate control signals.

A related research project [Ahmed 1998] compares the combined control topology with the feedback controller for laboratory HVAC applications. The comparison is made for the pressure control sequences commonly found in a laboratory with a VAV system. The control sequence for pressure is developed and a simulation model is built. Simulated results are then presented for the combined, feedforward only, and conventional feedback control approaches. The results indicate that the combined approach performs better than the feedback approach over widely varying operating conditions and different damper characteristics. The combined approach is stable and eliminates all steady-state errors.

To build the load model in our research, ANN could be constructed with previous states, e.g. zone temperature, controls, e.g. chiller status and thermostat set points, and current outdoor temperature and solar radiation as inputs and new states and energy performance as outputs. It can be trained offline by feeding the network simulation or experimental data. A large amount of data will be needed, which is a disadvantage. A neural network model can be hooked up with the optimizer fairly easily. An automatic training process with updated data is desirable.

- Time series

Gross [1987] gives a thorough and thoughtful review of the short-term load forecasting, which is the prediction of the system load over an interval ranging from one hour to one week. The paper discusses

the nature of the load and the different factors, including economic, time, weather and random effects, influencing its behavior. A detailed classification of the types of load modeling and forecasting techniques is presented. It reviews the peak load models and the load shape models. The latter is categorized into two basic classes: times of day, e.g. spectral decomposition models, and dynamic models, e.g. ARMA and state-space models. Dynamic models represent the stochastically correlated nature of the load process, meaning that the load is not only a function of the time of day, but also of its most recent behavior, as well weather and random inputs. [Papalexopoulos et al. 1990] presented a solid example of a linear regression-based model for short-term load forecasts. Its innovations include modeling holiday effects using binary variables, modeling temperature using heating and cooling degree functions and robust parameter estimation using weighted least-squares linear regression techniques.

The ASHRAE Application Handbook [1995] reviews some of load forecasting models specifically for buildings. MacArthur et al. [1989] presented a load profile predication algorithm that regresses the current power consumption to its past values and the time series of exogenous variables such as temperatures. The algorithm uses a series of recursive least-squares estimators with each having a sample time of one day, so that accurate predications are not limited to one sample time, e.g., an hour, and load profiles for at least a 24-hour period can be obtained. A very simple algorithm for forecasting either cooling or electrical requirements that does not use the 24-hour regressor was presented by Seem et al. [1989] and then further developed and validated by Seem and Braun [1991]. In [Seem and Braun 1991] the average time-of-day and time-of-week trends are modeled using a lookup table with time of day and type of day as the deterministic input variables. Entries in the table are updated using an exponentially weighted moving-average (EWMA) model. Furthermore, the forecasts are corrected through an improved peak load based on a correlation between peak demand and maximum daily temperature forecasts. Residuals are modeled using an auto-regressive moving average (ARMA) model.

Armstrong [2004] develops a transfer function model predicting the conductive cooling load. Together with the time series data of solar radiation, convective heat transfer and outdoor temperatures, and empirical models for chillers, the model relates the detailed dynamic heat transfer process to the plant power consumption.

c) Simplified models

Simplified models fall between full-scale simulation and statistical models. They consist of approximate functional relations for components and systems under study, which makes them more computationally

efficient than full-scale simulation while providing a fair amount of insight into the energy balance and transfer processes.

For chilled water systems that do not have significant thermal storage, a component-based nonlinear algorithm [Braun 1989a] was developed to optimize the system over continuous control variables. This constrained nonlinear procedure was then used as a simulation tool for investigating the optimal system performance. In this nonlinear optimization process, the operating cost and the output of each component in a chilled water system were approximated using a quadratic and linear form respectively. Results of this algorithm led to the development of a simpler system-based methodology for near-optimal control that is simple enough for on-line implementation.

In load control research, the transfer function plays an important role in simplified models. A simulation environment is described in [Braun et al. 2001] in which an inverse modeling approach is taken. The inverse model is based on a transfer function and uses measured data to 'learn' system behavior and provide relatively accurate site-specific performance predictions. Component (fan and chiller) power models are quadratic functions of flow or temperature variables.

A model used and validated by Morris [1994] is used to enrich a simulation tool [Keeney and Braun 1996] to develop and evaluate control simplifications and strategies. Keeney et al. set up a simulation environment by using the multi-zone building energy analysis subroutine of the dynamic simulation program TRNSYS [Klein et al. 1990] and the empirical functions developed by Braun [Braun 1989.1] for modeling cooling plant power consumption.

An inverse model [Braun 2001] was used to explore the effect of different building thermal mass control strategies on the energy cost. Models are built to represent the behavior of the building, cooling plant, and air distribution system. The transfer functions are used to predict sensible cooling requirements for the building. Empirical or regression results are used for power consumption. Particularly the plant power model is obtained statistically by regressing the power to a polynomial of chilled water temperature, ambient wet bulb temperature and their squares. Several thermal mass control heuristics with different set point adjustments are compared using this tool.

Armstrong [2004] developed a transfer function based discrete-time, linear and time-invariant system to characterize envelope thermal response, improved the model to preserve its physical feasibility, and estimated the updated model using a nonlinear least squares method. Internal loads are exogenous

variables. The chiller power is characterized by an empirical relation [Ng 1999], and is a function of the cooling load, which bridges the zone temperatures to the power consumption. Certain optimization processes can be applied.

[Wright and Farmani 2001] optimized simultaneously a building's fabric, the size of HVAC system, and the HVAC system supervisor control strategies using a genetic algorithm. A single zone lumped capacitance model was used to represent the thermal response of the zone, while the HVAC system performance has been simulated using steady component models.

[Constantopoulos et al. 1991] came up with a real-time consumer control scheme for space conditioning under spot electricity pricing. The key assumptions made in building the simulation model are: 1) single conditioned space - neglect circulation effects and assume uniform inside temperature and humidity; 2) lumped model – the shell, the air mass and the other contents of the space have a combined thermal mass; 3) no independent thermal storage is coupled to the main heating or cooling equipment - assume a single piece of equipment; 4) neglect humidity control and focus on temperature control alone; 5) neglect the cycling effect of the thermostat.

d) What simulation approach to use?

The first question is what the precise goals of the simulations. We need a model that has both dynamic building modeling and plant modeling. We need to take into consideration the plant component part-load performance which is important for load control. We need to be able to vary parameters such as temperature set point, supply air temperature, chilled water temperature, and chiller and fan status on an hourly basis for studying a variety of load control strategies. A full-scale simulation package like EnergyPlus offers all these, and therefore becomes our choice. Later in this research, we have built our own simplified model which preserves several important modeling aspects.

1.3.2 Optimization and load control

As the difference in [Morris et al. 1994] and [Conniff 1991] indicates, whether or not a control strategy is optimized has tremendous impact on the energy performance. This section is categorized by the optimization methods used in load control and related problems. As a more general field, optimal control is reviewed briefly first. Then attention is turned to the optimal control problems in the building industry and the ways optimization approaches have been applied. A variety of optimization methods have been applied in building control problems and only a few major ones are studied and discussed here: linear and non-linear optimization (LP & NLP), dynamic programming (DP), linear-quadratic optimal control (LQ)

and genetic algorithm (GA). As an indispensable part of optimal control research, different simulation techniques are also reviewed and the integration of simulation and optimization is emphasized. Some references, although not directly related to building industry, are discussed as well because they help understand the methodologies useful to the load control research. Comments are made during the discussion to relate the reference to the load control problem.

a) Optimal Control in general

From the point of view of control theories, optimal control is one particular branch of modern control that sets out to provide analytical designs of a specially appealing type. The system under optimal control not only satisfies the desirable constraints associated with classical control, but it is supposed to be the best possible system of a particular type.

From the point of view of mathematics, optimal control problems are among the most difficult of optimization problems with equality constraints in terms of differential/difference equations and various boundary conditions, while inequality constraints may involve boundary conditions, entire trajectories, and controls [Sage 1977]. The two major theoretical bases in the theory of optimal control are dynamic programming by Bellman and the minimum principle by Pontryagin. The dynamic programming approach is a natural fit for developing the basic relations in the discrete-time optimal control, whereas the minimum principle approach is more suitable for the continuous-time domain. Unfortunately, often times we have to face in complex engineering systems the problem of finding a global optimum for a non-linear optimization problem, which is algorithmically and computationally difficult. In practice, heuristic-based algorithms, such as GA, and direct search methods, such as the Hooke-Jeeves algorithm, are widely used due to their practical efficiency and ease of implement.

From the point of view of research in building industry, the term “optimal control” has been used rather loosely when referring to the control of building operations. Two major methods have been widely used. One is to follow the strict definition of optimal control by proposing optimization algorithms to minimize the cost function. For example, Keeney and Braun [1996] defined the cost as a combination of energy cost and penalized human comfort and minimized it by using the complex method, which is a direct search method that generates a shape in the control variable space that always encloses the minimum point. The other is to conduct extensive simulations with different parameter combinations; the comparison among those simulations indicates the optimal one, which is, to be precise, a suboptimal solution. For example, Henze et al. [1997] developed a simulation environment to investigate a wide

range of key parameters influencing the system's operating cost. The optimal control strategy to minimize the total electricity cost was validated based on the simulation results.

b) Linear and Non-Linear Programming

Two methodologies [Braun 1989a] were presented for determining the optimal control settings for chilled water systems that do not have significant thermal storage. A component-based nonlinear optimization algorithm was developed to optimize the system over continuous control variables. This constrained nonlinear optimization procedure was then used as a simulation tool for investigating the optimal system performance by optimizing over the feasible combinations of discrete controls. In this nonlinear optimization process, the operating cost and the output of each component in a chilled water system were approximated using a quadratic and linear form respectively. Nonlinear output, linear and nonlinear equality constraints, and inequality constraints were handled using LP or NLP techniques. Results of this algorithm led to the development of a simpler system-based methodology for near-optimal control that is simple enough for on-line implementation. The system approach involves correlating the overall system power consumption with a single function that allows for rapid determination of optimal control variables and requires measurement of only total power over a range of conditions. The estimating coefficients of this empirical system model involved regression on the results of the component-based optimization algorithm as a simulation tool. The system cost function led to a set of linear control laws for the continuous control variables. Separate control laws are required for each feasible combination of discrete controls. The number of controls in the system approach is greatly reduced compared to the component approach. With these models, general guidelines for near-optimal performance are developed. Braun's model optimized a snapshot of the plant, and thermal mass played no role in the analysis. Therefore, the results are time-invariant and can be applied to any time spot.

In another closely related work [Braun 1989b], the component-based optimization methodology developed in [Braun 1989a] was utilized as a tool for investigating chilled water systems under optimal control. With this tool, general guidelines for near-optimal performance are developed. These guidelines were incorporated in the system-based near-optimal control methodology, but they are also important to plant engineers for improved control practices. The important uncontrolled variables that affect system performance and optimal control settings are identified. Results and conclusions concerning both control and design under optimal control of chilled water systems are presented.

Braun [1990] studied the dynamic building control, dynamic adjustment of the indoor temperature set points in order to minimize overall operating costs by applying dynamic optimization techniques to

computer simulations of buildings and equipment. He pointed out that the optimization became complicated by the discontinuities associated with the different modes of operations. These modes include mechanical cooling with minimum outside air, mechanical cooling with 100% outside air and free cooling. The approach taken discretized the cost function and applied a non-smooth optimization algorithm to determine the set of controls that minimize the sum of costs over the specified time.

Determining dynamic optimal cooling control strategies that utilize building thermal mass is formulated as an optimization problem [Keeney and Braun 1996] with zone setting points as controls and a combination of energy cost and penalized human comfort as the cost. The complex method, an extension of the simplex method to constrained optimization problems, is used to solve this optimization problem over a 24-hour horizon. Based on detailed optimization, two simplified approaches are proposed for on-line implementations, where one approach takes two constant zone sensible precooling rates and the other applies a constant cooling rate for a given amount of time prior to building occupancy. With much less control variables, these two approaches successfully reduced the computation requirements for developing optimal strategies. Through the component-model-based simulation, these two approaches were tested with about 1000 different combinations of building, plant and weather. They achieved 95% and 97%, respectively, of the optimal savings relative to conventional control. These simplifications therefore could be used in the development of an on-line controller. In this work, zone set points are the only control, which makes the analysis and optimization easier. In general, developing cooling control strategies which utilize the thermal mass of a building is a formidable optimization problem, especially when on-line implementation is a consideration.

c) Dynamic Programming

Dynamic programming is used [Henze 1997] in determining the optimal control strategies for thermal energy storage systems and a predictive optimal controller for thermal storage systems is developed and simulated. An optimal storage charging and discharging strategy is planned at every time step over a fixed look-ahead time window utilizing newly available information. The certainty equivalence principle is used, which fixes the weather and internal gains at their expected values, to make it easier to solve the DP problem. Closed-loop optimization is employed, which means only the first control is executed at each time step although at each time step the optimization is conducted over the entire planning horizon using appropriate algorithms. The predictive optimal controller is compared to three conventional control heuristics: chiller-priority control, constant-proportion control and storage-priority control. The optimal controller was found to have a significant performance benefit over the conventional controls in the presence of complex rate structures. Compared to the load control problem, the thermal storage control

problem is not very complex in the sense that the system equations and the cost-to-go functions have explicit formulas and fewer controls.

Rossi and Braun [1996] used dynamic programming to obtain optimal service schedules and costs for cleaning the condensers and evaporators of air-conditioning equipment. Cost is defined as a combination of operating cost, human comfort, safety, and environmental criteria. The overall optimization problem is formulated in nested loops using key operating parameters. The innermost loop solves for optimal set of time stages between service tasks using DP. The next loop solves for total number of services in a service cycle using the golden section method by Rao. The outermost loop exhaustively searches for the duration of the service cycle and the time stage of the first service task. In addition, minimum operating costs are compared with regular service intervals (representative of current practice) and a strategy where service is only performed when a constraint is violated (e.g., a comfort reduction). It is found that optimal service scheduling reduced lifetime operating costs by as much as a factor of two over regular service intervals, and by 50% when compared to constrained only service. For practical implementation, a simple near-optimal algorithm for estimating optimal service scheduling is developed that does not require on-line forecasting or numerical optimization and is easily implemented within a microcontroller. Over the wide range of cases tested, the near-optimal algorithm gives operating costs that were within 1% of optimal.

A multi-criteria model is described [D'Cruz and Radford 1987] for assisting designers in the choice of form and construction of parallel-piped open plan office buildings at the schematic design stage of building design. The model considers four performance criteria: thermal load, daylight availability, planning efficiency, and capital cost. Pareto-optimal dynamic programming optimization is employed. The model's form and implementation and some typical results are described.

It is worth mentioning that dynamic programming is widely used in the field of operations management (OM). An optimal inventory purchasing policy is determined [Tsitsiklis 1998] with the DP cost-to-go approximated in neuro-dynamic programming (NDP), a method that uses neural nets to approximate the cost-to-go based on the features properly extracted in advance. NDP type of methodologies could possibly be applied in the load control problem. However, the systems in operations management scenarios are often less complex than most mechanical systems, so problem setup and computation would be more difficult in the load control scenario.

d) Linear-Quadratic Optimal Control

A problem with linear systems and quadratic cost is defined as a linear-quadratic (LQ) problem. The optimal controls can be obtained analytically, which is well known as the Riccati equation [Bertsekas 2000]. Linear and quadratic approximations are valid in many cases in building load research, and LQ is expected to be fairly useful. However, LQ has not been widely used, possibly due to the complexity in the real systems. Hopefully, research that focuses on solving LQ problems using nontraditional and more flexible methods such as neural nets [Lan 1990] and genetic algorithm [Michalewicz 1992] would help improve the situation.

e) Genetic Algorithm

[Wright and Farmani 2001] provided a brief introduction to the major optimization algorithms used in the "whole building optimization" problem. It first described several notable characteristics of the issue: problem variables are a mixture of integer and continuous variables; the problem has non-linear inequality and equality constraints; and the objective functions can be discontinuous. The authors reviewed previous work and suggested that neither traditional gradient-based methods nor direct search methods are effective for the whole building optimization problem. A genetic algorithm was recommended and used.

A PC-based supervisory controller is developed [Gibson 1997] for a building's energy management and control system to optimize cooling equipment operation. The system provides decision support to determine when to operate central cooling equipment to minimize costs under real-time pricing or conventional time-of-use electric rates. An artificial neural network is used to model the dynamic behaviors of the building and energy equipment while an evolutionary-based search routine, a genetic algorithm is used for optimization. In the GA-based operation schedule planning, Gibson used the bits of the chromosome to represent the status of the cooling equipment in 24 hours. Each chromosome is an operating schedule individual in a "population" of many possible operating schedules. The GA searches for optimal schedule by employing certain techniques of reproduction, crossover and mutation. The ANN-based modeling uses the current external stimuli (outdoor temperature, equipment status, etc.) in conjunction with the previous state of the building to predict the current state of the building. It provides a basic profile of building cooling needs against which each of the individual plans can be evaluated. The GA initializes and updates the control population. The ANN predicts for each individual its corresponding load and cost performances and evaluates the fitness which will be sent back to GA. The GA and ANN together form the planning module in this supervisory controller software. A prototype system is installed and operating at a high school in southern California to control a thermal energy

storage system: a conventional screw-type chiller, and a gas-fired, engine-driven chiller. Some lessons were learned during the controller development, and insights were gained in the practical application of both GA and ANN. Examples are how to balance the global optimum and the curse of computation in GA and how to maintain the accuracy of the ANN by applying a neural network monitor, which addresses the relationship between ANN accuracy and the optimization process itself.

Chow [2001] derivd an ANN model of a direct-fired double-effect absorption chiller system. In the paper is discussed the concept of integrating neural network and genetic algorithm in the system optimal control in achieving the final goal of minimizing the operation cost. It is pointed out that to obtain a well matched but reasonably simple ANN configuration of the system model, a systematic search on the family of architecture is mandatory. Testing should be well monitored and cross-validated. Adequate representation of test data is a prerequisite for a successful outcome.

Wright and Farmani [2001] simultaneously optimized the building's fabric constructions, the size of heating, ventilating and air-conditioning systems, and the HVAC system supervisory control strategy in order to account automatically for the thermal coupling between these building elements. The problem formulation is described in terms of the optimization problem variables, the design constraints, and the design objective functions. The optimization problem is solved using a GA search method. The conclusion is that the GA is able to find a feasible solution, and it exhibits an exponential convergence on a solution. The solutions obtained are near-optimal, the lack of final convergence being related to variables having a secondary effect on the energy cost objective function. Further research is required to investigate methods for improving the handling of equality constraints and to reduce the number of control variables (which will also improve the robustness of the algorithm).

Wright and Loosemore [2001] investigated the application of a multi-objective genetic algorithm (MOGA) in the search for a non-dominated (Pareto) set of solutions to the building design problems. Compared to the progressive approach that generates the trade-off curve by assigning different weights and repeating the optimization, MOGA employs the Pareto-ranking scheme to form the fitness of each solution and complete pay-off characteristic in one optimization of the building design. Constraint functions are aggregated by a normalized sum of their violations to form a single design criterion. The results indicate that the MOGA is able to identify the trade-off characteristic between daily energy cost and zone thermal comfort, and that between capital cost and energy cost. The MOGA exhibits fast progress towards the Pareto optimal solutions.

f) Control Heuristics in Building Industry

Engineering heuristics have been a major component in the current building control practice. Developed based on local optimization, system simplification, estimation and experiences, heuristics perform fairly well in many scenarios. The ASHRAE Applications Handbook [1999 chapter 40] describes in detail the control heuristics that have been widely used in operating HVAC systems and components, including cooling tower fan control, chiller water reset with fixed- and variable-speed pumping, sequencing and loading of multiple chillers, strategies for air-handling units, strategies for building zone temperature set points and control of thermal energy storage systems. Control heuristics could be used as a starting point in an optimization scheme. In addition, a heuristic type of suboptimal control is often desired as an extension from optimal control for on-line implementation purposes.

1.3.3 More load control strategies

Night cooling has shown in practice its cost-effectiveness for buildings with reasonably heavy thermal mass. In summer time, building thermal mass is cooled at night through natural or mechanical ventilation, stores cool energy, and discharges it the next day. Previous simulations and experiments have shown that significant operating savings and peak load reduction can be realized through night cooling.

Braun [2001] used an inverse model to represent the behavior of the building, cooling plant, and air distribution system and evaluated several building thermal mass-based control strategies, including a variety of pre-cooling schedules at night and discharge processes next day, at five locations with different weather and rate structures. The best strategy turned out to be the maximum discharge, which results in an average a 40% reduction in total cooling costs as compared to the conventional night set-up control.

Braun [2001] developed a tool that allows evaluation of thermal mass control strategies using HVAC utility costs as the baseline for comparison. Based on weather and solar inputs, as well as occupancy and internal gains schedules and utility rates, the evaluation tool predicts the total HVAC utility cost for a specified control strategy. Intelligent thermal mass control strategies can then be identified in a simulation environment using this analysis tool. The model was validated using the field-site data and applied in five cities across the country. The effects on the cooling load and energy cost of different control strategies, locations, utility rates and climates are evaluated using this simulation tool.

Rabl and Norford [1991] studied the load control strategy of subcooling a building a few degrees below its normal thermostat set-point during the preceding night, and controlling the warm-up and stored energy release through the thermostats to maximize the benefit. Several thermostat control strategies,

distinguished by their knowledge of the building dynamics, are described and simulated using a data-based dynamic model.

Keeney and Braun [1996] proposed a simplified version of the thermal-mass-based optimal control problem, by examining optimal cooling results covering a wide range of buildings, cooling plants, weather, utility rates, and internal gains. Two simplified approaches are based on two-variable approximations to the precooling portion of the day, and a set of comfort-based rules when the building is occupied. Those two variables are two constant zone sensible cooling rates in one approach and a constant cooling rate and a precooling time period in the other. The simplified strategy achieved 95% and 97% of the optimal savings relative to the conventional control.

Haves and Gu [2001] simulated several load-shedding scenarios in EnergyPlus and came up with guidelines for the building operators in commercial office buildings to reduce peak electricity demand by limiting buildings' HVAC operation. They describe the demand reduction and the increase in discomfort over time that can be expected from increasing temperature set points. They also give examples that demand reduction can be achieved by reducing fan capacity and increasing supply air temperature and chilled water temperature. The guidance is aimed at large buildings with built-up HVAC systems and chilled water plants, and applies to California and similar climates where humidity control is not a significant problem.

1.3.4 Scheduling issues

Two questions need to be answered to conduct load shedding: first, when load shedding should happen and for how long, which is more of a scheduling problem; second, what value the load shedding parameter should take.

[Jorge et al. 2000] presented a multi-objective decision support model which allows the consideration of the main concerns that have an important role in load management (LM): minimize peak demand as perceived by the distribution network dispatch center; maximize utility profit corresponding to the energy services delivered by the controlled loads; and maximize quality of service in the context of LM. [Hsu and Su 1991] used dynamic programming in developing an optimization technique to reach the optimal direct load control (DLC) dispatch strategy and system generation schedule. Similarly, [Chen et.al. 1995] used DP to study the optimal direct load control pattern which is defined by interruption starting time, maximum-interruption-time, minimum-connection-time, and payback energy and payback ratio.

[Kurucz et al. 1996] developed a linear programming (LP) model to optimize the amount of system peak load reduction through scheduling of control periods in commercial/industrial and residential load control programs at Florida Power and Light Company. The LP model can be used to determine both long and short-term control scheduling strategies and for planning the number of customers who should be enrolled in each program. Results of applying the model to a forecasted late 1990s summer peak day load shape are presented. It is concluded that LP solutions provide a relatively inexpensive and powerful approach to planning and scheduling load control. Also, it is not necessary to model completely general scheduling of control periods in order to obtain near best solutions to peak load reduction. Another related paper [Chen et al. 1995] that also focused on the scheduling part of load control used DP to study the optimal direct load control pattern which is defined by interruption starting time, maximum-interruption-time, minimum-connection-time, and payback energy and payback ratio.

[Effler et al. 1992] dealt with the procedures for energy import optimization based on the results of a load forecast program for the control center of the Pfalzwerke AG. Special emphasis is put on load management and the investigations for modeling the load behavior depending on season, day type and temperature.

1.4 Problem description

Our load control problem is a multi-objective optimal control one involving complex system (building and plant) dynamics and engineering constraints. It has two stages as follows.

First, optimize controls in a single building to minimize the cost, where the cost is the combination of electricity cost and penalized human comfort. Identify important load shedding strategies through parametric studies and propose guidance for building operators to shed the building load appropriately.

> The emphasis of our work is on short-term load control strategies that a building operator can easily implement. The intent in general is to curtail service in order to control costs. While building operators are familiar with possible set point adjustments for their plants and may be making them, it is important for them to have a basis for choosing among them and selecting the magnitude of adjustment. For example, adjusting supply-air temperature on a VAV plant may increase energy costs until service is curtailed. Turning off or restricting the output of a chiller may reduce service and costs for a short period, but may lead to a load (and cost) spike if service is restored while peak charges are still in effect.

Some issues we explore are: What characteristics does the building optimization cost function have? What are the important load shedding strategies for a specific building? How to connect a building simulator with an optimizer? What algorithms are efficient in solving building optimization given the specific cost function and constraints? We answer these questions through a large number of EnergyPlus simulations and simulation-based optimization.

There has been extensive work in the single building area. This research distinguishes itself in the following ways: 1) it applies a variety of optimization methods to EnergyPlus. The analysis framework is simple and potentially useful for practitioners; 2) as a new and powerful simulation tool, EnergyPlus offers great flexibility in studying building operations and controls. Some of the things such as scheduling cannot be fully accomplished by other tools. We look into a variety of issues by taking advantage of new functionalities of the software.

Second, optimize the load aggregation with multiple buildings by applying suitable controls to aggregation participants and matching their load profiles. Set up the structure of the overall optimization process. Build the underlying math model for the central optimizer and a statistical unit simulator for single building dynamics. Solve the optimization problem efficiently.

An optimization scheme is to be developed for building load aggregation. A brute-force algorithm supported by EnergyPlus simulations is less desirable for the multi-building case due to the size of the problem. Our experience with EnergyPlus in the single-building research suggests we stay away from full-scale simulation in the multi-building case for the following reasons: 1) it is too computationally intensive to manipulate a number of EnergyPlus models; 2) too much complexity from individual buildings would bury the system level dynamics and prevent us from gaining insight; 3) data from full-scale simulation tend to be nonlinear and discontinuous, which makes it difficult to solve the problem using nonlinear optimization algorithms . The idea is to separate simulation and optimization, as we will discuss in Chapter 5, where individual simulator serves as the building block of the optimization process. The optimization process looks at a group of such profiles and decides what to do at the macro level.

A generic math formulation of the problem is given in Eqn1.1, which minimizes the total electricity cost:

$$\min \quad \sum_{t=1}^{24}\left(\sum_{i=1}^{N} W_{i,t}\left(u_{i,t}, x_{i,t}\right)\right)\times R_t + \max_t\left(\sum_{i=1}^{N} W_{i,t}\left(u_{i,t}, x_{i,t}\right)\right)\times D$$

$$\begin{aligned} s.t. \quad & PPD_{i,t}\left(x_{i,t}\right) \le PPD^*, \ \forall i, t \in [t1, t2] \\ & u_- \le u_{c,i,t} \le u_+, \ c \in C, \forall t, \forall i \\ & u_{d,j,t} = 0 \, or \, 1, \ d \in D, \forall t, \forall i \\ & W_{i,t} = f_i\left(x_{i,t}, u_{i,t}\right), \forall i \\ & PPD_{i,t} = g_i\left(x_{i,t}, u_{i,t}\right), \forall i \end{aligned} \qquad \text{Eqn.1.1}$$

Where,

$W_{i,t}$	Electricity consumption by building i at time t, $i = 1,...,N$, $t = 1,...,24$
$PPD_{i,t}$	Predicted Percentage Dissatisfied values in building i at time t
R_t	Electricity rate at time t
D	Demand charge rate
$u_{i,t}$	Control variables, $u_{c,i,t}$ continuous variables, and $u_{d,i,t}$ discrete variables
$x_{i,t}$	State variables
f_i	Building i dynamics determining electricity consumption at hour t
g_i	Building i dynamics determining thermal comfort i is at hour t

We also need to look at the non-technical side of the load control issue because the energy market and energy policy play important roles in practice. With the utility market moving toward deregulation and information technology more advanced, opportunities and challenges emerge in designing an efficient and healthy deregulation system, communicating with the market and making informed decisions. It is also interesting to look at the increasingly complex energy-based financial instruments and how they might affect the market.

Overall, Chapter 2 is devoted to parametric studies that concern load shedding and control in a single building. Comparison between different strategies helps understand VAV system dynamics. Important operations parameters are identified and chosen to participate in an optimization process in Chapter 3 to optimize the load control strategies. Chapter 4 explores load control when multiple buildings are involved through two simplified approaches: educated enumeration and sequential-GA. Chapter 5 solves the multi-building optimization problem. A time-series linear model is used to simplify the building dynamics, on which a nonlinear algorithm is applied to capture the aggregation structure.

CHAPTER TWO
SINGLE BUILDING PROBLEM: LOAD CONTROL IN A SIMPLE VAV SYSTEM

In this chapter, we study load control opportunities in a single building through EnergyPlus-based simulation. A simple VAV system is used as the base model, and a variety of load shedding and control strategies are explored. The purposes of conducting single building research are as follows: 1) gain a thorough understanding of building dynamics and a mechanical system's partial load performance which is key to some load shedding strategies; 2) optimize load control in a single building's framework; and 3) prepare at the individual level for the multi-building problem. We first introduce in detail the building model used through the entire research, and test the model. Several load control strategies including night cooling are examined thorough parametric studies to identify important ones to this building. As a follow up, two optimization schemes are applied to load controls in the next Chapter.

2.1 Model description

- Building

The base model used in this research is a simple three-zone VAV system[1], which is located on a top floor of a commercial building with a gross area 126m^2. The system has no ground contact (all floors are "partitions"), and three zones are connected by inter-zone partitions. The roof is exposed to the outdoor environment. All of the zones are air-conditioned with the same temperature set point, and the middle zone has a south-facing window that is 10m^2. Figure 2.1 shows the floor plan. The external walls are common brick with a R-value of roughly 0.14. The floor is eight-inch concrete with a R-value of 0.12. The roof is two-inch concrete and its insulation layer has a R-value of 0.58. The estimated time constant is 44 hours for the walls and floor and 3 hours for the roof. Using the ASHRAE light, medium and heavy standards for room envelope construction [ASHRAE handbook 1997], this building is about the medium construction type. The window area is rather small in this building and the solar effect does not dominate. The internal loads including people, equipment, and lighting are about 13kW, roughly 0.1 kW/ m^2, and run at a typical-commercial-building schedule, meaning large load and operation differences exist between day time when the building is occupied 8am – 5pm and nighttime and weekends when the

[1] This model is based on one named vavsingleductreheat.idf in the free-downloaded EnergyPlus software package. We changed the model in a variety of ways including equipment sizing and chiller sequencing, control schedules, added an outside air system and disabled the reheat system. The EnergyPlus input file of the base model can be found in AppendixA.1. More changes will be made later to make available a variety of building systems for load aggregation research.

building is empty, and between weekdays and weekends. The load pattern schedule will be changed later in this research to come up with a model with less commercial load pattern.

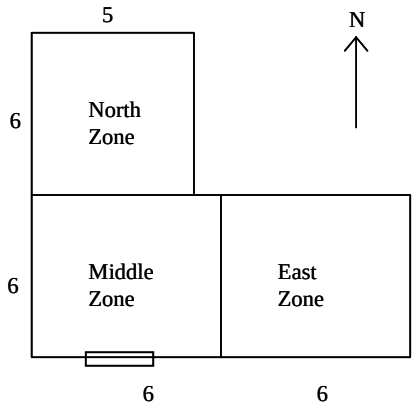


Figure 2.1 Floor plan of the three-zone VAV system

- Air-handling system

The building is served by a VAV air-handling system with a design supply airflow of 1.3 m^3/s. A dry-bulb temperature-controlled economizer is used for air-to-air heat recovery and is set at the return air temperature when night cooling is available or turned off other times. The air handling system operates normally from 8am to 5pm weekdays only.

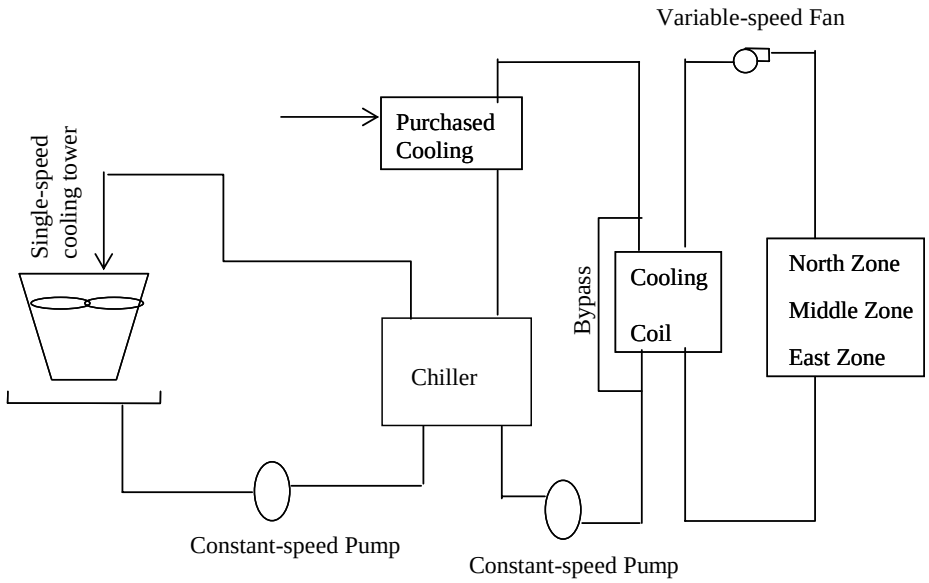


Figure 2.2 Plant layout of the three-zone VAV system

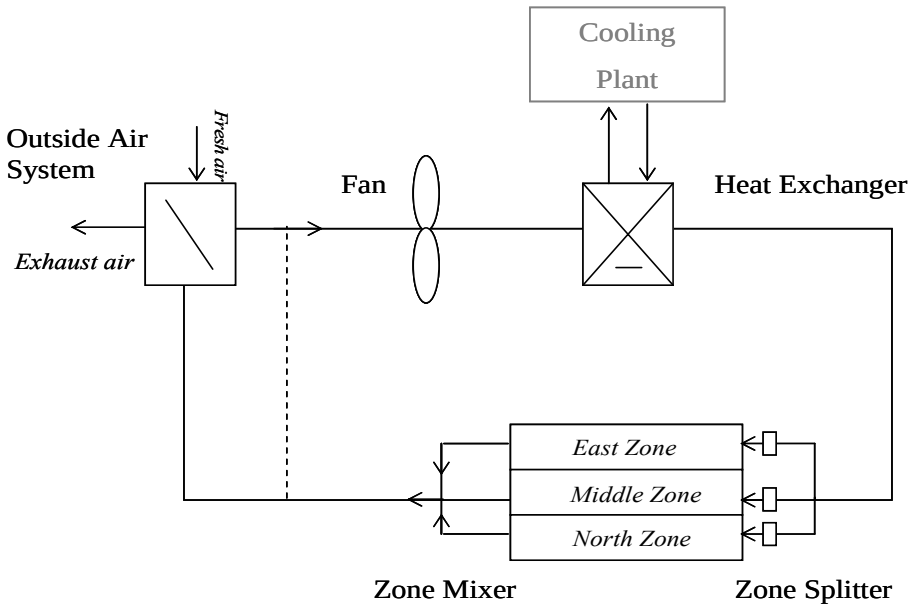


Figure 2.3 Air loop of the three-zone VAV system

The mechanical system is shown in Figure 2.2. It is a variable air volume system with a single air loop. The occupied hours are 8am – 5pm and unoccupied hours otherwise. The original system has purchased heating supply and runs all year. We focus only on the summer. To be energy efficient, the system has no reheat supply. The cooling set point is 24°C while occupied and 30°C otherwise. Fans and coils are scheduled to be unavailable during unoccupied hours. The air loop consists of a fan and cooling coil. The fresh air is supplied by an outside air system[2], as shown in Figure2.3. In the plant, the cooling loop is served by a 35kW electrical chiller and purchased cooling.[3] A variable-speed fan, constant-speed chilled water and condensed water pumps, and single-speed cooling tower fans are used in the system. From April 1 through September 30, the system runs in the summer operations condition, and winter condition with the remainder of a year.

Operation schedules are controllable and supervisory controls are adjustable. Set point managers are used to schedule supply air temperatures and mix OA (outside air) ratio. Fan and cooling coils are controlled

[2] The original model has no outside air system. We added one for purposes of studying a real building and implementing night cooling. See Appendix A.2 for the E+ code.
[3] In the original model, the cooling loop is served by a big chiller, a little chiller and purchased cooling, and priority-based controls determine which piece of equipment to be turned on to meet the demand. It is changed for simplicity purposes. The new capacity is chosen by design.

separately, so that we can implement fan-based night cooling or chiller-based night cooling. The reheat system is controlled in a way that it is always off at night time no matter which load control strategy is taken. The minimal outside air is controlled through min OA schedules under which system shuts down outside air at night for conventional night set-up but intakes as much fresh air as possible with night cooling available. All the schedules can be adjusted on an hourly basis.

Our research focuses on summer conditions. We simulate several typical summer days using San Francisco and Los Angeles weather data, and compare the average performance between two locations. Table 2.1 has more details about this three-zone VAV system we use.

Table 2.1 Basic building model

Building and Zones	Three zones, top floor commercial building, gross area 126 m^2 East and middle zones: 3 people and 3kW equipment North zone: 4 people and 4kW equipment
Construction and insulation materials	C4 - 4 in common brick ext walls - R value 0.14 m^2K/W, 1922 kg/m3, 0.84kJ/kgK C10 - 8in concrete floor slab – R value 0.12 m^2 K/W , 2242 kg/m3, 0.84kJ/kgK C12 - 2in concrete roof – R value 0.03m^2 K/W, 2242 kg/m3, 0.84kJ/kgK C6 - 8in clay tile partitions – R value 0.36 m^2K/W, 1121kg/m3, 0.84kJ/kgK B5 -1in dense insulation – R value 0.58 m^2K/W Other insulations include membrane, stucco, gyp board etc.
Plant and HVAC	Chiller: electric chiller, nominal capacity of 35kW, COP 3.0, minimal load ratio 0.15, optimal load ratio 0.65 extra cooling resource: purchased cooling Fan: variable-speed drive, 600Pa and 1.3 m^3/s design flow Pump: chilled water and condensed water pumps, constant-speed drive 300kPa and 0.0011 m^3/s rated flow Cooling tower: single speed, 1kW fan

2.2 Model test

As previously noted, the three-zone model described in the previous section originates from an EnergyPlus example model that comes with the software package. A few changes concerning plant equipment, air loop and schedules were made to make the model the right one for this research. Both the original model and our modified one, the base model, run smoothly and produce reasonable results. Because the base model and a few variations will be used extensively in our research, we consider it necessary to validate the model. We cannot and are not going to conduct the model validation in the traditional sense by using experimental data. Instead we will check the model to make sure it makes sense in physics. Details are as follows:

- Energy balance and mass balance are maintained for components and the system
    - Chilled water and condensed water loops separately
    - Chiller as a whole
    - Air loop heat and mass involving outside air system and cooling coil
    - Convective heat transfer vs. cooling provided on a real-time basis
    - Cooling load itemization
    - Building as a whole and load calculation vs. simulation results
    - Cooling demand and supply match on a daily basis
    - Energy flow in the entire system
- Component physics maintained
    - All physical parameters (temperatures, flow rates, thermal comfort et al.) take reasonable value considering equipment constraints
    - Components behave the way intended and part-load performance looks correct
    - Check components for the trend in power consumption with at different capacities
    - Transition between components is smooth and the system is reasonable (partially checked in balance)
- Input-output relation straight
    - Extreme condition tests
    - For those theoretically or empirically known relations, check if the simulation results meet the expected trends
    - Check model performance with different weather and internal loads

After going through all these checking and debugging procedures, the likelihood that the model has physical modeling errors is fairly small, although not perfectly guaranteed. In fact, some EnergyPlus

bugs and hidden modeling errors are identified later. What we have done above is to guarantee a relative correctness, meaning that the model by itself is correct in physics, but whether or not the model produces the similar results to those from measurements in the real building is less critical, because 1) we are conducting this research on a relative platform and the most important thing is the model is consistent in itself, so we can compare different load control strategies; 2) experiment-based validation is less viable. Real buildings are generally too complex for research purposes, but it is difficult to build an experimental chamber with its own complete plant system.

2.3 EnergyPlus Parametric Studies Part I: Basic VAV system dynamics

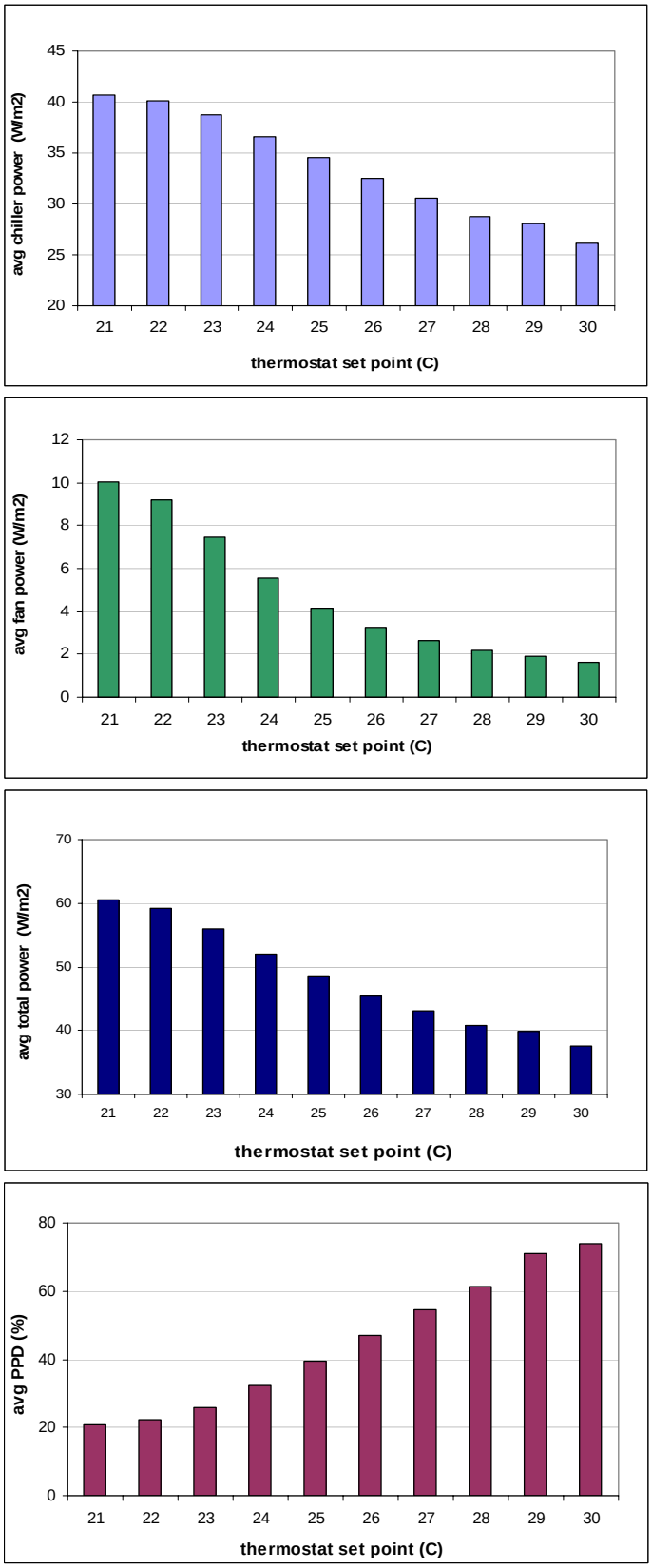
EnergyPlus parametric studies are carried out to see how the building system responds to changes of basic operation parameters such as thermostat setting points, supply air and chilled water temperatures. In this section, we simply run simulations with one parameter varying at a time.

2.3.1 Thermostat set points

The original thermostats are set at 24°C for occupied hours and 30°C for unoccupied hours. Here we vary the occupied-time thermostat from 21 to 30°C with 1°C time step on a typical summer day in Los Angeles to see how the thermostat set point affects the energy performance. Figures 2.4a, b, c and d show the daytime average fan power, chiller power, total power, and PPD values. There is no need to consider the unoccupied hours when system is off. It will be a totally different case if night cooling is available, which will be addressed late in this chapter.

Increasing the thermostat decreases monotonically all the power consumptions and reduces the service level. It has a large impact on both fan power and chiller power, which is due to the reduction in cooling demand. The marginal energy benefit shrinks with the further increase of the thermostat set point while thermal comfort keeps deteriorating. Therefore, increasing thermostats can only be a load control strategy within a certain level, which is a function of the building system and service level requirement.

The simple analysis here adopts new thermostat set points for the entire day, which is seldom the case in practice. In further parametric studies and optimization analysis late in this research, we will focus on thermostat set point change with a short-term, e.g. several-hour, horizon, where scheduling load shedding and reducing the setback peak will make the analysis more interesting.



Figures 2.4a, b, c and d Daytime average chiller, fan, total power, and PPD vary with thermostat set points in LA on a typical summer day of August 8

2.3.2 Supply air temperature, chilled water temperature and fan capacity reduction

The rationale behind varying the combination of supply air temperature, chilled water temperature and fan static pressure rise is as follows. In a VAV system, increasing the supply air temperature pushes up the fan power if the indoor air temperature remains the same. If the chilled water temperature can be increased accordingly, it will lead to better chiller efficiency and produce some savings. Two scenarios can happen.

- If we can shed the fan capacity, the fan power increase can be mitigated. Together with the savings from more favorable chiller working conditions under higher average chilled water temperatures, this strategy is likely to produce pure savings.
- If we cannot shed the fan capacity, the energy saving potential depends on the relative change between chiller power decrease and fan power increase. In most cases, it has to wait until the fan gets saturated and the fan power remains constant after.

The original supply air temperatures are set at 13°C for the entire day 24 hours. We run the simulation with the supply air temperature (Ts) varying from 10°C to 26°C with a 2°C step, and chilled water temperature (Tch) taking values of 6.67, 8, 10 and 12 on a typical summer day in LA. The changes apply to all 10 working hours.

Figure 2.5 shows how combinations of Tch and Ts affect the average chiller and fan power and PPD. Indoor air temperatures are kept at 24°C until the supply fan is saturated due to the increasing supply air temperature. We see a clear trend of chiller power decrease and fan power increase with the increase of Ts and the impact of varying Tch is not as important. Combining the total power and PPD plots, we see that power savings have to come with the price of severe comfort violation. The increase in fan power outweighs the savings in chiller power. This continues to be so until the fan gets saturated, around 16°C in this case, where we start seeing total power savings but the indoor temperatures are higher and PPD values have reached 60%. This rules out the possibility of relying on increasing Ts and Tch alone as a load control strategy.

To further improve the energy performance, we allow the fan capacity to change, which can be implemented by reducing the static pressure set point of the fan or the maximum speed. Figure 2.6 compares the full fan capacity with a 25% capacity reduction: Figure 2.6a shows that the fan power reduction helps reduce the average chiller power, which might be due to the fact that less fan heat is released to the air stream and therefore processed by the chiller. Figure 2.6b indicates a large fan power

drop by a 25% fan capacity reduction. Later in this research, we will quantify the impact of the short-term fan capacity reduction based on a few representative days in summer. Figure 2.7 compares the chiller and fan power consumptions with a full fan capacity and with a 75% fan capacity at different supply air temperature and chilled water temperature combinations. The fan capacity reduction leads to savings in fan power in all four cases, but the chiller power savings are more obvious when both supply air and chilled water temperatures are higher.

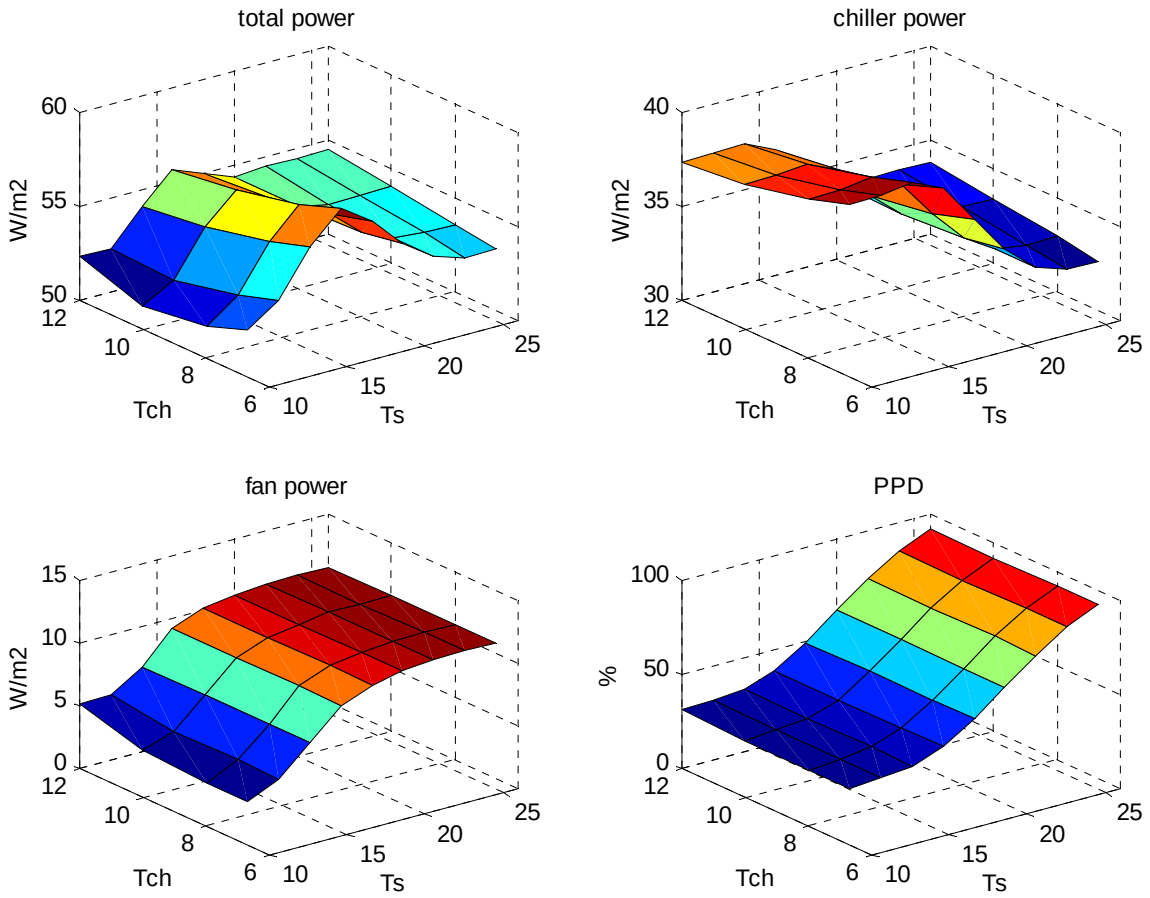


Figure 2.5 Total, chiller and fan power, and PPD vary with the combination of Ts (supply air temperature) and Tch (chilled water temperature) with fan capacity stays full and thermostats remain at 24°C before the comfort is degraded

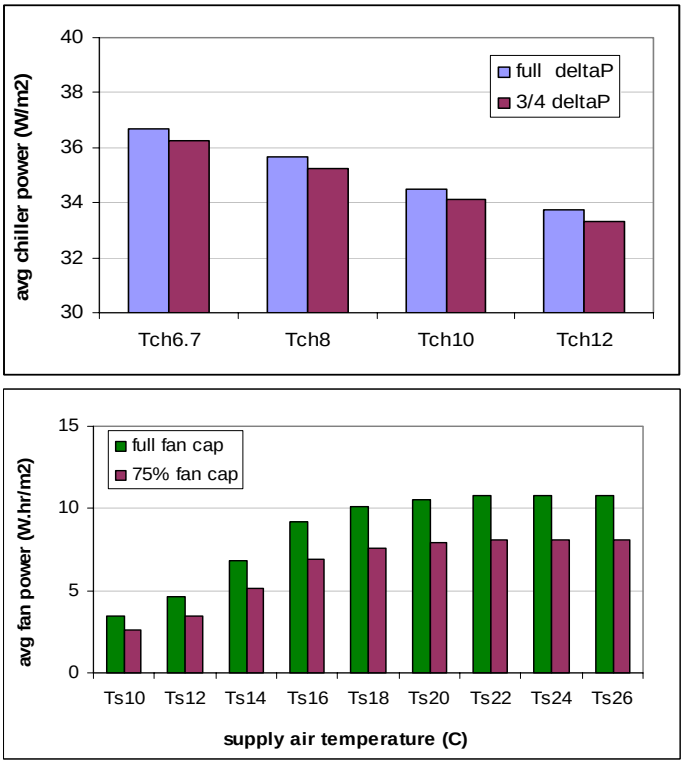


Figure 2.6 a) Chiller power varies with Tch at full and 75% capacity, Ts at 14°C (top) b) fan power varies with Ts at full and 75% capacity, Tch at 6.7°C (bottom). Parameters apply to all 10 hours

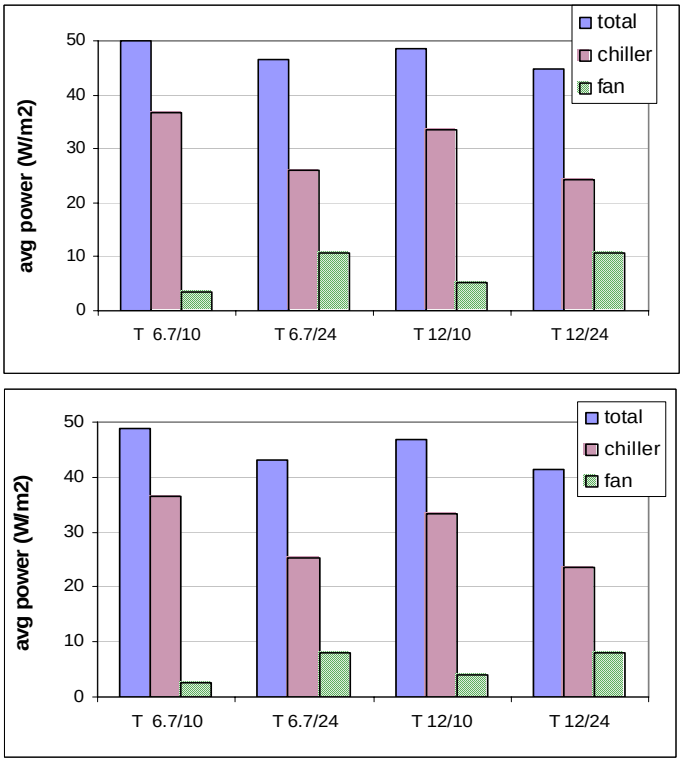


Figure 2.7 a) Total, chiller and fan power vary with combination of Ts and Tch at full fan capacity (top) and b) 75% fan capacity (bottom). Parameters apply to all 10 hours

2.3.3 Economizer set point

There is an economizer installed in our model, and we can control the fresh air intake by setting the economizer temperature. In previous two cases, the economizer was turned off so that we could focus on the impact of other parameters. In cases where natural ventilation is involved, the economizer will be set such that the system can take full advantage of free cooling. We now look at how the economizer temperature set point affects the energy performance. The base case is that the economizer is set at the return air temperature, 24°C, which is expected to be the most energy efficient as it maximizes the use of outdoor air. In other cases, the economizer temperature varies from 24 to 16°C. System remains off at night and all free cooling takes place during the day. Figure 2.8a) shows the impact of economizer on a rather cool day June 9 when the highest temperature is 20°C and the lowest is 15°C, the economizer performs the best with a set point of 24°C, which corresponds to the least chiller power. All the set points above 20°C perform equally well and take full advantage of fresh air. When the set point drops below 15°C, free cooling no longer comes in and mechanical cooling takes over. On a cool day like June 9, free cooling contributes more than 30% to the total cooling need. Figure 2.8b) shows the chiller power at different economizer set points on a warm day July 14 when the highest temperature is 23°C and the lowest is 18°C. In this case, operators should set the economizer no smaller than 23°C to capture the free cooling benefit, which is about 13%. In both cases, fan power remains unchanged, as the total airflow rate is the same. Therefore, varying economizer set point according to weather can be an easy way of saving energy for operators.

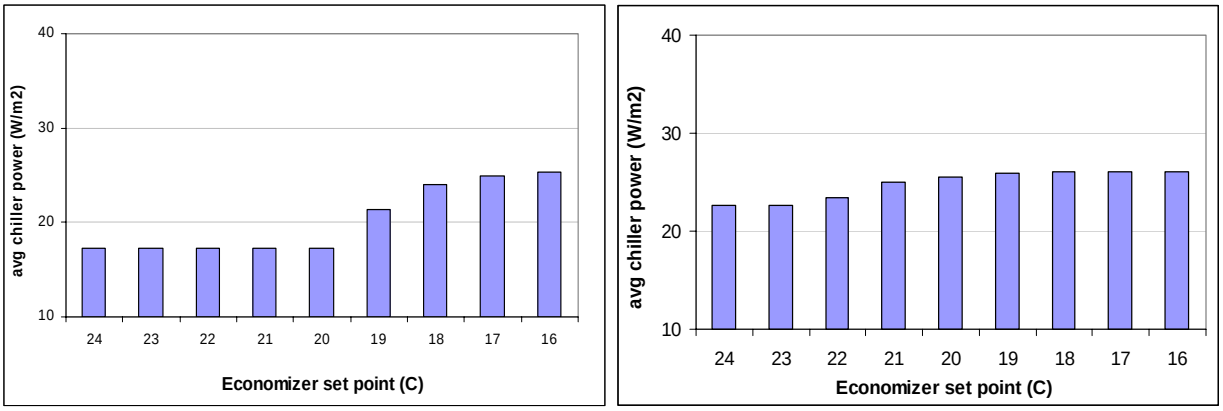


Figure 2.8 Chiller power vary with economizer set point on a) June 9(left) and b) July 14 (right) in LA

2.4 EnergyPlus Parametric Studies part II: short-term load shedding

We studied in part I the basic dynamics of a VAV system through parametric studies and identified the importance of certain operation parameters. More simulations are done in this section and the energy performance is quantified with the focus on the savings potential of short-term, e.g. several hours, load shedding. In reality, we can only afford sacrificing some comfort for a short period of time. In this section, we apply load shedding to hours 14-17, a four-hour period in the afternoon, and aim at reducing the peak demand and/or overall energy consumption. It is to be noticed that these two goals could yield different results, which depends on the specific load profiles and rate structures.

Because we choose the last four working hours as our load control period, any load-control-caused setback recovery wouldn't become the system load. However, the heat accumulated during the shedding period could be only partially dissipated by conduction through envelope at night, and what is left might lead to an early morning power pick-up. Because the convergence of the EnergyPlus simulation is on a single-day basis, this morning pick-up effect can be seen in some of our results. All the simulation in this research has a horizon of a day. During the parametric studies, we apply load shedding to six representative days in summer and present the average as the final savings value.

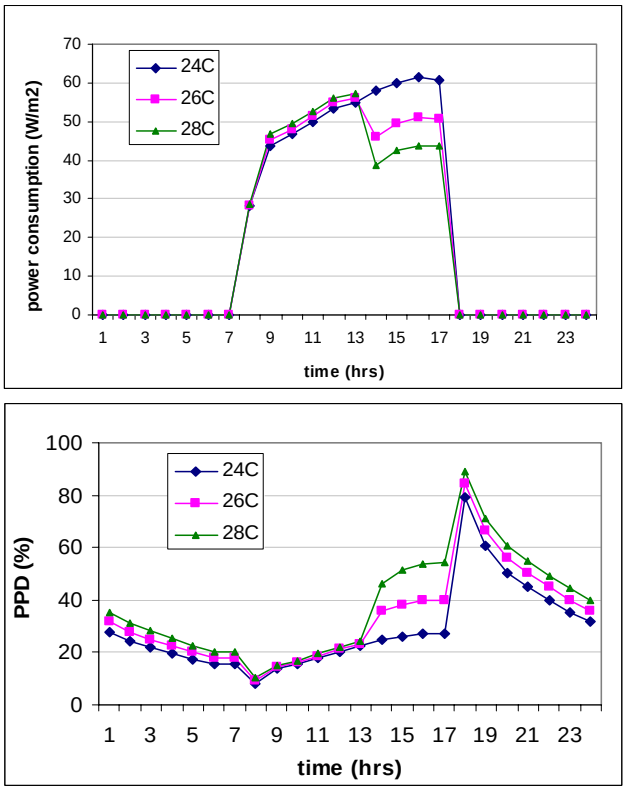
We look at three set-point related load control strategies in LA and SF: increasing thermostat set points, reducing supply fan capacity and increasing thermostat set points, and increasing both supply air temperatures and chilled water temperatures while reducing the fan capacity. All the adjustments are made during 14-17, last four working hours, except that the fan capacity reduction applies to a whole day. We also look at the effect of turning chiller off for an hour during the day. The impact of load shedding schedules is also explored.

The hourly power and PPD profiles in Figures 2.9 a) and b) show the potential demand reductions and decrease in service level. No load setback recovery takes place because the load control is applied in the last four work hours, which, however, corresponds to slightly higher power in the morning. Figures 2.10 a) and b) are the average daytime chiller and total power reductions for LA and SF. For thermostat-based load control, a 2°C thermostat increase in hours13-17 leads to an average power reduction of 2.8 W/m^2 in LA and 1.7 W/m^2 in SF, and a demand reduction[4] of 8 W/m^2 in LA and 4.8W/m^2 in SF; a 4°C increase corresponds to 4.4 W/m^2 average power savings in LA and 2.4 W/m^2 in SF, and a demand reduction of 11

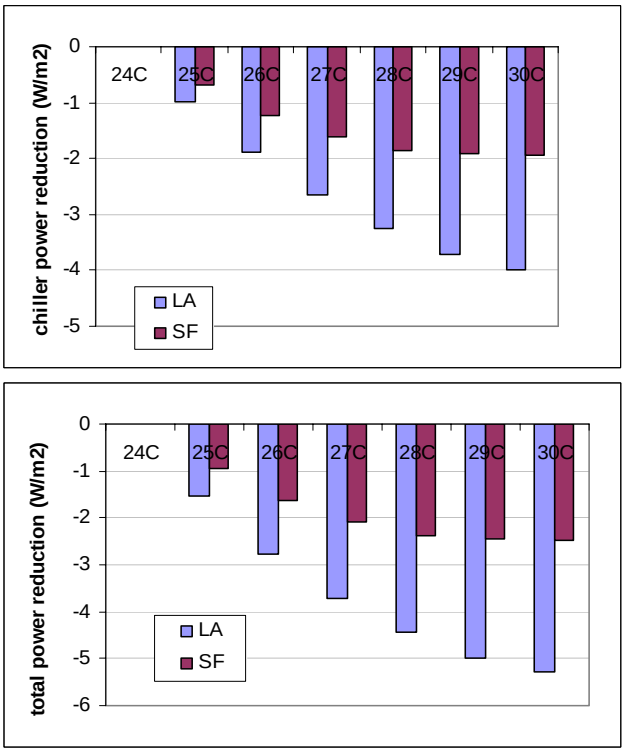
[4] Here we only look at the demand reduction within the control period of hours 13-17. Note that the new peak might be shifted to 12noon by applying load shedding to hours 13-17. Doing so, we produce demand reduction similar to Haves [2001] did, and two research projects are comparable.

W/m^2 in LA and 7 W/m^2 in SF. Haves [2001] reported a 5W/m^2 demand reduction for a 2 degree set-point change for a prototype office building in California, and 9.3 W/m^2 demand reduction for a 4 degree set-point change. According to Figures 2.9b, daily peak PPD reaches 50% with a 4°C increase and below 40% with a 2°C increase. Figure 2.10a) shows the total demand reductions in LA and SF and the contributions by chiller, which is the major source of the power reduction with thermostat set points increases. Thermostat-based load shedding works better in LA than SF because the base load in SF is rather small due to the mild weather and savings from load shedding are relatively less dramatic.

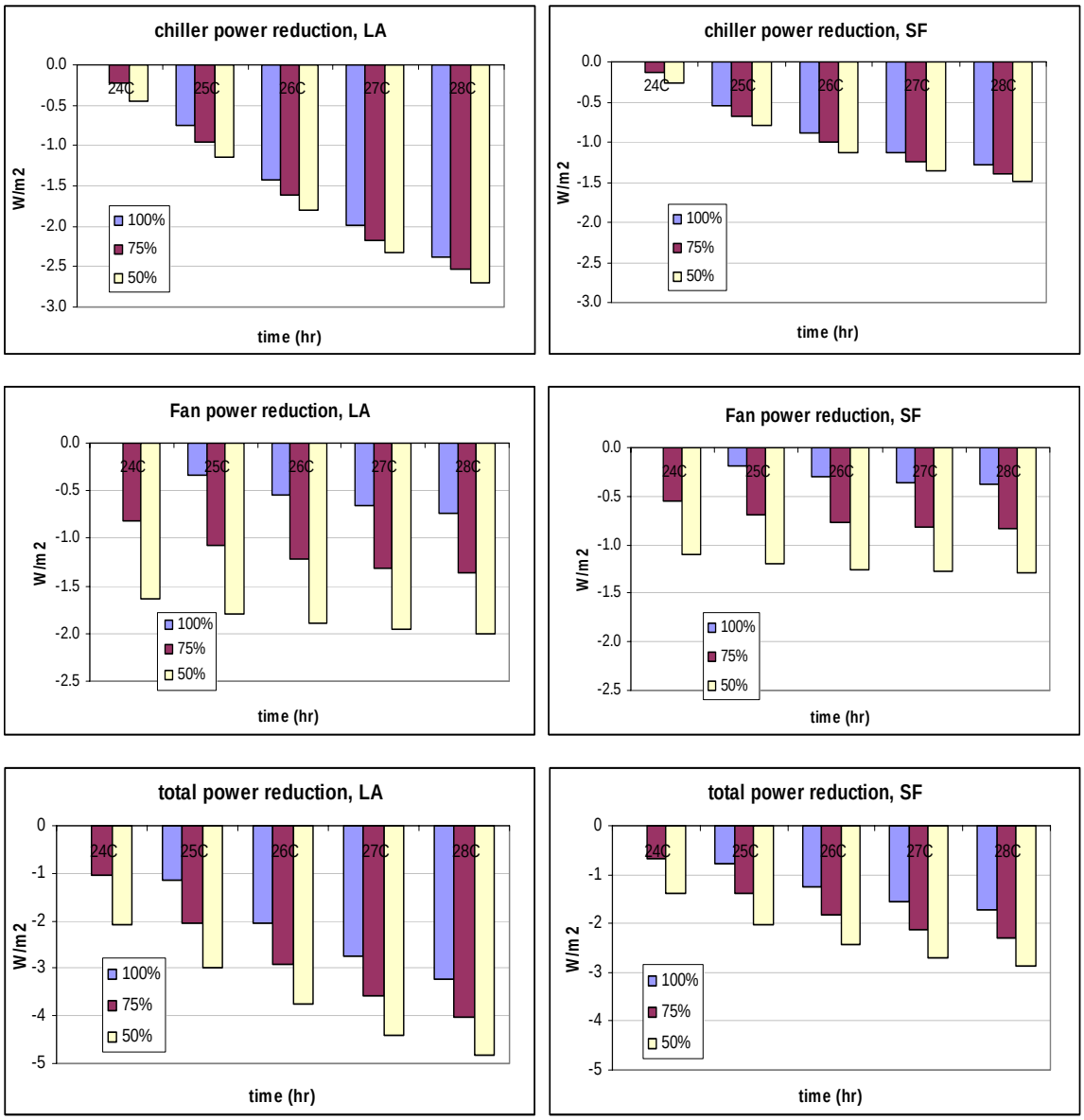
Fan capacity reduction by itself is a fairly effective load shedding method. In practice, fan often times runs at below full capacity. The very left bars in Figure 2.11 shows the shedding performance of controlling fan capacity. With 25% capacity reduction only, we see 1 W/m^2 average power savings in LA and 0.6 W/m^2 in SF, and a demand reduction of 1.4 W/m^2 in LA and 1 W/m^2 in SF; 50% fan capacity reduction alone leads to 2 W/m^2 average power saving in LA and 1.4 W/m^2 in SF, and a demand reduction of 2.8 W/m^2 in LA and 1.8 W/m^2 in SF. It does little harm to the service level as the indoor air temperature is maintained at 24°C. Combining fan capacity reduction with thermostat increase is more effective, as shown by the rest of Figure 2.11. A 25% fan capacity reduction along with a 4°C thermostat increase lead to an average power savings of 4.6 W/m^2 in LA and 2.5 W/m^2 in SF, and a demand reduction of 11.8 W/m^2 in LA and 7.5 W/m^2 in SF. A 50% reduction together with a 4°C thermostat increase produces an average power savings of 4.8 W/m^2 in LA and 2.8 W/m^2 in SF, and a demand reduction of 12.4 W/m^2 in LA and 8 W/m^2 in SF. Haves [2001] reported a 4.5 – 6.5 W/m^2 demand reduction for a 20% supply fan capacity reduction. The fan static pressure rise in Haves' research is about twice of that in our system. Therefore, discrepancies between two sets of results are expected when fan capacity reduction plays a role. The service level degradation is similar to that in the previous case, shown in Figure 2.9, where only thermostats are adjusted. Shedding fan capacity helps reduce the power profile and has little impact on the service level. We want to point out that the relative benefit of improving thermostat set points and shedding fan capacity is smaller than the sum of implementing them alone. The reason is that a VAV system will cut the flow rate to respond a thermostat set point increase. Shedding fan capacity with a smaller flow rate leads less energy savings than with a larger flow rate. However, the absolute energy performance is still improved as we combine these two. A 4°C temperature increase alone leads to an 11 W/m^2 demand reduction and a combination of 4°C and 25% fan capacity reduction to 11.8W/m^2.



Figures 2.9 a) Hourly power b) PPD profiles for 4-hour thermostat adjustment (24°C up to 28°C for hours 14-17) on a typical summer day in LA



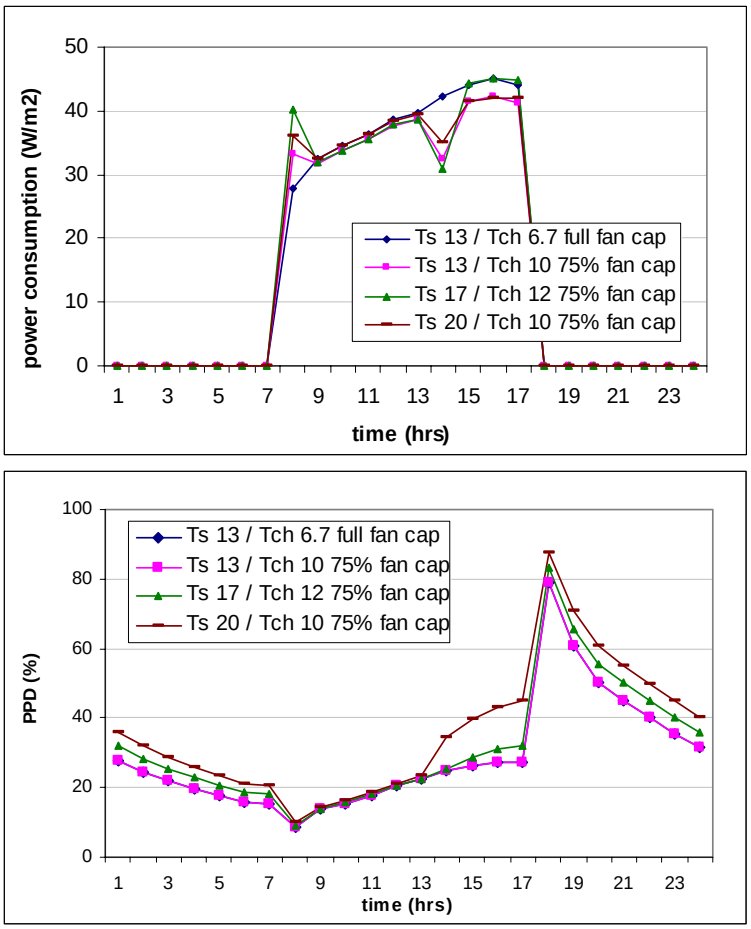
Figures 2.10 a) daytime average (top) and b) chiller (bottom) power reduction, for LA and SF on typical summer days with different thermostat set points during hours 14-17



Figures 2.11 c) Average total power reduction (Bottom), b) reduction contributed by fan (middle) and by a) chiller (top), for LA and SF in summer with supply air fan capacity reduction (50-75% of the original capacity) and 4-hour thermostat adjustment (hours 14-17)

Figure 2.12 gives the hourly load and PPD profiles corresponding to different combinations of supply air and chilled water temperatures and fan capacity reduction. In all shedding cases, those hours during and after shedding are seeing reduction, but the early morning pick-up can be a problem for systems without night cooling available when chilled water temperature is pushed too high, for example Tch of 12°C. In addition, the combination of Ts = 20°C, Tch = 10°C and 25% fan capacity reduction performs worse than Ts = 13°C, Tch =10°C and 25% fan capacity reduction, which is because the fan does not saturate until 20°C, and further increase will lead to energy savings and cause severe comfort problems. This is consistent with our findings before about the relationship between total power and supply air temperature. We conclude that there is an optimal combination of Ts and Tch, and simple parametric studies usually find a sub-optimum.

Haves [2001] suggested that a combination of supply air temperature increase and chilled water temperature increase during peak hours will bring in more demand reduction, and reported 7 - 9 W/m^2 savings by raising supply air temperature by 3°C and chilled water temperature by 4°C while reducing fan capacity by 20% for four hours in the afternoon. We also got substantial demand reduction by taking the similar short-term load control approach, but the average savings is still small, shown in Figure 2.13. For a 2°C supply air temperature increase and 2.3°C chilled water temperature increase, plus a 25% fan capacity reduction, we see about 1 W/m^2 average power savings in both LA and SF, and a demand reduction of 7.6 W/m^2 in LA and 4.5 W/m^2 in SF. The base case in Figure 2.13 has a supply air temperature of 13°C, a chilled water temperature of 6.7°C and full fan capacity. From the power breakdowns, we see that the savings in chiller power was partially cancelled out and in some cases overweighed by the fan power increase. The supply air temperature was not high enough to saturate the fan. Further shedding fan capacity might help if permitted. Overall, there are more savings in LA than in SF, which we believe due to the differences in base loads. The mild weather in SF leads to a low base load, and further reduction is therefore less helpful.



Figures 2.12 Hourly power (top) and PPD (bottom) profiles on August 10, LA for different combinations of supply air temperature increase and chilled water temperature increase, and fan capacity reduction

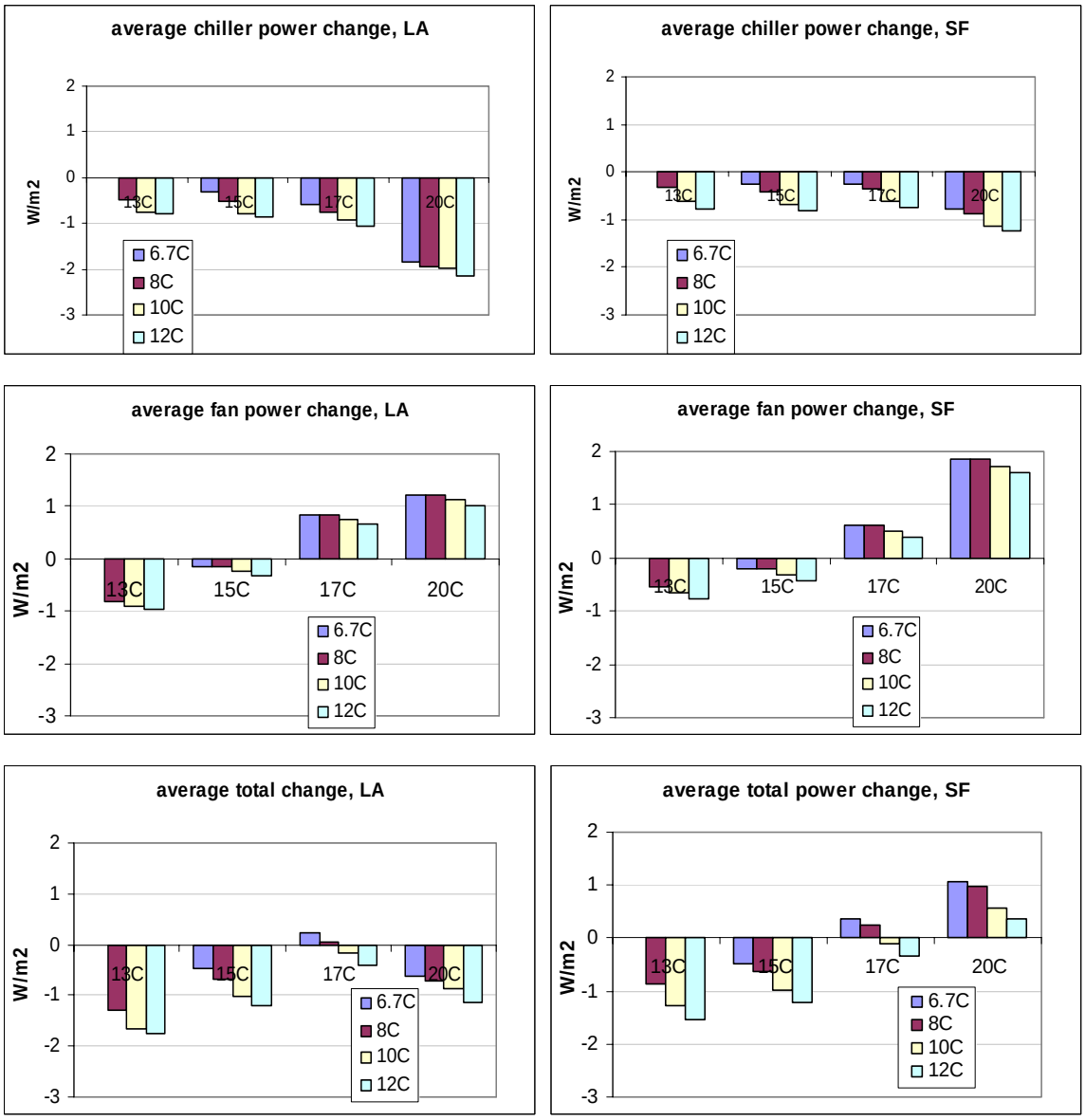


Figure 2.13 a) Average total demand reduction (top), b) demand reduction contributed by fan and pump (middle), and c) demand reduction by chiller (bottom), for LA and SF on typical summer days for combinations of supply air temperature increase (within the legend), chilled water temperature increase (along the x axis), and fan capacity with 25% reduction

Table 2.2 compares the demand reduction results in this research with Haves' results [2001] for different types of load shedding strategies in a few locations in California. The two sets of research are comparable because they both study commercial VAV systems in similar climates by taking similar load shedding approaches, and the peak loads per unit area are also close. The discrepancies are a function of the specific buildings under study.

Table 2.2 Demand reduction vs. Load shedding strategies

Load shedding measures Demand reduction	Our results (W/m^2)	Haves' results (W/m^2)	Implementation
Increase thermostat set points by 4°C	LA ~ 11.7 SF ~ 7	9.3	Network-addressable thermostat controller
Increase thermostats by 4°C and reduce fan capacity by 25%	LA ~ 11.8 SF ~ 7.5	11– 12	Adjust max speed or static pressure rise
Increase supply air and chilled water temperatures by 3°C while shedding fan capacity by 20%	LA ~7.6 SF ~ 4.5	7 -9	Alternative if thermostats are difficult to control

Norford et al. [2002] used a simplified analytical model the impact of changing thermostat, supply air temperatures and lighting on demand reduction and zone temperature for different internal load levels. Haves and Gu [2001] did a similar analysis in EnergyPlus for several cities in California and suggested some operation guidelines. Both pointed out VAV systems have more flexibility over CV systems in generating energy savings through load shedding. The exact amount of savings depends on the building thermal properties, load characteristics and the load shedding strategies applied.

We want to point out that often times it is the implementation issue associated with a load shedding strategy that determines whether it can be used. As our parametric studies show, increasing thermostats is very efficient and the analysis is simple. In order to implement it in a large commercial building, the zone temperature controllers need to be digital and network-addressable. It would be too difficult to adjust thermostats manually. If the zone temperature controllers are not network-addressable, load can be shed by reducing the air handling unit fan capacity, either on its own or while increasing the supply air temperature and chilled water temperatures. This partially motivates our looking at a variety of load shedding strategies. In the next several chapters, we assume thermostats can be adjusted easily.

In all the previous studies, we pre-specified the load shedding schedules, such as how long each period should be and when load shedding should start. In practice, however, scheduling itself is an important issue and needs to be determined through optimization. In this section, we illustrate the importance of the load shedding scheduling by simply comparing several thermostat set point-based cases with different schedules. It is worth in the future looking into the scheduling problem in detail and in a systematic fashion, and making suggestions on how building operators should time load shedding strategies based on rate structures, buildings, mechanical systems used, occupants' preferences, weather, and the shedding strategies to be applied.

With only thermostat set points as control variables, we look at the impact of shedding duration and starting time, with results shown in Figures 2.14 and 2.15. The base case, where thermostats are set at 24°C and no load shedding available, is compared with other three cases: 1) thermostats are set to 28°C for hours 13-16, 2) 28°C for hours 14-17, and 3) 28°C for hours 13-14. The comparison between 1) and 2) looks at when load shedding should start, and 1) and 3) looks at how long load shedding should last. Figure 2.14 compares the daytime average power consumption and the peak demand between these cases, and Figure 2.15 shows the corresponding hourly power and PPD profiles.

Because the building load goes down substantially after hour 18, a thermostat increase during hour 14-17 avoids the load setback recovery spike, although the overall power curve is pushed up for the next day, reflected in the increase early morning power. However, a similar 4-hour 4°C thermostat increase which happens two hours earlier incurs large load set-back recovery, which leads to a 1.3 W/m^2 average demand increase and a 12 W/m^2 peak demand increase compared to load shedding that takes place during hours 14-17. Therefore, the well-timed hour 14-17 load shedding offers demand charge advantage, giving the same the service level reduction. Compared to the base case, the load shedding case during hours 13-14 doesn't really offer any benefit: almost the same total consumption and the increased peak demand. This is because the load set-back recovery outweighs savings, which shows the importance of choosing the shedding period. The scheduling problem will get more complicated if the occupancy and a few operations schedules are subject to change.

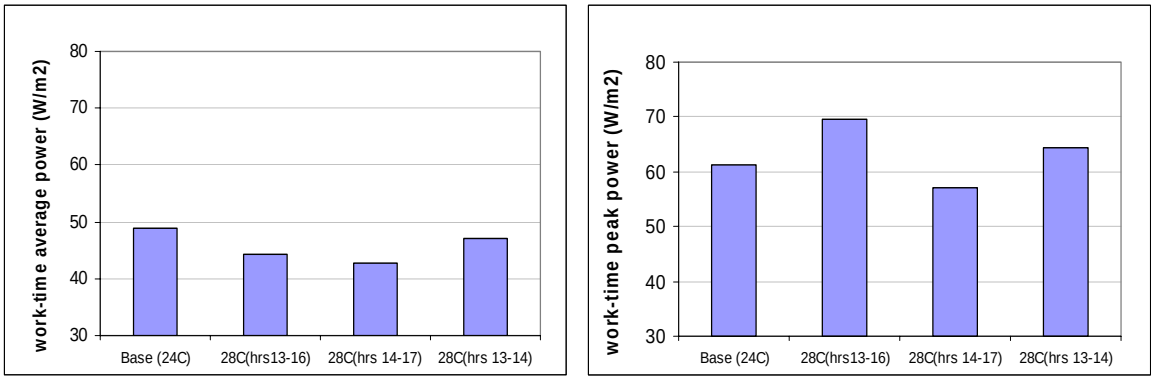
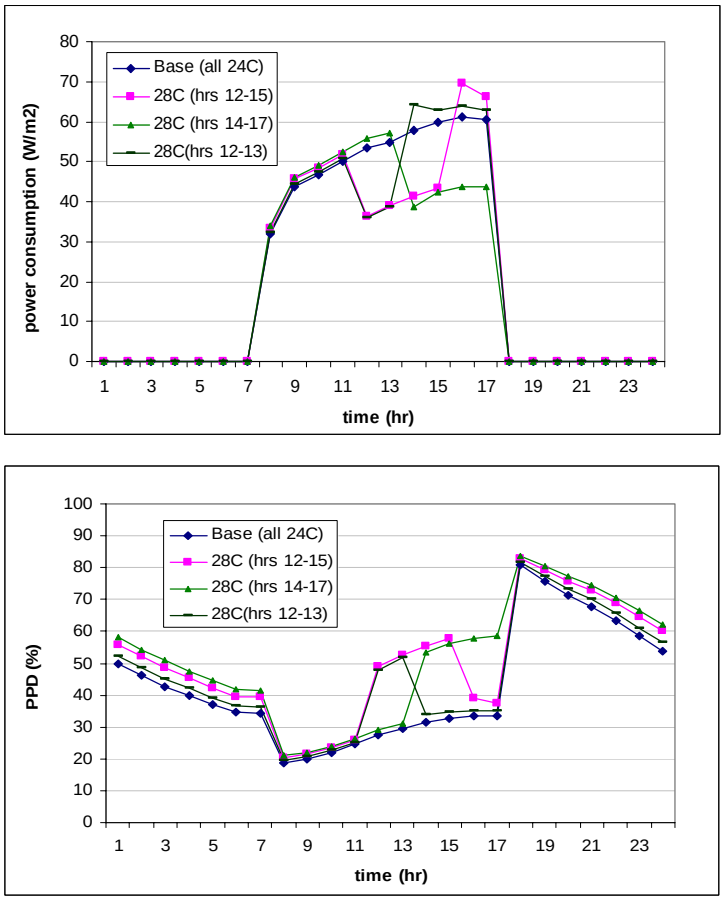


Figure 2.14 Daytime average power consumption (left) and peak power (right) with different scheduling durations and starting times under thermostat-based load control



Figures 2.15 Hourly power profile (top) and PPD profile (bottom) with different scheduling durations and starting times under thermostat-based load control

Chiller power is a big portion of the entire mechanical system energy consumption, 50-70% in this three-zone VAV case. Naturally, we'd like to turn off chillers during peak hours to cut peak demand and hopefully total power consumption, provided doing so wouldn't unduly hurt thermal comfort. In this three-zone VAV case, we compared the following cases: 1) chiller is on the whole time; 2) chiller off 15-16; and 3) chiller off 16-17. The simulation is done to three types of thermal masses on six individual summer days in LA, and averaged results are presented below.

Figure 2.16 shows the occupied-time average load and PPD for three chiller control strategies and three types of thermal mass. 1mass is the base case and 2mass and 3mass represent that the thermal mass is doubled and tripled. We simply double and triple the density of major construction materials and the system size remains the same. The details can be found in the material object description of the EnergyPlus model given in Appendix 1.2. As expected, turning chiller off leads to big energy savings associated with the much worse comfort condition. The peak demand chart in Figure 2.16 shows that high thermal mass is more tolerant of the temporary shut-off of the chiller, and setup recovery is smaller than that with low thermal mass; however, it is also more difficult for heat accumulated during the shedding period to dissipate over night when the system is off. The bottom graph in Figure 2.17 indicates that with the same chiller shedding strategy, high thermal mass leads to higher early day power consumption, although this doesn't affect the daily peak. We plot the hourly load and PPD curves for the 3mass scenario on a summer day in LA, as shown in Top and Middle of Figure 2.17. Even with the highest thermal mass, a 3mass model, the hour(s) when chiller is off corresponds to a PPD of 70-90%. Therefore, turning the chiller off for an hour is not acceptable in this system on a summer hot day.

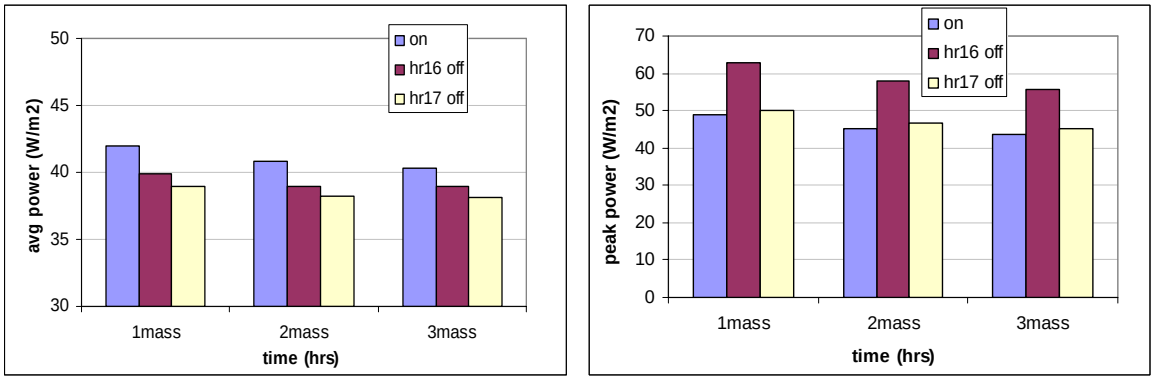


Figure 2.16 Occupied-time average power consumption and PPD for three chiller cases (always on, off during hour 15-16, and off during hrs 16-17) with three different types of thermal mass (1,2, and 3mass) on summer typical days (average of six summer days)

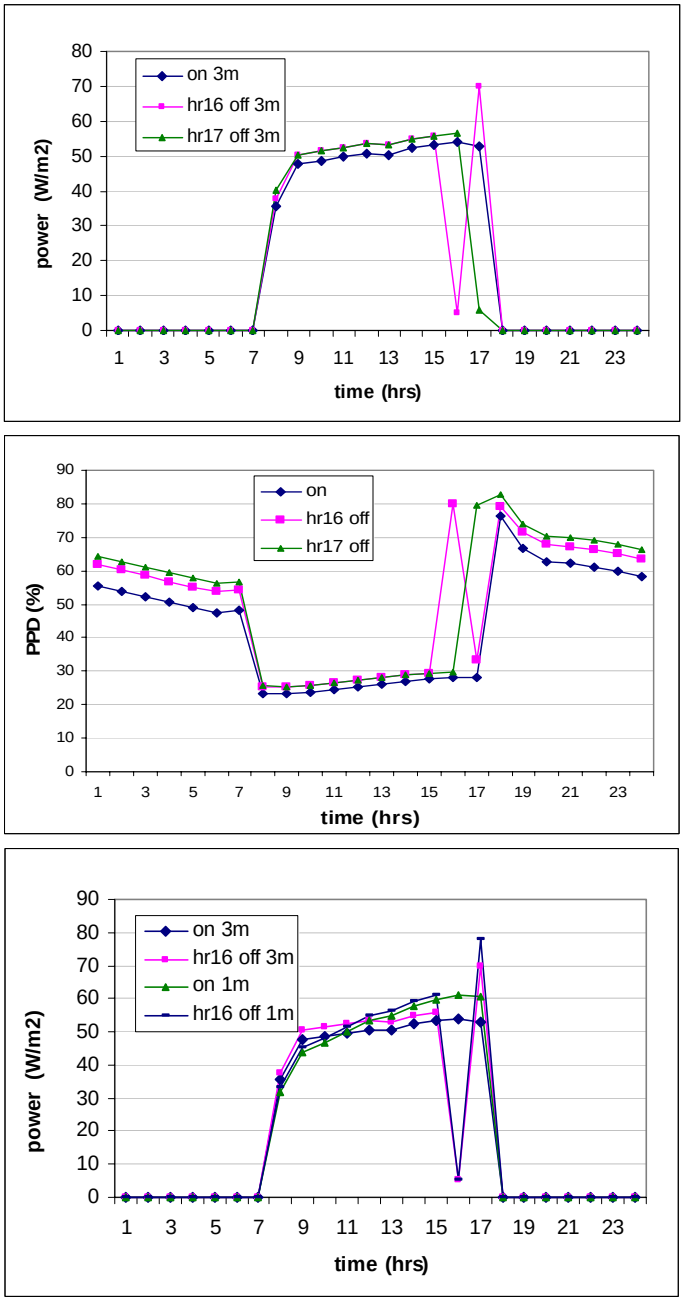


Figure 2.17 Hourly load (top) and PPD profiles (middle) vary with mass under temporary chiller-off strategies (on, off hr 15-16, off hr16- 17), and load setback recovery comparison between 1mass and 3mass (bottom), 8/8, LA

We use EnergyPlus version 1.1.0 in this research, and did not find a way to turn the chiller off for less than an hour. The most recent version of EnergyPlus 1.1.1 that came out when this thesis was prepared can schedule chillers on a time scale that is less than an hour. This would be an immediate next step of research because turning chiller off for a short period of time could offer good energy saving potential without hurting comfort much. Turning off chiller temporarily might be a more appropriate choice if the building thermal mass is cooled over night. In addition, increasing the thermostat set point and/or shedding supply fan capacity when chiller is off might also help.

2.5 Another load control strategy – night cooling

Previous load control strategies only concern the peak-load related hours in the afternoon. It will help to work on the entire next day's profile and take advantage of free cooling at night. We approach load control through night cooling in this section through parametric studies, and will come back to the same problem through an optimization approach in next chapter. We compare two night cooling schemes: mechanical ventilation – running fan at night and use the free cooling resource, and mechanical cooling – running chiller at night, for different thermal mass types and weather conditions. Scheduling night cooling to improve the load control performance is in the center of the discussion.

The base model described in section 2.1 remains in use, in which the system is turned off at night and the economizer is off. For fan-based night cooling, we allow fan starting time and discharge process, e.g. early morning thermostat set points, to change while keeping economizer set point at 24°C. For chiller-based night cooling, we allow chiller starting time and discharge process to change while keeping economizer off. The thermal mass in the base case is doubled and tripled and impact of thermal mass on night cooling performance is compared. Two weather types, LA and Austin, are compared as well regarding the night-cooling-based load control. Generally, night cooling benefits from large diurnal temperature differences, so its performance varies from day to day. Acknowledging this, we conduct parametric studies on the same single day for simplification purposes, and day to day differences are ignored.

Some of the following scheduling aspects in fan-based and chiller-based night cooling have been studied before [Braun 2001, Rabl and Norford 1991]. We look at them through VBA-automated parametric studies:

- Thermostat setting points during the night cooling time

  For fan-based night cooling, our strategy is to maximize the use of free cooling. Whenever the outside air temperature is below inside, the economizer is open to full, assuming that humidity is not a problem. Therefore, the nighttime thermostat set points have no impact on fan-based night cooling. For chiller-based night cooling, nighttime thermostat set points work the same way as daytime set points

- Thermostat set points early during the day

  These set points control the warm-up period and the way the stored energy is released, and affects the energy performance the entire next day. We define a set of early morning thermostat set points as a discharge process. Figure 2.18 examines the impact of discharge process on fan-based night cooling, and Figure 2.19 does so for chiller-based night cooling. The discharge process affects both in the same way. It would be ideal if we could track thermal mass temperatures and make operation decisions accordingly, but it is still difficult to implement this in EnergyPlus currently.

The names of the discharge processes in Figure 2.18.a and 2.19.a are borrowed from Braun's work [2001] which also compares the performance of these processes. Fast Linear and Slow Linear describe how thermostats change gradually, at different paces, from 18°C at night to 24°C after the day starts. Maximum discharge means thermostat set points turn to normal, e.g. 24°C or 25°C, right after the day starts and stay constant in the rest of the day. We add one more case named Until Peak which is to keep the temperature set points low until the peak is reached or very close. It starts at 22°C , increases to 23°C in an hour, remains flat until 3pm and goes up to 24°C. In both figures, Slow Linear and Fast Linear both cause load spikes early during the day due to the low temperature set points, and Slow Linear corresponds to a higher spike. The early spikes are partially paid off later during the day, but the late-day load reduction is fairly small. Until Peak has a flat power profile with the lowest peak of all the scenarios. Figures 2.18.b and 2.19.b have the constant day thermostat set points of 24°C as the base case, and the power differences are shown on the charts. Until Peak is the most efficient with close to a 7W/ m^2 peak reduction and similar total energy consumption to that in the base case. Maximum Charge to 25°C is also good and sees both peak and total load reduction. But unlike other four cases where the space is not heated up because the temperature set points are lower than in the base case, Maximum Charge to 25°C corresponds to an about 6% PPD increase in both fan-based and chiller-based cases.

Overall, reduction in early morning temperature set points helps reduce the afternoon peak but not the total load, and comfort is little affected. Keeping thermostat set points low before peak is very efficient: the thermal mass discharge process is controlled such that energy stored through night cooling carries

throughout the day and helps reduce the peak. This shows the benefit of looking at the day as a whole. The disadvantage is that most time of the day is probably cold and less comfortable for some occupants.

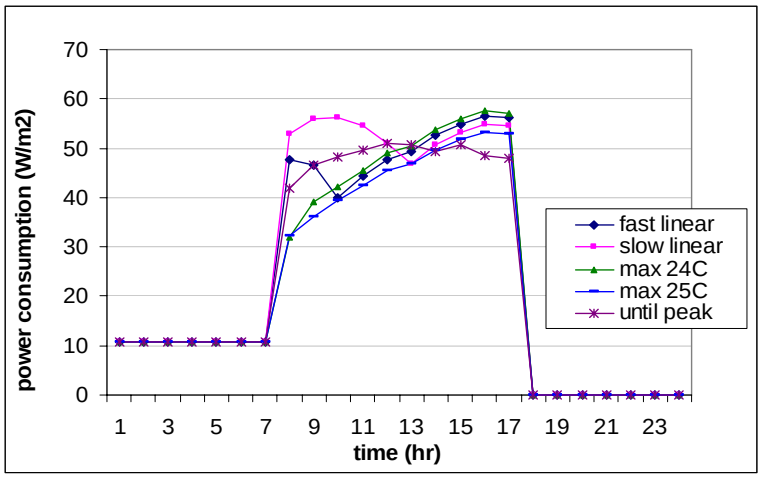


Figure 2.18.a) Load profiles of night cooling by mechanical ventilation with discharge processes, night cooling fan starts midnight with nighttime thermostat set points of 18C, and chiller starts 8am, August 8, LA

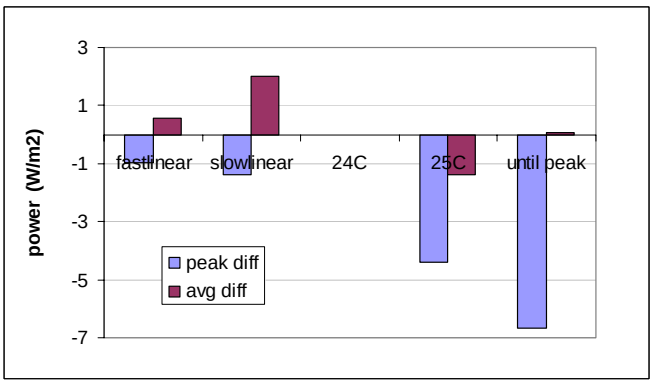


Figure2.18.b) Peak load and average load difference from the base case due to fan-based night cooling with different discharge processes

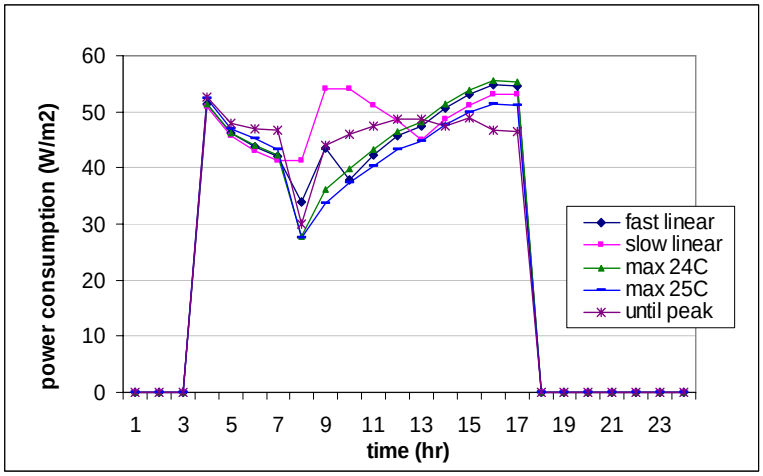


Figure 2.19.a) Load profiles of chiller-based night cooling, with different discharge processes in the early morning, and chiller starts at 4am with a nighttime thermostat set point of 18C, August 8, LA

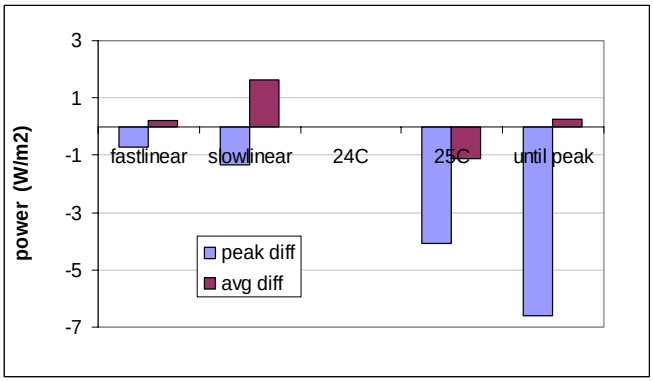


Figure2.19.b) Peak load and average load difference from the base case due to chiller-based night cooling with different discharge processes

- Night cooling start time

No matter which approach taken to night cooling, fan-based or chiller-based, when to start night cooling is a question. Early start can always help reduce the day peak load, but is likely to consume too much energy overall. We need a balance between getting enough low-cost cooling resource stored in thermal mass over night for next day use and consuming reasonable amount of energy over. Figure 2.20 is the parametric study results for fan-based night cooling, and Figure 2.21 for chiller-based. A rigorous optimization process will be set up in the next chapter.

Figure 2.20.a and Figure 2.21.a are power profiles with different fan and chiller starting times. Figure 2.20.b and Figure 2.21.b show the power and PPD difference between different night cooling start times and the base case, which is no night cooling at all. All cases have the same thermostat set points. From Figures 2.20, we see that for a fan-based night cooling strategy, early starting of the fan helps reduce the peak load and flatten the peak period and lower the maximum PPD and average PPD values. But early starting of the fan leads to the most energy use overall. It is a trade-off between getting enough energy stored and using less energy to run the fan at night. We need to figure out when the thermal mass gets fully charged. Starting the chiller early has a similar impact, according to Figure 2.21, but the chiller-based night cooling has more dramatic power and PPD impact.

It is clear that scheduling night cooling is an optimization problem with the peak load and PPD and the total load as conflicting goals. We can unite them under a total energy cost given a certain rate structure, which will be part of the next chapter.

The impact of weather and location on determining whether night cooling helps and which night cooling strategy is better is examined in Figure 2.22, where the chiller starts at 4am, fan starts at midnight and everything else remains the same. No night cooling, fan-base, and chiller-based night cooling strategies are applied to the same building in LA and Austin. LA has a high of 29°C, low 19°C and average of 23°C, while Austin has a high of 34°C, low 23°C and average of 28°C. In LA, chiller starting at 4am is almost equivalent to starting fan at midnight in terms of peak load and comfort but at the price of more total power consumption. However, in Austin, fan-based night cooling is less attractive as the outside temperature stays above the thermostat set point most of the time. Chiller-based night cooling reduces peak load and improves the overall load profile.

We also look at the impact of thermal mass combined with weather on the night cooling performance. In Figure 2.23, we simulate three types of thermal mass: 1mass, 2mass, and 3mass. 2mass and 3mass have

been defined before. For each thermal mass, we conduct three cases: traditional night set-up, fan-based night cooling and chiller-based night cooling. In the fan-based night cooling, we turn off the chiller at night and use mechanical ventilation with an economizer set point of 24°C from 12a to 7a. In the chiller-based night cooling, we turn the chiller on at 4am with the nighttime thermostat set point of 18°C, and let it run through the end of the working day. All the cases have the same indoor control target of 24°C during the occupied time. Figure 2.23 shows the results of three mass types, three strategies and two locations. Shown are power differences; the base cases for all charts are no night cooling and 1mass, with their figures at the bottom left corner of each chart. In LA, fan-based night cooling shows an advantage over chiller-based in term of peak load at higher thermal mass. In Austin, chiller-based constantly outperforms fan-based. Given the schedule used, chilled-based night cooling always consumes more total power than fan-based. 3mass chiller-based night cooling in LA has a higher peak than that of 2mass, which is due to the shift of peak from late afternoon to early morning pick up when the chiller starts at 4am. If we increase the night-time temperature set points from 18 to 20°C, the peaks shift back to around 4pm and 3mass has a smaller peak demand than 2mass. The 3mass case has the best daytime power profile due to the higher mass. For LA, with the same chiller schedule, 2mass outperforms 3mass in both peak and total load, which seems to contradict our expectation of better night cooling performance of higher thermal mass. It is only because chiller-related power is big enough to shift the daily peak and therefore leads us to a different problem. Which strategy and parameter combination is better depends on the rate structure, which could be Time of Use (TOU) energy charge, or demand charge plus flat rates or demand charge plus TOU energy charge. The load control strategy that gives the minimal total cost is the one should be used in operations.

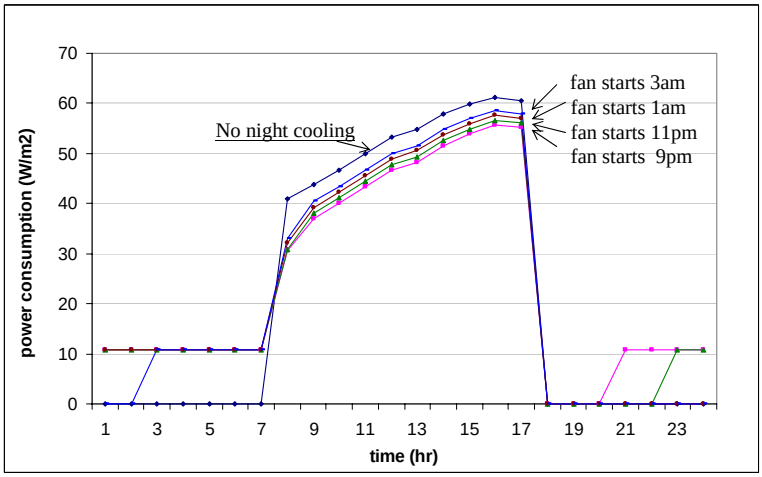


Figure 2.20.a) Load profiles of night cooling by mechanical ventilation with different fan starting times with nighttime thermostat set points of 18C, chiller starts at 8am and a maximum discharge to 24°C , August 8, LA

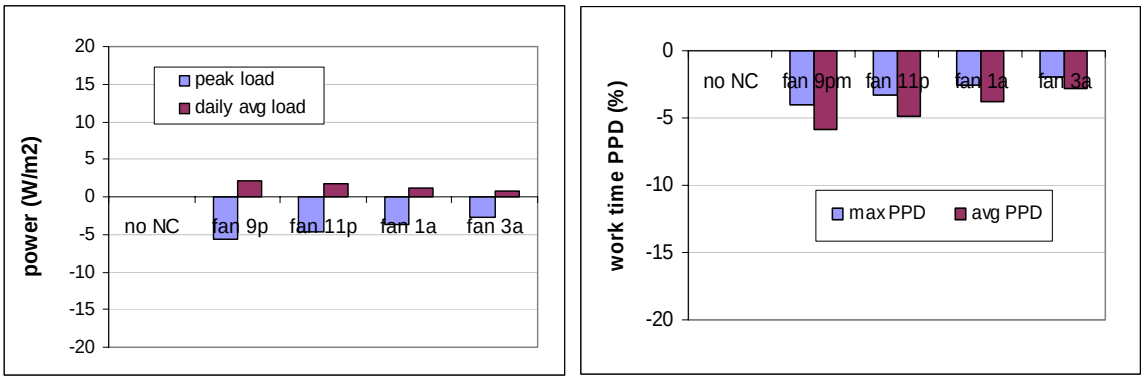


Figure 2.20.b) Summary of night cooling by mechanical ventilation with different fan starting times with nighttime thermostat set points of 18C and a maximum discharge to 24°C, 8/8, LA
Left: daily average load and peak load; Right: work time average PPD and peak PPD

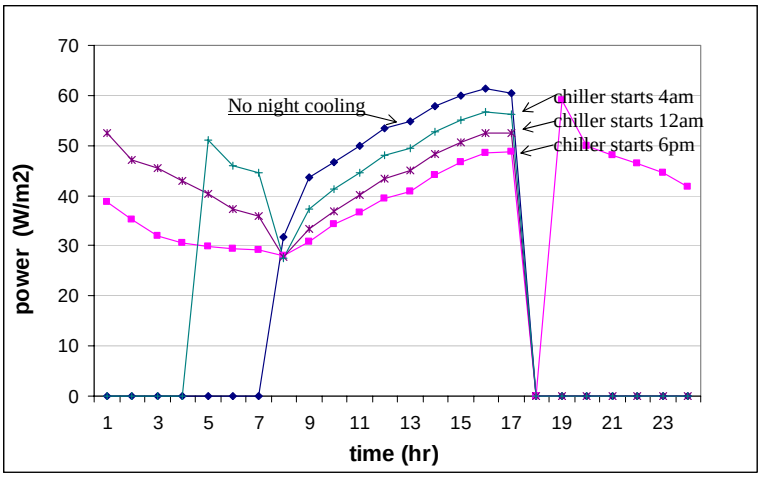


Figure 2.21.a) Load profiles of night cooling by running chiller at night with a nighttime thermostat set point of 18°C, for different chiller starting times, thermostats maximum discharge to 24°C, August 8, LA

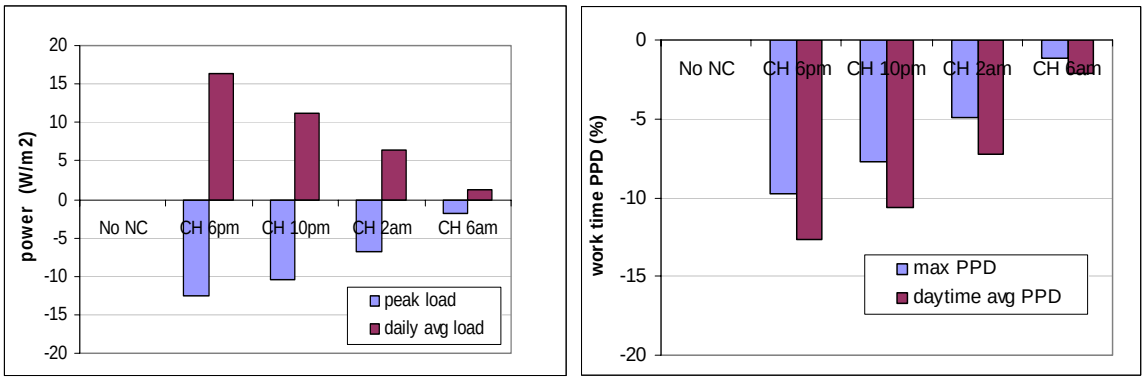


Figure 2.21.b) Summary of night cooling by chiller-based night cooling with different chiller starting times and a maximum discharge to 24°C, August 8, LA
Left: daily average load and peak load; Right: work time average PPD and peak PPD

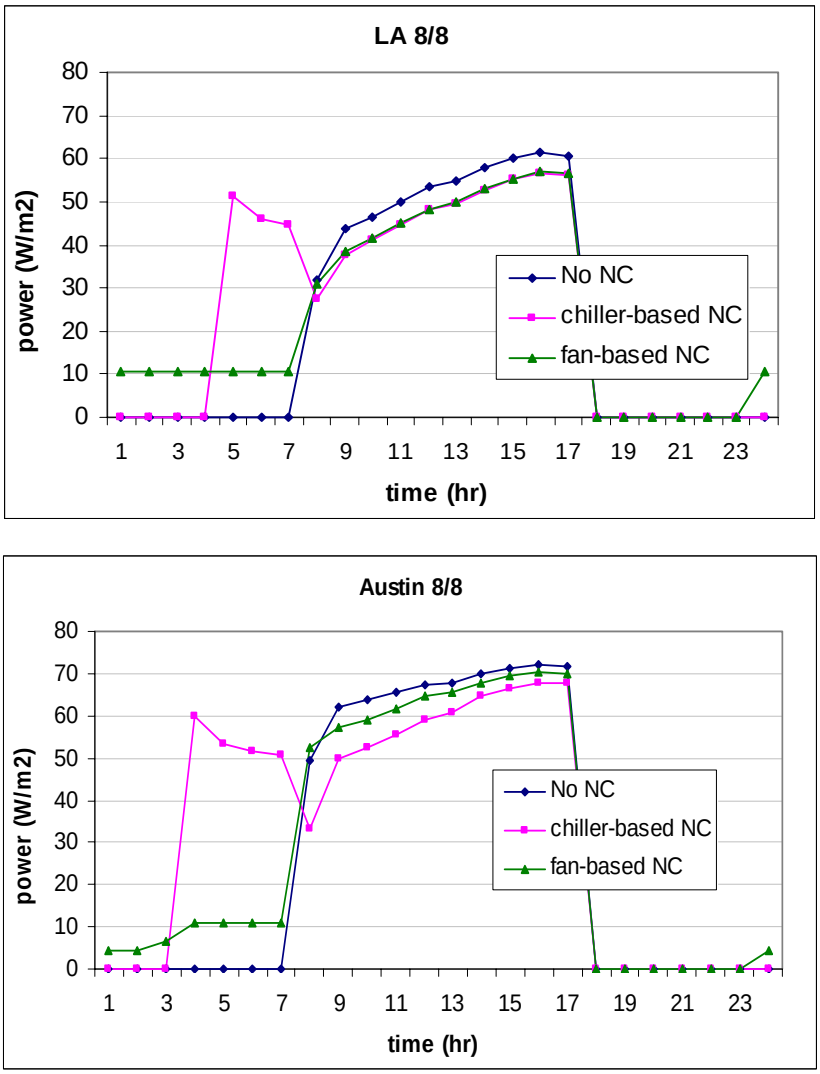
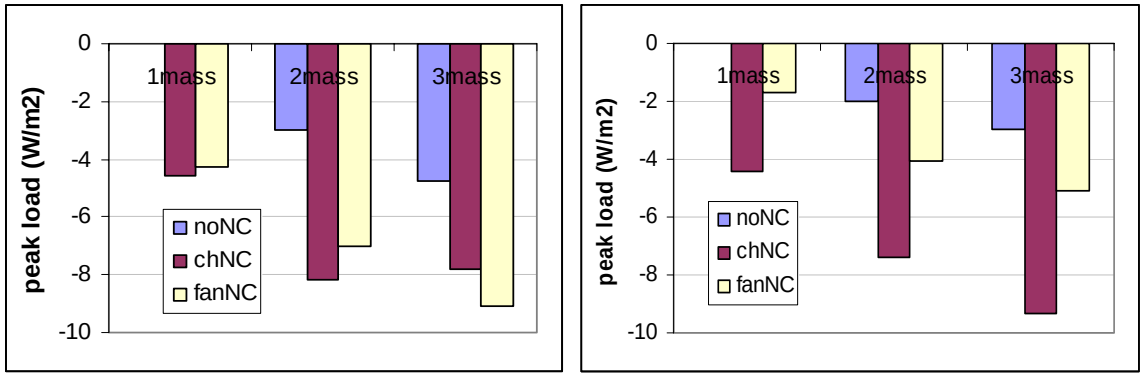
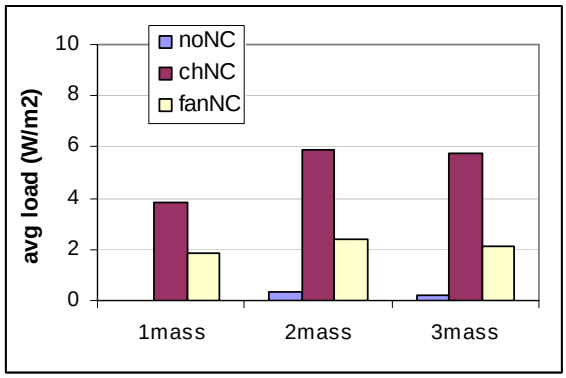


Figure 2.22 power profiles of no night cooling, chiller-based with chiller on at 4am and fan-based night cooling with fan on at midnight, nighttime thermostat set point 18°C, daytime maximum discharge 24, August 8, LA and Austin

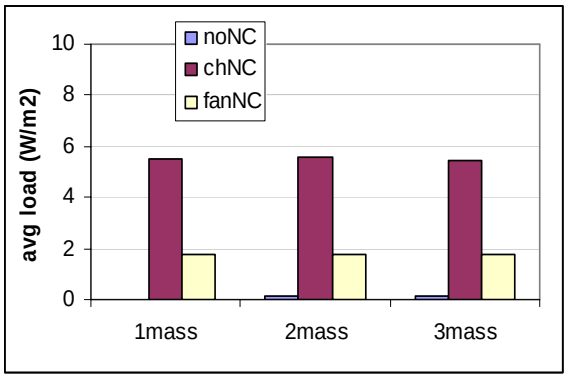


61 W/m^2

72 W/m^2



22 W/m^2



28 W/m^2

Figure 2.23 peak load and average load comparison between three thermal mass types, three night cooling strategies and two locations, left: LA; right: Austin, August 8
Base case given at the lower left corners of the charts

There are a few other factors that affect the night cooling performance as well, for example, the economizer set point in fan-based night cooling. In our studies, we set it to 24°C, which in most cases equals the indoor thermostat set point, so we already took great advantage of free cooling and further studies wouldn't be essential. Internal load patterns also affect night cooling, but the load pattern for a given building is assumed to be fixed throughout this research for simplification purposes. Table 2.3 summarizes briefly the night cooling parametric studies with several examples.

Table 2.3 Night cooling parametric studies summery

LA	Peak load reduction (W/m2)	Average load increase (W/m2)
Fan starts at midnight and early morning thermostats remain low	4 – 9	0 - 2.5
Chiller starts at 4am and early morning thermostats remain low	1 – 7	0.5 - 2.3
Double thermal mass in either fan-based or chiller-based night cooling	3 - 5	0.5

2.6 Single-building parametric study summary

Some preliminary operation guidelines are suggested based on extensive EnergyPlus simulations. Several load shedding strategies are proved to be applicable to this VAV system. The most efficient load shedding method is to increase thermostats and reduce fan capacity at the same time. A 4°C increase in four-hour afternoon thermostats and a 25% supply fan capacity reduction corresponded to 4.8W/ m^2 average power savings and 11.8 W/ m^2 peak demand reduction for this particular building in LA. Increasing thermostats alone is efficient as well. Increasing supply air temperatures and chilled water temperatures helps little, unless supply air temperatures are allowed to increase until the supply fan gets saturated, which, however, would hurt the service level much. Increasing supply air temperatures and chilled water temperatures by 4°C, together with a 25% fan capacity reduction, produces an average power savings of 1-2W/ m^2 and a peak demand reduction around 6 W/m^2. The duration of the load shedding period and when to start have impact on energy savings. Both fan-based night cooling and chiller-based night cooling are studied regarding the schedules and discharge processes. Both approach take advantage of thermal mass and help reduce the peak load of next day. Fan-based night cooling is shown to be more energy efficient for this model in LA. Starting fan early leads to 4 - 9 W/ m^2 peak load reduction for this three-zone VAV system in LA, around 10% savings. Chiller-based night cooling has

better peak-load reduction performance in Austin, about 12% savings. Which night cooling approach to use largely depends on weather, mechanical system availability and the building under study. A more general load-control scheduling problem is to be studied in next chapter with the help of an optimizer.

If thermostat set points can be easily adjusted, building operators should choose to increase thermostats for short term load shedding. Otherwise, reducing fan capacity is an efficient alternative. We recommend night cooling if the plant and/or fan are programmable and whenever weather permits.

Those buildings with high thermal mass should take full advantage of it: if outdoor temperatures at night are low, use fan-based night cooling; if outdoor temperatures at night are fairly high but the chiller has more than one stage, run the chiller with partial capacity at night; if the chiller consumes too much electricity at night and the consumption outweighs the benefit, thermostat set point adjustment may be made for a period of time during the day. Other options include shedding fan capacity reduction if possible and turning off the chiller for a short period of time. High mass would help maintain comfort.

Those buildings with light thermal mass have fewer options. Running fan or chiller at night may end up consuming more energy. Turning chiller off during the day might incur severe comfort problems. A modest adjustment of thermostat set point may still work. Fan capacity reduction is always an option if the system has the extra capacity to be shed. There are still things operators can do: if outdoor temperatures are low at night and during the day, leave windows open and create some cross ventilation if possible; if the building has large windows, shade during the day; if the humidity is low, a small indoor fountain would help reduce the temperature. Passive measures like these are not the target of this research, but operators are encouraged to use them as they cost little and can be quite efficient.

Load shedding generally work better in LA than in San Francisco. The mild weather in summer time in SF makes load shedding less helpful. We also point out that this three-zone VAV system is not among the most sensitive ones to load shedding strategies. It is important to identify buildings that are more appropriate for load shedding. Put another way, what kind of properties in terms of construction, load and operations does a building need to possess to be responsive to certain types of load shedding strategies? The load control performance depends on the properties of the building under study.

It is to be noticed that lighting is an independent factor and should be considered separately from HVAC. The reason is that though light energy increases the cooling load of the HVAC system, the control strategy for the lighting system remains the same, which is to always keep the lowest level of lighting as

long as doing so will not affect the building's normal functionality. Therefore, lighting is not considered when we developed the system operation strategies.

The effectiveness of the guidelines proposed in this research heavily relies on a large number of EnergyPlus simulations. We choose this three-zone VAV model in our research due to its simplicity and yet completeness as a VAV system. We have done basic testing such as heat and mass balances and qualitative checking on parameter trends, and fixed a few EnergyPlus problems along the way with the help of the software package improvement. For example, we had problems with the pump model at early stage of the research and the problem was fixed in a later version of EnergyPlus. Another thing worth pointing out is that the current version of EnergyPlus, as of November 2003 when this thesis is prepared, is imprecise in calculating PMV values. This was observed in our research, as the base case corresponding to an average daytime PMV of 1. A few EnergyPlus users also reported a deviation of 0.5 at the public mailing list of EnergyPlus maintained by Gard.com. The conversion between PMV and PPD values is correct and based on Fanger empirical results [EnergyPlus 2003]. Throughout this research, a PMV of 1.5 or a PPD of 50% is used when comfort is treated as a constraint, with the understanding that the true system might be cooler. Fortunately, the procedures are designed to carry out simulation and optimization tasks automatically throughout our research. Therefore a change in the PMV calculator can be easily adopted and wouldn't hurt the system being developed. Overall, as a complex software package still under improvement, EnergyPlus is a great help to our research and also a challenge in the sense that we need to understand and overcome the complexity and potential problems of the modeling and simulation process. We use the three-zone VAV model throughout this research and make sure that results are consistent. To explore load shedding opportunities, especially in a multi-building setting, we need a variety of buildings models that have reasonably good responses to load shedding. It is not realistic to build them all from scratch. Therefore, those models used in a multi-building setting are derived from the base three-zone VAV model and differences lie in thermal mass or internal load pattern.

CHAPTER THREE
SINGLE BUILDING PROBLEM: LOAD CONTROL OPTIMIZATION

We have conducted extensive parametric studies and compared the energy and comfort performance between different load shedding strategies. A natural extension is to find out a way to optimize those load control parameters. In this chapter, we develop a simulation-based optimization scheme for the single building problem, and use the framework to study a few strategies. The fact that we rely on simulation for objective function evaluation makes optimization difficult because simulation-based results provide a discontinuous search space, and, in addition, our problems are mostly nonlinear. The number of control variables in the problem is always a constraint. We limit control variables to those found important in parametric studies to simplify the problem.

Two types of optimization algorithms are used to solve the single building problem: direct search algorithms (DS) implemented at Lawrence Berkeley National Laboratory (LBNL) as a generic optimization package GenOpt, and GA implemented in a MATLAB freeware package GAOT. We will first review these algorithms and then present and compare the optimization results by applying DS and GA to the single building problem. We also examine the cost function structure of the single building problem to gain insight into the nature of building optimization. In the end of this chapter, we propose a hybrid optimization algorithm for single building optimization.

3.1 Direct Search Algorithms and GenOpt

3.1.1 Algorithm reviews

Direct search algorithms (DSs) have been replaced by more sophisticated techniques as numerical optimization has matured and globalized quasi-Newton methods have been successful. However, as Lewis [1997] pointed out, direct search methods still persist for three reasons: 1) they work in practice and the heuristics on which DSs are built remain sound. The convergence has been gradually proved in recent years [Polak and Wetter 2001] for pattern search methods under certain constraints; 2) Quasi-Newton methods are not applicable to all nonlinear optimization problems, and DSs have succeeded when more elaborate approaches failed; 3) DSs can be the method of first recourse. DSs are derivative-free methods, meaning neither compute nor approximate derivatives, and for unconstrained optimization, they depend on the objective function only through the relative ranks of a countable set of function values.

DSs can be organized into three basic categories: Pattern Search, Simplex Methods (not the simplex for linear programming) and Methods with Adaptive Sets of Search Directions. Pattern search methods are characterized by a series of exploratory moves that consider the behavior of the objective function at a pattern of points, all of which lie on a rational lattice. The Hooke-Jeeves algorithm is a good example of pattern search methods. Simplex methods construct a series of simplexes (a simplex is n+1 points in an N -dimensional space) and proceed in the search space by reflecting a simplex through the centroid of one of the faces, which doesn't depend on derivative information. The Nelder-Mead simplex method has enjoyed enduring popularity although its robustness has long troubled numerical optimizers. Methods with Adaptive Sets of Search Directions attempt to accelerate the search by constructing directions designed to use information about the curvature of the objective obtained during the search. Powell's method is such an algorithm, and it takes advantage of the previous results to construct a new search direction – a quasi-derivative.

Overall, DSs remain popular because of their simplicity, flexibility, and reliability. We use this type of method as the first course in the building optimization problem, and will compare it with GA later in this research.

3.1.2 GenOpt

To optimize building control using simulation programs such as EnergyPlus, we should be able to 1) modify the control variables in the input file and read the objective value from the output file automatically and continuously; 2) start simulation automatically; 3) keep improving the objective function toward the optimum. GenOpt, a generic optimization software package developed by the Simulation Research Group at LBNL, meets these requirements. GenOpt is designed to minimize the objective function that is calculated by an external simulation program. Several direct search methods, including Hooke-Jeeves and Nelder-Mead-O'Neil, were implemented in its algorithm library.

Besides having implemented several direct search methods, GenOpt provides a good interface to connect simulation and optimization. GenOpt allows any text-based simulation programs to be used, and it does all the data management work. Users only need to define the related parameters and inform GenOpt by defining initialization, command, and configuration files. In case users wish to use their own optimization algorithms, they need only to focus on the mathematics, and GenOpt takes care of data communication. Figure 3.1 shows the GenOpt schemes: a) is the overall scheme with the focus on the simulation setup and b) more on the optimization side. Details of how GenOpt works can be found in the GenOpt documentation [GenOpt Manual 2002]. The characteristics of GenOpt discussed so far

motivated its use in our building load control problem, especially in single buildings. Later in this chapter we will illustrate the process via a single building optimization example.

The Hooke-Jeeves algorithm [GenOpt 2002] is one of the direct search algorithms implemented in GenOpt, and is the one used in this research. It generates steps along the valley of the objective function. The algorithm requires neither the gradient of the objective function nor a line search. The original Hooke-Jeeves algorithm solves the unconstrained problem. It was modified to solve the box-constrained problem in GenOpt by redefining the objective function. The algorithm can be divided into 1) an initial exploration, 2) a basic iteration, and 3) a step size reduction. Steps 1) and 2) make use of so-called exploratory moves in order to get local information about the direction in which the function decreases. At each resulting base point, a sequence of orthogonal exploratory moves is made. The algorithm updates the base point once a small change in the objective function is found or reduces the search step otherwise.

Unfortunately, the cost functions evaluated in EnergyPlus and other simulation programs such as TRNSYS and DOE-2 are 1) nonlinear, which is difficult to deal with by a local optimizer; and 2) discontinuous with respect to the design parameter, although the discontinuities could be small. The problem gets worse when the structure of the objective function gets more complex, e.g. the combination of total energy consumption and thermal comfort. To prevent from getting stuck in local optima, which is the common problem for direct search methods and most nonlinear optimization algorithms, we can conduct several rounds of optimization with different initial guesses and pick the best local optimum as a global suboptimum. An alternative is to implement global-convergence oriented (but not guaranteed) algorithms such as genetic algorithms in the GenOpt framework. At the time this thesis was being written, GenOpt at LBNL released a new version with several global optimization algorithms implemented, including several global heuristic optimization algorithms that can be used to solve optimization problems with continuous and/or discrete independent variables, and a GA-based optimizer was also under development.

GenOpt provides a generic simulation-based optimization framework and automates lots of data processing work related to text-based simulators. It also has several optimization algorithms included. We are most interested in the data management framework GenOpt has and will take advantage of that in our work. We will also be using the direct search algorithms implemented in GenOpt for the single building problem.

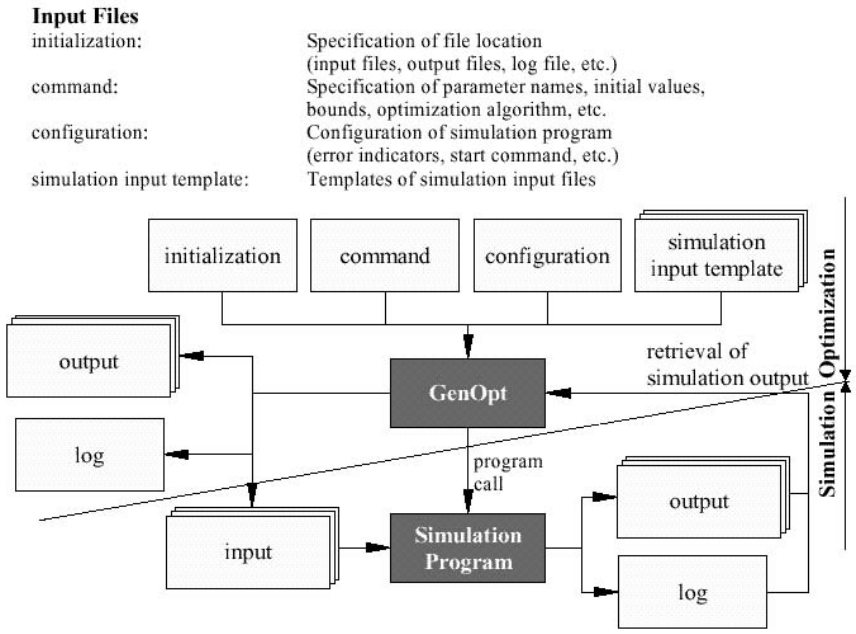


Figure3.1 GenOpt overall organization

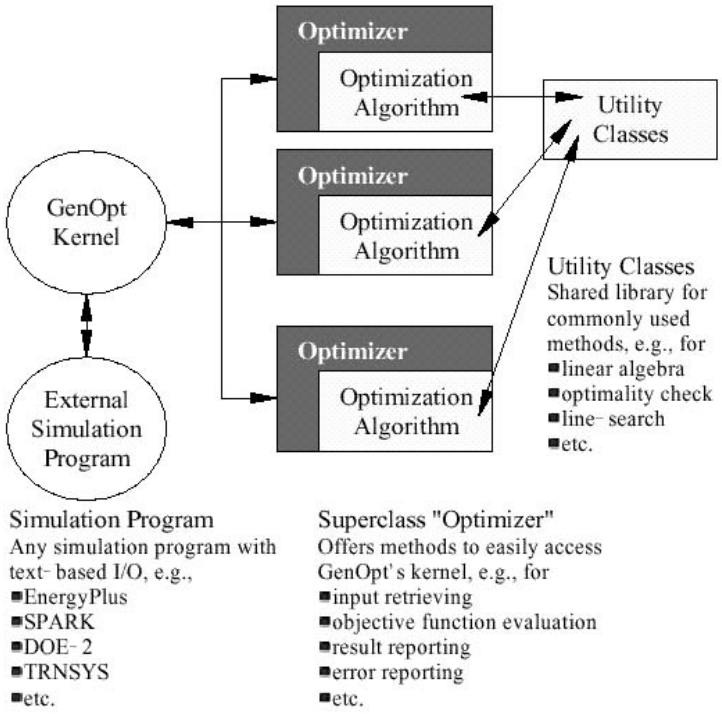


Figure3.2 GenOpt optimizer class

A detailed description of Hooke-Jeeves and other local-search algorithms implemented in GenOpt, and the general structure of GenOpt package can be found in the GenOpt manual [GenOpt 2002].

3.2 Genetic Algorithms and GAOT

3.2.1 Algorithm review

GAs have been reviewed in Chapter 1 as a general optimization technique. They were inspired by the natural evolution of species. They have been studied in the last twenty years as an evolutionary computation method along with simulated annealing. GAs are executed as a series of steps, called generations. They start with a population with a certain number of individuals, different states in the search space. In each generation, the individuals are evaluated with the fittest reproducing and continuing the next generations through fitness-based selection. The reproduction phase also introduces new individuals by applying genetic operators such as crossover and mutation to the current generation. A variety of genetic operators have been developed [Michalewicz 1992] to tackle certain aspects of different problems. The selection and/or invention of genetic operators are problem-specific and heavily depend on experience. As the process continues, the population converges to better individuals, which gives a higher likelihood of achieving global optimum.

3.2.2 GAOT: A GA Matlab toolbox

Genetic Algorithm Optimization Toolbox (GAOT) is a Matlab toolbox which implements simulated evolution in the Matlab environment using both binary and real representations [GAOT paper]. Ordered base representation has also been added to the toolbox. The implementation is flexible in the genetic operators, selection functions, termination functions as well as the evaluation functions that can be used. The toolbox was developed at the North Carolina State University and can be downloaded for free at http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/.

To use GAOT, users need to provide an evaluation function and several key parameters such as crossover and mutation probabilities, population size, and number of generations. Coding modifications are needed if evaluation is done by a simulation package, such as EnergyPlus, instead of an explicit function. In this research, Matlab is connected with EnergyPlus so that Matlab can modify EnergyPlus inputs as needed and start EnergyPlus runs within Matlab. The simulation results are post-processed in VBA for specific requirements of the cost function, e.g. certain rate structure, peak load or total load, and the final output is read in by Matlab as a fitness value for further optimization use. The scheme of this simulation-based GA is given in Figure 3.3, and the evaluation code myepeval.m is given in Appendix B.1. The Matlab-based GA framework is capable of dealing with 1) both single and multiple buildings; 2) different load control

strategies including thermostat control and night cooling; 3) a variety of objective functions and rate structures.

GAOT provides binary, real values and order-based representations, and a variety of selection functions and crossover and mutation operators for different representations. The real-value presentation is chosen in our problems as it matches the problem structure well, and is fairly computationally efficient [Michalewicz 1992].

In the following two sections, single building controls are optimized using GenOpt and GAOT respectively and with EnergyPlus as the evaluation function in both cases.

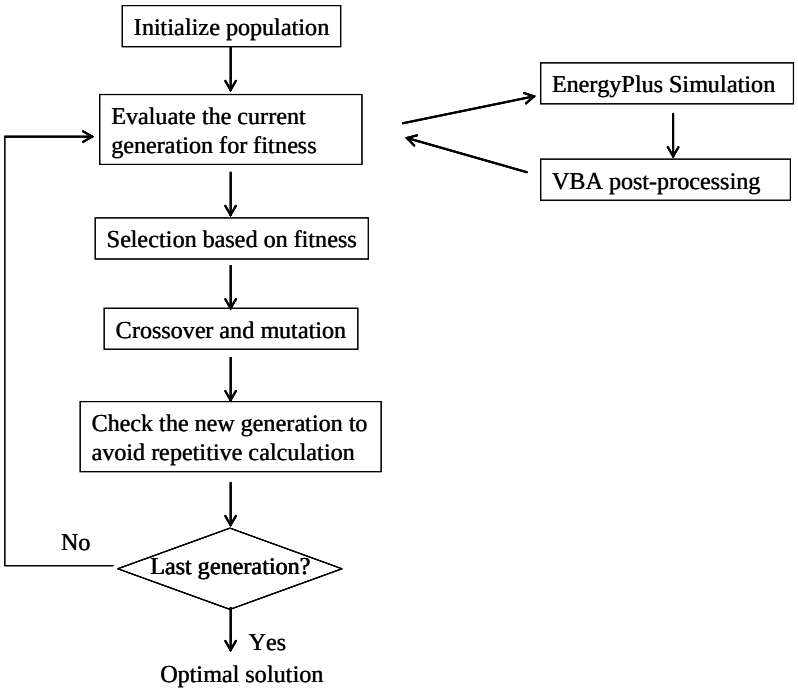


Figure 3.3 EnergyPlus-based GAOT scheme

3.3 Cost Function Structure

When dealing with an optimization problem, it is very helpful to have some sense of the search space we work on, although in most cases we can only learn limited characteristics of the cost function instead of the whole picture. If function evaluation is based on simulation, the cost function tends to have continuity issues, which makes optimization difficult. Furthermore, power consumption, especially when peak demand is involved, is a nonlinear function of operation parameters. To understand better the discontinuity and nonlinearity, we examine in this section cost function structures by visualizing the cost function surface over two randomly chosen control variables. To do so, we simply mesh the search space and compute for each grid point the cost function value, and there is no optimization involved. GenOpt provides such an algorithm named EqnMesh, which is used together with our VBA post-processor in computing the cost function surfaces. All the cases presented only have two control variables for visualization purposes. We enumerate 960 grid points over [22, 28] and find the best solution as the optimum. The GenOpt command file for EquMesh can be found in Appendix B.2.

Figure 3.4 illustrates the cost function surface when the total daily power is optimized over the thermostat set points in hour 16 and 17. The z-axis represents the total load while x and y stand for two thermostat set points. This is a monotonic case, as expected, since without comfort penalty, increasing thermostat set points will always reduce the total power. The variation in total power is rather mild because we are looking at the impact of two-hour thermostat set points on a whole day performance.

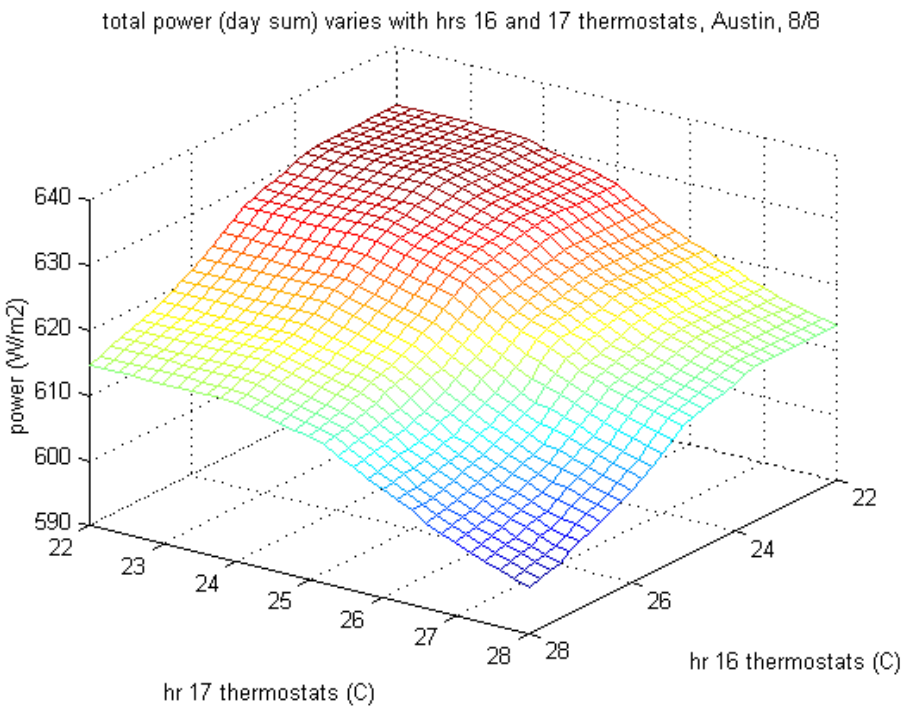
Figure 3.5 shows how the comfort level, measured by PPD, varies with the hour 16 and 17 thermostat set points. Similar to the total load case, PPD is also monotonic when two thermostat set points vary between 22°C and 28°C. Generally, 24°C is considered a comfortable set point for office buildings, but whether it leads to the lowest PMV and PPD values depends on other comfort-related factors such as clothes and activities. It could be the case that 22°C is more comfortable to occupants in this building than 24. In addition, the fact that EnergyPlus PMV calculations have been off by about 0.5, as previously noted, might play a role. Because we aim at load reduction, any temperatures lower than 24°C are of less interest and concern.

When peak load alone is to be minimized, the cost function structure has a different picture. As shown in Figure 3.6, starting from (22, 22) the peak load keeps dropping with the increase of thermostats and stops at (24.4, 24.2), a point in the middle of the field. Further increase of the thermostat will actually increase the peak. We explain why using Figure 3.7: three power profiles correspond to three sets of hour 16 and 17 thermostat set points: line 1 is the base case (24, 24), line 2 is the global minimum (24.3, 24.2) and line

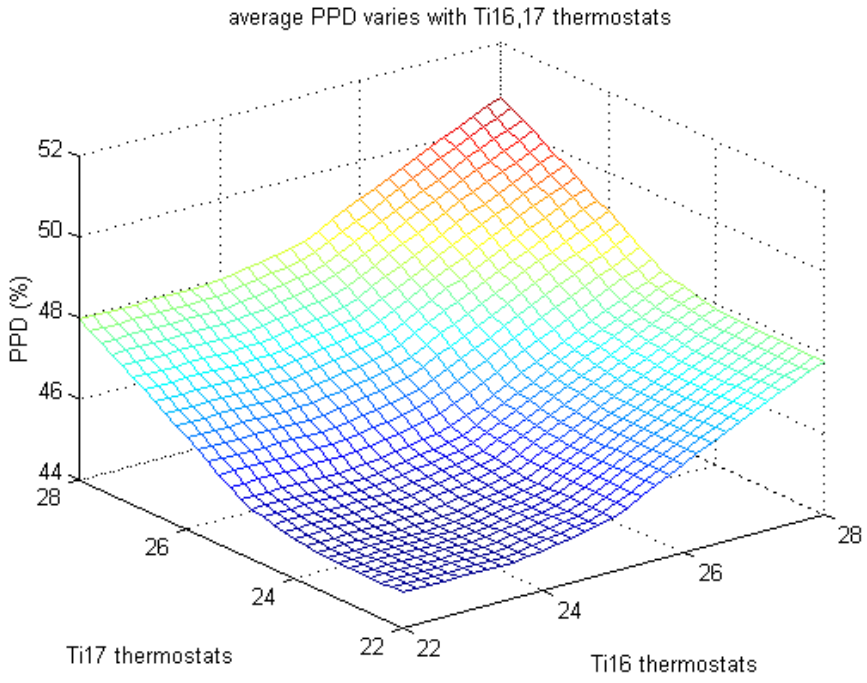
3 is a reference case (28, 28). Line 3 indeed sheds hour 16 and 17 power consumption, but the peak shifts to hour 15 from hour 16 after a modest increase of hour 16 and 17 temperature set points, and further increase would actually increase the new peak because more heat will be accumulated in the system. Since the system is turned off at hour 18, heat accumulated over night pushes up the power curve the next day. Therefore, the peak of (28, 28) at hour 14 on an inflated curve ends up higher than the peak of (24.3, 24.2) at hour 16 on a curve that is almost identical to the original one. This simple example shows that maximization complicates the minimization problem structure. We need to keep this in mind as many cases of our research have peak demand as part of or the whole cost function.

The cost function surface becomes more complex when both energy and comfort are taken into consideration. Figure 3.8 shows a weighted sum of total load and PPD varies with hour 16 and 17 thermostat set points. The surface has a clear, though not smooth, trend leading to the global minimum. But there are several "dips," local optima, on the surface. The optimizer might be trapped in one of those dips if starting the search from somewhere close to the border. In any case, getting to the global minimum is not guaranteed.

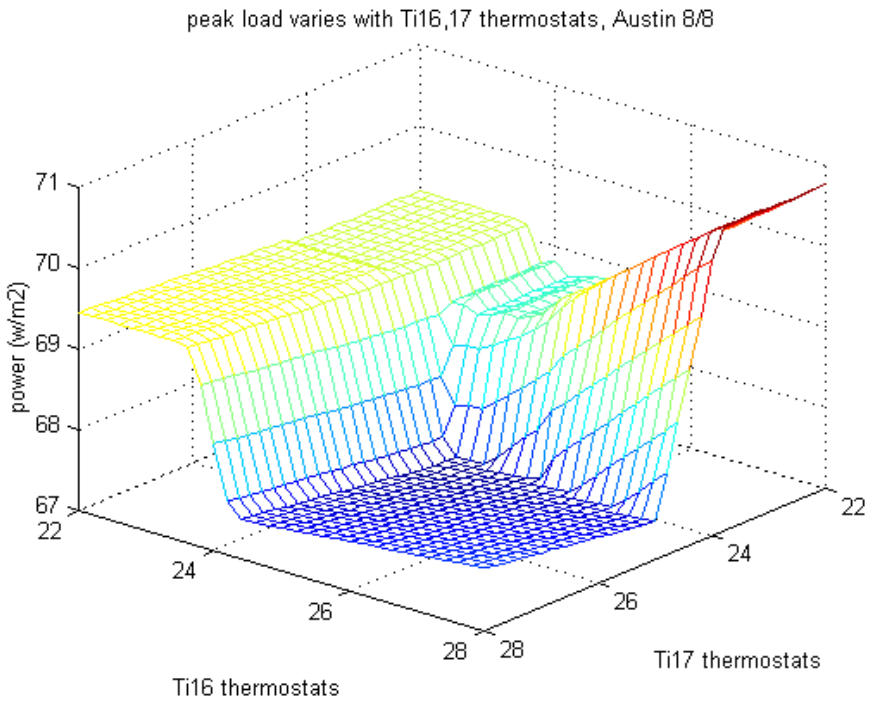
The examples we discussed here are all simple, but the point is clear that building load control problems can be discontinuous, nonlinear and have local optima. The complexity caused by these factors when the problem scales up, e.g. a ten-variable peak demand optimization problem, will become more challenging. We will illustrate how this affects the optimization results in the next two sections through two algorithms: Direct Search (DS) and Genetic Algorithms (GA). If, however, the problem is due to the existence of spurious local optima caused by simulation discontinuities, we need to refine our simulation models and try to eliminate those spurious local optima. We experienced spurious local optima at an early stage of our research. Increasing the hourly timestep in EnergyPlus simulation helps in some cases. The problem is improved in more recent versions of EnergyPlus.



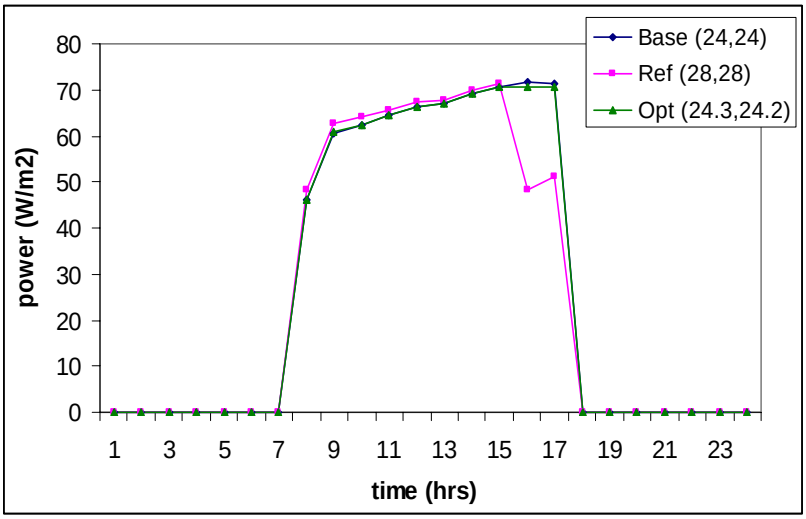
Figures 3.4 Surface of a cost function of total load with hours 16 and 17 thermostat set points as control variables varying between 22 to 28°C with 1°C temperature step



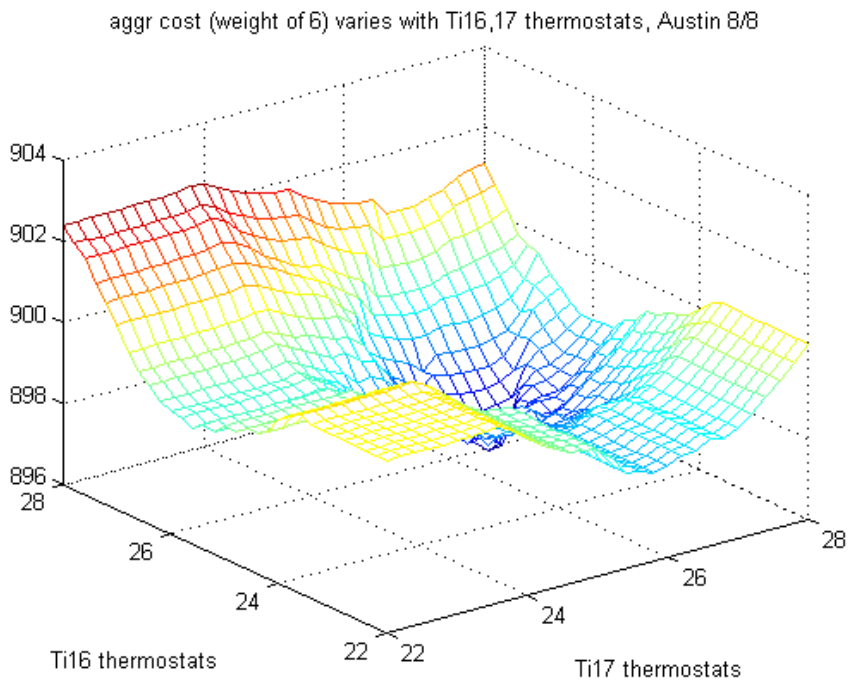
Figures 3.5 Surface of a cost function of PPD with hours 16 and 17 thermostat set points as control variables varying between 22 to 28°C with 1°C temperature step



Figures 3.6 Surface of a cost function of peak load with hours 16 and 17 thermostat set points as control variables varying between 22 to 28°C with 1°C temperature step



Figures 3.7 Power profiles of different scenarios when hours 16 and 17 thermostat set points can be varied between 22 to 28°C with 1°C temperature step



*Figures 3.8 Surface of a cost function of aggregated cost (total load + weight * PPD) varies with hr16, hr17 thermostat set points - a more complicated cost function structure with local optima*

3.4 Single Building Optimization Using Direct Search and EnergyPlus

We first study a simple case with the total energy consumption as the cost function and thermostat set points as control variables. Since we know the optimal solutions for the total load case, by doing so we aim at testing the algorithm and tuning the computation parameters. We then look at two more complicated cases with the cost function being peak demand and comfort-penalized energy consumption, respectively.

3.4.1 Simple cases with Direct Search

We continue to use the three-zone VAV system with the economizer off. Hours 14, 15, 16 and 17 thermostat set points are varied between 22°C and 28°C to minimize the total daily load without comfort constraint. We expect to see all four control variables end up at 28°C. Figure 3.9 is the GenOpt interface which displays the traces of the cost function value and control variable values over time. The downward line corresponds to the total load while other four upward lines are four temperature set points. Optimization evolves monotonically and reaches the optimum after 42 EnergyPlus runs. The Hooke-Jeeves algorithm is used in this case. The results are as expected because the total load is a monotonic

function of daytime temperature set points. The analysis and results should be different if the peak load is the cost function as the peak load has a highly nonlinear structure.

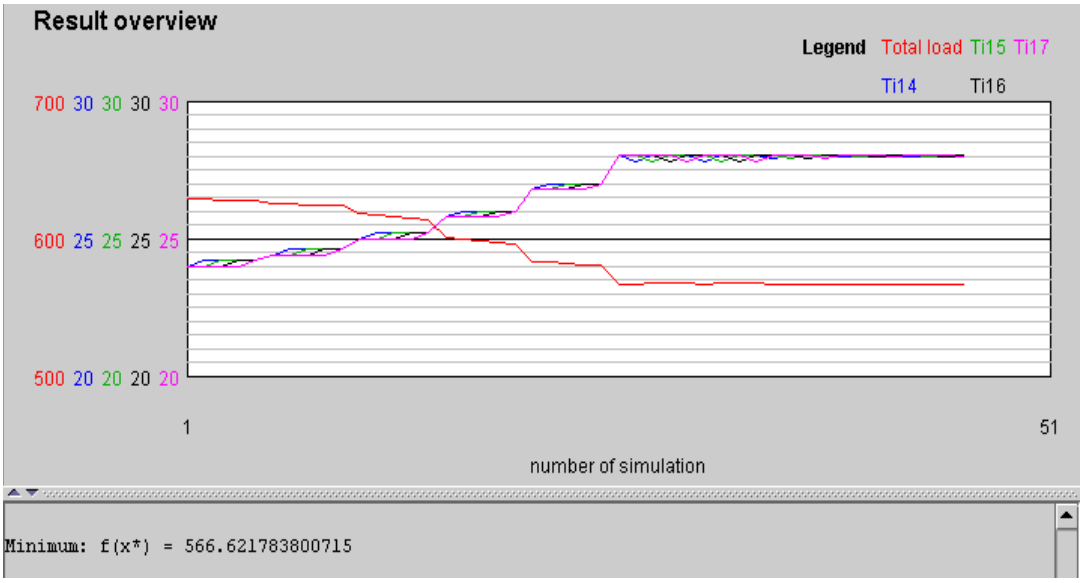


Figure 3.9 Direct Search in GenOpt optimizes total load over hours 14-17 thermostat set points between 22 and 28°C with 0.2°C time step and 24°C as initial point, without comfort constraint

As we argued previously, minimizing peak demand is a min max problem, and associated nonlinearity makes finding the global optimum difficult. With the same model used in the previous case, we optimize the daily peak demand without comfort constraints over hour 14-17 temperature set points with different initial points, shown in Table 3.1. Four cases are compared: base, reference, and two other cases as results of direct search. The reason we list the 28°C case is that without a comfort constraint, we expect setting afternoon temperatures to the upper bound will lead to good performance, which is the optimum when the total load serves as the cost function. As we see in the table, this reference case is not optimal. The peak load has shifted from hour 16 to hour 13, and the large temperature increase in the reference case pushes up the earlier hours' consumption, although the differences are small. With a search range of [22, 28], starting at 24°C is worse than starting at 26°C. However, if we narrow the search range to [24, 28], starting at 24°C performs equally well as starting at 26°C. In fact, 55.2W/m^2 is believed to be the global optimum in this case. A few things we learned from the table are:

- It is impossible for some complex problems to know the global optimum
- DSs don't handle well problems with discontinuous cost functions. The discontinuity brought in by simulation programs such as EnergyPlus makes the search stuck in local optima

- The initial point is essential in getting to the global or a good local optimum
- Search range also plays a role in whether a good sub-optimum can be found

One way to get around the dependency on initial points is to conduct optimization with different initial points and choose the best local optimum. While a global optimum is not guaranteed to be found, this way gets us a reasonably good sub-optimum and in general a satisfactory one for engineering purposes.

We continue on with the goal of understanding the cost function properties and compare the results of four cases shown in Table 3.2. These four cases correspond to a base and three different cost functions, indicated in the first column of the table. The optimal solutions and cost function values are shown in the rest of the Table. The optimization is done by the Hooke-Jeeves algorithm in GenOpt. By optimal, we mean the best sub-optimum. The control variables remain hours 14-17 temperature set points with a box range of [22, 28]. The GenOpt initialization, command and configuration files can be found in Appendix B.2 and the core VBA code in Appendix B.3. The difference in cost function is handled by the VBA post-processor.

In each case, the corresponding cost function value is highlighted in bold face. In the case with peak load as the cost function, the optimization shifts the peak from hour 16 to hour 13 over which the optimization has no control and reduces the peak from 61 to 55 W/ m^2. When minimizing the total load, optimization sets all four temperatures to the upper bound of 28°C and reduces the total load by 47 Whr/ m^2. The rate structure used to calculate the energy cost is shown in Table 3.3. The results show that the rate structure used favors minimizing the peak load. The power profiles of these cases are shown in Figure 3.10.

Table 3.1 optimize peak load with different start points through Direct Search

starting point	variable range	end optimum	peak load (W/ m^2)	E+ runs
base			61.3	
reference point		(28, 28, 28, 28)	57.1	
(24, 24, 24, 24)	[22 28]	(23.7, 24.1, 24.4, 24.3)	60.6	96
(26, 26, 26, 26)	[22 28]	(24.7, 25.0, 25.1, 25.1)	55.4	93
(24, 24, 24, 24)	[24 28]	(24.5, 24.8, 25.0, 24.9)	55.3	100

Table 3.3 Example rate structure [pge.com, 2003]

	On-peak $/kWh	Part peak $/kWh	Off peak $/kWh	Demand charge $/kW
Value ($)	0.19	0.11	0.09	6.5
Time (hours)	13-18	7-12, 19-21	22-6	N/A

Table 3.2 optimize different objective functions through Direct Search

Objective function	Ti hr14 °C	Ti hr15 °C	Ti hr16 °C	Ti hr17 °C	Peak Load W/ m^2	Total Load Whr/ m^2	Cost \$/ m^2.day		
							Total cost	Energy cost	Demand charge
Original	24.0	24.0	24.0	24.0	61.3 hr 16	520	0.479	0.081	0.398
Peak load	24.7	25.0	25.0	25.0	**55** hr 13	500	0.437	0.077	0.360
Total load	28.0	28.0	28.0	28.0	57 hr 13	**463**	0.440	0.069	0.371
Energy cost	24.7	24.8	25.0	24.9	55 hr 13	502	**0.436**	0.076	0.360

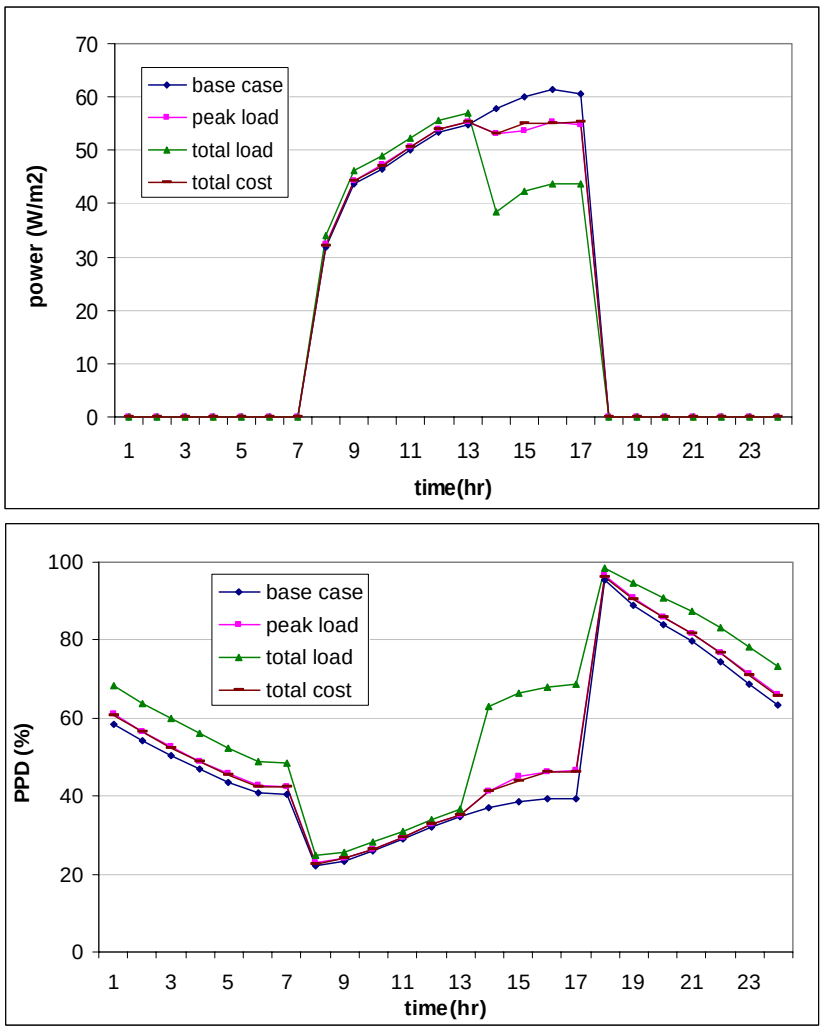


Figure 3.10 Hourly power and PPD profiles of several thermostat-based operation strategies, optimal under different cost functions set as in Table 3.2, 8/8, LA

3.4.2 Multiple Objective Optimization using Direct Search

We mentioned before that the load control problem is a multi-objective one aiming at reducing electricity consumption and maintaining a certain service level. Thermal comfort can be treated as a constraint so that an optimizer only searches within the feasible solution space. An alternative is to penalize the cost function in the way illustrated in Eqn.3.1 to take thermal comfort into consideration. Such a cost function is expected to achieve a balance between minimizing energy consumption and maximizing comfort. The importance of comfort depends on the operator or building occupants' preferences. This preference is reflected by the penalty coefficient. It helps to provide a Pareto front with varying coefficients so that we can have a better sense when it is worth sacrificing comfort because the return on load reduction is large. In this section, we take such a Pareto approach through Direct Search and study further the load control problem with hour 14-17 temperature set points as control variables.

$$Cost = Total\ Load_s + Coeff \times PPD\ values_s \qquad \text{Eqn.3.1}$$

The subscript of "s" on both total load and PPD values means they are standardized values. By standardizing the data, original magnitude has little impact, and the weighting coefficient can better represent the trade off. The optimization results are presented in Table 3.4 and the Pareto front is shown in Figure 3.11. It is clear that the competition between power and comfort has a great impact on the load shedding decisions. In reality, we can develop for a certain building a trade-off curve like this with more operating condition points on it. Building operators can decide where to be on this trade-off curve according to their expectation for energy savings and their knowledge of occupants' comfort preferences.

As the cost function surface in Figure 3.8 and the results in this section show, having comfort as part of the cost function brings more nonlinearity and likely discontinuity.

Table 3.4 Optimize trade off between energy and comfort through Direct Search

	P1	P2	P3	P4	P5
Weight Coeff.	0.1	0.25	0.375	0.5	1
Total load (Whr/ m^2)	464	480	504	515	520
Average PPD (%)	40	35	31	29	28
Thermostats in hrs 14-17 (°C)	27.4, 28, 28, 28	25.6, 26, 26.4, 27.2	24.3, 24.6, 24.8, 25.2	24, 24, 24.1, 24.6	24, 24, 24, 24

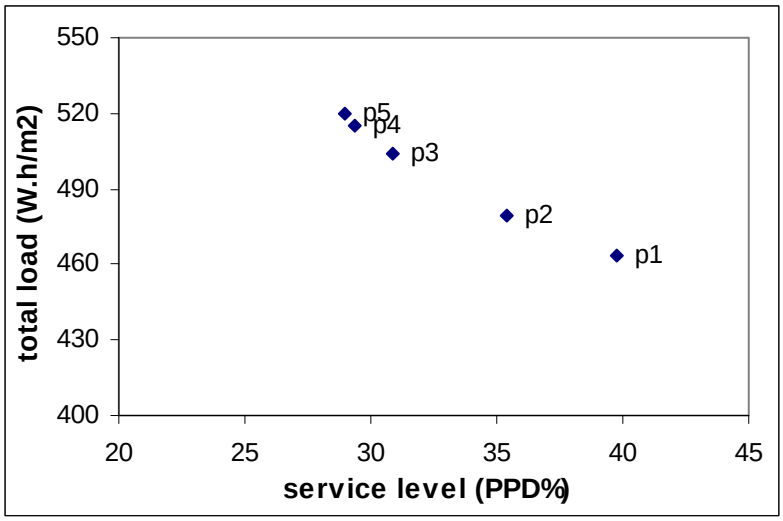


Figure 3.11 Direct Search finds multi-objective trade-off between power and thermal comfort by varying thermostat set points during hrs 14, 15, 16, and 17, 8/8, LA

3.4.3 Direct Search's difficulty with discontinuity when the number of variables is large

The number of control variables in previous cases is fairly limited. It is necessary to know how the optimizer does when the problem scale increases. For example, it might help to plan an entire day's temperature set points for an optimal profile, corresponding to 10 or 24 variables. When night cooling is available, we want to know how to control fan or chiller status, depending on whether natural ventilation or mechanical cooling is used at night, and how to control the discharge process the next morning or all temperature set points the next day to maximize the use thermal storage of the building mass. Again, we would have more than 10 control variables. We found that the Hooke-Jeeves algorithm has difficulty dealing with this – it stops at a local optimum which is very close to the starting point after a limited number of trials. We know that Hooke-Jeeves doesn't handle discontinuous cost function well. The damage of discontinuity seems to worsen when the dimension of the problem increases.

3.5 A GA-based optimizer for single-building study

A genetic algorithm is applied to the load control problem. Like DS, GA is derivative-free and must rely on simulation for function evaluation. Unlike DS, GA moves in the search space in a somewhat random fashion and therefore has a much better chance of approaching the global optimum. In fact, given enough time and appropriate parameters, GA can almost always find the global optimum, or more precisely, a very good suboptimum, as in most cases we do not know and cannot prove what the global optimum is.

In this section, we start with a test case further explaining the Matlab GA system we use. Then we look at two cases with more control variables or complex cost function structures.

The parameters for genetic operators, e.g. probability of crossover and mutation rate are determined through a trial-and-error process. The fact is that as long as these parameters are reasonably appropriate, GA can always converge and reach a good sub-optimum at least. They are adjusted to speed up the convergence. The number of total generations to run and the population size are decided based on the problem scale. For the night cooling case with five control variables, a population of 10 and a generation of 100 certainly suffice while the similar problem with eleven control variables is better done with a population of 20. The convergence speed is judged by an optimization/computation ratio, which aims at achieving the best trade-off between optimization performance and time taken.

This Matlab GA environment uses real-value coding, which means all the control variables are treated as real numbers. This is perfect for continuous variables such as temperatures but needs special treatment for discrete variables such as fan status. This will be discussed in 3.5.2 when night cooling is optimized. Note that genetic operators can be customized for the specific problems. It is worth noting that genetic operators can be customized for the specific problems.

3.5.1 Simple GA test cases

Again we vary hours 14, 15, 16 and 17 temperature set points to minimize the total daily load, a case we know the global optimum for and studied using DS in the previous section. Figure 3.12 traces the best and the average individual in each generation throughout the optimization process. A successful GA run should show visually that 1) best individual trace stabilizes and 2) the average solution trace asymptotically approaches the best solution trace, and mathematically, the individuals in the final generations are close enough in terms of fitness values. It is to be noticed that this GA is designed to maximize the objective function, so the y-axis can be understood as a constant minus power total load. Figure 3.13 explains the GA results in the EnergyPlus language by showing the power profile.

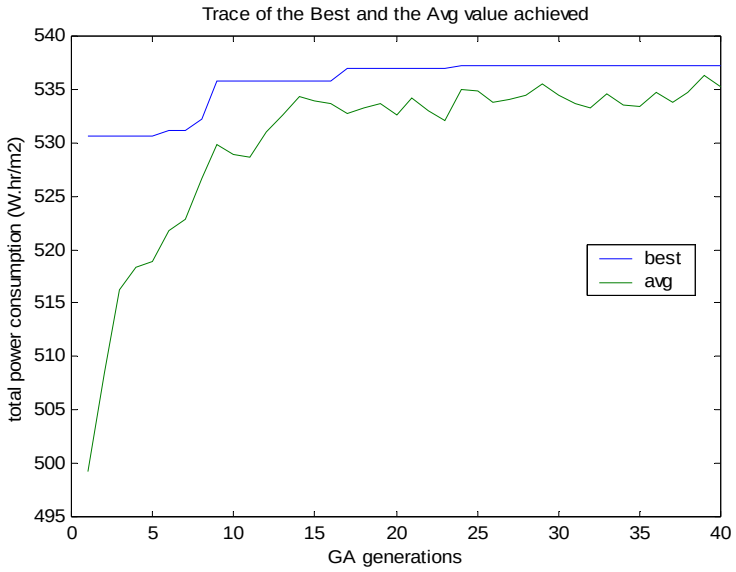


Figure 3.12 GA varies hrs 14-17 thermostat set points to minimize the total daily load. GA best and average individual traces throughout all generations

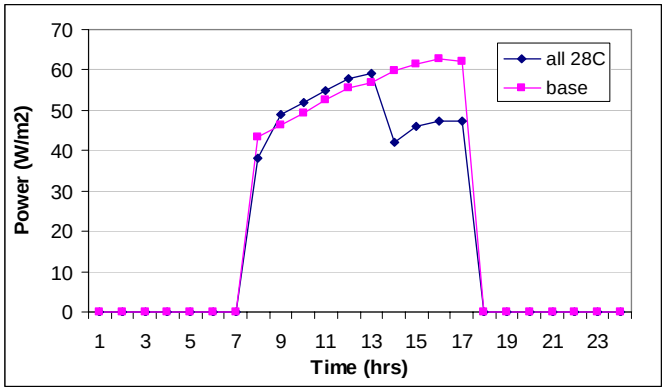


Figure 3.13 GA varies hrs 14-17 thermostat set points to minimize the total daily load. Base and optimal power profiles

3.5.2 More complicated GA cases

In this section, we look at what temperature set points should be set for the entire day so that the total electricity cost can be minimized with the specified comfort constraint. No night cooling is involved and the economizer is off. Therefore only 10 work-time temperature set points need to be optimized. The total electricity cost consists of demand change and energy cost and is calculated based on the rate structure in Table 3.3. The feasibility constraint is that PMV values have to be less than 1.5, which roughly corresponds to a PPD value of 50%. The base case, where all 10 temperatures are set to 24 °C, and the GA results are compared in Figure 3.14, where a) shows the temperature set point differences, b) power profiles, and c) PPD plots. It can be seen that the pricing information is reflected in the optimal results: the optimal GA case takes advantage of thermal mass by setting morning temperatures lower, and increases the peak time temperatures, so that the peak time power consumption is flattened as much as possible. The base case costs \$0.48/ m^2 while the optimal case \$0.44/ m^2 with the most benefit coming from the peak reduction.

Figures 3.15 and 3.16 present two night cooling cases that GA helps to improve the performance. We studied night cooling and associated scheduling issues in Chapter 2 and found that certain combination discharge processes and fan starting times perform better than others. Here, GA takes one step further and optimizes the related parameters. Two cost functions are compared: peak demand and an equally weighted sum of average and peak load (peak + total). In both cases, there are a total of five control variables: fan starting time and hour 8, 9, 10 and 11 thermostat set points. Instead of treating hourly fan status binary variables, GA takes fan starting time as a real value number varying between 6pm and 7am. It is then rounded to the closest integer before putting into EnergyPlus. Matlab GA sets the hourly fan status as on once the fan starts, and during the normal day cycle 8am to 5pm, the fan is always on. An important assumption made here, mostly for simplifying implementation purposes, is that once the fan is turned on, it will stay on until the end of the next day. This is a reasonable assumption for California weather as late nights and early mornings are colder than indoor temperature set points in most cases.

Figure 3.15 a) compares the daily temperature set points between the base case, the GA optimal case with peak load as the cost function, and the GA optimal case with a weighted sum of peak and total loads as cost function. The fan starts at 8am and all temperatures are set to 24°C in the base case; the fan starts at 6pm the day before and morning thermostat set points remain low in the GA case with peak load; and the GA case with a weighted sum cost function starts the fan at 5am and keeps morning temperatures low as well. Figure 3.15 b) shows the power profiles of these three cases. The peak load case shifts the overall peak to the early morning with a large reduction, while the peak + total case manages to reduce the peak a

little bit without incurring too much overall power consumption. Figure 3.15 bottom present the PPD profiles of these three cases. Overall, different from thermostat-set-point-based load control, night cooling improves both the energy performance and the comfort condition at the same time. The peak load case achieves a 9W/ m^2 and 14% peak load reduction, while the weight sum case has a 2W/ m^2 and 3% peak reduction, and 16Whr/ m^2 and 3% total load reduction.

Fan start time and four early morning thermostat set points are optimized in Figure 3.15. As we have seen in chapter 2 parametric studies, with night cooling available, more savings can be achieved if the day time temperatures are kept low before the peak is reached. Taking one step further, we will see how much better we can do if we allow all the day temperature set points to vary along with the night cooling start time. Put another way, how will the combination of night cooling based and thermostat set point based load control do? With exactly the same set up, we simply scale the problem by adding six more control variables for the rest of day temperature set points. We also run for two cases, peak load and the weighted sum of peak and total loads. Figures 3.16 report the results with the same set up as that in Figure 3.15. Table 3.5 summarizes three cases in terms of the peak load, total load and the cost. It can be seen that with peak load as the cost function, we achieve 14W/ m^2 peak load reduction but incur an increase in the total load; while with the weighted sum of peak and average load, we achieve reductions in both peak and total load. With the rate structure in Table 3.3 applied, we achieve \$0.1/ m^2 and \$0.02/ m^2 cost savings respectively. Compared to the savings in Figure 3.15 where only four early morning temperature set points are adjustable, Figure 3.16 achieves more savings by planning the day as a whole: 5 W/ m^2 and 8% in the peak load case and 8 W/ m^2 and 13% in the weighted sum case.

Apparently, with the entire day temperature set points changeable, the energy performance in terms of both peak load reduction and a weighted sum of peak and total consumption is improved. This matches the results from chapter 2 parametric studies that having more control flexibility improves the efficiency. Notice that we allow the late afternoon thermostats to change as well, and the system decides to set them above the base value of 24°C as it offers the direct peak reduction benefit. However, a certain amount of comfort is sacrificed, with the peak load case having 5% PPD increase and the peak + total load case on average 12% PPD increase. This is precisely what thermostat-set-point-based load control does to a system. In the mean time, night cooling still plays an essential role. With fan start time and ten temperatures, the peak load case achieves a 15W/ m^2 and 25% peak reduction, while the peak + total case corresponds to a 30Whr/m^2 and 6% total load reduction and a 10W/m^2 and 17% peak load reduction.

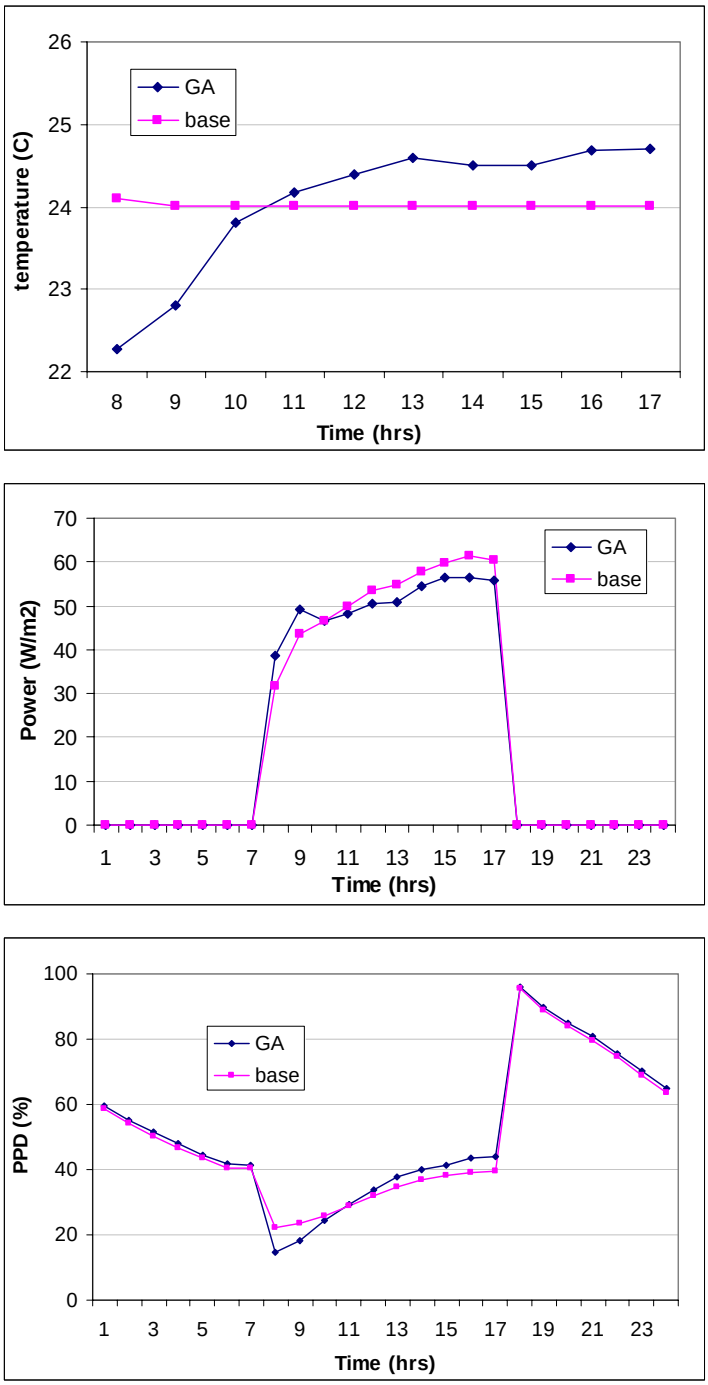


Figure 3.14 GA varies 10 work time thermostat set points to minimize the total electricity cost a) Set point profile (top), b) power profiles (middle), and c) PPD plots (bottom)

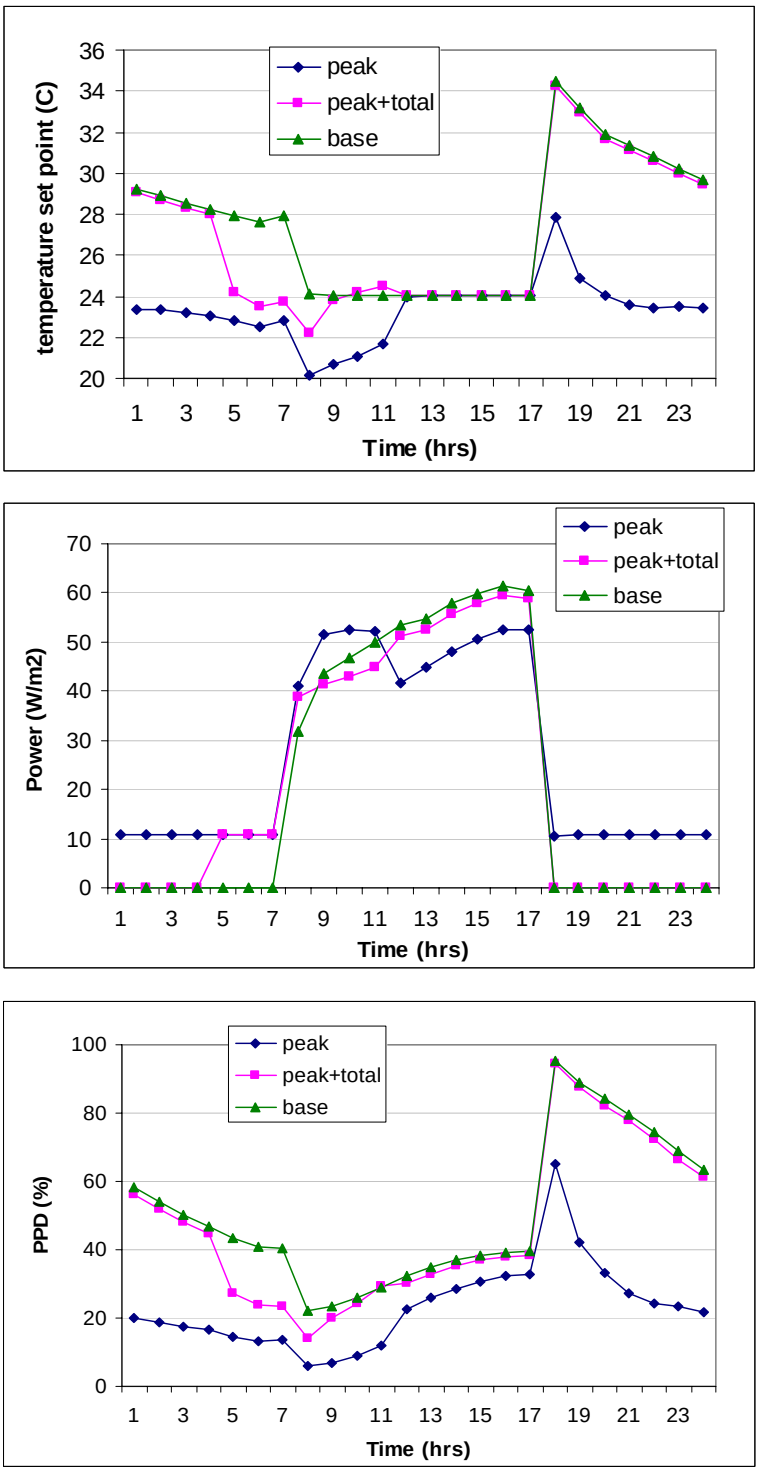


Figure 3.15 a) Thermostat set points (top), b) Power profiles (middle) and c) PPD plots (bottom) of the base case, the GA optimal case for peak demand, and the GA optical case for a weighted sum of total and peak demand, by varying fan starting time and 4 early morning thermostat set points, 8/8, LA

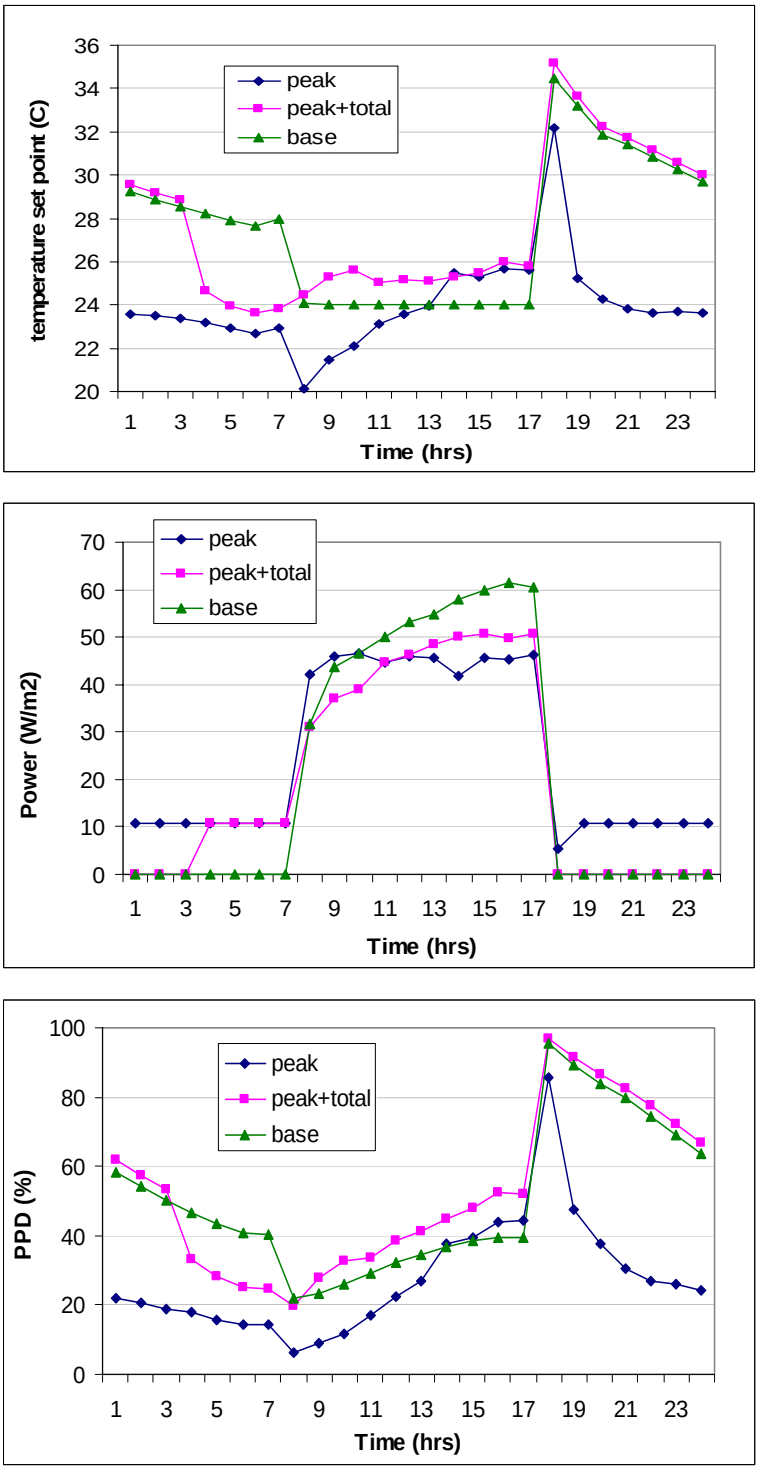


Figure 3.16 Thermostat set points (top), Power profiles (middle) and PPD plots (bottom) of base case, GA optimal case for peak demand, and GA optical case for a weighted sum of total and peak demand, by varying fan starting time and all ten day thermostat set points, 8/8, LA

Table3.5 Performance of the Matlab GA optimal results – fan-based night cooling

	base	peak	peak + total
peak load (W/ m^2)	61	47	51
total load (W/ m^2)	520	596	491
peak demand ($/$m^2$)	0.40	0.30	0.33
energy use ($/$m^2$)	0.08	0.08	0.07
total cost ($/$m^2$)	0.48	0.38	0.4

3.6 Algorithm comparison

This chapter focuses on the single building simulation-based optimization. Two derivative-free algorithms are implemented: direct search algorithms (DS) and genetic algorithms (GA). Compared to the majority of linear and nonlinear algorithms, DS and GA are both generalists in the sense that in theory they do not have favorable problem structures on which they perform particularly well. However, we still observed certain differences between these two when they are applied to building optimization problems. In Table 3.6, we highlight these differences through three simple test cases. As we can see, DS is more efficient than GA at small problems, but seems to have difficulty to make progress when the dimension increases. This is partially because the particular DS method we used, the Hooke-Jeeves algorithm, assumes continuity in the cost function, and discontinuity might be a bigger concern for simulation-based optimization at a large scale. GA is truly a generalist but pays a price of intensive computation in almost any scenario, which is especially intolerable when the problem is small.

Braun [Keeney and Braun 1996] solved an HVAC supervisory control problem with 24 variables using direct search. However, direct search algorithms were found to be less effective in dealing with 48 variables [Ren 1997]. Wright et al [2001] pointed out the GA's advantage over gradient-based and direct search algorithms is that GA's relative effectiveness increases with the size of the solution space, which matches what we found in previous sections. Wright concluded that GA is the best for the whole building optimization problem that involves a wide range of design, construction and HVAC operations parameters. In terms of when direct search starts to break down or become less effective, we believe it depends on the cost function structure, the building under study, and the specific direct search algorithm implemented. Again, there is no single optimization algorithm that works the best in all scenarios, and it takes experience and trial-and-error efforts to tune an algorithm for a specific problem. GA is more general in this sense, but it requires accurate function approximation and more computation.

3.7 A hybrid algorithm – Future work

As we have seen in previous sections, GA is capable of locating the area in which the global optimum lies, but its local search performance is not as efficient, reflected by the fast improvement of overall performance at the beginning generations and slow convergence approaching the end. This is illustrated through the asymptotic behavior of the GA optimization process, shown in Figure 3.12. In addition, we observed that direct search tends to get stuck at points close to initial points but still offers modest improvement. A natural thought is to combine GA with a local search algorithm, such as Hooke-Jeeves used in this chapter – taking advantage of the GA fast search for the optimal sub-area but avoiding its slow convergence when the search space becomes small, and taking advantage of the local algorithm's efficiency for locating a local optimum but avoiding its lack of sense for big directions. Due to time constraints, we did not implement this idea, but think it is an interesting problem and worth exploring in future work.

Table3.6 Comparison of GA and DS

Cases	E+ simulation runs and Results	
	DS (Direct Search: Hooke-Jeeves)	GA (Genetic Algorithm)
Test case 1 Peak load varies with hr 16, 17 thermostat set points without comfort constraint	Computation complexity depends on initial points. Takes 30 – 40 E+ runs. Converges to global minimum (24.3, 24.2)	Takes 20 generations and a population of 10, a total of 200 E+ runs. Converges to global optimum (24.3, 24.2)
Test case 2 Total load varies with hr 14-17 thermostat set points without comfort constraint	Takes on average 60 E+ runs. Converges to the global optimum (28,28,28,28)	Takes 40 generations and a population of 10, a total of 400 E+ runs Converges to the optimum (28,28,28,28)
Test case 3 Total load varies with 10 working time thermostats	Gets stuck at points very close to starting points Mostly cannot deal with 10 variables and/or problems with complicated cost functions	Takes 60 generations and a population of 20, a total of 1200 E+ runs Gets to a good point (24, 26, 28, 28, 28, 28 , 28, 28, 28, 28)
Summary and Comparison	DS is efficient when the problem size is small. It fails to deal with discontinuity at higher dimensions	As a generalized method, GA can make progress in almost any scenario, but is computationally intensive, especially with small-scale problems

CHAPTER FOUR
SIMULATION-BASED MULTI-BUILDING OPTIMIZATION

Our scope in previous two chapters is limited to single buildings. In this chapter and the next, we will look at the load control problem involving more than one building, discuss the essential difference that having multiple participants in the system brings to analysis, and propose three different approaches to tackling the multi-building problem. Two simulation-based methods will be discussed in this chapter: engineering-rule based enumeration and a multiple-GA. Throughout this chapter, we illustrate the aggregation concept through two-building or three-building examples and learn from simple cases before scaling up the problem. In addition, as we will discuss late in this chapter, there are computation concerns of applying these two methods to large-scale multi-building problems. A less scale-sensitive approach will be discussed in the next chapter.

Section 4.1 looks at an enumeration approach, through which the number of feasible EnergyPlus simulations for each aggregation participant is reduced. The optimal combination of a two-building problem is found by matching two sets of feasible solutions in Matlab. Section 4.2 takes a GA approach with the chromosome consisting of control variables from all aggregation participants. EnergyPlus simulations are run sequentially for all the participants, and the GA evaluation is done at the end of the simulation. Section 4.3 compares the two simulation-based approaches, discusses their limitations and points out the need of further research on a more efficient method for the multiple building problem. Section 4.4 takes a break from the main road of developing computationally efficient aggregation methods and looks at how the aggregation decisions vary with the size of the aggregation pool at a small scale.

4.1 Smart Enumeration - A Rule-based Engineering Approach

Reddy and Norford [2002] discussed load aggregation through a portfolio optimization type of approach. Four different building profiles, office, retail, grocery and school, are combined exhaustively to find the best aggregation effect. The diurnal load profiles are generated using existing data. No load controls are applied to any of the participants. Their research examines which buildings are more appropriate to participate in the aggregation, and the appropriateness is a function of both individual building systems and correlations between participants. We take one step further in our research and look at how to make a group of existing buildings more appropriate for aggregation by applying load control to each of them.

In this section, we take a trial-and-error approach based on a reasonably sized trial set. Instead of solving directly the optimization problem, Eqn.1.1, we first run for each building a number of simulations that cover a wide range of load control possibilities, then match the individual results and search exhaustively for the best one with a specified cost function. It is impossible to enumerate all load control possibilities, and we make simplifications according to the load control strategies used.

Following previous chapters, two types of controls are studied. In Section 4.1.1, thermostat set points in the afternoon hours are increased in each individual building to achieve an overall "optimized" power profile. Night cooling schedules are the control variables in Section 4.1.2. Both cases study simple two-building or three-building cases.

We reduce the number of simulations partially by narrowing down the scope of the questions studied. With the thermostat control, one simplification comes from the fact that we are dealing with commercial buildings in summer time. The peak takes place somewhere between 1pm and 6pm. Therefore, for short-term load control with little emphasis on thermal mass, we only need to focus on the afternoon instead of a 24-hour horizon, which greatly reduces the number of control variables. With the night cooling control strategy where thermal mass plays an important role, we have one more control variable which is the night cooling start time.

4.2 Set thermostat set points for multiple buildings by smart enumeration

4.2.1 EnergyPlus models and expert rules

Three types of building models are used through out the multi-building studies. All three building models are derived from the base VAV model introduced in Chapter 2, with the thermal mass and/or the load pattern changed. Table 4.1 summarizes the main differences between these models, and a detailed description about the base model can be found in the Appendix A.1. The key inputs, including material thermal properties, supervisory control schedules and load patterns for models in Table4.1 are given in Appendix C.1.

For each of three models, we run extensive E+ simulations on the summer day to which the load aggregation is applied. Thermostats between 1pm and 5pm are allowed to vary. Temperature set points have a maximum hourly change of 3°C and a maximum change period of four hours based on possible schedules: one-hour thermostat change, two-hour, three-hour and four-hour changes including both consecutive cases and separate ones. Therefore, we have for each building a total of

$$1+3\times\left(C_5^1+C_5^2+C_5^3+C_5^4\right)=1+3\times\left(5+10+10+5\right)=91$$

simulation cases where each of the three integer temperature increase corresponds to 30 cases and the base case involves no change. Notice the big assumption we made: the temperature increase is the same across the load control period regardless of the schedule. Doing so is only to make the analysis simple, and we understand that it would lead to a sub-optimal solution. The trade-off between accuracy and computation will be addressed in Section 4.3 when Enumeration and multiple-GA are compared.

Table 4.1 Building model types for aggregation studies

E+ models	Thermal mass[5]	West wall	Load pattern
Base model	original mass	No window	original: peak in late PM
Model 1	½ mass	6 m^2 window	original: peak in late PM
Model 2	½ mass	6 m^2 window	new: peak in early PM
Model 3	2 mass	No window	original: peak in late PM

All the simulations are done automatically through a data processing engine written the Visual Basic Application (VBA) for Excel. For each simulation, the hourly electricity consumptions, PMV and PPD values are computed, processed by the VBA code with certain rate structure applied, and results are output to a matrix which is further processed in Matlab to search exhaustively the combination with the best cost function value. A simple Matlab code to do the search can be found in Appendix C.2. We present the graphical results in the following section.

To summarize what we have done in reducing computation efforts in the following pre-defined expert rules:

- Run simulations with integer temperature increases and those integers are a good discrete represent of the search space, e.g. 1, 2 and 3°C in this case
- All the hours involved in load shedding experience the same amount of temperature change
- Starting the enumeration from the smallest temperature increase and gradually approach the upper bound. If a 1°C thermostat setting point increase violates thermal comfort at a certain hour, stop searching beyond 1°C at this hour, and the same to 2°C and 3°C increase
- If a short-period shedding scheme violates comfort at certain hours, no need to search a long-period scheme with the same amount of temperature increase at the same hours

[5] The definitions for 1/2mass and 2mass are the same as those in Chapter 2 – the densities of main construction materials for walls, floor and roof are half and double of what it originally is for ½ and 2masss respectively.

- No need to consider a short-term load shedding scheme if a long-term one includes it at the beginning and hours after remain below the base
- For night cooling, instead of enumerating the possibilities that night cooling could start any hour, we simulate those starting at 6pm, 8pm, 10pm, midnight, 2am, 4am, and 6am only. Our research results will show that the computation savings is worth the minor loss of optimality

By violating comfort, we mean the specified PMV or PPD is not satisfied. Here we adopt a single standard of comfort and keep the working-period PMV values below 1.5. All these rules are implemented in the VBA engine in Excel to check automatically a solution's feasibility. The size of feasible set is reduced by 64% for buildings 1 and 2 and by 17% for buildings 1 and 3 with thermostats-based strategies. Refer to Appendix C.3 for the VBA code and the detailed savings.

4.2.2 Matching results for thermostat set point-based load control

1) Peak demand

Figure 4.1 shows individual load profiles for three models before aggregation, each of which has all thermostats set at 24°C. Corresponding to the building description in Table 4.1, models 1 and 2 have lower thermal mass and higher peak load in late afternoon around 4pm and around 1 pm respectively. Model 3 has the highest thermal mass and the same load pattern as in model 1, and its load curve is rather flat due to the high thermal mass. The feasible solutions for each of these three are computed. Based on the expert rules proposed in Section4.1.1, model 1 and 2 end up only having 32 feasible solutions, which corresponds to a 65% computation savings, and model 3 has 75 feasible solutions and a 17% savings. The difference is caused by thermal mass: in building 3, a wide range of temperature increases are feasible while in buildings 1 and 2 higher increases violate the comfort constraint. The optimal solution for each pair is found through exhaustive matching. Figures 4.2 to 4.4 are the "optimal" match results for buildings 1 and 2, 2 and 3, and 1 and 3 respectively, while Figure 4.5 illustrates the match results for buildings 1, 2 and 3. In Figures 4.2 to 4.5, peak demand is minimized.

In Figure 4.2, the peaks for models 1 and 2 are comparable but they happen at different times, so that the original aggregated load, represented by the solid line on the top of the figure, fairly flat for most of the afternoon hours. In the "optimal" match, building 1 lifts temperature set points at hours 13, 14, 15, and 16 by 1°C and building 2 increases those at hours 14, 15, 16 and 17 by 1°C. This change leads to a peak load reduction of 5.2%, but the peak remains at hour 13 as any tempt to shed hour 13 load would lead to an even higher peak load overall. The total load profile is flattened, which makes sense. The high peak of building 2 at hour 13 is compensated by the reduction of building 1 at this hour, while the load setback

recovery at hour 17 by building 1 is compensated by load shedding of building 2 at the same time. If we look at two buildings separately, they both incur a higher peak after aggregation, however, the total peak is reduced. This is the essence of the aggregation load control: by communicating with each other in the aggregation pool, individuals take actions that favor a centralized goal. The amount of reduction is a function of weather, rate structures and specific buildings under study.

Building 3 by itself needs little load shift because its high thermal mass helps maintain the afternoon load profile flat. However, it is no longer the case when building 3 and 1 are to be aggregated, shown in Figure 4.3. Building 3 chooses to increase thermostats at hours 14, 15, 16 and 17 by 2°C to help cut the aggregated peak, and building 1 pushes up thermostats at hours 13, 15, 16 and 17 by 1°C accordingly. This case enjoys a 10.4% peak reduction with the peak shifted from hour 16 to 12. This is a very good example of how multiple buildings can collaborate to achieve an overall performance increase that cannot be achieved by individuals alone.

Building 3 also plays a complimentary role in Figure 4.4. Building 2 peaks at hour 13, so building 3 increases hour 13 and 14 temperature set points by 1°C while building 2 pushes up hours 13, 15 and 16 by 1°C. After both buildings cut as much as possible, hour 13 is no longer the aggregated peak, and the small setback recovery of building 3 at hour 15 and 16 is cancelled out by the savings of building 2. Therefore, the new peak is shifted to hour 12 with 4.8% reduction.

For all three pairs, individuals done worse or just equally well after aggregation than before, but the aggregated performance is improved. Again, this is because two buildings communicate with each other during the matching process and decide jointly to shift and/or shed the total peak load.

Aggregation with three buildings is shown in Figure 4.5, where building 1 increases thermostat set points at hours 14, 15, 16 and 17 by 1°C, building 2 at hours 13, 14, 15 and 16 by 1°C, and building 3 at hours 13, 15, 16, 17 by 1°C. This leads to a 6.3% peak reduction and the peak is shifted from hour 14 to hour 12. Recall that our criterion for admitting any load control strategy into the feasible set is that PMV less than 1.5, which corresponds to PPD[6] below 50%. Figure 4.6 shows the PPD profiles for the three-building case. We consider this less comfortable but acceptable. Table 4.2 summarizes three two-

[6] The PMV and PPD values provided by the current EnergyPlus version are generally high. In the base case with indoor air temperatures at 24C, the average PMV is 0.8 and PPD is 20%. Therefore, PPD below 50% is not a bad service level requirement.

building cases and one three-building case. In most cases, the new peak at 12pm is a consequence of applying load control to hours 13-17.

Table 4.2 Summary of simple aggregation cases

Building group	original peak load (W/ m^2)	peak time	new peak load (W/ m^2)	peak time	peak load reduction
1 and 2	135	13	128	13	5.2%
1 and 3	125	16	112	12	10.4%
2 and 3	125	13	119	12	4.8%
1, 2 and 3	190	14	178	12	6.3%

We have seen the potential of peak reduction by applying load control to the aggregation pool. It is interesting to see how much of this reduction comes from load control and how much from aggregation. For each of the four cases studied before, we compare the aggregated load control results in Table 4.2 with the results from individual load control. In individual load control, each building minimizes its own peak before aggregation and there is no communication between participants in the pool. Figure 4.7 shows for buildings 1 and 2, the aggregated load control achieves 2.4% more peak load reduction than the individual load control does. Notice that in individual load control, both 1 and 2 choose to increase the temperature set point at their own peak hours. In aggregated load control, building 1 increases the temperature set point at building 2's peak hour of 1pm while building 2 does the same at building 1's peak hour of 4pm. It is easier to shed load during non-peak hours, which explains why the aggregated load control achieves more peak reduction as a result of communication between aggregation participants. Figure 4.8 combines Figures 4.7 and 4.2: there is a 2.8% peak reduction from the base case to the individual load control, and a 2.4% reduction from the individual load control to the aggregated load control. Table 4.3 summarizes this comparison for all three two-building cases and one three-building case. It can be seen that the contribution by aggregation is significant and ranges from 30 to 50% in these small-scale cases.

When two buildings are identical, aggregation might not help as much. Figure 4.9 aggregates two identical type-1 buildings and Figure 4.10 two identical type-2 buildings. For two type-2 buildings, the aggregated load control is identical to the individual load control; while for two type-1 buildings, the peak reduction difference is only 0.3%. This is due to the lack of diversification in load profiles and limited choices in load shedding strategies.

Table 4.3 Individual load control vs. aggregated load control

	Bldg 1 and 2	Bldg 1 and 3	Bldg 2 and 3	Bldg1,2,3
original total peak load (W/m^2)	135	125	125	190
Total peak after individuals minimize their peak load (W/m^2)	131	116	122	184
total peak load with aggregation (W/m^2)	128	112	119	178
Peak load reduction from individual load shedding	2.8%	7.5%	2.8%	3.4%
Peak load reduction from load-control-enabled aggregation	5.2%	10.5%	5%	6.1%
Contribution of aggregation to the total peak load reduction	46%	29%	44%	44%

It is to be noticed that although we minimized the peak energy consumption above, the methodology would remain the same if the cost function is the total energy cost. We only need to apply a rate structure to the hourly energy consumption to convert the peak demand problem to a cost-based one. Throughout this research, we are energy cost-oriented with the recognition that peak demand charge plays an important role in cost calculation. The majority of our research focuses on the peak demand only, which is meant to simplify the problem so we can get more insight into the nature of the aggregation. As we will argue in the next chapter, minimizing peak demand captures the essential mathematical structure of the problem. The next section looks at the same problem with total energy cost as the cost function aiming to illustrate the difference in optimal control strategies caused by a cost-based cost function.

2) Rate structure

Based on the information at the PG&E website [pge.com, 2003], we quote the rate structure listed in Table4.3 for the small commercial building type in California. We apply the rate structure to the feasible individual simulation results obtained in the previous section and rerun the matching process. We present in Figures 4.11 to 4.14 three two-building cases and one three-building cases under this rate structure, as a comparison with peak load oriented counterparts in Figures 4.2-4.5, to learn how rate structures change the analysis results.

To explain each of Figures 4.11-4.14 the same way we did to Figures 4.2-4.5, the reasoning would be more subtle as the cooperation between different buildings is now determined not only by the load profiles, but by variations in the rate structure. The operations details will not be analyzed here, but we want to emphasize the differences 1) brought in by considering hourly energy cost on the top of demand charge and 2) caused by different demand charges. There are two figures in each of Figures 4.11-4.14 where the top one corresponds to a demand charge of $6.5 /kW and the bottom one an extreme case without demand charge. A general trend is that aggregated power profiles tend to be flat when the peak demand is $6.5/kW; they tend to have dramatic variation so that the area below the profile, which represents the total energy use, is minimized when there is no peak demand. Table 4.5 gives the summary statistics of the load aggregation performance with the rate structure specified in Table 4.4. The savings are comparable to those in Table 4.2, but the actions taken by these participants are different, depending on the demand charge applied. For example, for the no demand charge cases in Figures 4.11 to 4.14, the aggregated peak load is not reduced much, and even increased such as in Figure 4.13. A demand charge of $6.5/kW leads to similar results as those in the peak load case, which means that a rate structure with a $6.5/kW encourages peak reduction.

Table 4.6 presents the cost-based aggregation results in the same way as Table 4.3 did to the peak-demand- sum of individual costs. It compares four cases: the sum of individual costs, the cost of aggregated load without load control, the sum of individual costs with individual load control, and the cost of aggregated load with aggregated load control results. Notice that the reference case is the sum of individual costs in Table 4.6, which says simply adding two buildings together would do better than calculating cost separately. The case of with buildings 1 and 3 does not see improvement by purely aggregating loads because two peaks are at the same time. Pure aggregation brings 0-4% cost savings while individual load control brings in 5-6% savings. Aggregated load control performs the best and achieves 7-10% savings. Figure 4.15 illustrates the cost savings increases.

Table 4.4 Example rate structure

	On-peak $/kWh	Part peak $/kWh	Off peak $/kWh	Demand charge $/kW
Value ($)	0.19	0.11	0.09	6.5
Time (hours)	13-18	7-12, 19-21	22-6	N/A

Table 4.5 Summary of simple aggregation cases, cost-based

Part I: demand charge = $6.5/kW

Building group	Original total cost ($/m^2)	Ori. peak load (W/ m^2) / time	New total cost ($/m^2)	New peak load ($/m^2) / time	Total cost reduction
1 and 2	1.06	135 / hr 13	1.00	128 / hr 13	5.0%
1 and 3	0.98	125 / hr 16	0.88	112 / hr 12	9.7%
2 and 3	0.98	125 / hr 13	0.94	119 / hr 12	4.6%
1,2 and 3	1.49	190 / hr 15	1.41	178 / hr 12	5.7%

Part II: demand charge = $0/kW

Building group	Original total cost ($/m^2)	Ori. peak load (W/ m^2) / time	New total cost ($/m^2)	New peak load ($/m^2) / time	Total cost reduction
1 and 2	0.18	135 / hr 13	0.17	130 / hr 16	4.4%
1 and 3	0.17	125 / hr 16	0.15	114 / hr 16	7.3%
2 and 3	0.17	125 / hr 13	0.16	128 / hr 13	7.1%
1,2 and 3	0.26	190 / hr 15	0.24	182 / hr 13	6.2%

Table 4.6 Individual load control vs. aggregation and load control, cost-based

	Bldg 1 and 2	Bldg 1 and 3	Bldg 2 and 3	Bldg1,2,3
sum of individual costs ($/m^2)	1.10	0.98	1.01	1.55
cost of aggregated load without load control ($/m^2)	1.06	0.98	0.98	1.49
sum of individual costs with individual load control ($/m^2)	1.04	0.92	0.96	1.46
cost of aggregated load with aggregated load control ($/m^2)	1.00	0.88	0.94	1.41
cost reduction from pure aggregation	3.6%	0%	3%	3.9%
cost reduction from individual load control (peak-reduction-based control)	5.5%	6.1%	5%	5.8%
cost reduction from aggregated load control	9.1%	10.2	6.9%	9%

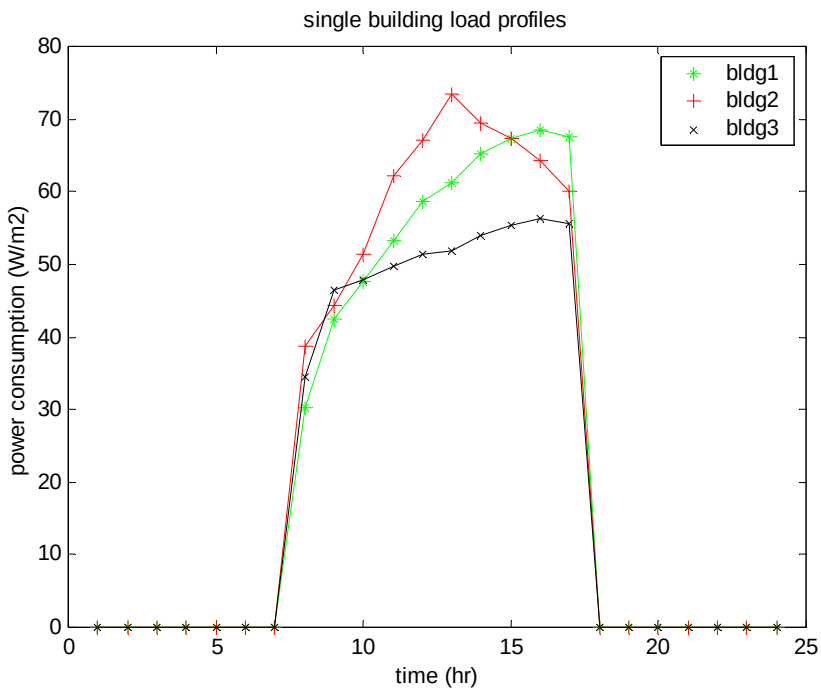


Figure 4.1 Individual load profiles of buildings 1, 2 and 3

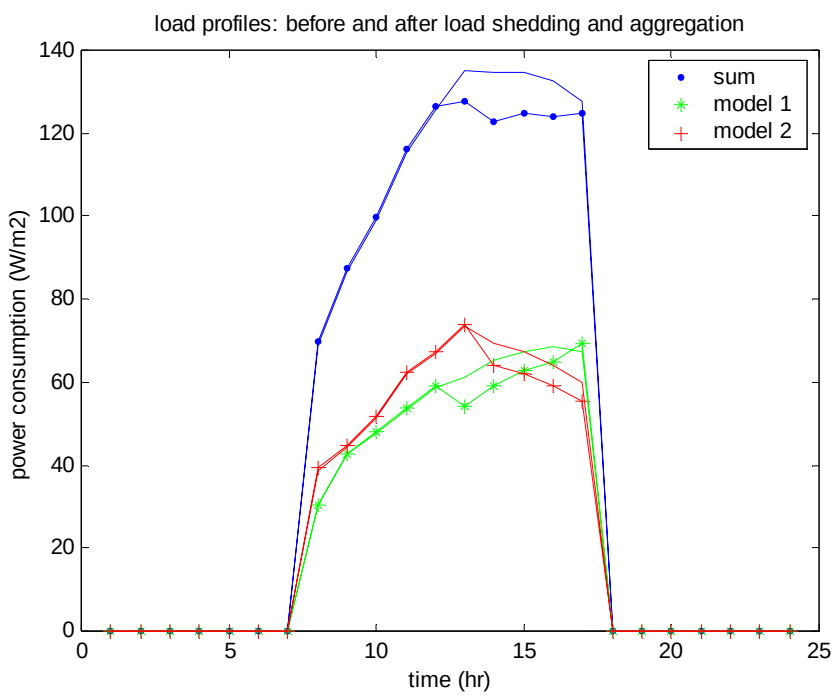


Figure 4.2 "Optimal" load aggregation between buildings 1 and 2

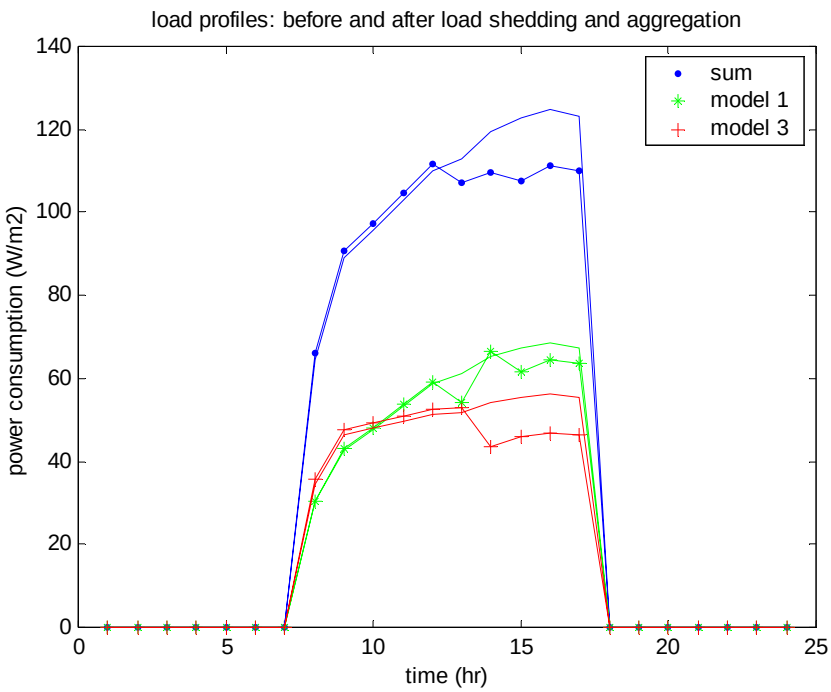


Figure 4.3 "Optimal" load aggregation between buildings 1 and 3

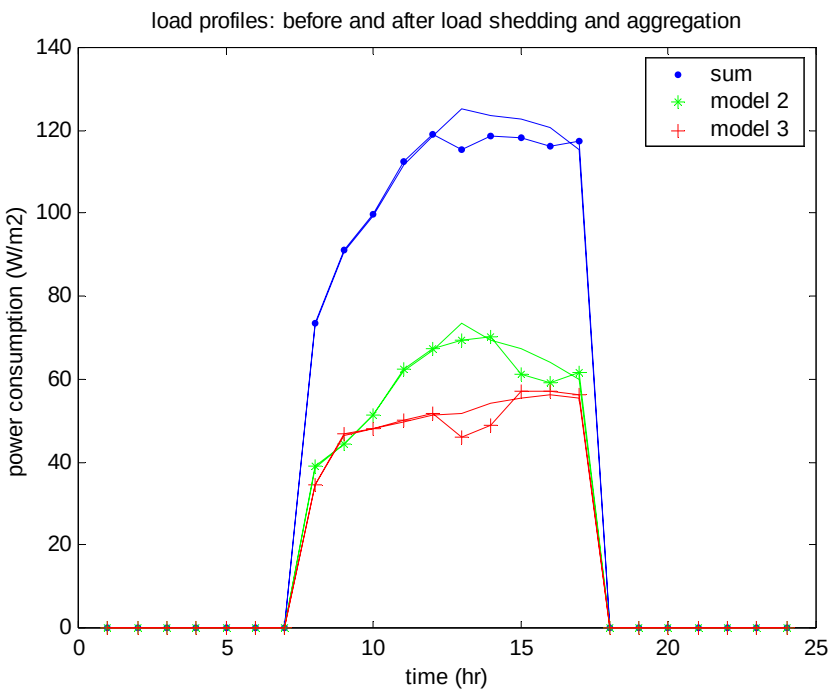


Figure 4.4 "Optimal" load aggregation between buildings 2 and 3

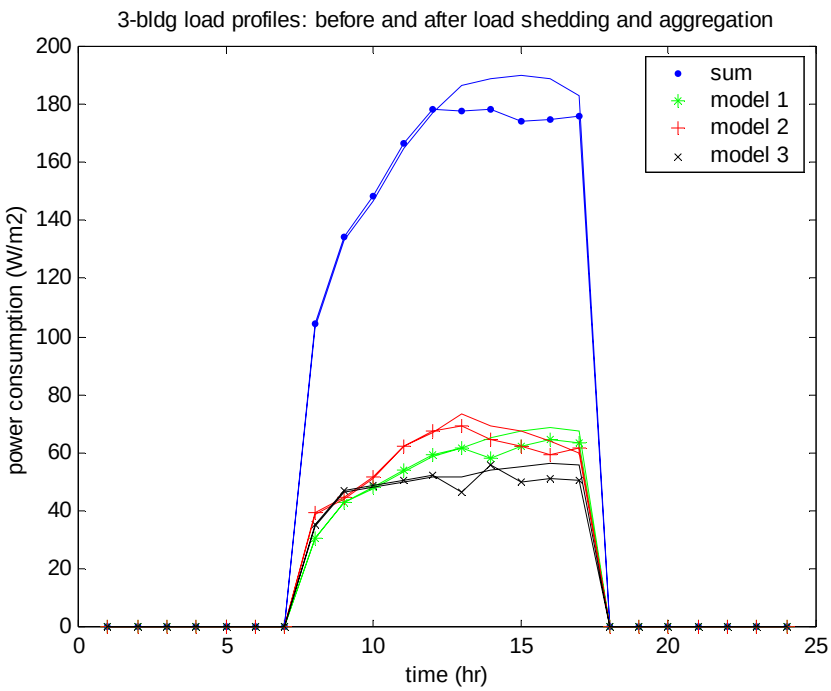


Figure 4.5 "Optimal" load aggregation between buildings 1, 2 and 3

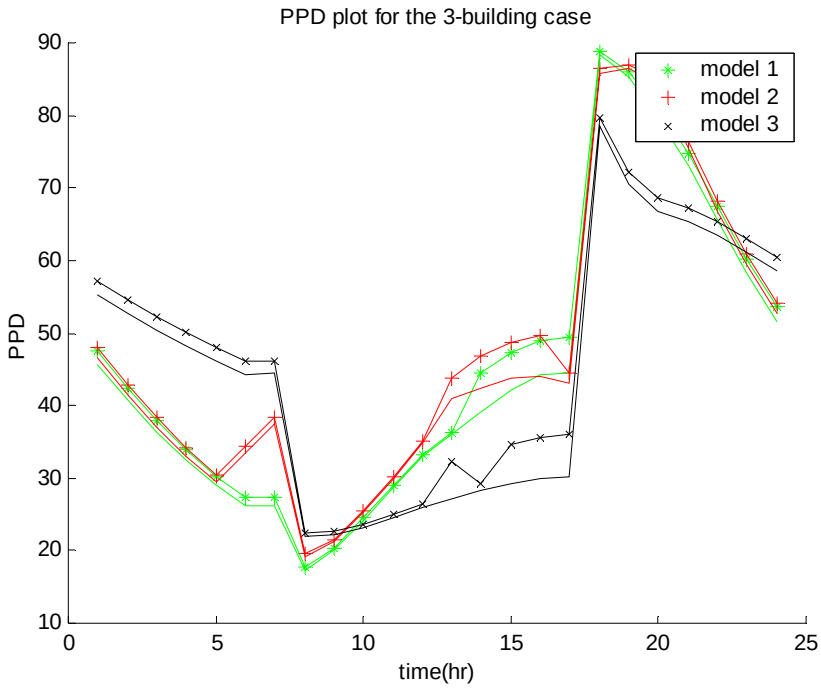


Figure 4.6 PPD profiles corresponding to three-building optimal" load aggregation

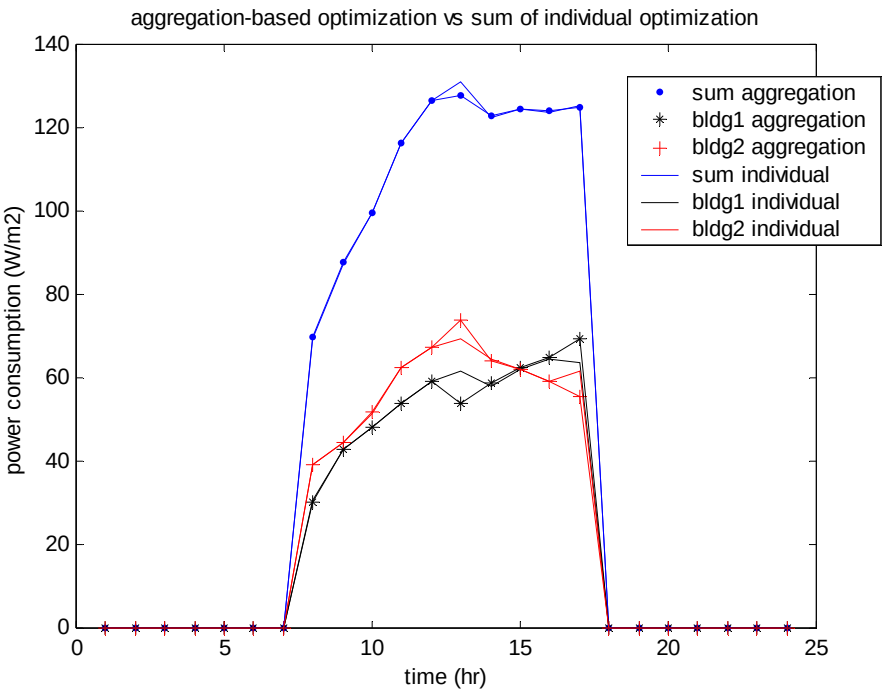


Figure 4.7 Comparisons between aggregated load control and individual load control, buildings 1 and 2

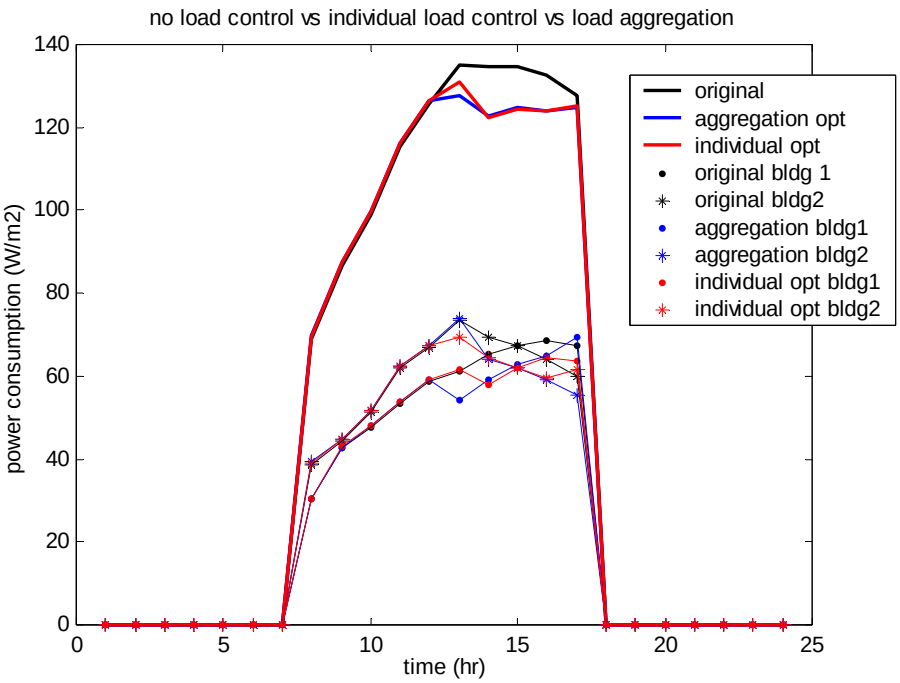


Figure 4.8 Comparisons between the base case, individual load control and aggregated load control, buildings 1and 2

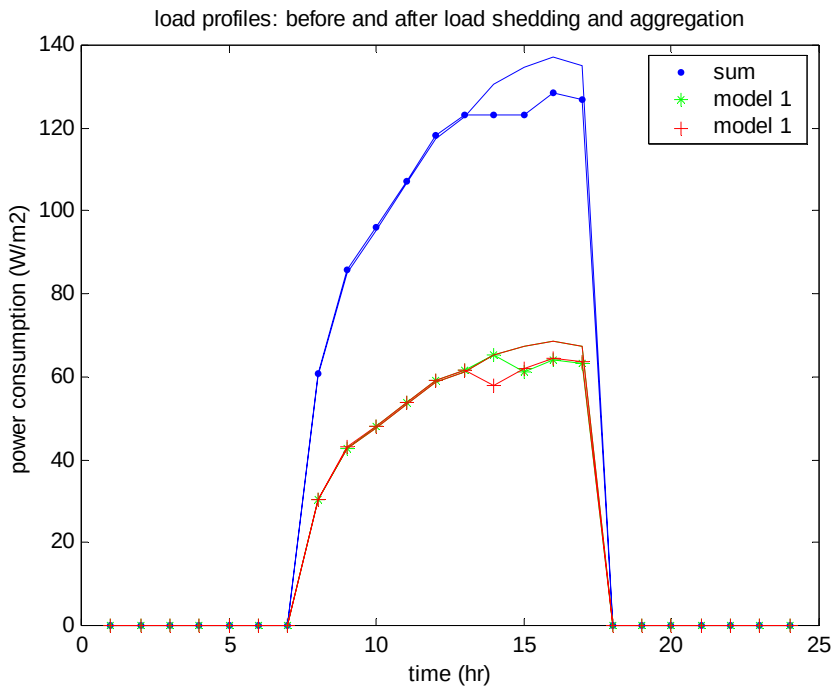


Figure 4.9 "Optimal" load aggregation between buildings 1and 1

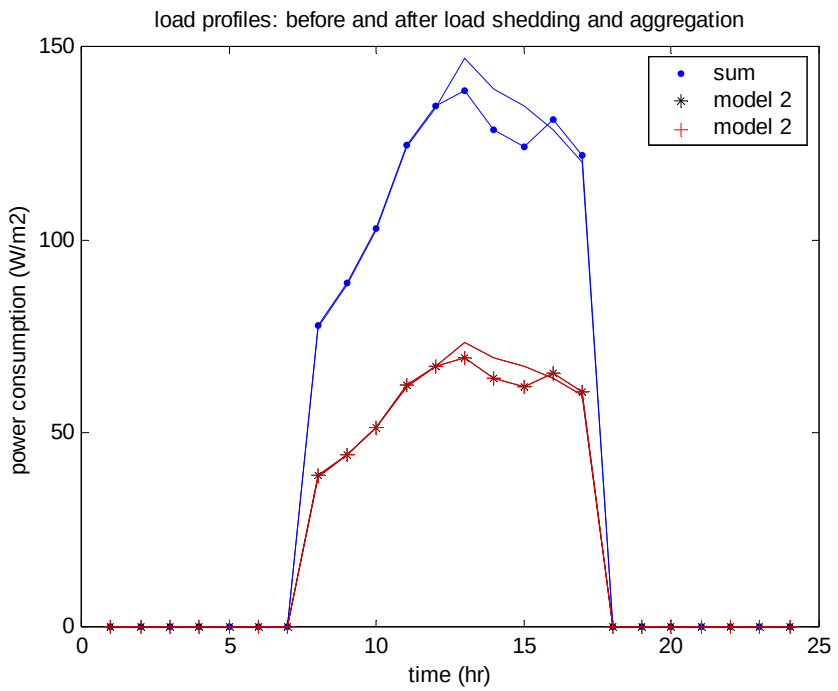


Figure 4.10 "Optimal" load aggregation between buildings 2 and 2

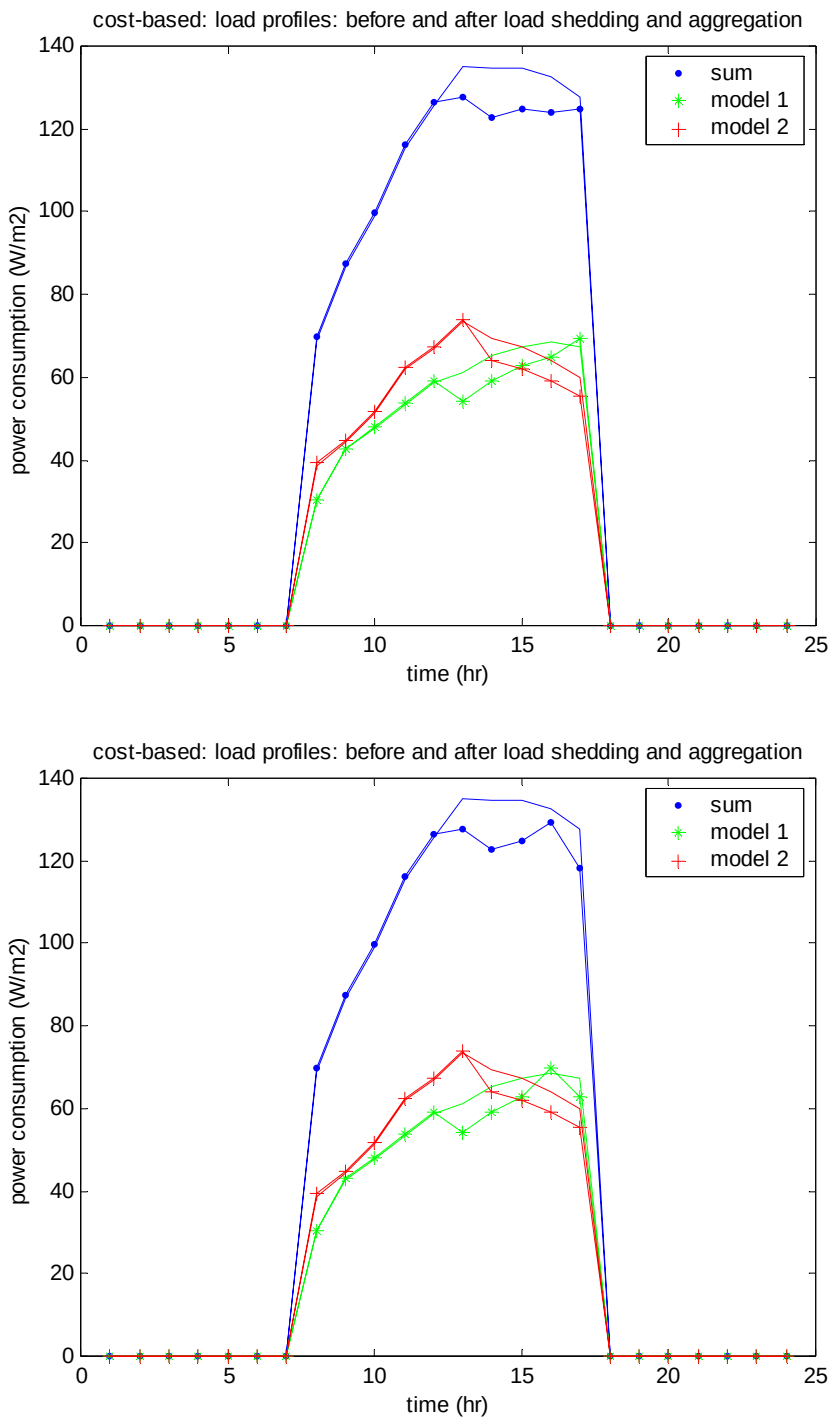


Figure 4.11 "Optimal" load aggregation between buildings 1 and 2, cost-based
Top: demand charge $6.5/kW. Bottom: no demand charge

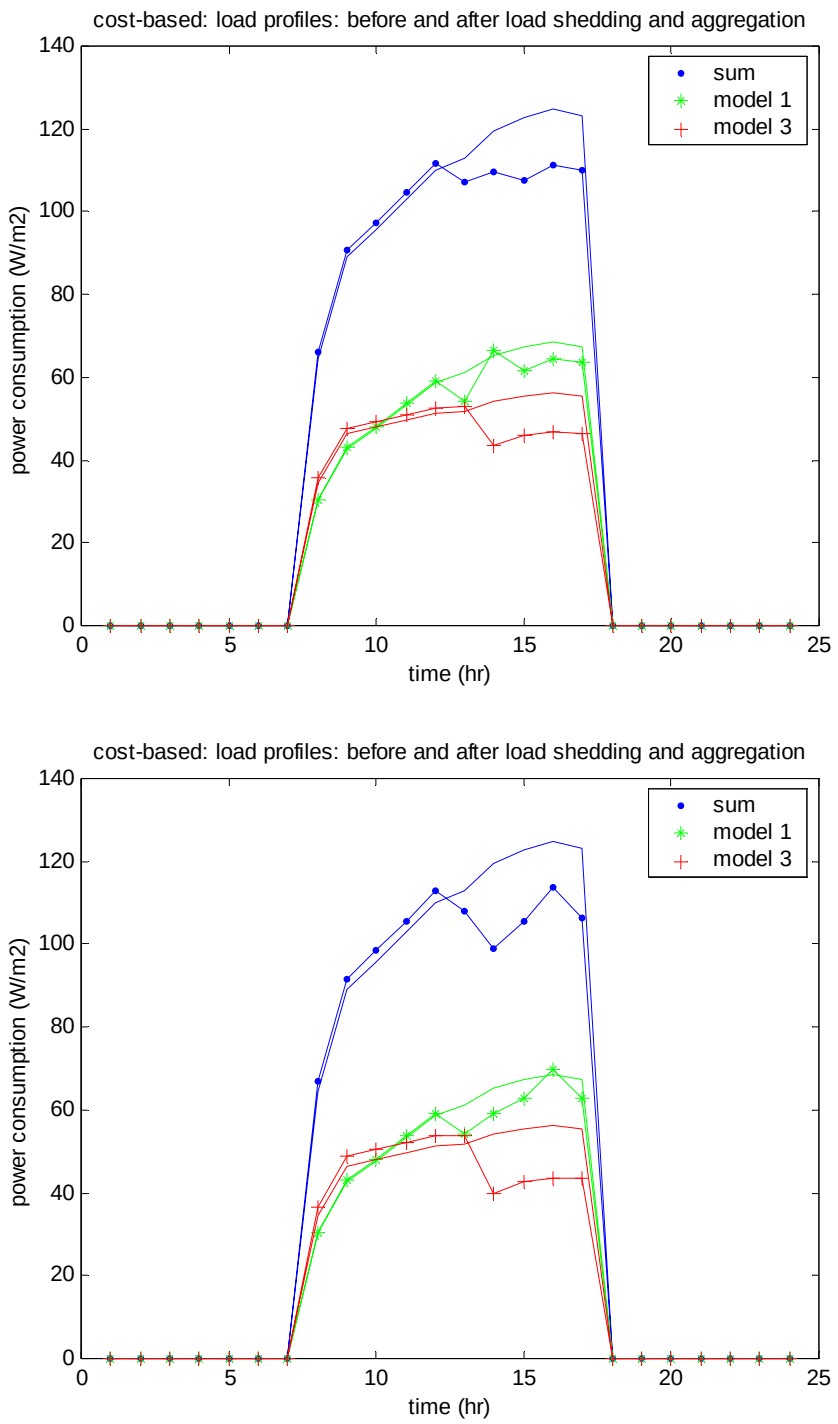


Figure 4.12 "Optimal" load aggregation between buildings 1 and 3, cost-based

Top: demand charge $6.5/kW. Bottom: no demand charge

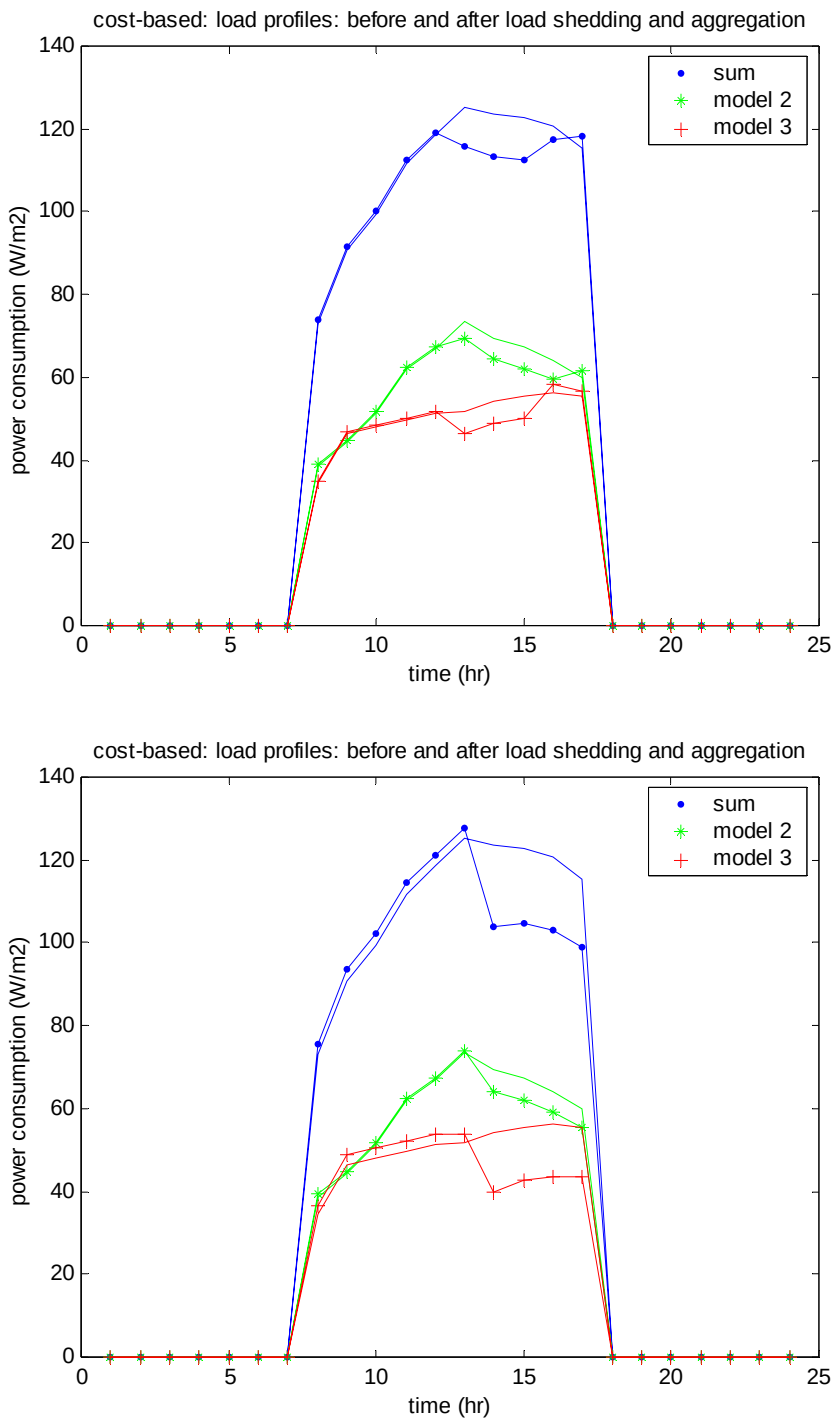
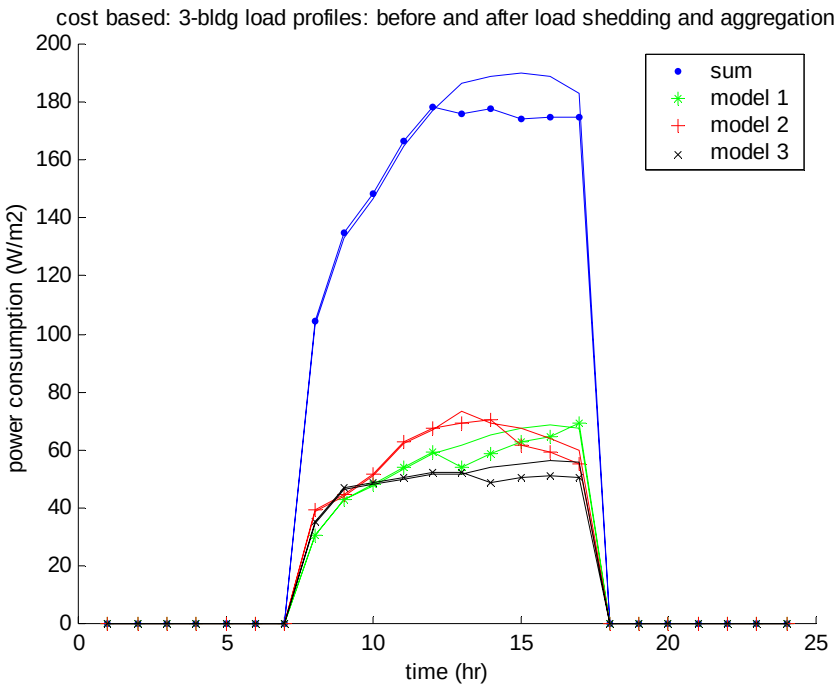


Figure 4.13 "Optimal" load aggregation between buildings 2 and 3, cost-based

Top: demand charge $6.5/kW. Bottom: no demand charge



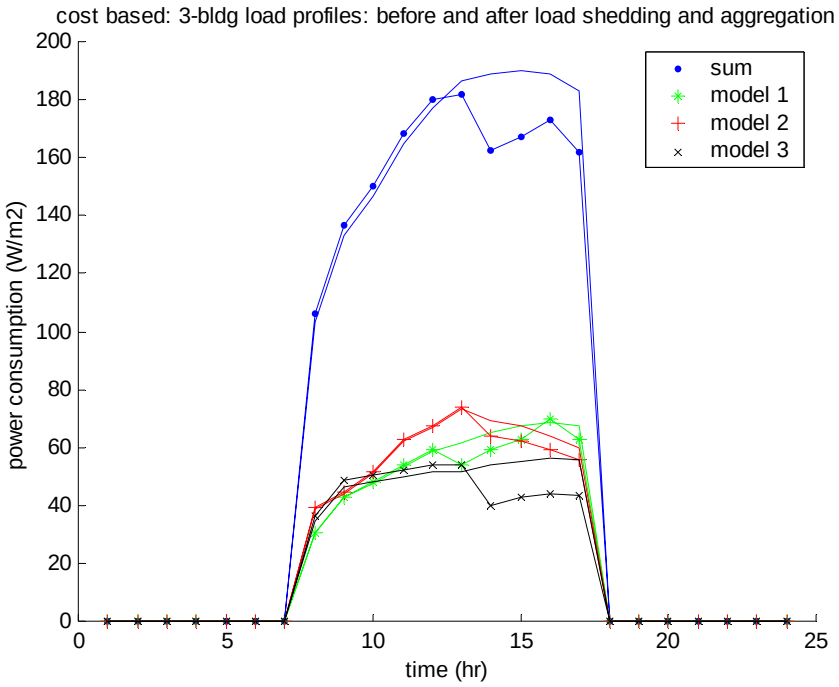


Figure 4.14 "Optimal" load aggregation between buildings1, 2 and 3, cost-based

Top: demand charge $6.5/kW. Bottom: no demand charge

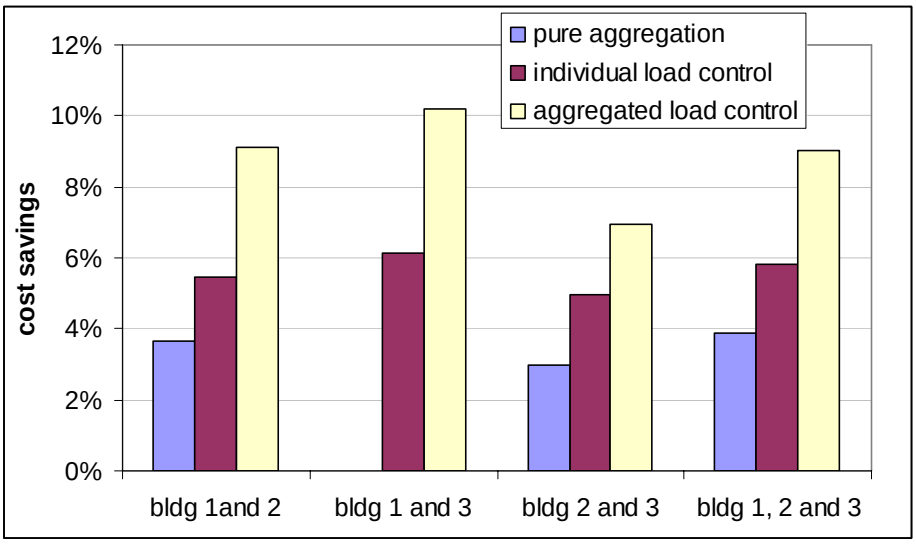


Figure 4.15 cost comparison for two-building and three-building aggregation. Base case is the sum of individual costs

4.3 Match Results of a Two-Building Case for Night Cooling-Based Load Control

As we discussed and our parametric studies showed in previous chapters, night cooling can help shift and shed the peak load and improve the overall load profile with comfort maintained at a satisfactory level. We study in this section the optimal night cooling strategies in a multi-building setting, and learn how the correlations and interdependence among buildings affect night cooling scheduling. We focus on the fan-based night cooling performance in Los Angels using models 1 and 2 from Table 4.1. The economizers in both buildings are turned on. For each of these two models, we run a total of 32 scenarios corresponding to eight fan start times and four discharge processes, shown in Table 4.7. Detailed explanations of these terms can be found in Section 2.4, the night cooling parametric studies. We match the results in Matlab in the way as we did to the thermostat-based aggregation. Of those 1024 combinations, we find the one with the smallest peak load and the one with the lowest total cost by applying the rate structures in Table 4.4. Table 4.8 gives a summary of these two "optimal" cases, compared to two other cases: 1) base1: simply add up two building load profiles without any night cooling control at all; 2) base2: night cool two buildings individually before adding up their load profiles.

Figure 4.16 shows the difference between the base case and the optimal base with peak load as the cost function. In the optimal case, both building participants start fans as early as possible, e.g. 6pm in this case, and keep early morning thermostats low to slow down the discharge of thermal mass. The peak load is reduced from 135 to 121 W/ m^2. Comparison of hourly power consumptions throughout the day shows

that in the base case, night power consumption is zero while day consumption is 1.18kWhr/ m^2 on this particular day; in the optimal case, night power is 0.3kWhr/ m^2 while day consumption drops to 1.14kWhr/ m^2. With night cooling available, power is consumed when it is cheap and the peak is reduced. The PPD plots in Figure 4.17 indicate significant improvement of comfort by pre-cooling. It can be seen that minimizing peak load through night cooling improves daytime thermal comfort: the average PPD down by more than 10%.

Figures 4.18 and 4.19 present the matching results with the objective function being the total electricity cost: 4.18 corresponds to a $6.5/kW demand charge; demand charges between $3.5 and $6.5/kW give the same aggregation results, and so do demand charges between $2 and $3.5/kW, although the energy performance varies from 2 to 3.5 and from 3.5 to 6.5; parts a) to e) in Figure 4.19 show the matching results with the demand charge varying from $2/kW to zero with $0.5/kW step. We can see that the impact of the increasing impact of hourly energy cost on the optimal aggregation operations: a demand charge of $6.5 leads to a similar control strategy to the peak load case in Figure 4.16, and the cases with a smaller demand charge, for example $1.5/kW in Figure 4.19 (b), pay more attention to the overall energy use instead of the load timing. In the $6.5/kW case, both fans start at the earliest possible times and early morning thermostats are chosen to be slow linear – as low as possible. In the $1.5/kW case, fans start late compared to the peak load case, at 10pm and 8pm respectively, and the discharge process is chosen to be fast linear to consume less energy during the discharge period. Table 4.9 summarizes differences by the cost function type. The comfort condition is similar to that in Figure 4.17.

We look at the results in Table 4.9 from a different angle, and the analysis is shown in three parts of Table 4.10 where the aggregated performance is compared to the individual performance without aggregation. In part I, the combination of buildings 1 and 2 achieves an 11% peak load reduction, which building 1 and 2 can achieve 9% and 11% respectively if they simply act alone according to the "optimized" operating schedules. The individual savings would be even better if they act based on individual load control optimization. We drew the similar conclusion in Part II, where the aggregated cost reduction of 7% is comparable to that of individuals', 5% and 8% respectively. Night cooling offers large energy benefit to individual buildings already, and the extra contribution by aggregating these individuals is rather small.

Although not significant, aggregating individual loads still helps by offering the pool diversification opportunities and energy/cost saving potential. We compare in Part III of Table 4.10 four cases: 1) sum of the individual costs without night cooling; 2) cost of the aggregated load without night cooling; 3) sum of the individual costs with night cooling applied, and 4) cost of the aggregated load with night cooling

applied. The fact that 2) is better than 1) and 4) is better than 3) is precisely due to the diversification provided by aggregation. The aggregation contributions are 4% for base case and 5% for night cooling case respectively.

Table 4.7 Two EnergyPlus models used for night cooling

E+ models	Thermal mass	West wall mass	Load pattern
Model 1	1/2 mass	with glass window	peak around 4pm
Model 2	1/2 mass	with glass window	peak around 1pm

Table 4.8 Night cooling schedules

fan-based night cooling 32 scenarios	fan starting time	8am no NC, and NC starts at 6pm, 8pm, 10pm, 12am, 2am, 4am and 6am			
	Thermostats Discharge process	24°C constant	fast linear increase	25°C 8-11am	slow linear increase

Table 4.9 Night cooling based load aggregation

cost function	original cost function value	new cost function value	cost function reduction	Load control strategies fan starting time and discharge processes
peak load (W/ m^2)	135	120	11%	bldg 1: fan starts 6p, slow linear bldg 2: fan starts 6p, slow linear
total cost with a peak demand of \$6.5/kW (\$/m^2)	1.06	0.99	7%	bldg 1: fan starts 8p, slow linear bldg 2: fan starts 6p, slow linear
total cost with a peak demand of \$1.5/kW (\$/m^2)	0.385	0.380	1%	bldg 1: fan starts 10p, fast linear bldg 2: fan starts 8p, fast linear

Table 4.10 Aggregated night cooling details: contribution by individual participants

Part I peak load as cost function

	base peak load (W/m^2)	peak time	new peak load (W/ m^2)	peak time	peak load reduction
sum of 1,2	135	13	121	16	11%
bldg 1	69	16	63	16	9%
bldg 2	74	13	65	13	11%

Part II: total electricity cost as the cost function with demand charge of $6.5/kW

	base total cost ($\$/m^2$.day)	demand charge / energy cost	new total cost ($\$/m^2$.day)	Demand charge / energy cost	total cost reduction
sum of 1,2	1.06	0.88 / 0.18	0.99	0.79 / 0.2	7%
bldg 1	0.53	0.45 / 0.09	0.51	0.41 / 0.10	5%
bldg 2	0.57	0.48 / 0.09	0.53	0.42 / 0.10	8%

Part III: Itemized contributions

	cost of bldg1 + cost bldg2	cost of sum of bldg1 & bldg2	cost of bldg1 with NC + cost bldg2 with NC	cost of sum of bldg 1with NC & bldg 2 with NC
cost ($\$/m^2$)	1.10	1.06	1.04	0.99
reduction		4%	5%	10%
aggregation contribution		4%		5%

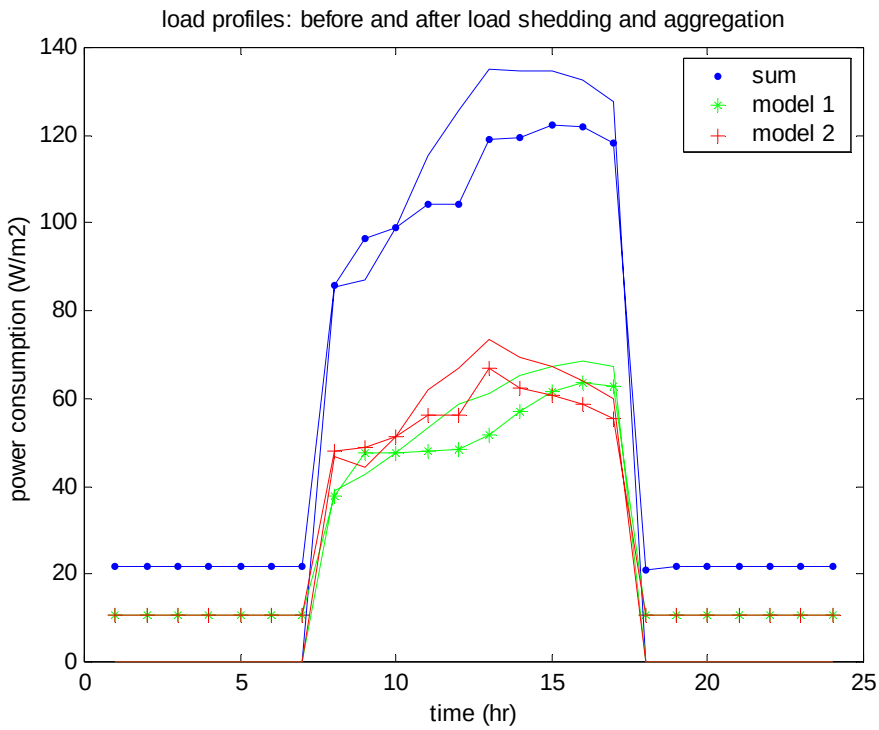


Figure 4.16 "Optimal" load aggregation between models 1 and 2 with night cooling available, peak-load based

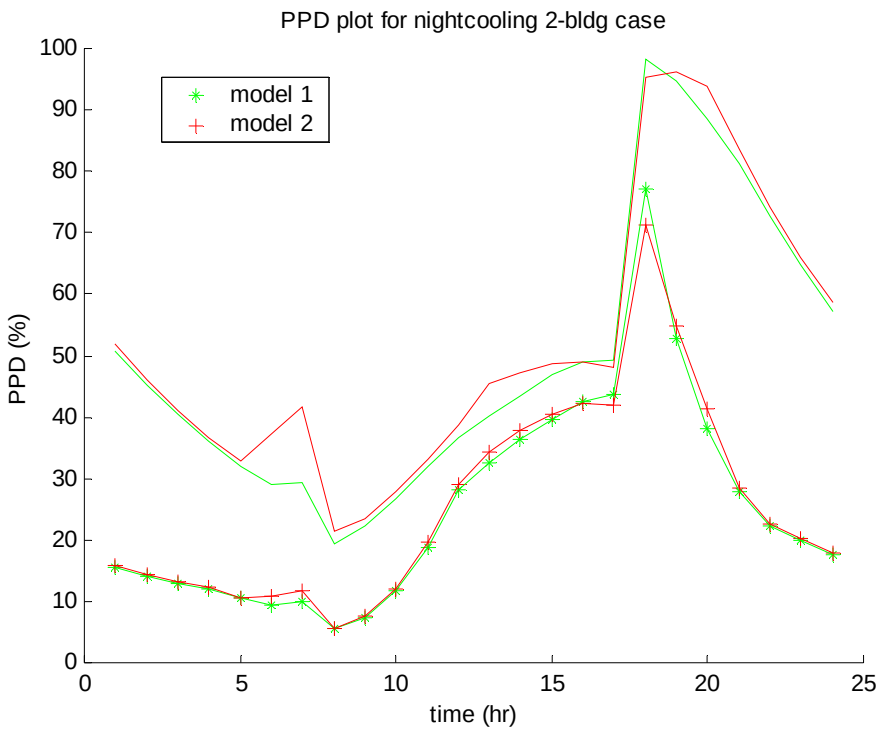


Figure 4.17 PPD plots corresponding to "Optimal" load aggregation between models 1 and 2, peak-load based

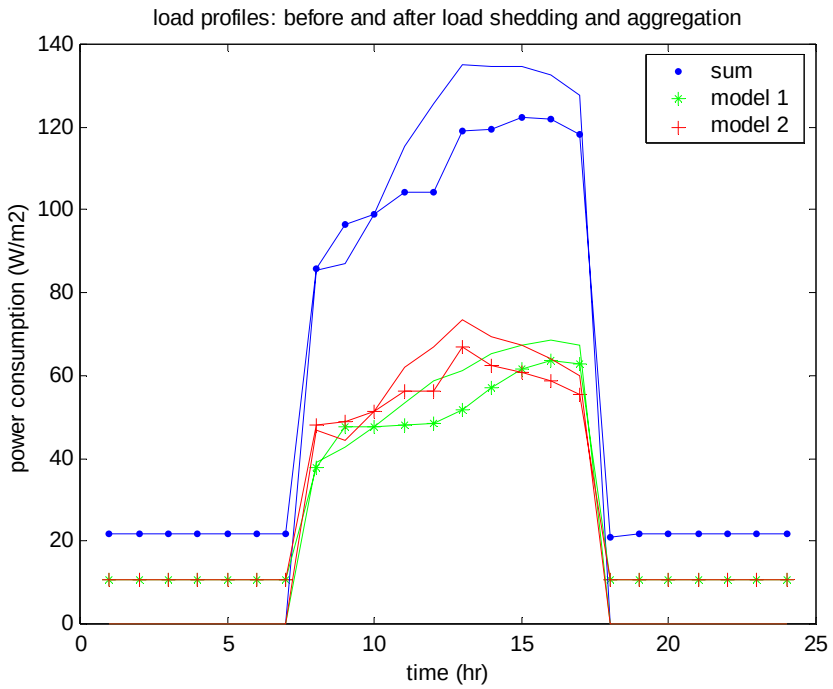


Figure 4.18 Optimal load aggregation between models 1 and 2 with NC, cost based with $6.5/kW demand charge (same until demand charge drop below $4/kW)

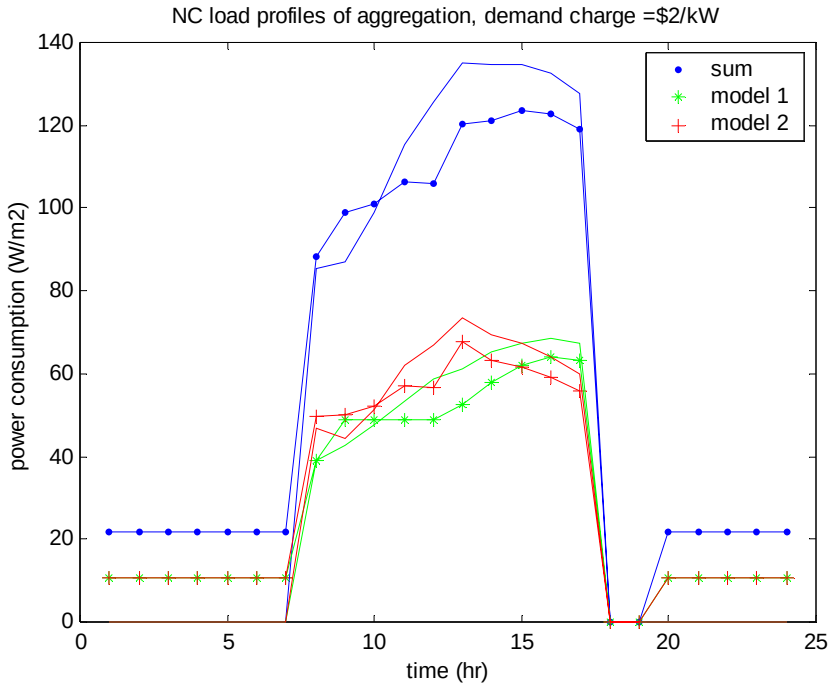


Figure 4.19 a) Optimal load aggregation between models 1 and 2 with NC, cost based with $2/kW demand charge (same until demand charge goes beyond $3.5/kW)

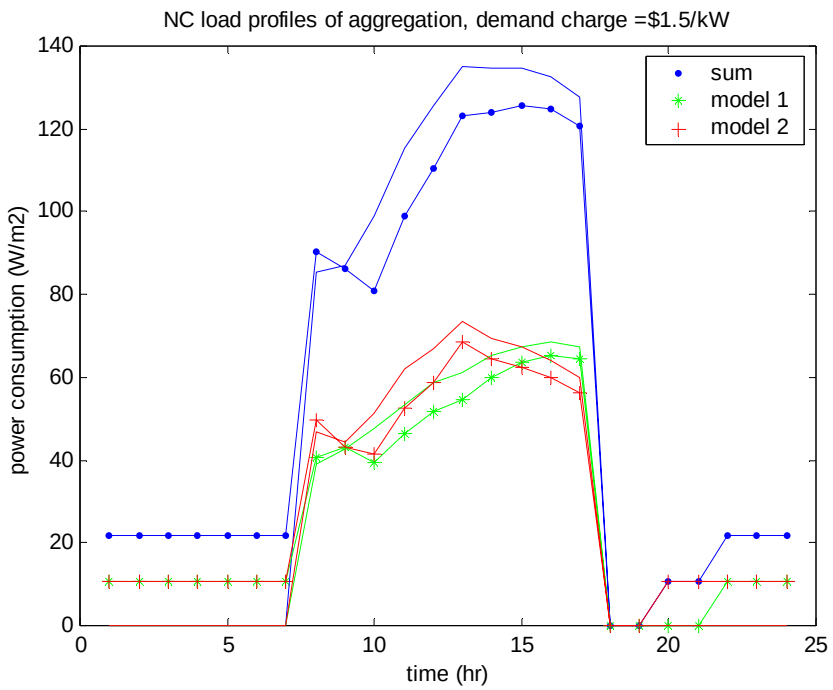


Figure 4.19b) optimal load aggregation between models 1 and 2 with NC, cost based with $1.5/kW demand charge

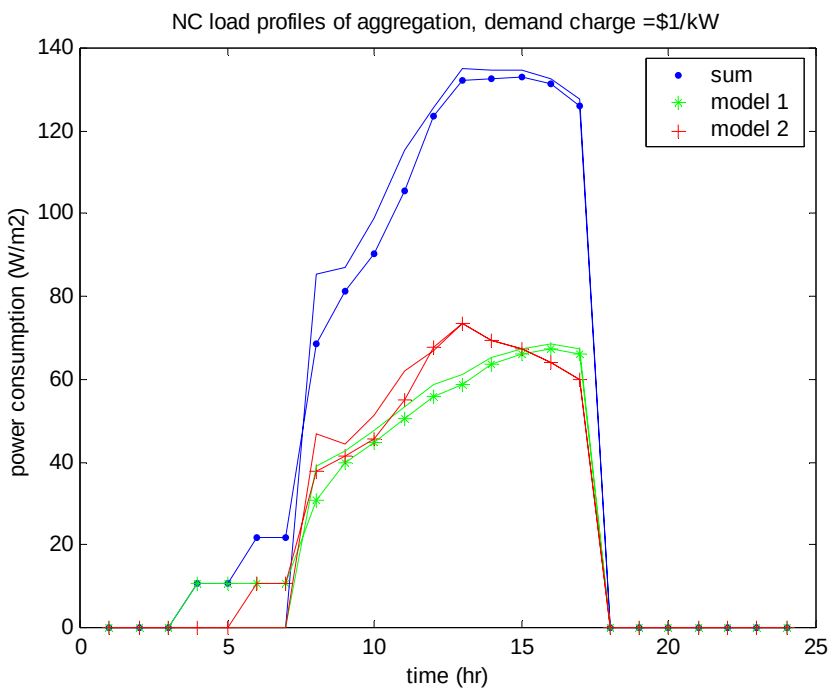


Figure 4.19c) optimal load aggregation between models 1 and 2 with NC, cost based with $1/kW demand charge

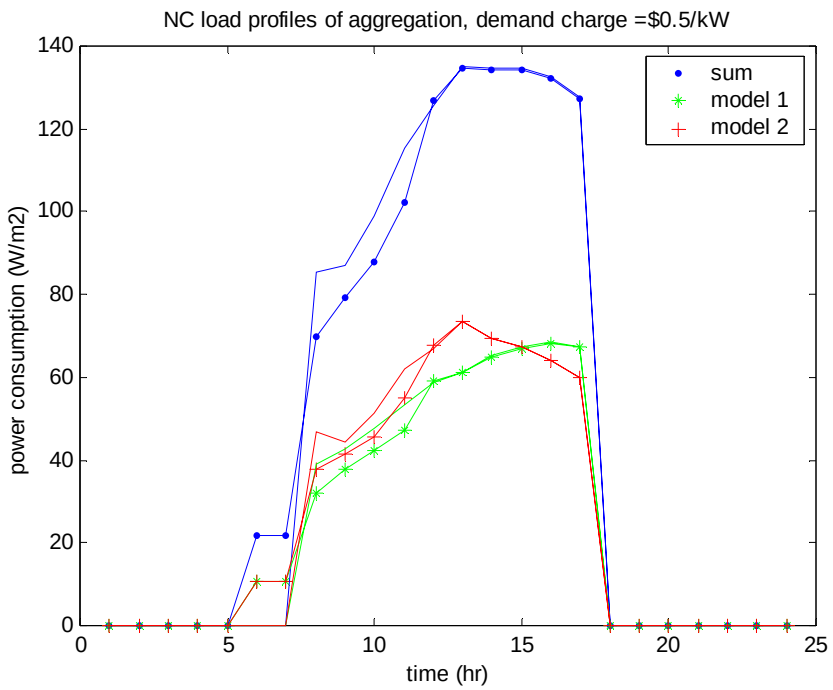


Figure 4.19d) optimal load aggregation between models 1 and 2 with NC, cost based with $0.5/kW demand charge

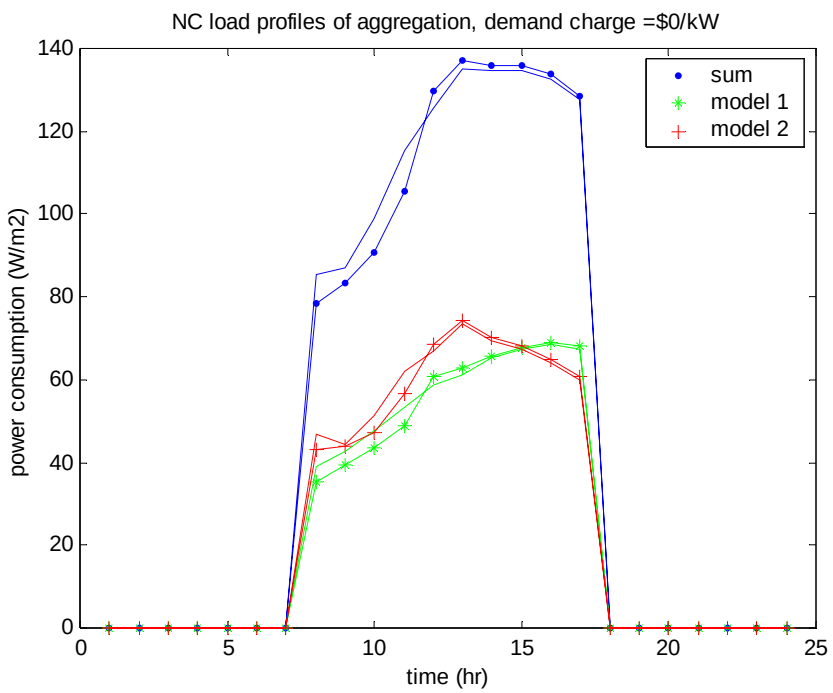


Figure 4.19e) optimal load aggregation between models 1 and 2 with NC, cost based with $0/kW demand charge

4.4 A GA approach to the multi-building problem

The enumeration approach works fairly well, except that the process requires a certain amount of expert knowledge to set up. To generalize the solution, we can solve the problem by expanding the GA framework used in the previous single building studies. This section explores how GA works with multiple buildings and compares the GA performance with that of enumeration. We run EnergyPlus in a sequential manner with each simulation corresponding to a single building participant, shown in Figure 4.20. The process is illustrated using a two-building example in this section.

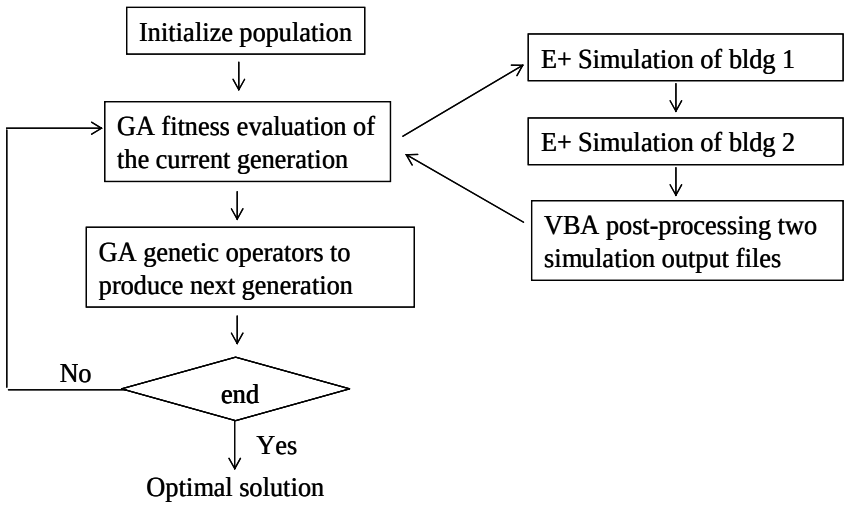


Figure 4.20 Sequential GA process for a two-building aggregation case

Similar to the single GA case, the chromosome in the multi-GA consists of control variables, e.g. thermostat set points or fan starting time plus early AM temperatures, from multiple buildings that are being aggregated. The EnergyPlus batch file is modified so that sequential simulations can be done automatically. At the end of the last EnergyPlus simulation, the VBA post-processing is activated, all the EnergyPlus simulation results get processed in Excel and sent back to the Matlab GA. The system is designed to be able to handle any cost function and theoretically any number of individual buildings. The bottle neck with a large number of participants lies in the computation, not implementation.

For easy comparison purposes, we redo using the multi-GA framework the three thermostat-set-point-based cases in Table 4.2, and present the results in Table 4.11. Recall that in the enumeration setting, the maximum thermostat increase is 3°C and PMV is limited to below 1.5 for feasible solutions, which is strictly followed by the multi-GA framework: five afternoon thermostat set points are varied within [24, 27], and results are checked for feasibility to make sure that only those with less than 1.5 PMV values

survive. This is realized by assigning a very small fitness value to those infeasible solutions that will have a higher probability being thrown out at the next generation. Recall that the Matlab GA used in this research does maximization. Table 4.11 shows that GA performs slightly better than enumeration while taking a lot more EnergyPlus runs. Figure 4.21 plots the traces of best individual and average individual throughout all the generations. We can see a very good convergence in the end. In addition, it only takes 100 generations to reach the stable solution, although 200 generations have been run. This saving is considered in Tables 4.9 – 4.12 when the computation intensity is compared between Enumeration and GA. An E+ simulation takes 5 to 10 seconds. The matching time for Enumeration when there are only two buildings in aggregation is very short and can be ignored.

Tables 4.12 to 4.14 compare GA with Enumeration for the night cooling two-building aggregation case, where Table 4.12 minimizes the aggregated peak demand, Table 4.13 minimizes the total electricity cost with a $6.5/kW demand charge, and Table 4.14 targets the total cost with $1.5/kW demand charge. In all three cases, fan starting time and early morning thermostats are varied the same way as in the single building case in Chapter 3. There are several major observations: 1) similar to the thermostat-based load control case in Table 4.11, GA does perform better, but pays a high price of intensive computation. Take the peak load case in Table 4.12 as an example: GA takes 17 times more EnergyPlus runs for a mere 1.5% more peak load reduction, a 17% increase from the Enumeration case; 2) the total cost case with $6.5/kW demand charge works almost the same way as the peak load case does, which means a $6.5/kW demand charge is really peak-load control oriented; and 3) the cost reduction is more significant when the demand charge is higher, as can be seen by comparing Tables 4.13 and 4.14, which is due to the fact that we apply to daily power profiles the rate structure quoted from the PG&E website and meant for monthly power usage or even longer horizon. Therefore, the role of hourly energy use is underestimated and the peak demand ends up having more impact on the results.

With the target of minimizing the aggregated peak load, Figure 4.22 compares the GA results with the no-night cooling base case in terms of aggregated and individual power profiles, and Figure 4.23 compares the GA results with the Enumeration results. Both GA and Enumeration recognize that it helps to consume more power at night and early in the morning in order to bring down the peak in the afternoon, while GA stretches further in this direction due to the flexibility and therefore achieves more peak load reduction. The total cost case with a $6.5/kW demand charge has the similar load profiles as those in Figures 4.22 and 4.23. Figures 4.24 and 4.25 present the case with a $1.5/kW demand charge, where both GA and Enumeration still decide to turn the fans on at night, but at later times. In addition, they both

keep the early morning temperatures at a level lower than normal but not as low as that in the peak-load case, as a compromise between total and peak consumption.

Table 4.11 Optimizing two-building thermostat set points with the Matlab GA

bldg 1 and 2	peak (W/ m^2)	peak time	peak reduction (W/ m^2)	E+ runs[7]	Time(min)
Base	135	13			
Enum	128	13	5.2%	64	12
GA	126	15	6.7%	340	57
bldg 1 and 3	peak (W/ m^2)	Peak time	peak reduction (W/ m^2)	E+ runs	Time (min)
Base	125	16			
Enum	112	13	10.4%	107	20
GA	111	14	11.2%	536	90
bldg 2 and 3	peak (W/ m^2)	Peak time	peak reduction (W/ m^2)	E+ runs	Time (min)
base	125	13			
Enum	119	12	4.8%	107	20
GA	118	14	5.6%	514	86

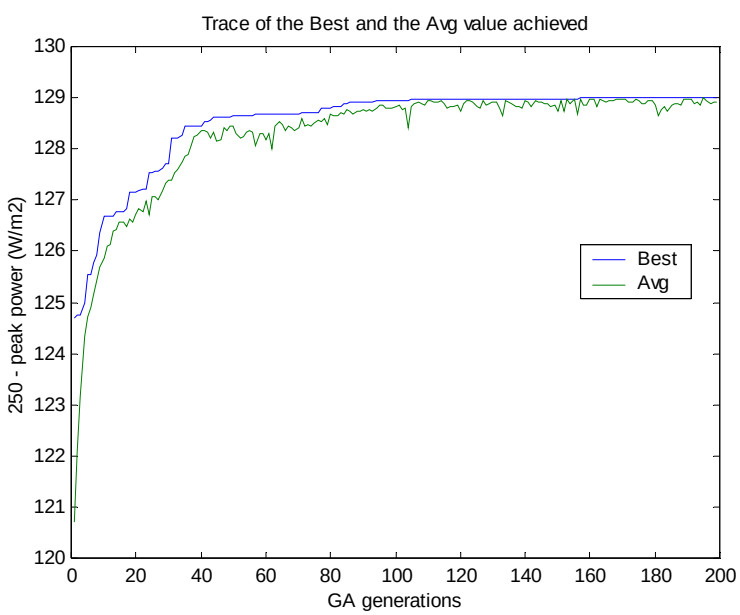


Figure 4.21 Traces of the Matlab GA for a two-building aggregation case

[7] The numbers of EnergyPlus runs are precise for enumeration, and are approximates for GA runs. We run a large number of GA generations, cut off where the best solution of GA is asymptotically stable, and use this value as the necessary simulations by GA.

Table 4.12 Optimizing two-building night cooling schedules with Matlab GA- peak load as cost function

	peak (W/m^2)	peak time	peak reduction	E+ runs	Optimal operations
base	137	13			both fans start at 8am early AM temperature set points: 24°C
Enum	123	15	8.9%	64	both fans start at 6pm both slow linear 20 / 21 / 22 /23
GA	121	15	10.4%	1163	both fans start at 6pm 20 / 20 / 20/ 20, 20 / 20 / 20 / 22

Table 4.13 Optimizing 2-building NC schedules with Matlab GA- total cost with $6.5/kW demand charge

	cost ($\$/m^2$)	cost reduction	peak (W/m^2)	peak load reduction	total load (Whr/m^2)	total load change	E+ runs	Optimal operations
base	1.059		135 1pm		1177			both fans start at 8a all 24°C
Enum	0.994	6.2%	122 3pm	9.3%	1323	18.4%	64	fan1 starts at 8am 20 / 21 / 22 / 23 fan2 starts at 8pm 20 / 21 / 22 / 23
GA	0.988	6.7%	121 3pm	10.3%	1315	20.9%	950	fan1 starts at 10pm 22 / 22 / 23 / 23 fan2 starts at 10pm 22 / 23 / 23 / 21

Table 4.14 Optimizing 2-building NC schedules with Matlab GA- total cost with $1.5/kW demand charge

	cost ($\$/m^2$)	cost reduction	peak (W/m^2)	peak load reduction	total load (Whr/m^2)	total load change	E+ runs	Optimal operations
base	0.385		135 1pm		1177			both fans start at 8a all 24°C
Enum	0.380	1.3%	126 3pm	6.6%	1323	12.5%	64	fan1 starts at 8am 20 / 21 / 22 / 23 fan2 starts at 8pm 20 / 21 / 22 / 23
GA	0.379	1.5%	126 3pm	6.6%	1315	11.7%	1212	fan1 starts at 10pm 22 / 22 / 23 / 23 fan2 starts at 10pm 22 / 23 / 23 / 21

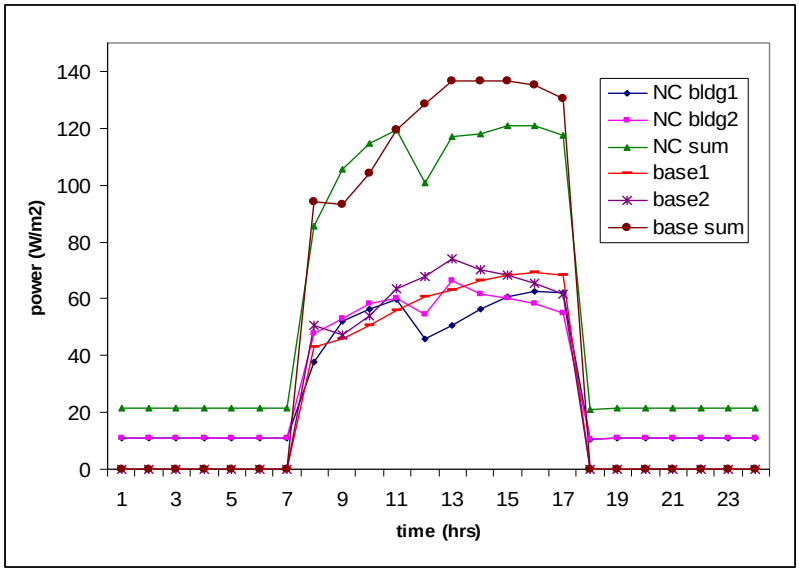


Figure 4.22 Aggregated and individual power profiles for base and GA optimal cases, two-building fan-based night cooling to minimize the aggregated peak , 8/8, LA

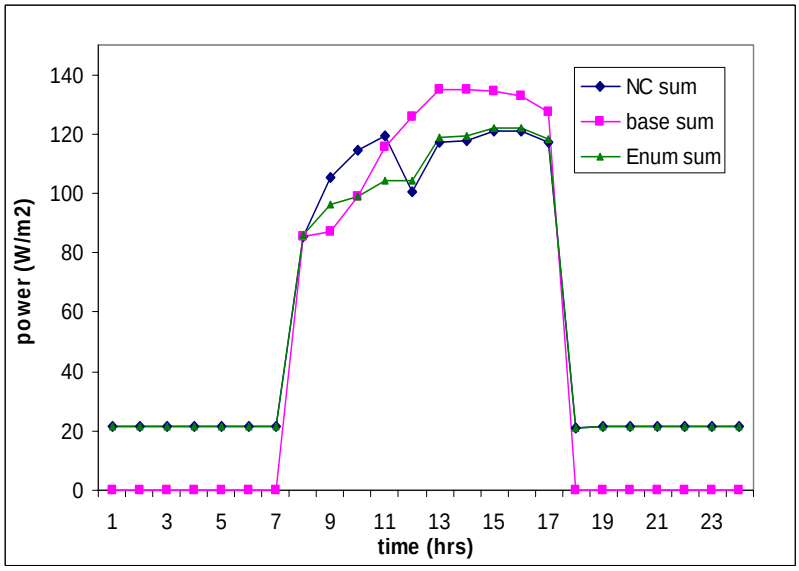


Figure 4.23 Aggregated and individual power profiles for GA optimal and Enumeration optimal cases, two-building fan-based night cooling to minimize the aggregated peak , 8/8, LA

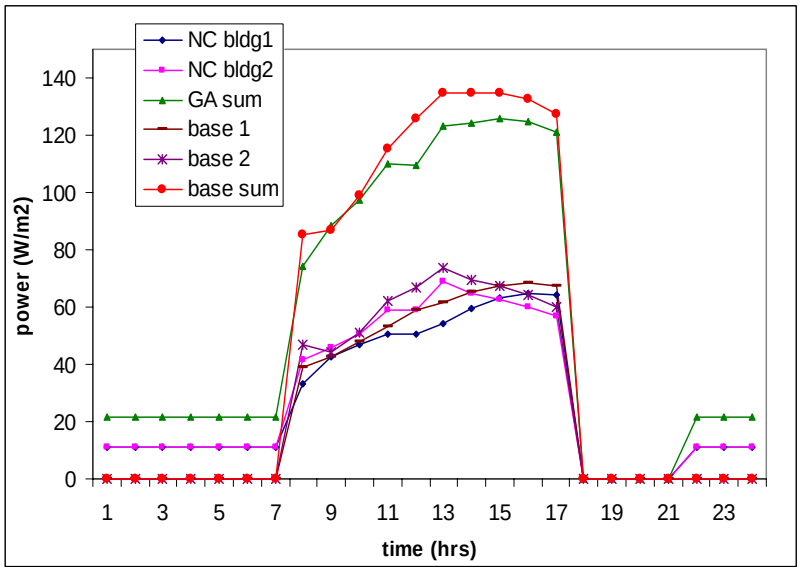


Figure 4.24 Aggregated and individual power profiles for base and the GA optimal cases, two building fan-based night cooling to minimize the total cost with $1.5 demand charge, 8/8, LA

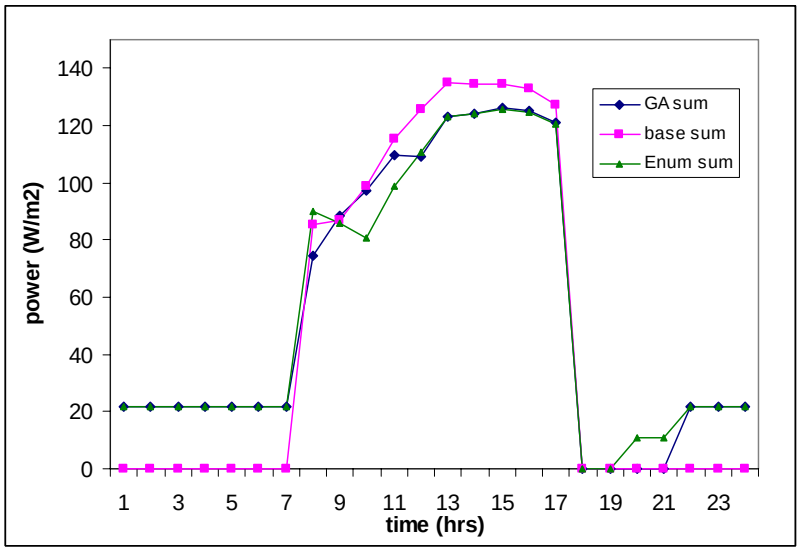


Figure 4.25 Aggregated power profiles for base, GA optimal and Enumeration optimal cases, 2bldg fan-based night cooling to minimize the total cost with $1.5 demand charge, 8/8, LA

4.5 Computation concerns of simulation-based approaches

The computation of the enumeration approach consists of two parts: individual simulation and match. The matching process simply adds up individual load profiles in Matlab and has little computation intensity involved when the number of buildings is small. However, its complexity increases exponentially with the number of individuals. With the simple exhaustive matching, five buildings with 30 feasible solutions each take 30^5, about 24 million matches. Although each match only involves summation and maximization, the number of matches makes it impossible to handle more than five buildings in a reasonably amount of time. Such a five-building case takes 2.5 hours on a Pentium4 1.8GHz machine, although a four-building case only takes 2 minutes.

Educated enumeration performs better than a generalized optimizer, but requires expert knowledge to set up. For models and load control strategies concerned in this research, the simulation for individual buildings in the enumeration approach takes about an hour for a typical summer day. Building operators can do it off line the night before to come up with strategies for tomorrow if the aggregation scale is small, say computation can be done in minutes. The system can be used flexibly in practice. For example, building operators can classify those days with similar weather conditions, so that apply the same load control strategies to a group of similar days instead of every single day. In all the analysis, we assume that tomorrow's weather forecast is accurate, which is true in most cases. However, the system is fast enough to rerun in case there are sudden weather, load or price changes, assuming that those changes can be forecast couple of hours in advance, which is true in practice.

The enumeration approach is efficient and simple, but its dependence on expert knowledge could be a problem in order to generalize the approach. For those scenarios we know little about it is difficult to cover most of the input space by enumerating a limited number of possibilities. Therefore, the best solution from enumeration and match might not be a good sub-optimum overall.

GA is computationally intensive in almost all the cases, but also generally efficient and can always find a better operation strategy given enough time. The process is generic and takes little expert knowledge to set up. The bottleneck is a large number of EnergyPlus simulations are needed for function evaluation. As a way to reduce computation, Matlab GA saves the computation results throughout all generations to a lookup table. For every new chromosome, the code searches in this Table first and if this chromosome has been calculated before, the result is taken and recalculation can be avoided. This saves about 30% - 60% EnergyPlus runs in our Matlab GA studies.

Notice that with the multi-GA set up, the amount of computation needed increases linearly with the number of buildings. In reality, GA does slightly better than this due to its advantage of handling large parameter sets. Two-building aggregation cases take anywhere about an hour and the specific number depends on the cost function and the convergence requirement. Five-building cases take about 2.5 hours. Table 4.15 shows the E+ runs taken by two-building cases and five-building cases. Table 4.16 compares the computation between Enumeration and GA in terms of the total time. Enumeration's exhaustive search part takes little time in the two-building case but increases exponentially to 2.5 hours in the five-building case, while GA's computation time increases approximately linearly. Therefore, GA has advantage when the size of the problem gets large.

Table 4.15 EnergyPlus runs[8] taken by multi-GA

Cases	2 buildings 1/ 2	5 buildings 1/1/1/2/2	2 buildings 1/ 3	5 buildings 1/1/1/3/3
E+ runs	340	840	536	1236

Table 4.16 total computation time comparison between Enumeration and multi-GA

	Enumeration			Multi-GA
	E+ run time (min)	Matching time (min)	Total time (min)	Total time (min)
2bldg 1 and 2	12 (64 E+ runs)	<0.1	12	57 (340 E+ runs)
5bldg 1/1/1/2/2	12 (64 E+ runs)	150	162	140 (840 E+ runs)

4.6 Economy of scale

We've looked at several simple aggregation cases involving two or three buildings. It is worth studying how aggregation efficiency varies as the number of buildings increases. Without any load control, aggregation still achieves some savings due to the diversification effect, as shown in both thermostat-based and night-cooling-based strategies early in this chapter. However, the savings from pure aggregation will approach asymptotically an upper bound as the number of buildings goes up because the marginal benefit brought to diversification by a new building will drop after some point. Being able to alter the load profiles through load control could make a difference. We will illustrate the size effect at a small scale, up to six buildings, using the Enumeration approach. The load control strategy is thermostat-based with the peak load as the cost function. We continue to use three models in Table 4.2 from Section

[8] As in Table4.11, the E+ runs here are the numbers that meet convergence and are chosen after a longer run.

4.2, and duplicate those building if necessary. The same Matlab program is used for matching multiple buildings.

Table 4.17 summarizes the peak load reduction with different building types: by total peak reduction, we mean the aggregated peak. The average peak reduction is a conceptual number by assuming all individuals make equal contribution to the total reduction, which is not the case in reality because different buildings peak at different times. We make the following observations: 1) the more buildings in the aggregation pool, the more total reduction achieved compared to the base case; 2) the profile mix plays an important role in determining the aggregation performance.

To focus on the effect of size, we fix the building mix and simply increase the number of buildings in the pool. Table 4.18 part I shows the aggregation results from one to five buildings, all based on building 1. The aggregation results in Table 4.18 part II are based on two or three buildings and we simply duplicate one or more pairs in the pool. In both tables, aggregated load control is compared with individual load control and the contribution of aggregation is quantified. Notice that the contribution of aggregation stabilizes as the size increases: 12% peak reduction contributed by aggregation in the one-building case and 56% in the two-building case with 1 and 3. The reason is that with the limited number of load profiles, a new building into the pool can only choose to cooperate with others in a limited number of ways. When the size increases to some point, new buildings start to repeat what old ones do, and the performance reaches an upper bound.

Figures 4.26 and 4.27 present the aggregation results for a three-building case and a four-building case respectively. The related two-building case can be found in Figure 4.3. We have the following observations based on these three cases: 1) the peak reduction potential has been largely exploited even with three buildings, indicated by the flat control period of the aggregated profile in Figure 4.26. A 4-building pool flattens the aggregated peak even further; 2) two buildings in an aggregation pool with the same type tend to behave differently when the size is small, which enhances the aggregation performance.

In reality, we can adjust the control period, e.g. assign different buildings different control periods, and try to maximize the reduction potential that can be captured by a group of buildings. Notice that the savings and the scale effect depend on the buildings under study. Load aggregators should choose those buildings that work well together to create a higher reduction potential.

Table4.17 Aggregation performance – mix matters

	2-bldg	3-bldg		4- bldg		5-bldg	
individual model type	1/3	1/3/3	1/1/ 3	1/3/1/3	1/1/1/3	1/1/1/3/3	1/1/1/1/3
total peak reduction (W/ m^2)	13.1 10.5%	17.6 9.7%	21.7 11.2%	26.4 10.6%	26.1 10.0%	34.9 11%	30.4 9.2%
average peak reduction (W/ m^2)	6.6 5.2%	5.9 3.2%	7.2 3.7%	6.6 2.6%	6.5 2.5%	7.0 2.2%	6.1 1.8%

Table4.18 Aggregation performance – economy of scale

Part I: size effect with a limited number of profiles – an extreme case

building mix	1bldg 1	2bldg 1/1	3bldg 1/1/1	4bldg 1/1/1/1	5bldg 1/1/1/1/1
peak reduction from individual load control	5.9%	5.9%	5.9%	5.9%	5.9%
peak reduction from aggregated load control	5.9%	6.2%	6.7%	6.7%	6.7%
savings from aggregation	0%	0.3%	0.8%	0.8%	0.8%
contribution of aggregation	0%	5%	12%	12%	12%

Part II: size effect with a limited number of profiles

building mix	1/2	1/2/1/2	1/1/1/ 2/2/2	1/3	1/3/1/3	1/2/3	1/2/3/ 1/2/3
peak reduction from individual load control	2.8%	2.8%	2.8%	7.4%	7.4%	3.4%	3.4%
peak reduction from aggregated load control	5.2%	6.3%	6.3%	10.5%	10.6%	6.1%	6.2%
savings from aggregation	2.4%	3.5%	3.5%	3.1%	3.2%	2.7%	2.8%
contribution of aggregation	46%	56%	56%	30%	30%	44%	45%

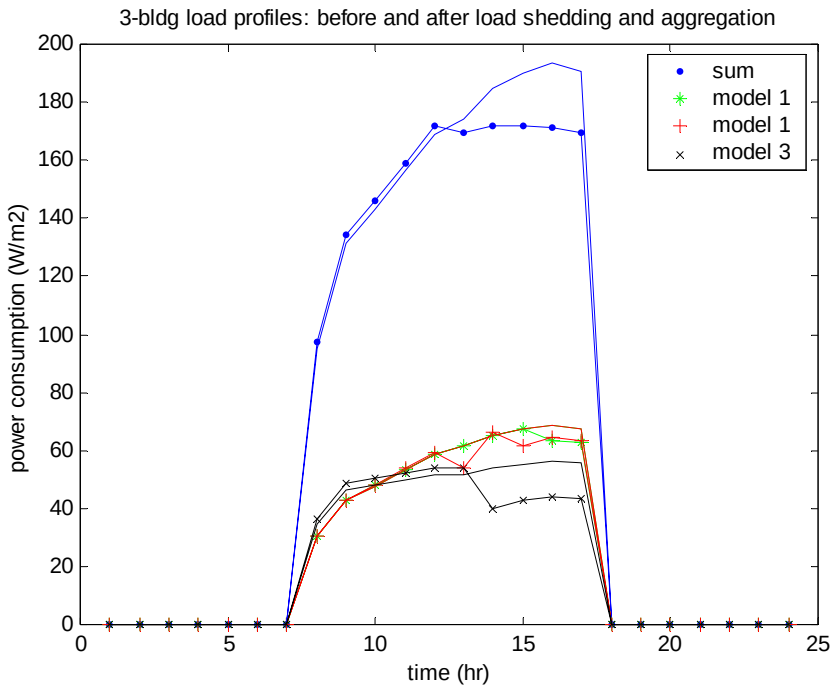


Figure 4.26 "Optimal" load aggregation between buildings with model types of 1, 1 and 3

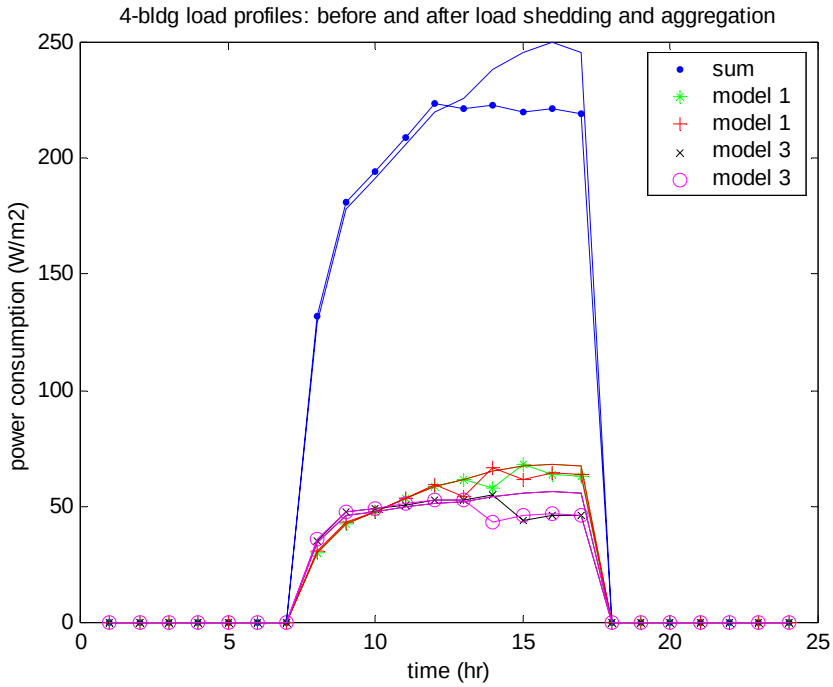


Figure4.27 "Optimal" load aggregation between buildings with model types of 1, 1, 3 and 3

CHAPTER FIVE
A MODEL-BASED NONLINEAR OPTIMIZATION APPROACH TO THE MULTI-BUILDING PROBLEM

In the previous chapter, we explored the multi-building problem through Enumeration and a multi-GA. Both approaches involve tens or hundreds of EnergyPlus simulations and the decision-making processes are slow. One natural development would be separating optimization from simulation, and providing the optimizer with simplified building dynamics learned from the simulation data. This is expected to improve the computation efficiency. We will explore such an approach in this chapter by building a time series model to predict building load profile in the base case. On the top of that, we use linear regression and model the load reduction and the service degradation due to load shedding. The time series model and regression models together form the individual load profile prediction model. The multi-building aggregation is then formulated as a nonlinear optimization problem supported by individual load prediction models. The problem is modeled in AMPL (A Mathematical Programming Language) and solved by LOQO [Vanderbei 1997], a nonlinear commercial solver. Optimization results are evaluated and compared to those from simulation and Enumeration.

5.1 Problem formulation

In this section, we review briefly the general mathematical problem of multi-building optimization. In the remaining sections, a detailed load predication and optimization process will be presented.

The general problem is to come up with an optimal scheme for a multi-building aggregation pool to minimize the total electricity cost or peak demand while maintaining a certain comfort level in all the buildings. We formulate this problem in two ways that are related: 1) treat comfort as a constraint, shown in Eqn.5.1; 2) treat comfort as part of the cost function by penalizing the violation through a Lagrangian multiplier, shown in Eqn.5.2. In both formulations, we have represented the hourly power consumptions and the PPD values in each individual building as functions of the individual system, load control parameters and time. In theory, these functions are complicated and nonlinear, and are what full-size simulation software packages such as EnergyPlus try to compute. We will take a simpler approach of computing hourly power and PPD, which is key to this chapter.

$$\min \quad \sum_{t=1}^{24}\left(\sum_{i=1}^{N} W_{i,t}\left(u_{i,t}, x_{i,t}\right)\right)\times R_t + \max_t\left(\sum_{i=1}^{N} W_{i,t}\left(u_{i,t}, x_{i,t}\right)\right)\times D \quad \textit{Minimize the total cost}$$

$$\begin{aligned}
s.t. \quad & PPD_{i,t}\left(x_{i,t}\right) \le PPD_i^*,\ \forall i, t \in [t1, t2] && \textit{comfort requirement} \\
& u_- \le u_{c,i,t} \le u_+,\ c \in C, \forall t, \forall i && \textit{continuous control variables} \\
& u_{d,j,t} = 0\, or\, 1,\ d \in D, \forall t, \forall i && \textit{discrete control variables} \\
& W_{i,t} = f_i\left(x_{i,t}, u_{i,t}\right), \forall i && \textit{building dynamics} \\
& PPD_{i,t} = g_i\left(x_{i,t}, u_{i,t}\right), \forall i && \textit{comfort dynamics}
\end{aligned}$$

Eqn.5.1

$$\min \quad \sum_{i=1}^{N}\left(\sum_{t=1}^{24}\left(W_{i,t}\times R_t + \alpha \times PPD_{i,t}\right)\right) + \max_t\left(\sum_{i=1}^{N} W_{i,t}\right)\times D \quad \textit{Minimize the penalized total cost}$$

$$\begin{aligned}
s.t. \quad & u_- \le u_{c,i,t} \le u_+,\ c \in C, \forall t, \forall i && \textit{continuous variables} \\
& u_{d,j,t} = 0\, or\, 1,\ d \in D, \forall t, \forall i && \textit{discrete variables} \\
& W_{i,t} = f_i\left(x_{i,t}, x_{i,t-1}, \ldots, u_{i,t}\right), \forall i && \textit{building dynamics} \\
& PPD_{i,t} = g_i\left(x_{i,t}, u_{i,t}\right), \forall i && \textit{comfort dynamics}
\end{aligned}$$

Eqn.5.2

Where,

$W_{i,t}$	Electricity consumption by building i at time t, $i = 1,\ldots,N$, $t = 1,\ldots,24$
$PPD_{i,t}$	PPD values in building i at time t
PPD_i^*	PPD requirement in building i
R_t	Electricity rate at time t
D	Demand charge rate
$u_{i,t}$	Control variables, $u_{c,i,t}$ continuous variables, and $u_{d,i,t}$ discrete variables
$x_{i,t}$	State variables
f_i	Building i dynamics determining electricity consumption at hour t
g_i	Building i dynamics determining thermal comfort i is at hour t

A mathematically similar problem is presented in Eqn.5.3a where the peak load, instead of the total cost, is minimized for the same N-building pool. The sum of hourly energy use is a linear[9] term to this optimization problem, and taking it out wouldn't change the mathematical nature of the problem. Although optimizing peak load and optimizing total cost will lead to different results, two problems are mathematically similar and the peak load problem has already captures the nonlinearity in Eqn.5.1 and Eqn.5.2, so the method developed for the peak load problem can be applied to the total cost problem without any extra complexity. This chapter focuses on the peak load problem.

$$
\begin{aligned}
\min \quad & \max_{t}\left(\sum_{i=1}^{N} W_{i,t}\right) && \textit{Minmize the peak demand} \\
s.t. \quad & W_{i,t} = f_i\left(x_{i,t}, x_{i,t-1}, \ldots, u_{i,t}\right), \forall i && \textit{building dynamics} \\
& PPD_{i,t} = g_i\left(x_{i,t}, u_{i,t}\right) \le PPD_i, \forall i && \textit{comfort requirement} \\
& u_- \le u_{c,i,t} \le u_+,\ c \in C, \forall t, \forall i && \textit{continuous control variables} \\
& u_{d,j,t} = 0 \text{ or } 1,\ d \in D, \forall t, \forall i && \textit{discrete control variables}
\end{aligned}
\qquad \text{Eqn.5.3a}
$$

The key characteristic of all three formulations is that they are all min max problems: first the program looks for the peak demand, which is a maximization problem over a 24-hour period on the aggregated load profile; then minimizes this peak demand over the control variables specified in both Equations. Min max problems are difficult to solve in general due to the nonlinearity. This min max problem can be converted to a typical minimization problem by adding a new variable to the original problem, a linear programming technique, as shown in Eqn.5.3b, where the maximization term is replaced by 24 inequalities in the constraint. Let $z = \max_{t}\left(\sum_{i=1}^{N} W_{i,t}\right)$, and we have

$$
\begin{aligned}
\min \quad & z && \textit{Minimize the overall peak or its upper bound} \\
s.t. \quad & \sum_{i=1}^{N} W_{i,t} \le z, \forall t && \textit{Min Max problem} \\
& W_{i,t} = f_i\left(x_{i,t}, x_{i,t-1}, \ldots, u_{i,t}\right), \forall i && \textit{building dynamics} \\
& PPD_{i,t} = g_i\left(x_{i,t}, u_{i,t}\right) \le PPD_i, \forall i && \textit{comfort requirement} \\
& u_- \le u_{c,i,t} \le u_+,\ c \in C, \forall t, \forall i && \textit{continuous variables} \\
& u_{d,j,t} = 0 \text{ or } 1,\ d \in D, \forall t, \forall i && \textit{discrete variables}
\end{aligned}
\qquad \text{Eqn.5.3b}
$$

[9] By linear, we mean the hourly energy use term to the cost function, not control variables to the energy use. We know that building dynamics are nonlinear, which, however, is not a concern at the top level of model structure.

Eqn.5.3b is the model on which this chapter is based. For simplicity, we only implement the thermostat-based load control strategy. The control variables are hourly thermostat set points, therefore, no discrete variables will be considered. This, indeed, is a big simplification, as discrete variables need special treatment during optimization. The next three sections are building blocks: Section 5.2 discusses a time series model for predicting base load profiles, responsible for one part of $W_{i,t}$; Section 5.3 presents an approximation model for load reduction due to load shedding, responsible for the other part of $W_{i,t}$; similarly, section 5.4 builds an approximated PMV increase model, responsible for $PPD_{i,t}$ in Eqn.5.3b. Section 5.5 brings these three together and solves this simplified multi-building problem via nonlinear optimization.

5.2 Base load predictor – a time series model

To separate EnergyPlus simulations from optimization, we need a simplified model to capture the building dynamics. We take the approach of function approximation in this section and build a time series model by learning from the EnergyPlus simulation data. The reason why we choose time series models is three-fold: 1) load data can be nicely described by a time series with seasonal patterns; 2) time series models represent single building dynamics by a small number of parameters, which is important for the multi-building problem at a large scale; 3) the linearity of time series models make it easy to solve a nonlinear optimization problem for which the time series model is a constraint. Chapter 1 reviews some function approximation approaches including artificial neural networks and time series models.

5.2.1 Data preparation

We use the models 1 and 3 from Table 4.1 as two participant types, and will duplicate them when the size of the pool increases. For each of these two model types, we run EnergyPlus simulation for the entire summer, from June 1 to August 31, and on each day, we simulate a total of nine scenarios: base without temperature set point change, four cases with different increase amount: hours 13-17 set point increase by 1°C to 4°C respectively, and four cases with temperature increase of 2°C and increasing shedding window length from 1 to 4 hours respectively.

June data are used as the training set and July and August as the testing set. To train the base load model, the base case data are enough. The rest is for training load reduction model due to thermostat-based load control.

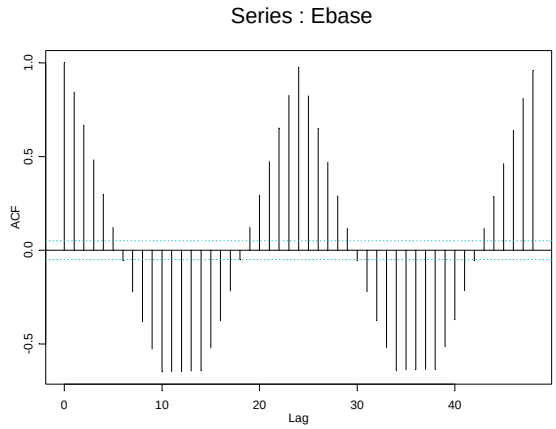
For simplicity, we take out all the weekend data out of the summer time series and use the weekday only data for model identification. Throughout this research, all building systems are completely off over the weekends. Therefore, taking out weekend data points wouldn't lose any information. However, weekend data are part of the complete time series, and taking them out could possibly distort the embedded dynamics in this time series and lead to incorrect model identification and poor estimation, especially for Mondays. However, we argue that this is not a big concern for the model used in this research because weekday dynamics dominate. In addition, Mondays will be given special treatment in prediction. We will show late in this chapter that the model based on purely weekday data achieves a satisfying prediction performance. The time series model only provides a starting point and we have other model components to further improve the prediction.

Our goal is to predict load profiles using as fewer inputs as possible. During EnergyPlus simulations, we output hourly total power consumptions (sum of power uses by chillers, fans and pumps), PMV values, and indoor air temperatures as system responses and hourly outside temperatures and solar radiations on walls and roof as exogenous features. The internal load patterns remain constant throughout the entire summer in this study; therefore weather is the only exogenous factor. This is not a very accurate but still reasonable assumption in reality. Although internal loads are stochastic by nature, they vary with low standard deviations.

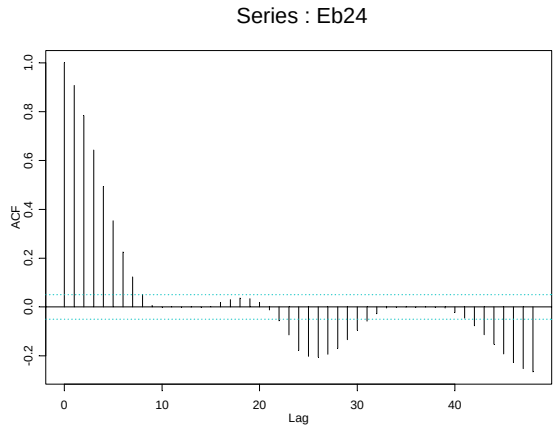
By now, we have hourly simulation data over a total of 66 workdays and are ready to conduct model identification.

5.2.2 Model identification and estimation

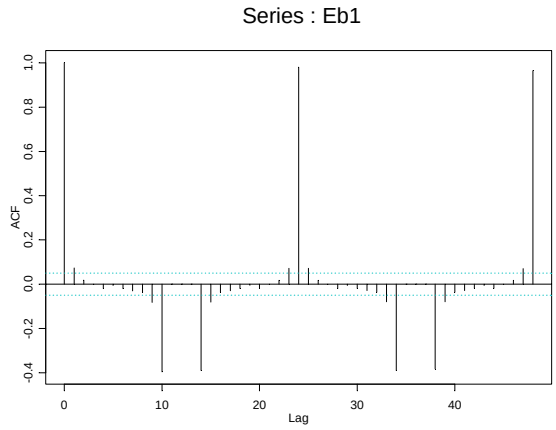
We start model identification by visualizing and analyzing statistically the time series of power consumptions in SPlus, a statistics software package. We first look at the autocorrelation function (ACF) plots of the power time series. ACF plots [Box and Jenkins 1976] are a commonly-used tool for checking randomness in a data set. This randomness is ascertained by computing autocorrelations for data values at varying time lags. If random, such autocorrelations should be near zero for any and all time-lag separations. If non-random, then one or more of the autocorrelations will be significantly non-zero. ACF plots are often used in the model identification stage for autoregressive moving average time series models, and can be used to access the seasonality (or periodicity) of a data series.



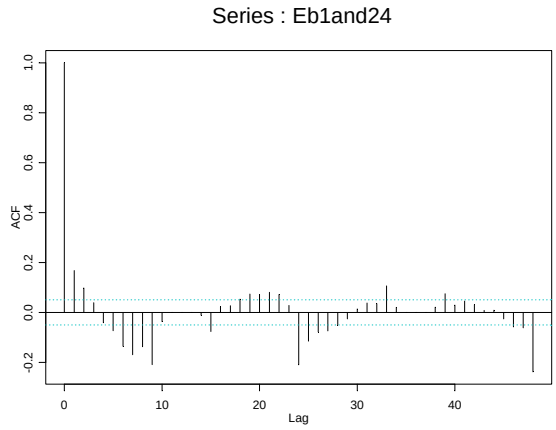
a) acf of original power series, diff_0



b) acf of diff_24



c) acf of diff_1



d) acf of diff_1_and_24 (top and bottom)

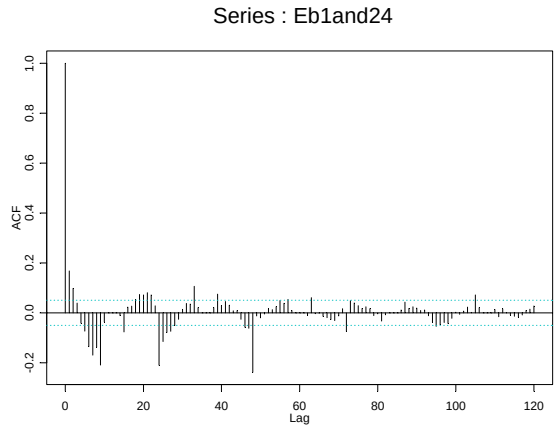


Figure5.1 ACF plots of differencing schemes

Figure5.1a) shows the autocorrelation coefficient plot of the power time series. The plot strongly suggests a seasonal pattern in the data with a period of 24 hours, which should be the case because the building operates periodically on a daily basis. The oscillatory and nondecaying function indicates a nonstationary process. In order to obtain a stationary process, a number of differencing schemes are tested, as illustrated in b), c), and d). The one that differences twice with 1 and 24 as periods, indicated by d), is finally chosen as it has the least overall autocorrelation. The small-lag autocorrelations, although not within the specified uncorrelated range, are fairly low. The fact that the differencing scheme of 1 and 24 best describes the data series indicates that the lagged terms by 1 hour, 24 hours, and 25 hours play important roles in predicting the current hourly power consumption. We understand that weather-related variables certainly play a role in prediction as well. As the first attempt, we bring in all possible exogenous variables and conduct a regression over the lagged power terms and these factors to decide the relative importance of all these inputs.

The exogenous variables here include outdoor temperature and external surface solar incident on each of the four relevant surfaces: south wall, east wall, west wall and roof. All the lagged terms of these exogenous variables are also included in this regression. Eqn.5.4 gives the general format of this regression. Notice that Eqn5.4 is only a symbolic expression of which factors might have impact on the current power consumption, and it is a linear relationship. The coefficients of all these factors are determined by regression.

$$\begin{aligned} W \sim\ & W_1 + W_{24} + W_{25} + T_{out} + SS + SW + SR + SE \\ & + T_{out_1} + SS_1 + SW_1 + SR_1 + SE_1 + T_{out24} + SS_{24} + SW_{24} + SR_{24} + SE_{24} \\ & + T_{out_{25}} + SS_{25} + SW_{25} + SR_{25} + SE_{25} \end{aligned} \qquad \text{Eqn.5.4}$$

Where,

W Current hourly power consumption

T_{out} Current outdoor temperature

SS, SW, SR, SE Current solar incident on South wall, West wall, Roof and East wall respectively

$With_{1,24,25}$ 1-hour, 24-hour and 25-hour lagged terms of the corresponding parameters

All the data available from the simulation are standardized before put in regression. A generalized linear model named glm in SPlus is used for regression. A glm model is fit using Iterative Reweighted Least Squares (IRLS) [SPlus 2001]. The SPlus code is given in Appendix D.1.

Table 5.1 gives the regression coefficients and their t values for Eqn.5.4. Statistically, variables with absolute t value greater than 2 are considered significant. However, y1, y24 and y25 are much more significant than others, although the west all solar incident, SW, might also have some minor impact with t values slightly greater than 2. Of the three lagged terms, the same hour yesterday has the biggest impact. Note that although three lagged power terms dominate the prediction and there are no exogenous factors in the final formulation, it does not mean exogenous variables are not important. In fact, the lagged terms are results of weather factors acting on the system. These exogenous variables will be used later to enrich the model.

Table5.1 Coefficients and t- values - A first cut

Factors	Value	Std. Error	t value
(Intercept)	0.00244	0.00362	0.7
W1	**0.73846**	0.02513	29.4
W24	**1.00619**	0.00542	185.6
W25	**-0.74007**	0.02512	-29.5
Tout	-0.00765	0.01159	-0.7
SS	0.00313	0.02889	0.1
SW	0.01574	0.00924	1.7
SR	0.00933	0.02073	0.5
SE	-0.00529	0.00797	-0.7
Tout1	0.01799	0.01149	1.6
SS1	0.05267	0.02874	1.8
SW1	-0.02729	0.00856	-3.2
SR1	0.04520	0.02161	2.1
SE1	-0.00500	0.00905	-0.6
Tout24	0.01117	0.01185	0.9
SS24	-0.01891	0.02931	-0.6
SW24	-0.01867	0.00928	-2.0
SR24	-0.00301	0.02136	-0.1
SE24	0.00519	0.00834	0.6
Tout25	-0.01920	0.01164	-1.6
SS25	-0.05092	0.02894	-1.8
SW25	0.02279	0.00855	2.7
SR25	-0.03891	0.02145	-1.8
SE25	0.00699	0.00847	0.8

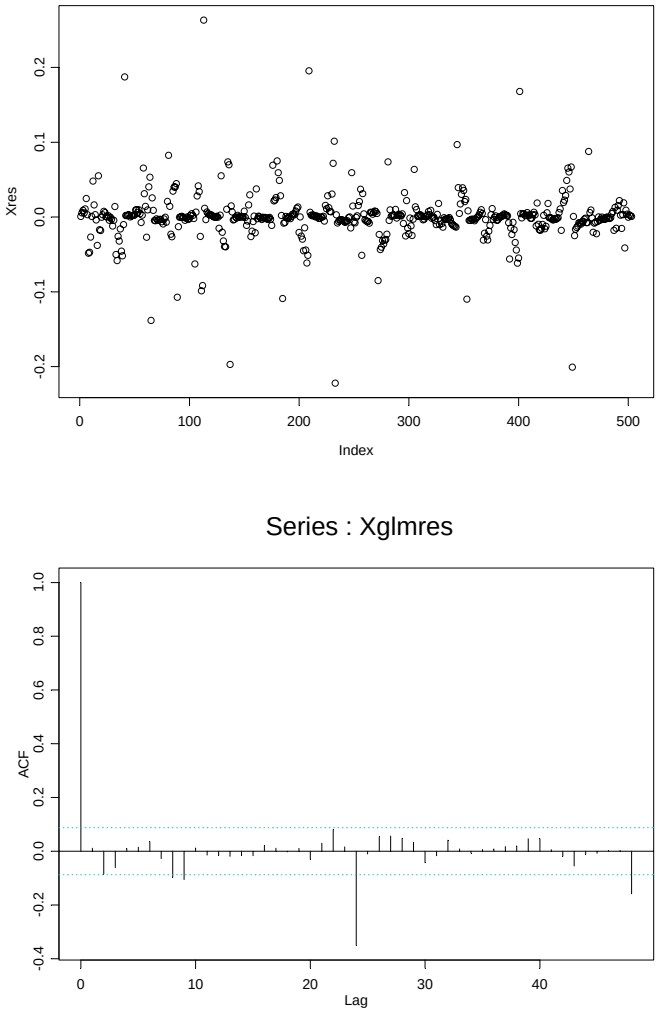


Figure5.2 In-sample residual info: a) residual plots b) ACF plots of glm residuals

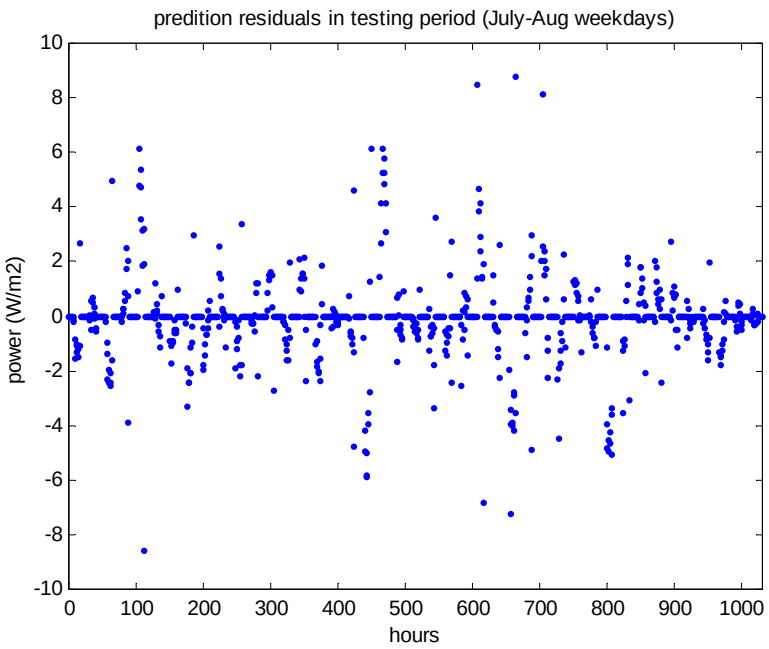


Figure5.3Testing data residuals with the model of Eqn.5.4 and Table 5.1, building type model 1

Table5.2 Model coefficients and their t values for building 1- with only significant variables

Factors	Value	Std. Error	t value
(Intercept)	0.00129	0.00186	0.7
yt1	0.87396	0.02175	40.2
yt24	1.00711	0.00374	269.4
yt25	-0.87953	0.02227	-39.5

Figure 5.2 a) plots the residuals of Table 5.1 and Figure 5.2b) shows the ACF of the regression residuals. It only serves as a reference as we cannot judge a model's predictability using in-sample (training) data. However, bad in-sample performance can kill a model without testing it in the testing set. Figure 5.3 plots the residuals on testing data using the model in Table 5.1. The residuals have a mean of 0.1 and a variance of 2.2, which is fairly good as a starting point of the prediction. Since all the exogenous variables are less significant at this point, we rerun the regression with only three lagged power terms and obtain Table 5.2. The lag-only model has better prediction performance. On the testing set, both models center around zero as they should, but the full-model has a variance of 2.2 and the lag-only model 1.88, a 15% variance reduction. Further analysis is based on Table 5.2.

The ACF plot in Figure5.2 b) shows a few spikes at non-zero lags, indicating that residuals are not purely white noise and certain system features have not been captured. To further explain the residuals, we could build an ARIMA model for the residuals. This approach is typical in time series regression studies. An alternative is to extract features from the exogenous factors to build a nonlinear regression model for residuals. Here, we decide to ignore the residuals for now and focus on other aspects to improve the prediction for simplicity and also for the following reasons: 1) the variance of residuals in testing data and the autocorrelations between regression residuals are rather low. It would not be too far off without a residual model; 2) more importantly, the time series analysis did not take into consideration an important aspect of this problem, which is we don't have hourly updated power data for tomorrow, but our predictor still tries to predict the power uses at all hours next day. The prediction mechanism used here is basically a one-step system, meaning that for the next step to be precise, the current step needs to be updated with the true information instead of estimates. However, we aim at predicting at the end of today tomorrow's power profile; 24-hour and 25-hour lagged data are available, but one-hour lagged data can only be predictions. How to improve this one-step model to better predict tomorrow's profile is a far more important issue than building a residual model based on assumed perfect data.

We have obtained a load profile for tomorrow based on Table5.2. Because we do not have the real-time one-hour lagged power data, there will be a gap between the predicted and real values. Especially if there is a large prediction error in the morning, it will be carried on all day and lead to even larger errors later in the day. There is not much we can do to the time series model due to the information constraint, but extra information from exogenous variables should help. There could be two approaches to further improve the prediction: correct the errors and correct the peak.

Eqn.5.5 presents a model to correct tomorrow's prediction errors based on previous prediction errors: Eqn.5.5 1) is the ideal model consisting of three lagged power terms and a residual term; 2) is the model used in reality due to unavailability of one-hour lag term; and 3) is the error prediction. This approach is again another time series modeling problem, as indicated in 3) where f is a time series function predicting today's prediction errors from yesterday's. To implement this, we need to use two days' real power consumptions, yesterday and the day before, as initial data and conduct prediction for both yesterday and today.

$$W(t) = a_1 \times W(t-1) + a_2 \times W(t-24) + a_3 \times W(t-25) + e(t) \qquad 1)$$

$$\hat{W}(t) = \tilde{a}_1 \times \hat{W}(t-1) + \tilde{a}_2 \times W(t-24) + \tilde{a}_3 \times W(t-25) + \hat{e}(t) \qquad 2) \qquad \text{Eqn.5.5}$$

$$\hat{e}(t) = f\left(W(t-24) - \hat{W}(t-24)\right) \qquad 3)$$

Where,

$W(t), \hat{W}(t)$	- True and estimated power consumption at hour t
$W(t-1),\ \hat{W}(t-1)$	- True and estimated power consumption at the previous hour
$W(t-24), \hat{W}(t-24)$	- True and estimated values at the same hour on the previous day
$W(t-25)$	- True power consumption, previous hour previous day
$e(t), \hat{e}(t)$	- True and estimated residual at hour t

The error time series analysis in Eqn.5.5 makes statistical sense. Physically, the model says that the power consumption at the current hour is a function of previous power values and previous prediction errors. We believe that past power consumptions and errors carry the information about building dynamics, and past errors have information about the predictor.

This error-correcting model described above is dynamic and complicated. The peak-correcting model that we will describe below is a static and simpler one. The peak-correcting model takes advantage of everyday specific exogenous information and models the daily peak load as a function of maximum temperature and maximum solar incidents on related surfaces. Figure 5.4 shows the relationship between maximum daily temperatures and peak loads in the summer months in LA. There is clearly an upward pattern and close to being linear. We have seen the similar trends between peak loads and maximum solar radiations. The peak loads in the training set are regressed on corresponding maximum temperatures and south, east, west wall and roof solar radiations. We tried both linear and quadratic models in SPlus, and found the linear model, Eqn.5.6, has the best performance – the best balance between the coefficient significance, residual variation and model simplicity. Table 5.3 gives the model's statistics indicting coefficients of the maximum temperature, south wall and east wall solar incidents are significant. In reality, the model identification results depend on the specific building under study. We assume that we have the perfect weather forecast for tomorrow's maximum temperature and solar radiations. In reality, detailed solar radiation forecast might not be easily obtained and temperature forecast could be off too. We need to keep in mind these potential constraints.

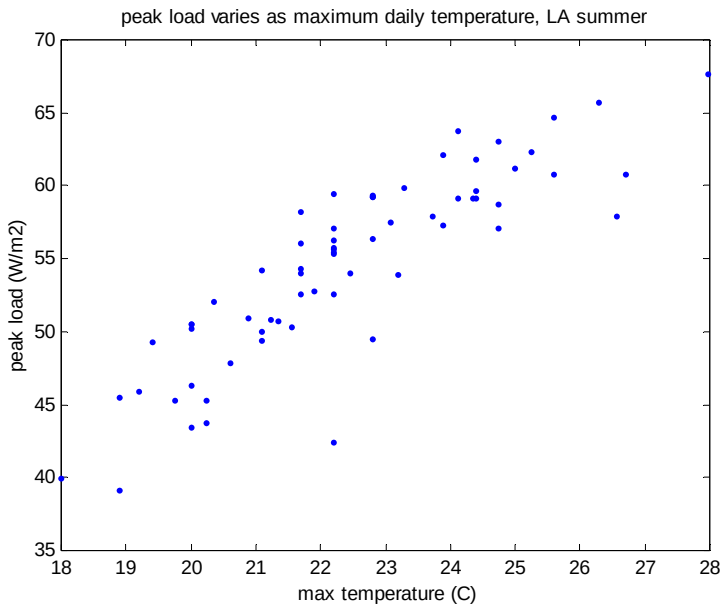


Figure5.4 Relationship between peak loads and maximum temperatures in summer, LA

$$W_{\max} = 1.468 \times T_{\max} + 0.032 \times SS_{\max} + 0.015 \times SE_{\max} \qquad \text{Eqn.5.6}$$

Where, $W_{\max}$ is the peak demand of the day, $T_{\max}$ peak temperature, $SS_{\max}$ south wall peak solar radiation, and $SE_{\max}$ east wall solar radiation.

Table5.3 Peak load as a function of exogenous variables

Factors	Value	Std. Error	t value
(Intercept)	3.913	3.048	1.3
T_max	**1.468**	0.165	8.9
SS_max	**0.032**	0.008	4.3
SW_max	0.005	0.004	1.1
SR_max	-0.007	0.004	-1.6
SE_max	**0.015**	0.002	5.9

The maximum temperatures have been used to predict peak load for profile adjustment before [Seem and Braun 1992]. We compare the full model in Table5.3 with the maximum-temperature-based-only model on the testing set. The model in Table5.3 has a mean of -0.8 and a variance of 2.8, corresponding to a two standard deviation range of [-4.2, 2.5], while the maximum-temperature-only model has a mean of 3.84 and a variance of 3.20, corresponding to a wider range of [2.6, 10.2]. The full model works better.

It is to be noticed that we choose the training data set for this model slightly differently. As we only have one data point for each day, using only June data as the training set is not sufficient to yield a convincing result. In addition, we observe the following time-sensitive trend throughout the summer: high temperatures lead to higher peaks more easily in June than in July and August. Therefore we choose half-month data from June, July and August to form the training set. It is acceptable to distort the time series in this case because we are looking for a static relationship rather than a dynamic and time-related one.

With the newly estimated peak load, we update the previous prediction from Table 5.2 and scale the entire power profile to obtain the final base load profile prediction. The prediction performance is examined on several representative days, shown in Figures 5.5, 5.6 and 5.7. Figure 5.5 emphasizes the importance of adjusting peak load where the true load profiles, predictions without peak corrections and predictions with peak corrections are compared. Peak correction helps improve the prediction on most days, especially on July 27 and August 17. In Figure 5.6, a total of eight days are examined. Figure 5.7 looks at the same hour prediction over a few days in the testing period. Overall, the prediction performs well. Afternoon hours are predicted better than early morning ones. This is because we scale the entire curve based on the peak load adjustment which takes place in the afternoon.

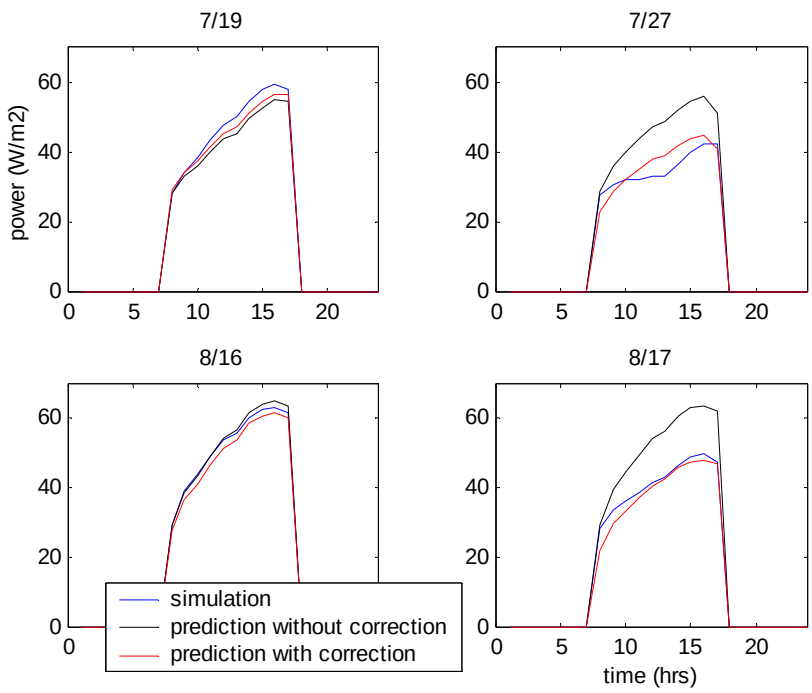


Figure 5.5 based load profile prediction and peak correction

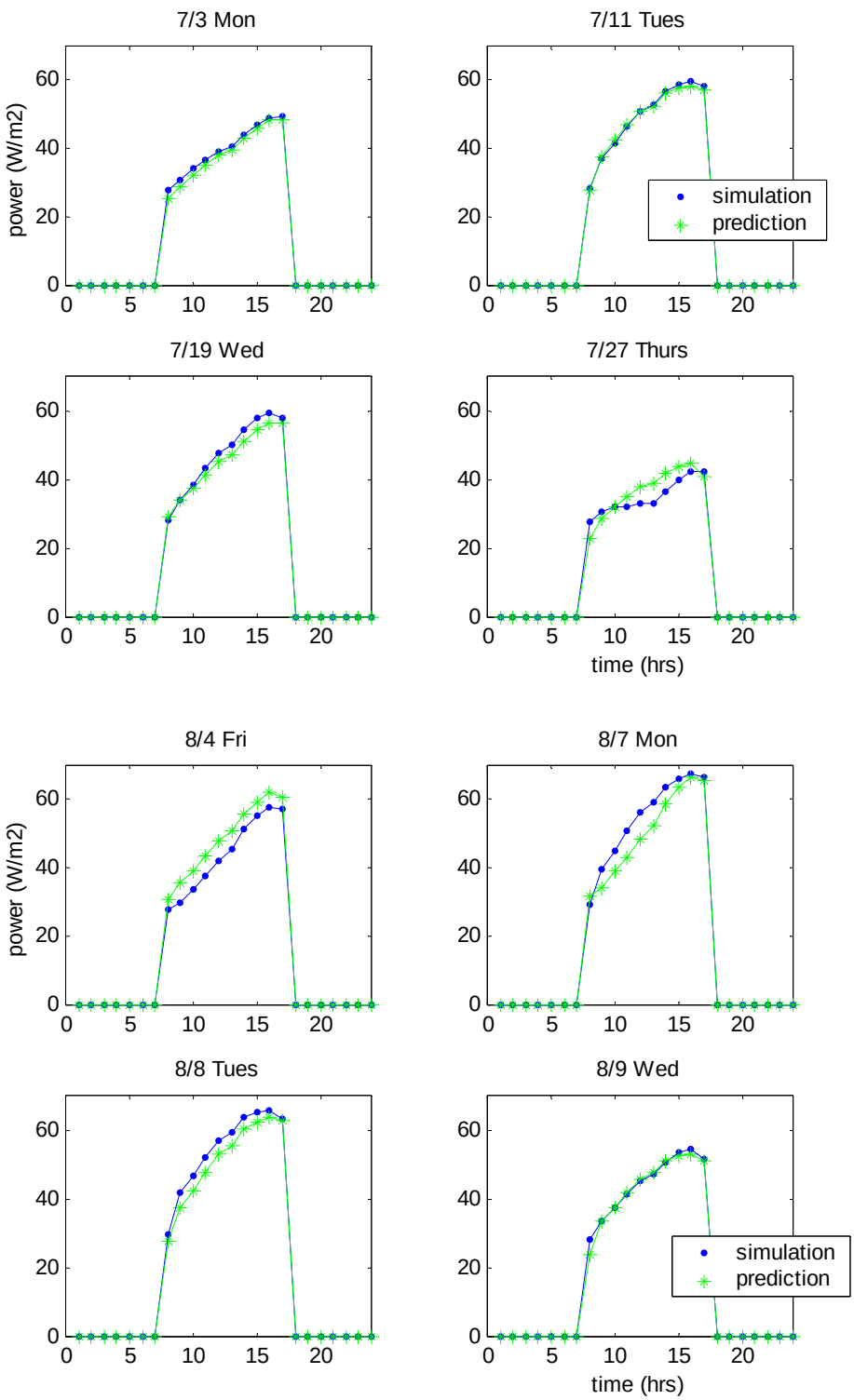


Figure5.6 Base load profiles prediction vs. simulation

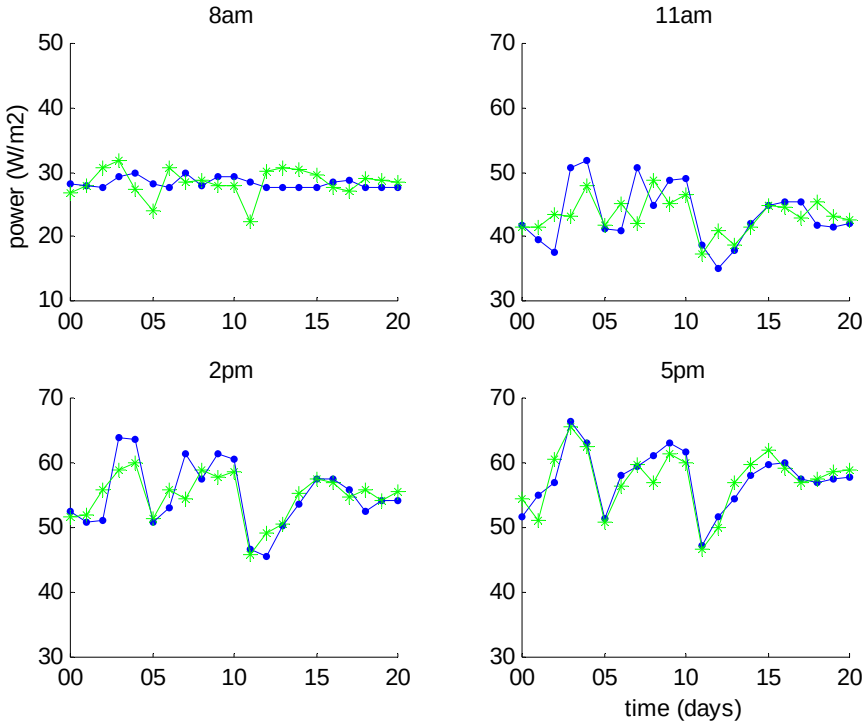


Figure5.7 Prediction performance for specific hours over the last 20 days in August

An important reason why the prediction is good is that the data used for both training and testing are from EnergyPlus simulations and have little noise. In addition, there are no stochastic aspects associated with our system; notably, the load pattern remains the same throughout the entire summer, which keeps the load profile shape similar between different days. We expect to see higher errors and degraded performance with data from the real world.

Although the prediction performance for Mondays is acceptable, we could make it better by giving it special treatment. For a Monday, a weighted average of predictions by the previous day, which is a Friday in our data set, and by the previous Monday would give a better prediction. In general, a prediction considering both the previous day and the day type performs better than either of the single ones. Day type can be just calendar days such as Monday through Friday. It can also be the classified groups such as “hot day”, “warm day”, and “cold day” based on temperatures. A more sophisticated way is to classify days using an unsupervised classification algorithm, such as the K-means algorithm [Johnson and Wichern 2002]. A classification algorithm can consider a lot more exogenous variables other than temperatures.

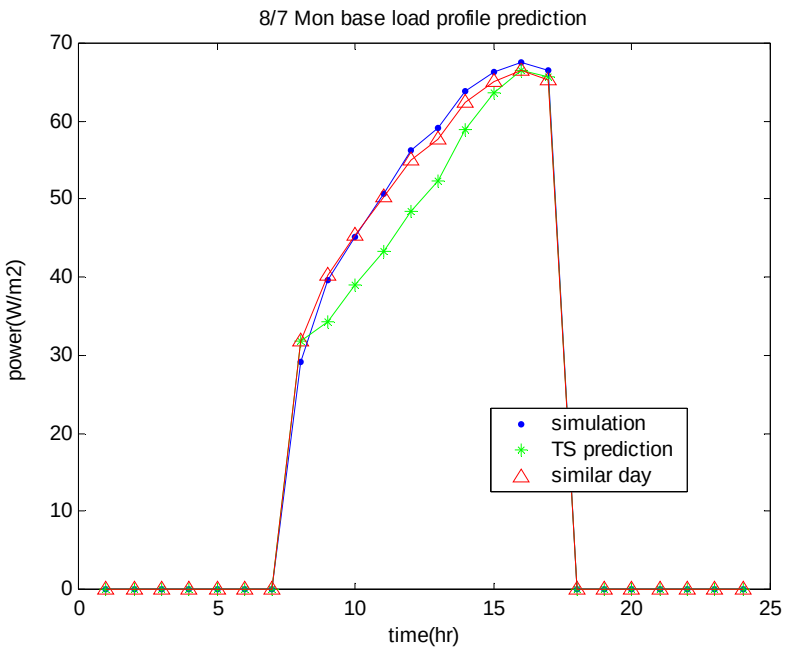


Figure 5.8 base load profile prediction improvement based on similar day

For example, the time series prediction for Monday August 7 is based on the previous day which is a Friday August 4. However, the prediction is less satisfying because in reality there is a weekend between August 4 and 7 but weekends' impact on Mondays is not considered in the base time series model, Eqn.5.5. We choose to use a previous Monday July 10 as the equivalent previous day and build the prediction on the combination of July 10 and August 4. Another reason we choose July 10 is that both July 10 and August 7 are fairly warm day: the highest is 26.5°C on July 10 and 26.7°C on August 7. Although we developed a model to correct the peak load, it still helps to use a previous day with the similar temperature range. We present the base prediction and the new similar-day-based prediction in Figure 5.8: the new prediction based on the combined day is better.

Another nice by-product of classifying and storing a set of typical days is that a profile from this set can be used as an approximation of another load profile to be predicted, given that they belong to the same category. A building operator, with weather forecast info, has the approximated load shape available, and a suboptimal load aggregation can be done based on operation experience.

5.3 Load reduction model

The time series model in section 5.2 provides us a base load profile prediction based on the previous day's real power profile, but there is nothing in the model that can explain the impact of the load control.

In this section, we will develop a simple model to describe how thermostat-based load shedding distorts the base load profile. We run the base case, together with four other cases with 1, 2, 3 and 4°C thermostat increases during hours 13 – 17. The system is off at night and no other load control strategies are applied. Then we analyze the hourly power difference as a function of thermostat increase and the base load. As previous analysis showed, increasing thermostat set points in the afternoon will reduce greatly the shedding-period power consumptions while slightly increasing the power uses at early hours due to the heat accumulated in the system at night. Figure 5.9 shows the power reduction from the base case for specific hours 11, 12, 13 and 14 over the entire training set. Hours before 11am gain little power increases and can be ignored. All the hours during the shedding period have the similar patterns: power reductions increase with base loads and thermostat increase. Late hours such as 16 and 17 incur more reduction than early hours such as 13 and 14. Figure5.10 looks at hour 14 in detail where power reductions are close to being linear with base loads. Figure5.11 provides a way to observe the relationship between power reductions and thermostat increases, where five specific hours on a summer day are presented with different thermostat increase in each case. It can be seen that for different hours, the impact of thermostat increases is different, but load reductions are an approximately linear function of the square root of thermostat increases.

Based on the observations, we propose a simple linear load reduction model shown in Eqn.5.7, where

$$pdiff = b_{i,0} + b_{i,1}(\Delta T)^{0.5} + b_{i,2}B \qquad \text{Eqn.5.7}$$

Where

$pdiff$	Power reduction from the base case
ΔT	Thermostat set point increase
B	Base load

We reorganize the training data and make it specific-hour oriented. Hours before noon are affected little, so we focus on the period of hours 12-17. Recognizing that load reduction potential differs from hour to hour, training data are regrouped targeting each hour during the period of 12-17. Again, the glm function in SPlus is used for regression. The process is automated so that multiple hours can be studied easily. Appendix D.1 has the SPlus code. Table 5.4 shows the regression results. The first three columns are regression coefficients and the last column of residual variance says that how much of the training variance can be explained by the model. A small number is a necessary condition for good predictability. Table 5.5 gives the two standard deviation range of the model's prediction over the testing set for each of

hours 12-17. We consider ± 2.2 W/ m^2 is acceptable given that the average load reduction magnitude over all thermostat increases is 4 W/ m^2. It is very difficult to make the load prediction error and the load reduction error one magnitude less than the load itself. Figure5.12 illustrates the prediction performance through hour 14 by comparing the prediction and simulation values in the testing set for different thermostat increase. A perfect prediction would lead to a 45 degree line. The model did fairly well with large temperature increases, but not as well with a 1°C increase.

Table5.4 SPlus regression results for load reduction model at different hours

	Intercept	deltaT^0.5	base load	residual variance
hour 12	-2.40	0.98	0.04	19.5%
hour 13	19.47	-6.43	-0.42	9.2%
hour 14	17.31	-7.35	-0.31	5.4%
hour 15	16.53	-8.01	-0.26	3.7%
hour 16	14.39	-8.43	-0.20	2.9%
hour 17	12.56	-8.22	-0.16	3.0%

Table5.5 Load reduction model prediction errors in testing data set

testing data prediction errors	mean - 2*stdev (W/ m^2)	mean + 2*stdev (W/ m^2)
hour 12	-1.2	0.8
hour 13	-2.9	3.5
hour 14	-2.2	2.5
hour 15	-2.0	2.1
hour 16	-2.1	1.9
hour 17	-1.9	1.6
overall	-2.2	2.2

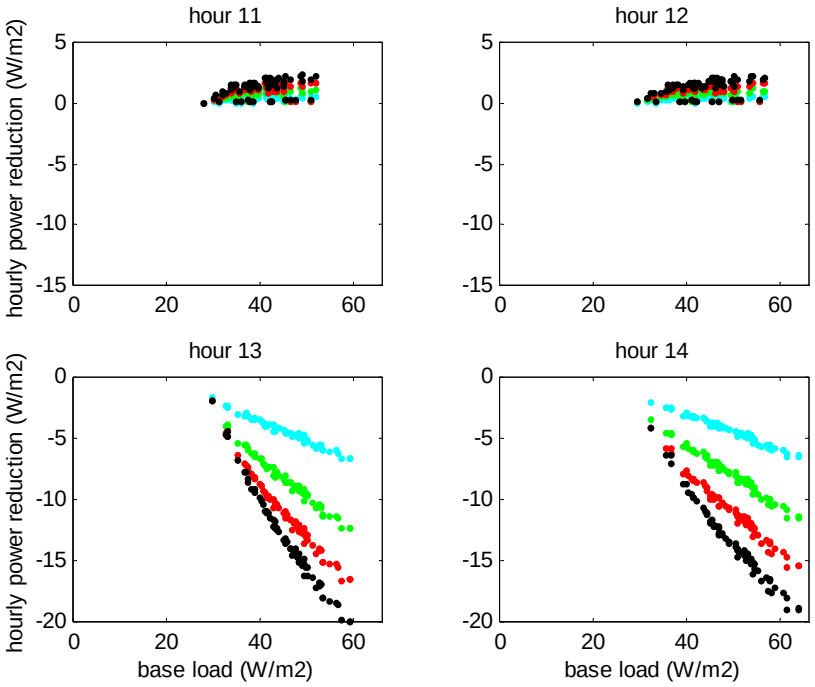


Figure5.9 Power difference from the base case at hours 11, 12, 13 and 14 vary with base load and thermostat change

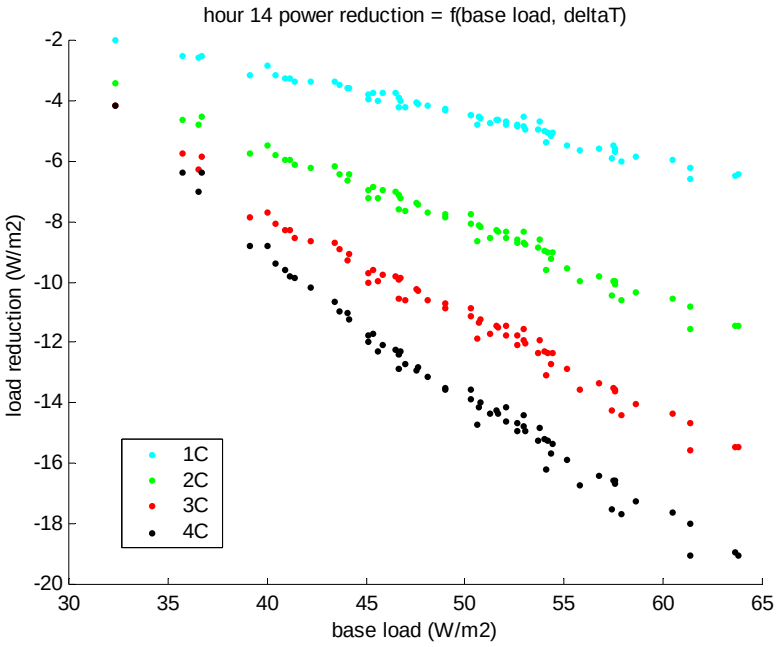


Figure5.10 Power difference from the base case at hour 14 vary with base load and thermostat change, a blow-up of the fourth graph in Fig.5.9

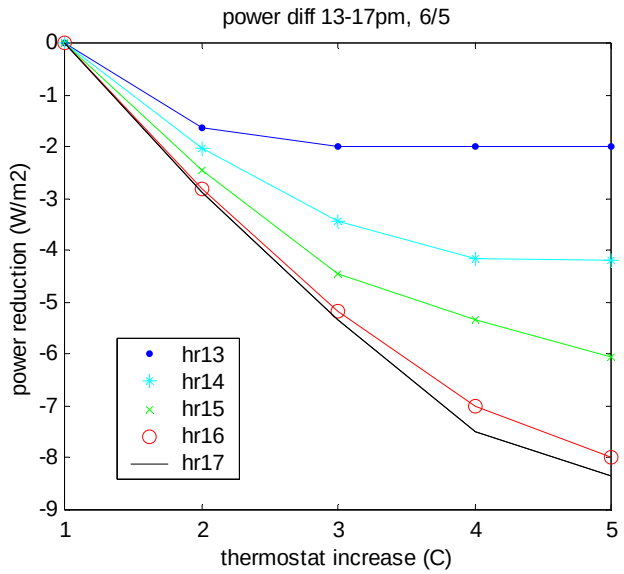


Figure5.11 Power difference from the base case at hours 13-17 vary with thermostat change, June 2, a section view of Figure5.10 with fixed base load

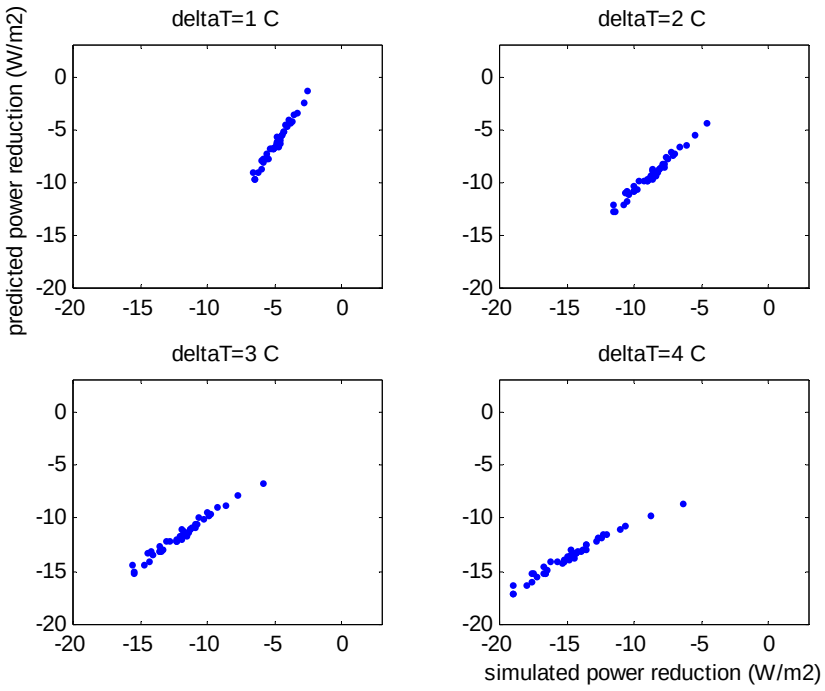


Figure5.12 Hour14 power reduction prediction vs. simulation (real values)

Modeling power reductions for a general load control strategy is more complicated than in this specific thermostat-base case. The key thing is the correlation and interdependence between hours. For example, if the thermostat set point is increased at hour 14 alone, hour 15 will certainly see a set-back spike, and hours 16 and 17 will experience a small amount of power increase as well. The challenge is how to represent the large number of possible schedule combinations, especially when a longer control period is involved. Integer programming might be used to assign an integer variable to each hour indicating whether this particular hour incurs a set-back spike. Then we could run regression to find the responding factor of this specific hour as a function of a variety of inputs. We ignore the set-back issue in this model by applying the load shedding, regardless the magnitude, to all the afternoon hours. Therefore, the scheduling modeling is not an issue for this case, but we understand that we are trading optimality for simplicity: the constraint we apply to the process could cost us optimal or good solutions.

5.4 Comfort model

Similar to load reduction, we will in this section develop a model that describes how thermostat-based load shedding degrades the service level compared to in the base case. Different from the load models, we will not develop a full model of predicting base comfort level on an hourly basis throughout the day. Instead, we will develop a maximum PMV estimate, which, together with an assigned PMV upper bound, will set an upper bound for the maximum PMV increase. Different from load profiles, PMV values are determined mostly on the individual hour basis as a function of air and wall temperatures, and scheduling is of little importance. That is why limiting the maximum PMV is enough to guarantee that the overall service level is under control. Therefore, the model is to bridge load shedding parameters with PMV values change.

Correlations between hours are not an issue for PMV modeling. For any hour that is not in the load shedding period, its PMV value is in line and is not affected by previous PMV changes. For hours during the shedding period, scheduling matters. For example, a person feels worse in the hour 17 when the thermostat-based shedding applied to hours 14-17 than at the same hour 17 but with load shedding only applying to 16-17, given the same temperature set point increase. This scheduling impact is rather subtle and is not considered explicitly in the current research. But a longer shedding period corresponds to higher mean radiant temperatures, which is in turn reflected in the PMV increases. Overall, PMV increases are mainly a function of thermostat set points, provided that thermostat set points can be maintained at the desired level and indoor air temperatures are under control.

We continue to use June data as training data set and July and August as the testing data set. We first look at how PMV increases vary with afternoon hours' thermostat set points increased by 1 to 4°C. Figure 5.13 shows the PMV changes on June 21 as a function of time and thermostat set point increases; the shape of the graph is the same for most summer days. Hour12 is used as a reference because load shedding is applied to hours 13-17 and hour 12 is hardly affected. We assign the same temperature increase to all five hours. It turned out that the first hour of the load-shedding period has the smallest PMV increase while hours after incur increasing PMV changes. However, the differences between hours are not obvious, which makes it possible to ignore the specific hour and build a model for the entire shedding period as a whole. If we do so, the PMV increase estimate at the beginning of the shedding period would be overestimated. It is not a concern because overestimating PMV increases guarantees conservative and feasible results. We put an upper bound of PMV increase on all hours and the service level is maintained as long as the worst hour with the highest PMV increase is in line. In this sense, treating all hours as a whole is an over safe design.

Figure 5.14 shows how PMV increases vary with the base load. Data in the training set are regrouped according to the temperature change over the training period. The PMV increases have little to do with base loads, except when temperature increases are large and base loads are small, which is a rather small area and can be ignored.

We regress PMV increases to temperature set point increases according to Eqn.5.8, and the t values of the coefficients for building 1 are given in the parentheses. Table 5.7 reports the model prediction performance in the testing set in terms of the residual two-standard deviation range. A change of 0.3 in PMV is less likely to cause any comfort concern, so the model is sufficient for this case.

$$\Delta PMV_i = \underset{(-14.2)}{c_{i,0}} + \underset{(76.9)}{c_{i,1}} \times \Delta T_i \qquad \text{Eqn.5.8}$$

Where, ΔPMV represents the PMV increase and ΔT the thermostat set point increase

A further look at how maximum PMV values vary with peak loads in Figure 5.15 suggests a strong linear relationship between the two. We take advantage of this linear relationship and project tomorrow's maximum PMV based on today's maximum PMV, today's peak load and tomorrow's peak load. Note that tomorrow's peak load is an estimate from previous load models.

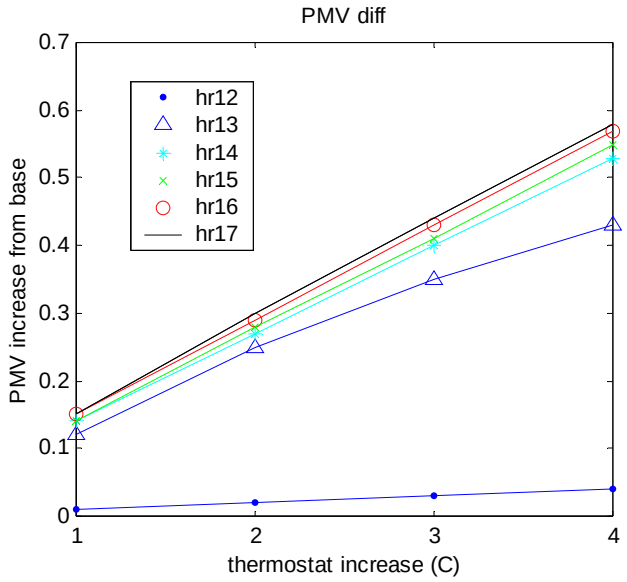


Figure5.13 PMV difference from the base case at hours 12-17 vary with thermostat change,6/21

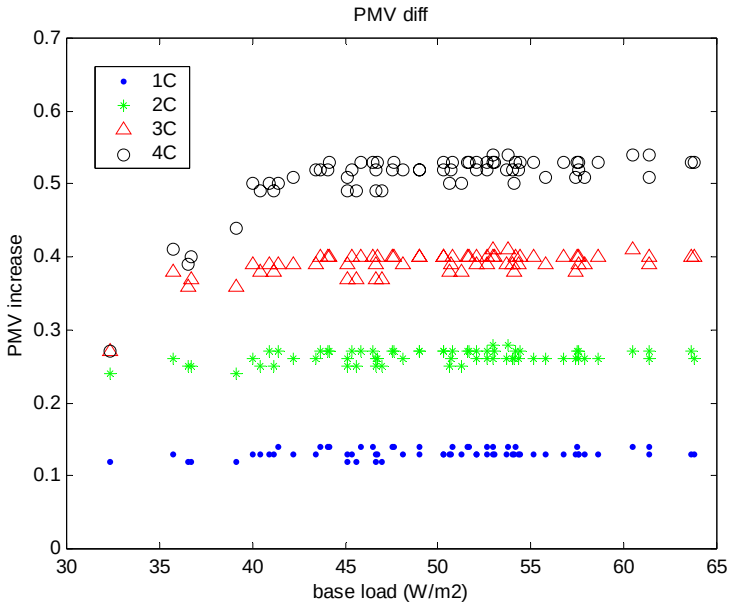


Figure5.14 PMV differences from the base case at hour 14 vary with base load under different thermostat change

Table5.6 PMV model prediction errors in testing data set

Delta PMV prediction errors	Mean - 2*stdev	mean + 2*stdev
deltaT = 1	-0.25	0.15
deltaT = 2	-0.21	0.23
deltaT = 3	-0.19	0.34
deltaT = 4	-0.18	0.47
overall	-0.25	0.34

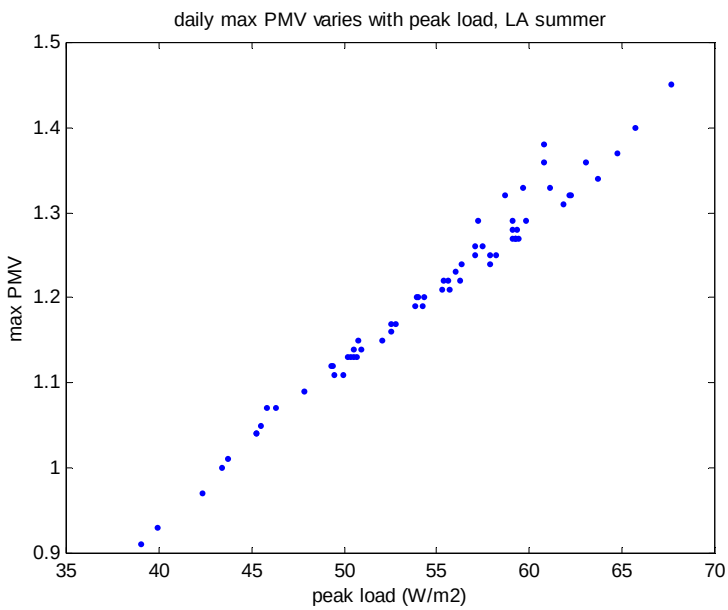


Figure 5.15 Relationship between daytime peak PMV values and peak load in summer, LA

5.5 A nonlinear central optimizer

The previous three sections are devoted to developing building blocks of a simplified building simulator. Now we are ready to assemble these parts and based on them build a nonlinear optimization solver. These building blocks, including the base load predictor, the peak load corrector, the load reduction model and the PMV increase model, are plugged into Eqn.5.3 and we obtain the final model in Eqn.5.9. All statistical results are written into the code of this mathematical formulation.

Eqn.5.9 is fairly easy for a commercial nonlinear solver to handle. The cost function and most constraints are linear, and the only nonlinear constraint is quadratic, a “soft” and computationally friendly one. We

code this model in AMPL (A Mathematical Programming Language), which is a platform for optimization problems with complicated constraints [ampl 2003] and is supported by a variety of linear and nonlinear solvers such as CPLEX, LOQO and MINOS. Programming in AMPL helps avoid coding an optimization algorithm from scratch. One only needs to formulate the problem as Eqn.5.9 does, and describe the cost function and constraints in APML, similar to Matlab programming.

There are plenty of commercial and/or research packages that can be linked to AMPL and to used in a certain model. We choose LOQO as the solver for the multi-building problem. LOQO [Vanderbei 1997] is a system for solving smooth optimization problems. It is based on an infeasible primal-dual interior-point method applied to a sequence of quadratic approximations to the given problem. The problems can be linear or nonlinear, convex or nonconvex, constrainted or unconstrained. The only real restriction is that the functions defining the problem be smooth, which is, they should be twice continuously differentiable at every point visited by the algorithm. See Appendix D.2 for the AMPL code.

$$\min \quad z \qquad \textit{total peak load}$$

$$s.t. \quad \sum_{i=1}^{N} W'_{i,t} \le z, \forall t \qquad \textit{together with the}\ \text{cost}\ \textit{function to handle a}\ \min\max\ \textit{problem,}$$

$$W'_{i,t}\ \text{and}\ W_{i,t} - \textit{the real and base power use of building } i \textit{ at time } t$$

$$W'_{i,t} = \left(a_{i,1}W_{i,t-1} + a_{i,2}W_{i,t-24} + a_{i,3}W_{i,t-25}\right)\times \alpha_i \qquad \textit{linear base load prediction and peak load correction}$$

$$+ f\left(\left(\Delta T_{i,t}\right)^{0.5}, W_{i,t}\right), \forall i, \forall t \ge 2 \qquad \textit{nonlinear power reduction}$$

$$dPMV_{i,t} = g\left(\Delta T_{i,t}\right) \le \Delta PMV_i, \ \forall i,t \in [14,17] \qquad \textit{pmv prediction and comfort requirement}$$

$$\Delta T_{i,t} \in \left[\Delta T_{low}, \Delta T_{high}\right], \forall t, \forall i \qquad \textit{thermostat set point range}$$

Eqn.5.9

Where, α_i is the peak adjustment correction factor for building i, and ΔPMV_i is the PMV increase target, determined by the tomorrow's maximum PMV estimate and an assigned PMV upper bound. A target of PMV less than or equal to 1.5 is used throughout this research and is consistent with all previous studies.

To see how this model-based nonlinear optimizer performs, we first run the program for a single building without comfort constraints – total demand as the cost function and hour 14-17 thermostat set points as

control variables – as a test case. With an all-zero starting point, the AMPL code takes no time to get to (4, 4, 4, 4), the already-known global optimum for the temperature increases in hours 14-17.

We then run the program for two two-building cases: 1) two identical buildings of model type 1; 2) buildings 1 and 3. Results are reported in Table 5.7. Enumeration and model-based optimizer lead to comparable amount of peak load reduction and similar operations in terms of thermostat set point changes, although load predictions by our model are consistently off by about 10%. Figure 5.16 compares four profiles for the case with two type-1 buildings: the sum of two base loads from simulation, the aggregated load control results by Enumeration, the sum of two predicted base loads, and the sum of aggregated load control results by the optimizer. The predicted base and aggregated load profiles are off compared to the simulation and enumeration results, but the differences before and after aggregated load control are essentially captured.

We move to a case with five buildings in the aggregation pool: four building 1s and one building 3. Here Enumeration experiences some computational difficulties. The exhaustive matching process takes about 2 hours in Matlab, while the model-based optimizer still takes no time to do the computation. For a commercial solver, several tens of control variables together with linear and quadratic constraints are very easy to handle and the optimization is very efficient. The savings again are reasonably close: simulation and Enumeration together produce 9.2% peak reduction while our model is 6.8%, shown in Table 5.7. Overall, the operations decisions between these two methods are similar, but our model failed to show the fact that same models in an aggregation pool can act quite differently. All building 1s in our model act similarly and their thermostats increases are close, but building 1s in Enumeration choose to act on different schedules and therefore achieve more reduction. The nonlinear optimization model needs to be improved to handle the scheduling aspect better.

Table5.7 Model-based optimizer vs. Enumeration

		Peak shift & (W/ m^2)	Peak Reduction	Final results	Computation Intensity
bldg1 and bldg3	Enumeration	125 / hr 16 -> 112 / hr 12	10.4%	bldg1 (0,1,1,1) bldg3 (2,2,2,2) PMV <= 1.5	123 E+ simulations Requires simulation every time to do aggregation
	Model-based nonlinear solver	114 / hr 16 -> 104 / hr16	8.6%	bldg1 (0.5,0.5,0.5,0.5) bldg3 (3, 3, 3, 3) PMV <= 1.5	460 E+ simulations to prepare training data One time only optimization takes no time
bldg1 and bldg1	Enumeration	137 / hr 16 -> 128 / hr 16	6.6%	1st bldg1 (1,1,1,1), 2nd bldg1 (0,1,1,1) results very close to those with both at (1,1,1,1) PMV <= 1.5	64 E+ simulations Requires simulation every time to do aggregation
	Model-based nonlinear solver	128 / hr 16 -> 121 / hr 16	5.7%	both buildings (0.5, 0.5, 0.6, 0.6) PMV <= 1.5	460 E+ simulations to prepare training data One time only optimization takes no time
bldg1 bldg1 bldg1 bldg1 bldg3	Enumeration	330 / hr16 -> 299 / hr 16	9.2%	(0,1,1,1) (1,1,1,1) (0,1,1,1) (0,0,1,1) (2,2,2,2)	123E+ simulations takes 2 hours to do matching in matlab
	Model-based nonlinear solver	305/ hr16 -> 280 / hr16	6.8%	All 4 type-1 buildings (0.6, 0.5, 0.5, 0.5) and bldg 3 (2, 2.5, 3, 3) PMV <= 1.5	460E+ simulation optimization takes no time

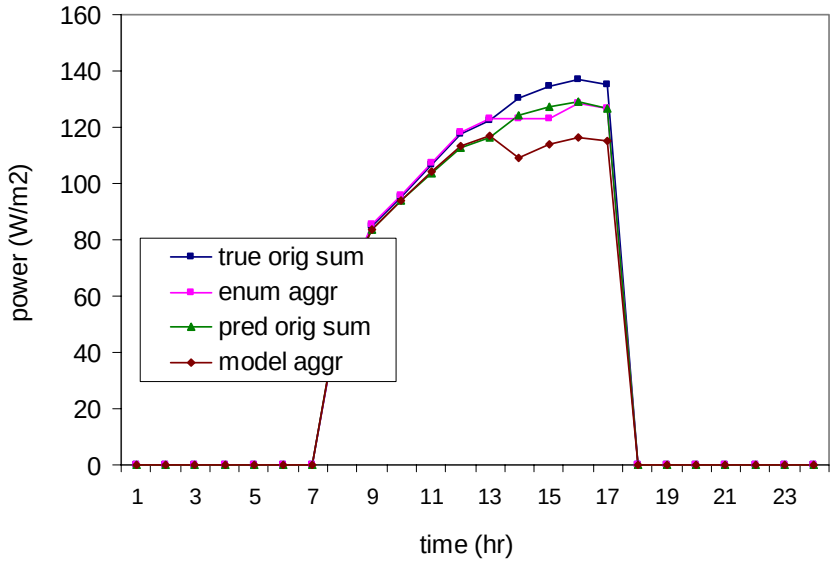


Figure 5.16 Aggregating 2 identical buildings with model type 1 model-based optimizer vs. simulation + Enumeration

We show in Chapter 4 that both multi-GA and Enumeration take hours when the size of the problem reaches 5. Running an aggregation case with ten buildings would be computationally formidable for both of them, but the model-based approach remains equally efficient. Table 5.8 shows a few 10-building cases with different mix based on buildings 1 and 3. All ten cases take little computation. We make the following observations: 1) the mix of a pool matters. Having equal number of building 3 and building 1 achieves 2% more peak reduction than having nine building 1s and one building 3; 2) individuals play different roles as a consequence of individual load profiles and thermal mass etc. The pool favors more building 3s than building 1s. To run the model-based optimizer, we need to have either simulation or experiment data, from which we can learn and train a unit simulator for each participant following the procedures developed in this chapter. We only trained models for buildings 1 and 3 in this research. An immediate next step is to train more models with different load profiles, so we can study the aggregated load control at a larger scale.

Table 5.8 Aggregation using model-based approach

cases	1 bldg1 / 9 bldg3	3 bldg1 / 7 bldg3	5 bldg1 / 5 bldg3	7 bldg1 / 3 bldg3	9 bldg1 / 1 bldg3
peak reduction	8.4%	8.6%	8.7%	8.0%	6.5%

5.6 Comparison of three approaches to the multi-building problem

To end this chapter, we compare all three approaches developed in this and the last chapters for solving the multi-building problem. Issues of computation, accuracy, implementation and future research opportunities of Enumeration, multi-GA and the model-based optimizer are presented in Table 5.9.

Table 5.9 Comparison between three approaches to the multi-building problem

approaches	Computation	Accuracy	Future research opportunities	Implementation
Enumeration	Efficient when n is small (<= 5) $\sim O(N^n)$ 2-bldg cases: 2mins 5-bldg cases: 2.5 - 8hrs	Good. Trade-off between reducing size of feasible sets and achieving accuracy	Speed up the matching process	Requires expert knowledge to form feasible sets
Multi-GA	Intensive in most cases $\sim O(n)$ 2-bldg cases: 1-5 hrs 5-bldg cases: 2.5 - 10hrs	Good. GA search performs better than Enumeration	Code the chromosome more efficiently. Both binary and real-value based	Requires input files, GA parameters, and function evaluator
Model-based optimizer	Efficient with any size of n $\sim O(1)$ both 2bldg and 5bldg cases solved immediately	Ok. Simplicity affects accuracy, but the improved operations are achieved and are close to being optimal	Improve the prediction models. Develop a load profile classification system	Train a set of models using either simulation or experimental data, and run the optimization program

CHAPTER SIX
NON-TECHNICAL ASPECTS OF THE LOAD CONTROL PROBLEM

Although we spent most of this thesis developing a technical analysis framework, we understand that policies, the economic environment, and human behavior play important roles in the energy problem. We will begin our discussion in this chapter with two disastrous events: the California Energy Crisis in 2000 and the New York Blackout in 2003. Then we briefly discuss non-technical issues with special attention to current load aggregation practice.

6.1 Energy crisis review

a California Energy Crisis 2000

In summer 2000, with severe weather leading to a huge demand in electricity for air-conditioning, and with a booming economy leading to fast-increasing use of electricity, California experienced the largest unplanned blackout of electricity since World War II. Electricity supplies fell to dangerous levels, and utilities cut power to more than 100,000 homes. From spring 2000 to the beginning of 2001, California's two biggest utilities spent $12 billion [Holson 2001, Egan 2001] more for power than they collected from ratepayers, and paid about $4 billion more for electricity in summer 2000 than they did in summer 1999. Utilities were in danger of bankruptcy, companies put off expansion plans, and residents braced for rate increases. The search for explanations for where money went reached a fever pitch.

The California power market went through deregulation in the late 1990's, among the first such experiments in the nation. As part of deregulation, utilities such as PG&E and SCE sold off power plants to outside power companies such as Duke Energy of Charlotte, N.C., and Reliant Energy of Houston, and became middlemen [Greenwald 2001]. The intent was to take advantage of plentiful power supply within and outside of the state through a bidding system to reduce the utilities' expense. Electricity was bought and sold in the California power exchange, where buyers and sellers bid for electricity to be used the next day. Demand was matched to supply by a new state agency, the California Independent System Operator.

However, a deregulated market is only partially free – the state did not allow utilities, the new intermediaries, to enter into long-term purchasing agreements for fear they would be locked into fixed-price contracts as deregulation market developed and prices dropped. Utilities could only buy power on the spot (cash) market and were exposed to spot pricing risk. At the beginning of deregulation, the prices at the spot market were low. The utilities willingly accepted the limitation, as well as a rate freeze with

customers until 2002, which made it impossible to transfer the buying risk to the selling side, or to hedge the downside risk. Unfortunately, the rapidly increasing demand, lack of generation capacity within the state, unusual weather, high natural gas prices and environmental restrictions pushed prices to a record high in mid-2000. The fact that utilities could only buy power at the last minute left California utilities and Independent System Operator in a desperate position with little negotiating power. In sum, California dismantled its private power-generating industry without securing adequate power supplies.

Much of the blame went to a few power companies that generate most of the power for California, such as Reliant Energy, Dynegy and Duke Energy. Whether these power companies manipulated the market and artificially pushed up the prices was disputed. These companies denied doing so. Houston-based Dynegy Inc. had one of the best performing stocks in the S&P 500 in 2000 with its shares more than tripling. Reliant Energy made $90 million in operating income from California alone during the third quarter of 2000 - more than twice what the entire subsidiary made in the period the year before [Holson 2001]. Although the companies argued that they faced high natural gas costs to run power plants for California and did not make much profit in the fourth quarter of 2000, a handful of studies suggested [Joskow 2001] that these power generators "engaged in behavior to drive prices above competitive levels."

There are many aspects to blame for this crisis. The deregulation plan was a very complicated but flawed one. Lacking thorough and objective analysis for the future power market, California utilities were not fully prepared to compete in a more finance-oriented market and did not possess in many situations a trader's mind which is very necessary. A power surplus in the early 1990's made the utilities hungry for deregulation, but it turned into a shortage, with the state's booming economy straining both generating capacity and the natural gas supplies that run many of the power plants. The state's onerous environmental regulations made it difficult and slow to build new power plants. Adding a few untimely generator shutdowns and cold, dry weather in the northwest that reduced the supply of hydroelectric power from that region, and the result was a recipe for a debacle.

Texas started its own power regulation after learning the lessons from California and paid more attention to designing a better market system. As many sources in Texas pointed out, Texas is different from California: it has a higher electricity surplus, plenty of natural gas and a diversity of power generations, and environmental regulations make it easier to build new power plants. After a year of deregulation implementation, surveys at the end of 2002 [TEC 2003] showed that the deregulation had been so far moderately successful but many customers were still not convinced that changing providers can bring much benefit because 1) lack of in-depth knowledge of deregulation and positive perceptions of electric

competition was in their way of making decisions; 2) most customers were satisfied with current power providers and rates and thought there was little need for competition. For deregulation, there is still a fairly long way to go in both improving the system and overcoming consumer inertia.

b Power blackout 2003

In August 2003, a massive power outage hit New York and most of New York state as well as other northeast cities including Toronto, Ottawa, Detroit and Cleveland. The blackout affected a total of 50 million people in the U.S. and Canada, and it took anywhere from two to 16 hours to restore the power supply for U.S. commercial and residential areas. By researching references [CNN 2003, Gibbs 2003], we try to understand why this happened and what associated problems are.

The exact reason was difficult to pinpoint, but experts had a fairly good idea why: the electrical system in the northeast and midwest consists of much capacity to generate power and too few means of moving it around smoothly. Over the past 10 years, electricity demand has jumped 30%, but transmission capacity has increased only half that much. Because everything is tied together, too much strain in one place can cause the whole system to malfunction. There have been a few disastrous blackouts in the past few years and the transmission system had been improved, but the safety margin built into the system has been eaten away by lack of investment in modernization.

A combination of market forces, political foot-dragging and the reluctance of residents to welcome high-voltage lines or towers in their neighborhoods has made it almost impossible to create a transmission system that can keep up with demand. Energy policy tends to have too much regulation in certain areas and too little in others: no one was requiring the utilities to upgrade the grid, and utilities were worried that if they did so voluntarily, they might not be allowed to charge enough to cover their cost. The government and the energy industry focused more on whether to deregulate or not deregulate. The blackout crisis looked sure to change the landscape where lawmakers decide on new energy bills.

Environmentalists and industry groups have very different opinions toward this multi-state summertime blackout. Environmentalists emphasize the importance of homeowners generating their own power, courtesy of clean, renewable energy sources. Industry officials speak instead about building new nuclear or conventional power plants or improving existing ones and delivering the power through a modernized distribution system.

The blackout helped draw more attention to the idea of "power parks" [Gibbs 2003] —communities or groups of homes that would generate their own energy courtesy of solar panels, wind turbines, fuel cells or natural-gas generators. The little clusters could be almost entirely self-sufficient and would have the freedom to disconnect from the larger network entirely if a regional crash was threatening to knock them off-line along with the bigger consumers. What has always kept this kind of energy free-lancing from becoming more than environmentalist daydreaming is that the necessary technologies have remained unreliable and prohibitively expensive.

Proponents of the policy hope that it will boost energy independence, but not everyone thinks that is a good idea. Because so much of the American economy is involved in the coal, petroleum and nuclear industries, walking away from them would set off severe economic dislocation. Many believe that decentralization will play some role in the energy industry of the future but could well be a minority player. After the 2003 fiasco, however, attitudes might change and plenty of consumers would be happy to see the whole system replaced—or at least dramatically improved. What the changes should be and how to implement them are the challenges.

6.2 Non-technical issues

a. Human behavior

The change of human energy-use behaviors would have a huge impact on building energy performance. For example, the saving s would be large if occupants shut off lights when lighting is not needed. We recognize that it may be difficult to implement behavior-related energy saving due to the associated inconvenience, but we believe it is worthwhile to let end users know the options available and ease of implementing those. The following measures are from the California Independent Systems Operator's website [CA ISO 2002]:

- Consider replacing old HVAC systems with new energy-efficient systems
- Install time clocks or setback-programmable thermostats to maximize efficiency
- Install locking covers on thermostats to prevent employee tampering with temperature settings
- Perform scheduled maintenance on units including cleaning condenser coils, replacing air filters regularly, and checking ducts and pipe insulation for damage
- Clean condenser coils and replace filters regularly
- Install ceiling fans
- Install blinds, or solar screen shades to cool the office
- Install reflective window film or awnings on all south-facing windows

- Close shades or blinds during early morning and late evening to reduce solar insulation heat gain
- Consider installing an air conditioning economizer to bring in outside air when cool outside
- For optimal energy savings, set thermostats at 78 degrees F for cooling in the summer and 68 degrees F for heating in the winter
- Install ceiling and wall insulation
- Insulate water heaters and supply pipes

b. Aggregation policy

Aggregation can be a promising business when the aggregator follows the policy, understands the electricity reconstructing market, and is able to identify potential customers and know them well. We look at the current situation, policies applied and potential development, and problems through aggregation cases in Ohio and Massachusetts [Brown 2002] [Alexander 2002]. Both states have opt-out aggregation which allows a municipality or a local branch of government to aggregate the load of some or all of its customers within its jurisdiction. The Texas electricity reconstructing site [TEC 2003] also has some information for aggregation where opt-out is not an option.

In Ohio, aggregation accounts for 85 percent of residential customer switching, 50 percent of commercial customer switching, 25 percent of industrial customer switching, a 17-percent discount on power prices in one town in northern Ohio, and discounts of between 1 percent and 15 percent as well as a guarantee of a "greener" power mix for another aggregated group of more than 300,000 people in northern Ohio.

Ohio's Northeast Public Energy Council (ONPEC) is the nation's largest aggregated group. It took advantage of the Ohio restructuring law's aggregation provisions by combining not only the load of the citizens of a single municipality, but the combined load of many municipalities. It is now a buying group representing 97 cities or townships and more than 300,000 people. Green Mountain Energy serves this buying group on a six-year contract that offers a single price option at a discount from what customers would otherwise pay for power. The savings vary from one customer to another, ranging from a high of 15 percent for a few customers to as low as 1 percent for others. Green Mountain's product is guaranteed to be cleaner than the average Ohio electricity product. It is a combination of 98 percent natural gas and nuclear and 2 percent alternate sources, such as wind.

In Massachusetts a smaller scale pilot aggregation program has yielded approximately 45,000 participants, discounts of 11 to 22 percent, or $3.50 to $7 per month for an average customer and a set of green power options available to participants

An opt-out aggregation is a new concept in power markets and, like any new concept, encounters policy and others barriers. The efforts in Ohio and Massachusetts have required tremendous patience, sophistication and dedication on the part of its organizers. Aggregation, in general, requires that policymakers make what can be controversial decisions about how to manage their competitive power markets. It does, however, offer a possibility of bringing the benefits of competition to smaller power users, who thus far have not seen much benefit to choosing a new power provider.

Brown [2002] concluded that the success of aggregation is tied to regulated retail prices, and wholesale prices remain an important determinant of how successful aggregation can be. Aggregation is likely to be most successful in higher priced areas, just as retail competition has been more successful in the parts of the country with the highest electricity rates. Wholesale power markets affect aggregation, just as they affect any retail power market. Massachusetts' situation demonstrates that rising or volatile wholesale prices can make it as difficult for a marketer to serve an aggregated group as it can be to serve an individual customer.

Aggregation can produce savings and can benefit adoption of alternative power sources. Aggregation appears to have given all participating Ohio customers in the aggregators' jurisdictions at least some access to competitively determined electricity prices. Price reductions have not been dramatic, although the benefits to participants in the ONPEC group have been broad, including access to alternative sources. The Massachusetts program has served as a new way to offer a portfolio of green products to consumers and to offer a new set of efficiency programs to the consumers.

One question we need to answer is how to choose participants in order that all participants benefit as much as possible. Porter [1998] pointed out that a company with a good load profile would probably avoid becoming involved in an aggregation group unless it is aggregating with businesses that have similarly favorable profiles. To the utility, the most important consideration is when you buy power rather than how much or at what load factor. For individual participants who pay separately, algorithms have been developed to allocate savings according to a customer's usage profile and volume. There is some debate as to whether aggregation – especially aggregation without load-profile preferential treatment – is a good deal for even the smallest end users.

c. Market Innovation and advance in financial markets

The power industry is undergoing rapid and significant change with the advent of deregulation. Electric power marketers have emerged, and consumers can shop around for best suppliers. The industry is adopting internet-based electronic reservation and trading systems that provide open access to all transmission services information for all market participants. The ideas of intelligent software agents and auction services have been explored by some researchers [e2i 1999, Reticular 1999, NYISO 2002]. The agents communicate and cooperate with each other and their owners to buy and sell electric power on their behalf. Ideas that have been explored include the dynamics of the electronic marketplace, proper vehicle for investigating the appropriate agent behaviors, buying and selling strategies and market algorithms necessary for use in an automated power marketplace.

Electricity is a tradable financial product. Major investment banks and many investment firms are investing in the energy market and trading electricity and natural gas. Some of them have been quite profitable. Aggregators are in fact playing a broker role in the sense that they are middlemen between end users and utilities and they can shop around for the best deal for the customers in the aggregation pool. More energy-based products and derivatives have been created, which helps expand the energy market. Through the use of electricity futures, consumers or load aggregators can purchase futures contracts to offset or hedge power prices for deliveries in advance. By doing so, the energy price volatility in a deregulated energy market can be mitigated. One of the biggest obstacles in dealing with the California energy crisis in 2000 is that the regulation policies required electricity only be purchased on a spot market. The California utilities had no way to avoid the volatility and ended up purchasing from the power companies at sky-high prices for the next day's use. Other energy-related financial instruments can help utilities, power companies and eventually end users better achieve their goals.

CHAPTER SEVEN
CONCLUSIONS

7.1 Research Conclusions

Summary

The entire thesis is devoted to improving building HVAC system operations through a variety of simulation and optimization methods. Based on extensive EnergyPlus simulations we have proposed some guidelines that can be applied to practical operations. We have developed a simulation-based Matlab GA environment and a model-based nonlinear optimization scheme that can be potentially used by sophisticated operators, and we have illustrated through our research that optimization methods have good potential in building research. Overall, with peak demand and/or real time pricing applied, we expect load aggregation to offer diversification opportunities among participants and to improve the overall load profile, and load shedding to change individual load profiles and enhance the aggregation performance.

Chapter 2 conducts EnergyPlus parametric studies of load shedding in a single building and compares different strategies in a VAV system. The simple and efficient load shedding method is to increase thermostats, which can lead to about 10W/m^2 peak load reduction. An alternative is to reduce fan capacity and if necessary increase supply air and chilled water temperatures. Temporarily shutting off the chiller for an hour in this building is infeasible as it leads to severe comfort problems. The duration of the load shedding period and the start time affect energy savings. Fan-based night cooling is shown to be more energy efficient than chiller-based for this VAV model in Los Angeles, corresponding to 4-9 W/ m^2, or 10% peak load reduction. Chiller-based night cooling achieves 12% peak-load reduction in Austin, better than fan-based. Night cooling is recommended if the plant and/or fan are programmable and whenever weather permits. It is important to identify buildings that are more appropriate for load shedding.

In Chapter 3, thermostat-based and night-cooling-based load control strategies are optimized for a single building. Two simulation-based optimization algorithms, Direct Search and Genetic Algorithm, are implemented and compared regarding their convergence, complexity and accuracy. Optimization studies in this chapter are highly dependent on the simulation performance. The cost function structure for single building optimization presents nonlinearity and discontinuity. When few variables are involved and cost

functions are relatively smooth, DS converges faster than GA, but GA has the advantage when the size of the problem becomes large. For the particular problem of setting up thermostats for the base model, DS works well with up to five variables but fails when the size increases to ten. The user-friendly platform of GenOpt makes it possible for building operators to use it in practice, but it is important to include in the package global-oriented optimization algorithms such as GA. Being able to handle a variety of cost functions is also necessary. An EnergyPlus-based Matlab-GA environment is developed, in which we conduct a variety of GA optimizations in a systematic way.

Chapter 4 shows the effectiveness of load aggregation through two simplified approaches: smart Enumeration and multi-GA. Smart Enumeration enumerates representative feasible solutions from EnergyPlus simulation but avoids unnecessary computation and reduces the size of the feasible sets with the help of expert rules. Multi-GA is an extension of single-building GA, and the function evaluation is a series of EnergyPlus simulations for all the aggregation participants. Aggregation is applied to a small pool of two or three buildings to reduce the total peak demand or the total cost. For the specific building under study, we achieve a peak load reduction of 2 – 14% with thermostats as control variables. This peak load reduction consists of the reduction from individual load control and from aggregation. The contribution of the aggregation ranges from 30% to 50% for the two-building and three-building cases. We see a 27% peak load reduction and approximately 20% cost reduction in a two-building case with fan-based night cooling enabled. Exact numbers for savings depend on the correlation and interdependence of the individual participants. For night cooling, individual improvement is huge, and the benefit from the pure aggregation can be ignored. Enumeration is more efficient at a small scale, but the computation of the exhaustive search increases exponentially with the size of the pool. Multi-GA has relative advantage when the problem size becomes large because its computation increases approximately linearly with the size. For the three building models used, 5 buildings is the break point between Enumeration and Multi-GA.

Chapter 5 solves the multi-building optimization problem using a model-based approach. A simple time-series model is used to represent building dynamics, and regression is applied to correct the peak load and the entire power profile. Two regression models are developed to describe load reduction and comfort degradation respectively from the base case due to load control. In the end, a nonlinear optimization scheme brings all these parts together and solves this soft-nonlinear-constraint problem in seconds. For the peak load optimization, the nonlinear optimizer manages to achieve similar optimal actions as those in Enumeration and GA, but the base load and savings predictions are off by a certain amount as a price of

the simplicity. For a large number of buildings, the model-based nonlinear optimization method remains computationally efficient, given that the constraints remain linear or soft nonlinear.

Key issues

a. EnergyPlus simulations

Most of this thesis is based on EnergyPlus simulation. Therefore, its accuracy, consistency and computational efficiency are very important to us. As a complex software package still under development and improvement, EnergyPlus is a great help to our research and also a challenge in the sense that we need to understand the complexity and overcome the potential problems. To the base VAV model, we have done basic validation such as heat and mass balances and qualitative checking on parameter trends. In reality, field test will be desired to further verify the model. We fixed a few EnergyPlus problems along the way with the help of the EnergyPlus development team [EnergyPlus Support 2003]. EnergyPlus so far is still a computation engine. We developed a data post-processor in VBA to handle a variety of cost functions.

b. The building dynamics model in multi-building optimization

Developing building dynamics models is not a new topic. The difficulty here is to come up with a modeling scheme that can consider both the time dimension at the individual building level and the space dimension at the aggregation level. Quite a few research results available from references focus on the time dimension and have great details of a specific system and its components. It is not simple to handle a number of such models at the same time. On the other hand, aggregation examples are plentiful: load patch and water heater control in the direct load control field [Chen et al.1995, Kurucz et al.1996] and mean-variance portfolio optimization in the investment community [Markowitz 1952], etc. In these aggregation examples, individuals are extracted to a static point with several representative properties, and the individual dynamics are ignored. Timing is key to our research – when the peak shows up in the aggregated case compared to in the individual cases. We choose the structure of a central optimizer and time-series based individual simulators and believe it has a good balance in handling both time and space. Each of the steps is quite simple, but enough to meet the goal discussed above.

c. Which buildings are more appropriate to be aggregated?

Without load control, the aggregation performance is determined by the diversity of individual profiles and differences between individuals. Diversification guarantees the aggregated profile is flatter than any individual ones, but large performance discrepancies between individuals might hurt certain individuals' initiatives of participating in aggregation. With aggregated load control, deciding whether or not a

building is appropriate for aggregation is two-fold: at the individual level, whether the building responds well to load control and shedding; at the aggregation level, whether this building can cooperate with others to achieve good overall savings. This thesis focuses on developing an analytic process and assumes that all the individual participants are given, but choosing the right buildings is a very important task for a load aggregator in reality.

7.2 Future work

a. Load prediction model as a whole

Currently, the load prediction errors are comparable in magnitude to energy savings. The savings of the models in this research are moderate due to the specific system, but further improvement with the prediction is still needed. We simply build today's profile on yesterday or a similar day's, and correct the predicted profile using an improved peak load. The base load prediction model could incorporate more physics but still be useful for multi-building optimization. The load reduction model does not consider the setback recovery situations, and it is an immediate next step.

b. Discrete variables

Being able to deal with discrete variables is key to some load control strategies such as scheduling night cooling and turning off the chiller for a short period of time, e.g. less than an hour. In addition, discrete variables can be used to represent different operation modes. For example, we can present different day types, Monday through Friday, using five 1/0 variables within the same model and require the sum of these five 1/0 variables to be 1 as a constraint. In this research, we converted the fan status in the night cooling problem to a continuous variable specifically for California. But we need a more general approach to deal with discrete variables. It would help to make GA handle both binary and real value based coding, which can be an immediate research topic. In addition, the nonlinear optimizer cannot deal with integer variables through the interior point method. Algorithms such as branch-and-bound can be brought into the framework.

c. Peak demand generalized

In practice, demand charge is applied on a horizon longer than a day, such as a month or a year. We simplify the analysis by assuming the rate structure applied to a single day. We wish to point out: 1) this simplification overestimates the peak demand because a monthly rate structure is applied to a day's profile; 2) the analysis can be generalized to a monthly scale. The peak load pricing research [Raymond 1971] in economics will help this expansion. Peak load pricing deals with the stochastic demand in a time series fashion.

d. Sensitivity analysis

Most of this thesis work assumes accuracy in weather and load data. It does not quantify how the results will change and how robust the analysis is when the inputs, such as outdoor temperature measurement, change. Sensitivity analysis is critical in practice and is one of the next steps.

A related topic is that whether our results and/or analysis can be generalized to other building types. Much of our work is based on short-term oriented load shedding strategies, such as thermostat set points. It might not be appropriate to expand into the situations where mid to long-term load planning is required. For example, aggregating hotels, offices and grocery stores would have a much longer control period than in our research. The methods proposed in this research will need to be improved to handle a larger parameter space.

e. Stochastic factors

Throughout the research, we assume all the factors static in order to focus on the major issue of aggregation and optimization. However, we recognize that weather, human behavior and equipment use are all stochastic factors. An ideal solution should have the stochastic aspect in the optimization framework. We argue that for many commercial buildings, the load patterns are relatively stable over days and weather forecast, given our setting of predicting a day in advance, is fairly accurate. Therefore, ignoring the stochastic aspect is acceptable at this point, but further studies incorporating stochastic factors would be helpful. Stochastic programming [Freund 2002] can be helpful as long as we categorize the stochastic factors in advance with assigned probability.

REFERENCES

Ahmed, O., J.W.Mitchell, and S.A.Klein. 1996. "Application of general regression neural network in HVAC process identification and control," *ASHRAE Transactions,* 102(1).

Ahmed, O., J.W.Mitchell, and S.A.Klein. 1998. "Feedforward-feedback controller using general regression neural network for laboratory HVAC system: part I - pressure control," *ASHRAE Transactions*. 104(2).

Alexander, B., 2002. "An analysis of residential energy markets in Georgia, Massachusetts, Ohio, New York and Texas," National Energy Affordability and Accessibility Project (NEAAP).

AMPL, 2003. http://ampl.com.

Anstett, M., and J.F. Kreider. 1993. "Application of neural networking models to predict energy use," *ASHRAE Transaction,* 99(1).

Armstrong, P., 2004. "Robust inverse models for fault detection and optimal thermal control," Ph.D. dissertation. Department of Architecture, Massachusetts Institute of Technology, Cambridge, MA.

ASHRAE. 1999. *Handbook of HVAC Applications*. ASHRAE, Atlanta.

Bertsekas, D.P. 2000. *Dynamic Programming and Optimal Control*, Vol.1. Athena Scientific.

Box, G. and G. Jenkins. 1976. *Time Series Analysis: Forecasting and Control.* Holden-Day.

Braun, J. S. Klein, W. Beckman, and J. Mitchell. 1989a. "Methodologies for optimal control of chilled water systems without storage," *ASHRAE Transactions*. 652-662.

Braun, J. S. Klein, J. Mitchell, and W. Beckman. 1989b. "Applications of optimal control to chilled water systems without storage," *ASHRAE Transactions*. 663-675.

Braun, J., K. Montgomery, and N. Chaturvedi. 2001. "Evaluating the performance of building thermal mass control strategies," *Intl. J. of HVAC&R Research.* Vol.7, 403-428.

Brown, M., 2002. "An analysis of Opt-Out aggregation in Massachusetts and Ohio," NEAAP, 2002. http://www.ksg.harvard.edu/hepg/Papers/

CAISO, *California Independent System Operator*, 2003. http://www.caiso.com/

Chen, J., F. Lee,, and R. Adapa. 1995. "Scheduling direct load control to minimize system operational cost," *IEEE Transactions on Power Systems.* Vol.10, No.4.

Chow, T., Z. Lin, C. Song, and G.Zhang. 2001. "Applying neural network and genetic algorithm in chiller system optimization". *Seventh International IBPSA Conference*, 1059-1065, Brazil.

CNN, "Major power outage hits New York, other large cities," *CNN.com*, Aug 2003.

Crawley, D., F. Winkelmann, L. Lawrie, and C. Pedersen. 2001. "EnergyPlus: new capabilities in a whole-building energy simulation program," *Seventh International IBPSA Conference*, Brazil.

Conniff, J. 1991. "Strategies for reducing peak air-conditioning loads by using heat storage in the building structure.," *ASHRAE Transactions* 97(1): 704-709.

Constantopoulos, P., F. Schweppe, and R.C. Larson. 1991. "ESTIA: A real-time computer control scheme for space conditioning usage under spot electricity pricing," *Computers and Operation Research*. 18(8) 751-765.

Curtiss, P., J. Kreider, and M. Brandemuehl. 1993. "Adaptive control of HVAC processes using predictive neural networks," *ASHRAE Transactions. 99(1).*

D'Cruz, N. and A. Radford. 1987. "A multi-criteria model for building performance and design," *Building and Environment*. Vol22, No.3, 167-179.

E2i, Prototype intelligent software agent for trading electricity, Electricity Innovation Institute, 1999.

Egan, T. 2001. "California's panic was moneymaker for energy sellers," The *New York Times*, Feb.

Effler, L. and G. Wagner. 1992. "Optimization of load procurement and energy management, *IEEE Transactions on power systems*," Vol.7, No.1, 327-333.

EnergyPlus Manual, 2001. http://simulationresearch.lbl.gov/

EnergyPlus Support, 2003. http://groups.yahoo.com/group/EnergyPlus_Support/

Freund, R. 2002. Systems Optimization lecture notes. MIT.

GenOpt Manual, 2002. http://simulationresearch.lbl.gov/GO/index.html

Gibbs, N., "Lights out," *Times magazine*, Aug 2003.

Gibson, G. 1997. "A supervisory controller for optimization of building central cooling systems," *AHSRAE Transactions*. 486-493.

Greenwald, J., "The new energy crunch," *Time magazine*, Jan. 2001.

Gross, G. and F.Galiana. 1987. "Short-term load forecasting," *Proceedings of the IEEE*. Vol.75, No.12.

Haves, P., and L. Gu. 2001. "Guideline for the operation of demand-responsive HVAC systems," LBNL.

Henze, G., R. Dodier and M. Krarti. 1997. "Development of a predictive optimal controller for thermal energy storage systems," *Intl. J. of HVAC&R Research*. Vol.3, No.3, 233-264.

Holson, L. "Trying to follow the money in California's energy mess," T*he New York Times*, Jan 2001.

Hsu. Y. and Su. C., 1991. "Dispatch of direct load control using dynamic programming," *IEEE Transactions on Power System.* Vol.6, No.3.

Hung, S.-Y., T.Liang, and V.Liu. "Integrating arbitrage pricing theory and artificial neural networks to support portfolio management," *Decision Support Systems*. Vol.18, 301-316, 1996.

Ilic, M, J. Black, and J. Watz. "Potential Benefits of Implementing Load Control," *Proceedings of the IEEE PES Winter Power Meeting*, New York City, NY. January 27-31, 2002.

Jorge, H., C. Antunes, and A. Martins. 2000. "A multiple objective decision support model for the selection of remote load control strategies," *IEEE Transactions on Power Systems.* Vol.15, No.2.

Joskow P., 2001, "California's electricity crisis," *Oxford Review of Economic Policy*. Nov, 2001.

Kaastra, I., and M. Boyd. "Design a neural network for forecasting financial and economic time series," *Neurocomputing*. 1996 (10) 215-236.

Keeney, K. and J. Braun. 1996. "A simplified method for determining optimal cooling control strategies for thermal storage in building mass," *Intl. J. of HVAC&R Research*. Vol.2, No.1, 59-78.

Kluger, J. 2003. "Can America free itself from the grid and democratize energy?" *Times magazine*. Aug.

Kurucz, C., D. Brandt, and S. Sim. 1996. "A linear programming model for reducing system peak through customer load control programs," *IEEE Transactions on Power Systems.* Vol.11, No.4.

Lan, M.-S., and S. Chand. 1990. "Solving linear quadratic discrete-time optimal controls using neural networks," *Proceedings of the 29th Conference on Decision and Control*, Hawaii.

Lewis, R., V. Torczon, and M. Trosset. 2000. "Direct search methods: then and now," *Journal of Computational and Applied Mathematics.* Vol. 124, No.1-2, 191-207.

MacArthur, J., A. Mathur, and J. Zhao. 1989. "On-line recursive estimation for load profile prediction," *ASHRAE Transactions* 95(1): 621-628.

Markowitz, H. 1952. "Portfolio selection," *Journal of Finance*. Vol.7, No.1, 77-91, March.

Michalewicz, Z. C. Janikow and J. Krawczyk. 1992. "A modified genetic algorithm for optimal control problems," *Computers Math. Application.* 23(12), 83—94. Hawaii.

Morris, F., J. Braun, and S. Treado. 1994. "Experimental and simulated performance of optimal control of building thermal storage," *ASHRAE Transactions* 100(1): 402-414.

Narendra, K. and Parthasarathy, K. 1990. "Identification and control of dynamical systems using neural networks," *IEEE Trans. on Neural Networks*, 1(1):4, 4-27.

Neuralworks Professional II/PLUS, 2003. http://www.neuralware.com/products_pro2.jsp

Neural networks toolbox, 2003. http://mathworks.com

NeuroSolutions, 2003. http://www.nd.com/

Rabl, A., and L. Norford. 1991. "Peak load reduction by preconditioning buildings at night," *Int. J. of Energy Research*. Vol.15. 781-798.

Norgaard, M., O. Ravn, and N.K.Poulsen. 2001. "NNSYSID and NNCTRL tools for system identification and control with neural networks," *Computing and Control Engineering Journal*, Feb. 2001, pp 29-36.

NYISO, New York Independent System Operator, 2003. http://www.nyiso.com/

Papalexopoulos, A. and Hesterberg, T. 1990. "A regression-based approach to short-term system load forecasting," *IEEE Transactions on Power System.* Vol.5, No.4.

Polak, E., and M. Wetter. 2001. "Generalized Pattern Search Algorithms with Adaptive Precision Function Evaluations," LBNL Technical Report 52629.

Porter, A., 1998. "On power load aggregation," *The Magazine of Total Supply Chain Management*.

Raymond, J. 1971. "Airport noise and congestion: a peak load pricing solution," *Applied Economics*. Vol.3, No.3.

Reddy, T., and L. Norford. 2002. "Building operation and dynamics with an aggregated load," ASHRAE 1146RP Final report.

Reticular System Inc., 1999. "Using intelligent agents to implement an electronic auction for buying and selling electric power," http://www.aesc-inc.com/download/epri.pdf

Rossi T. and J. Braun, "Minimizing operating costs of vapor compression equipment with optimal service scheduling," *HVAC&R Research*. Vol.2, No.1, 3-25, 1996.

Sage, A.P. and C.C.White III, *Optimum Systems Control*, Prentice-Hall, Inc., 1977.

Seem, J. and J. Braun. 1991. "Adaptive methods for real-time forecasting of building electrical demand," *ASHRAE Transactions* 97(1): 710-721.

Seem, J., P. Armstrong, and C. Hancock. 1989. "Comparison of seven methods for forecasting the time to return from night setback," *ASHRAE Transactions* 95(2): 439-446.

Specht, D.F. 1990. "Probabilistic neural networks," *Neural networks*, Vol.3, 109-118, 1990.

Specht, D.F. 1991. "A general regression neural network," *IEEE Transactions on Neural Networks 2(6).*

TEC, *Texas Electric Choice*, 2003. http://www.powertochoose.org/

Tsitsiklis, J., B. Van Roy, D. Bertsekas, and Y. Lee. 1997. "A neuro dynamic programming approach to retailer inventory management". *Proceedings of the IEEE Conference on Decision and Control*.

Johnson, R. and J. Wichern. 2002. *Applied Multivariate Statistical Analysis.* Prentice Hall; 5th edition.

Wright J. and H. Loosemore. 2001. "The multi-criterion optimization of building thermal design and control," *Seventh International IBPSA Conference*, 873-880, Brazil.

Wright, J. and R. Farmani. 2001. "The simultaneous optimization of building fabric construction, HVAC system size, and the plant control strategy," *Seventh International IBPSA Conference*, 865-872, Rio de Janeiro, Brazil.

Vanderbei, R. 1997. "LOQO users' manual. v.3.10," Princeton University.

Xing, H. and L. Norford. 2003. "Load-control assessment methodologies: multi-building optimal control," deliverable 3.5.3(a) (b) reports. Submitted to the Architectural Energy Corporation.

Appendix A.1 EnergyPlus code for the base model

Download at http://mit.edu/hxing/www/3vav.idf or email to hxing@alum.mit.edu for the base model.

Appendix A.2 Core EnergyPlus code

1. We vary the densities of major construction materials to create models with different thermal mass
2. An outside air system is added to enable free cooling
3. Thermostat set point schedules, fan schedules and chiller schedules are varied during parametric studies and simulation-based optimization
4. reduce the fan static pressure rise for capacity reduction

```
MATERIAL:Regular,
  C4 - 4 IN COMMON BRICK,  !- Name
  Rough,  !- Roughness
  0.1014984,  !- Thickness {m}
  !- Conductivity {W/m-K}, !- density, !- Specific Heat {J/kg-K}
!- base material
  0.7264224, 1922.216,   836.8000,
!- high thermalmass (2mass)
! 0.7264224, 3844.432,   836.8000,
!- low thermal mass (halfmass)
! 0.7264224, 961.108,   836.8000,
  0.9000000,  !- Absorptance:Thermal
  0.7600000,  !- Absorptance:Solar
  0.7600000;  !- Absorptance:Visible

MATERIAL:Regular,
  C10 - 8 IN HW CONCRETE,  !- Name
  MediumRough,  !- Roughness
  0.2033016,  !- Thickness {m}
  !- Conductivity {W/m-K}, !- density, !- Specific Heat {J/kg-K}
!- base material
  1.729577,  2242.585, 836.8000,
!- high thermalmass
!   1.729577,  4485.17, 836.8000,
!- low thermal mass
!   1.729577,  1121.293, 836.8000,
  0.9000000,  0.6500000, 0.6500000;

MATERIAL:Regular,
  C12 - 2 IN HW CONCRETE,  !- Name
  MediumRough,  !- Roughness
  5.0901599E-02,  !- Thickness {m}
  !- Conductivity {W/m-K}, !- density, !- Specific Heat {J/kg-K}
!- base material
  1.729577,  2242.585,  836.8000,
!- high thermalmass
!   1.729577,  4485.17,  836.8000,
!- low thermal mass
!   1.729577,  1121.293,  836.8000,
  0.9000000, 0.6500000, 0.6500000;
```

```
!nightcooling
  CONTROLLER:OUTSIDE AIR,
    OA Controller 1,  !- Name
    ECONOMIZER,  !- EconomizerChoice
    RETURN AIR TEMP LIMIT,  !- ReturnAirTempLimit
    NO RETURN AIR ENTHALPY LIMIT,  !- ReturnAirEnthalpyLimit
    NO LOCKOUT,  !- Lockout
    FIXED MINIMUM,  !- MinimumLimit
    Mixed Air Node,  !- Control_Node
    Outside Air Inlet Node,  !- Actuated_Node
    0.2,  !- minimum outside air flow rate {m3/s}
    1.3,  !- maximum outside air flow rate {m3/s}
    10,  !- temperature limit {C}  !!! how to set this temp needs more
thoughts
    ,  !- temperature lower limit {C}
    ,  !- enthalpy limit {J/kg}
    Relief Air Outlet Node,  !- Relief_Air_Outlet_Node
    Air Loop Inlet Node,  !- Return_Air_Node
    Min OA Sched;  !- Minimum Outside Air Schedule Name

DAYSCHEDULE,
    Zone Hi Temp Day Sch, Temperature,
      30.,30.,30.,30.,30.,30.,30.,24.,24.,24.,24.,24.,
      24.,24, 24, 24, 24.,30.,30.,30.,30.,30.,30.,30.;
!    ,  ,  ,  ,  ,  ,  ,20,20,20,20,24,24,24,24,24,24,  ,  ,  ,  ,  ,  , ;

  DAYSCHEDULE,
    FanAndVAVOperatingDaySched,  Fraction,  !- ScheduleType
  night setup - fan not running at night
     0.,0.,0.,0.,0.,0.,0.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,0.,0.,0.,0.,0.,0.,0.;
 !   1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.;


DAYSCHEDULE,
    CoolingcoilOperatingDaySched,  Fraction,  !- ScheduleType
 !  original chiller schedule - start at 8am
    0.,0.,0.,0.,0.,0.,0.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,0.,0.,0.,0.,0.,0.,0.;


  FAN:SIMPLE:VariableVolume,
    Var Vol Supply Fan 1,  !- Fan Name
    FanAndVAVAvailSched,  !! FanAndCoilAvailSched,  !- Available Schedule
    0.7,  !- Fan Total Efficiency
    600.0,  !- Delta Pressure {Pa} ! originally 600 delta pressure
    1.3,  !- Max Flow Rate {m3/s}
    0.20,  !- Min Flow Rate {m3/s}
    0.9,  !- Motor Efficiency
    1.0,  !- Motor In Airstream Fraction
    0.35071223,  !- FanCoefficient 1
    0.30850535,  !- FanCoefficient 2
    -0.54137364,  !- FanCoefficient 3
    0.87198823,  !- FanCoefficient 4
    0.000,  !- FanCoefficient 5
    Mixed Air Node,  !- Fan_Inlet_Node
    Cooling Coil Air Inlet Node;  !- Fan_Outlet_Node
```

Appendix B.1 Matlab GA code for both single and multiple buildings

B.1.1 Main program

```
%function mymain(pop, gen, model_index, weather)
%function mymain(pop, gen, model_index, weather)
%pop - population in GA, gen- total generations to run
%model_index  - which model to run
%               1: single bldg, 14-17p 4hr thermostats
%               2: single bldg, 10 worktime hrs thermostats
%               3: single bldg, fan starting time + 8-11a 4 thermostats
%               4: single bldg, chiller start time + 8-11a 4 thermostats
%               5: single bldg, fan on fixed period, 10 thermostats
%               6: 2 bldgs, 14-17p thermostats, (8 vars)
%               7: 2 bldgs, fan starting time + 8-11a 4 thermostats (10 vars)
%               8: 5 bldgs, 13-17 thermostats (or could be all 10 worktime
hrs)
% weather - "SF", "LA" and "Austin" etc.
% VBA codes apply comfort constraints

%cost function - peak load, or total cost or a mix of total cost and comfort
%control variables - thermostats, nightcooling schedules, and ...

clear all
close all
%load nextPop.mat
global history_array_new  array_temp
global evaluation_ctr epEval_ctr simsave_ctr
global model_index

model_index = 7;

num_of_gen=150;
num_in_pop=20;

evaluation_ctr=0;
epEval_ctr=0;
simsave_ctr=0;


% Crossover Operators
xFns = 'arithXover simpleXover';
xOpts = [1 0; 1 0];

% Mutation Operators
mFns = 'boundaryMutation multiNonUnifMutation nonUnifMutation unifMutation';
mOpts = [2 0 0;3 200 4;2 200 4;2 0 0]; %change multiNonUnif and nonUnif paras
3 ->2

% Termination Operators
termFns = 'maxGenTerm';
termOps = [num_of_gen]; % number of generations before program terminates

% Selection Function
selectFn = 'normGeomSelect'; %could be 'roulette' too
selectOps = [0.08];
```

```
% Evaluation Function
evalFn = 'myepeval';
evalOps = [];  %consider putting base file name in here? [2, 24, 15, 16]

% GA Options [epsilon float/binar display]
gaOpts=[1e-3 1 1];

%4-hr thermostats 14,15,16,17pm
if model_index ==1
   bounds = [22 28; 22 28; 22 28; 22 28];

 %10-hr thermostats
elseif model_index ==2
   bounds =[22 28;22 28;22 28;22 28;22 28;22 28;22 28;22 28;22 28;22 28];

%fan-based night cooling, fan starting time, early AM thermostats
elseif model_index ==3
   bounds =[17 31; 20 26; 20 26; 20 26; 20 26];

%fan-based night cooling. fan starting time and all day thermostats
elseif model_index ==30
   bounds=[17 31; 20 26; 20 26; 20 26; 20 26; 20 26; ...
            20 26; 20 26; 20 26; 20 26; 20 26];

%chiller-based night cooling, chiller starting time, early AM thermostats
elseif model_index ==4
   bounds= [17 31; 20 26; 20 26; 20 26; 20 26];

%with fan on 12a-5a, 10-hr thermostats, other times float with economizer
elseif model_index ==5
   bounds =[22 28;22 28;22 28;22 28;22 28;22 28;22 28;22 28;22 28;22 28];

%two bldgs, 4-hr thermostats 14,15,16,17pm (8 vars total)
elseif model_index ==6
   bounds = [24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26;
24 26];

%two bldgs with night cooling, fan start time, early AM thermostats (10vars)
elseif model_index ==7
   bounds = [17 31; 20 26; 20 26; 20 26; 20 26; 17 31; 20 26; 20 26; 20 26;
20 26];

elseif model_index ==8
   bounds = [24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26;
...
24 26; 24 26; 24 26; 24 26; 24 26; 24 26; ...
 24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26; 24 26];

 end


% Generate an intialize population of size 20
startPop = initializega(num_in_pop, bounds, evalFn, evalOps, [1e-3 1])
%startPop=endPop;

history_array_new=[floor(startPop(:,1:end-1))*10 startPop(:,end)];%put into
array
```

```
save history_array_new history_array_new;
array_temp = history_array_new;

%run the GA to create next generation
%end of GA when the criteria are satisfied or limit is reached
start_time=cputime;

[x,endPop,bestPop,trace]=ga(bounds,evalFn,evalOps,startPop,gaOpts,...
    termFns,termOps,selectFn,selectOps,xFns,xOpts,mFns,mOpts);

 % x is the best solution found
x;
% endPop is the ending population
endPop;
% bestPop is the best solution tracked over generations
bestPop;
% trace is a trace of the best value and average value of generations
trace;

% Plot the best over time
% clf
plot(trace(:,1),trace(:,2));
hold on
plot(trace(:,1),trace(:,3));

num_of_epEval =  epEval_ctr
num_of_simSave = simsave_ctr
%num_of_epEval_GA = evaluation_ctr
timeuse_in_min=(cputime-start_time)/60
time_per_Eval = timeuse_in_min / num_of_epEval

save history_array_new history_array_new
```

B.1.2 EnergyPlus-based function evaluator

```
function [sol, val]= myepeval(sol,options)

global epEval_ctr
global model_index

T0=24;

cd C:\EnergyPlus1.1.0\ExampleFiles

%4hr thermostat control only, no nightcooling
if model_index ==1

    dos('copy Tibase.idf Tivary.idf');
    file_id=fopen('Tivary.idf','A');

    fprintf(file_id,'\n %s ', 'DAYSCHEDULE, Zone Hi Temp Day Sch,
Temperature');
    for i=1:7
       fprintf(file_id, '%s%s', ',', ' ' )
    end
    for i=8:13
        fprintf(file_id, '%s%f ', ',', T0);
    end
    for i=14:17  %hours 14,15,16,17
        fprintf(file_id, '%s%f ', ',', sol(i-13));
    end
    for i=18:24 %hours 18-24
        fprintf(file_id, '%s%s ', ',', ' ');
    end
    fprintf(file_id,'%s ',';');

    fclose(file_id);


%10hr thermostat control only, no nightcooling
elseif model_index ==2

    dos('copy Tibase.idf Tivary.idf');
    file_id=fopen('Tivary.idf','A');

    fprintf(file_id,'\n %s ', 'DAYSCHEDULE, Zone Hi Temp Day Sch,
Temperature');
    for i=1:7
        fprintf(file_id, '%s%s', ',', ' ' )
    end
    for i=8:17  %hours 8-17
        fprintf(file_id, '%s%f ', ',', sol(i-7));
    end
    for i=18:24 %hours 18-24
        fprintf(file_id, '%s%s ', ',', ' ');
    end
    fprintf(file_id,'%s ',';');

    fclose(file_id);
```

```
   % nightcooling enabled - fan start time and discharge process
   elseif model_index ==3

       dos('copy fanTibase1.idf Tivary.idf');
       file_id=fopen('Tivary.idf','A');

       fprintf(file_id,'\n %s ', 'DAYSCHEDULE, FanAndVAVOperatingDaySched,
Fraction');
       tfan=round(sol(1));
       if tfan > 24
          tfan=tfan-24;
          for i=1:tfan
             fprintf(file_id, '%s%d ',',',0);
          end
          for i=(tfan+1):17
             fprintf(file_id, '%s%d ',',',1);
          end
          for i=18:24
             fprintf(file_id, '%s%d ',',',0);
          end
          fprintf(file_id,'%s ',';');
       elseif 17 <tfan <24
          for i=1:17
             fprintf(file_id, '%s%d ',',',1);
          end
          for i=18:tfan
             fprintf(file_id, '%s%d ',',',0);
          end
          for i=(tfan+1):24
             fprintf(file_id, '%s%d ',',',1);
          end
          fprintf(file_id,'%s ',';');
       elseif tfan ==24
          for i=1:17
             fprintf(file_id, '%s%d ',',',1);
          end
          for i=18:24
             fprintf(file_id, '%s%d ',',',0);
          end
          fprintf(file_id,'%s ',';');
       elseif tfan ==17
          for i=1:24
             fprintf(file_id, '%s%d ',',',1);
          end
          fprintf(file_id,'%s ',';');
       end

       %% only specify four temperatures in the earlyAM discharging process
       fprintf(file_id,'\n %s ', 'DAYSCHEDULE, Zone Hi Temp Day Sch,
Temperature');
       for i=1:7
          fprintf(file_id, '%s ',' ,');
       end
       for i=8:11
```

```
            fprintf(file_id, '%s%f ', ',', sol(i-6));
        end
        for i=12:17
            fprintf(file_id, '%s%f ', ',', T0);
        end
        for i=18:24
            fprintf(file_id, '%s ',' ,');
        end
        fprintf(file_id,'%s ',';');

        fclose(file_id);
        %%-----------------------------------------------------

       elseif model_index ==30

        dos('copy fanTibase10.idf Tivary.idf');
        file_id=fopen('Tivary.idf','A');

        fprintf(file_id,'\n %s ', 'DAYSCHEDULE, FanAndVAVOperatingDaySched,
Fraction');
        tfan=round(sol(1));
        if tfan > 24
            tfan=tfan-24;
            for i=1:tfan
                fprintf(file_id, '%s%d ',',',0);
            end
            for i=(tfan+1):17
                fprintf(file_id, '%s%d ',',',1);
            end
            for i=18:24
                fprintf(file_id, '%s%d ',',',0);
            end
            fprintf(file_id,'%s ',';');
        elseif 17 <tfan <24
            for i=1:17
                fprintf(file_id, '%s%d ',',',1);
            end
            for i=18:tfan
                fprintf(file_id, '%s%d ',',',0);
            end
            for i=(tfan+1):24
                fprintf(file_id, '%s%d ',',',1);
            end
            fprintf(file_id,'%s ',';');
        elseif tfan ==24
            for i=1:17
                fprintf(file_id, '%s%d ',',',1);
            end
            for i=18:24
                fprintf(file_id, '%s%d ',',',0);
            end
            fprintf(file_id,'%s ',';');
        elseif tfan ==17
            for i=1:24
                fprintf(file_id, '%s%d ',',',1);
            end
            fprintf(file_id,'%s ',';');
```

```
        end

        %% all 10 thermostats
        fprintf(file_id,'\n %s ', 'DAYSCHEDULE, Zone Hi Temp Day Sch,
Temperature');
        for i=1:7
           fprintf(file_id, '%s ',' ,');
        end
        for i=8:17
           fprintf(file_id, '%s%f ', ',', sol(i-6));
        end
        for i=18:24
           fprintf(file_id, '%s ',' ,');
        end
        fprintf(file_id,'%s ',';');

        fclose(file_id);
        %%-------------------------------------------------

   %nightcooling enabled - chiller starting time and discharge process,
   elseif model_index == 4
   %with fan on 12a-5a, 10-hr thermostats, not used
   elseif model_index ==5

   %two bldgs, 4-hr thermostats 14,15,16,17pm (8 vars total)
   %thermal comfort constraints applied by VBA codes
   elseif model_index ==6

        dos('copy Tibase1.idf Tivary1.idf');
        dos('copy Tibase2.idf Tivary2.idf');

        file_id1=fopen('Tivary1.idf','A');
        fprintf(file_id1,'\n %s ', 'DAYSCHEDULE, Zone Hi Temp Day Sch,
Temperature');
        for i=1:7
           fprintf(file_id1, '%s%s', ',', ' ' )
        end
        for i=8:12
            fprintf(file_id1, '%s%f ', ',', T0);
        end
        for i=13:17  %hours 14,15,16,17 of bldg1 (1,2,3,4 of 8 vars)
            fprintf(file_id1, '%s%f ', ',', sol(i-12));
        end
        for i=18:24 %hours 18-24
            fprintf(file_id1, '%s%s ', ',', ' ');
        end
        fprintf(file_id1,'%s ',';');


        file_id2=fopen('Tivary2.idf','A');
        fprintf(file_id2,'\n %s ', 'DAYSCHEDULE, Zone Hi Temp Day Sch,
Temperature');
        for i=1:7
           fprintf(file_id2, '%s%s', ',', ' ' )
        end
        for i=8:12
            fprintf(file_id2, '%s%f ', ',', T0);
```

```
        end
        for i=13:17  %hours 13,14,15,16,17 of bldg2 (6,7,8,9,10 of 10 vars)
            fprintf(file_id2, '%s%f ', ',', sol(i-7));
        end
        for i=18:24 %hours 18-24
            fprintf(file_id2, '%s%s ', ',', ' ');
        end
        fprintf(file_id2,'%s ',';');

        fclose(file_id1);
        fclose(file_id2);

    %two bldgs, nightcooling enabled, fan start time, early AM thermostats
    elseif model_index ==7

        dos('copy NC_fan1.idf Tivary1.idf');
        dos('copy NC_fan2.idf Tivary2.idf');

        file_id1=fopen('Tivary1.idf','A');
        file_id2=fopen('Tivary2.idf','A');

        % bldg 1 input --------------------------------
        fprintf(file_id1,'\n %s ', 'DAYSCHEDULE, FanAndVAVOperatingDaySched,
Fraction');
        tfan=round(sol(1));
        if tfan > 24
           tfan=tfan-24;
           for i=1:tfan
              fprintf(file_id1, '%s%d ',',',0);
           end
           for i=(tfan+1):17
              fprintf(file_id1, '%s%d ',',',1);
           end
           for i=18:24
              fprintf(file_id1, '%s%d ',',',0);
           end           fprintf(file_id1,'%s ',';');
        elseif 17 <tfan <24
           for i=1:17
              fprintf(file_id1, '%s%d ',',',1);
           end
           for i=18:tfan
              fprintf(file_id1, '%s%d ',',',0);
           end
           for i=(tfan+1):24
              fprintf(file_id1, '%s%d ',',',1);
           end
           fprintf(file_id1,'%s ',';');
        elseif tfan ==24
           for i=1:17
              fprintf(file_id1, '%s%d ',',',1);
           end
           for i=18:24
              fprintf(file_id1, '%s%d ',',',0);
           end
           fprintf(file_id1,'%s ',';');
        elseif tfan ==17
           for i=1:24
```

```
            fprintf(file_id1, '%s%d ',',',1);
         end
         fprintf(file_id1,'%s ',';');
      end

      %% only specify four temperatures in the earlyAM discharging process
      fprintf(file_id1,'\n %s ', 'DAYSCHEDULE, Zone Hi Temp Day Sch,
Temperature');
      for i=1:7
         fprintf(file_id1, '%s ',' ,');
      end
      for i=8:11
         fprintf(file_id1, '%s%f ', ',', sol(i-6));
      end
      for i=12:17
         fprintf(file_id1, '%s%f ', ',', T0);
      end
      for i=18:24
         fprintf(file_id1, '%s ',' ,');
      end
      fprintf(file_id1,'%s ',';');
      % bldg 1 input ends ---------------------------------

      % bldg 2 input ---------------------------------------
      fprintf(file_id2,'\n %s ', 'DAYSCHEDULE, FanAndVAVOperatingDaySched,
Fraction');
      tfan=round(sol(6));
      if tfan > 24
         tfan=tfan-24;
         for i=1:tfan
            fprintf(file_id2, '%s%d ',',',0);
         end
         for i=(tfan+1):17
            fprintf(file_id2, '%s%d ',',',1);
         end
         for i=18:24
            fprintf(file_id2, '%s%d ',',',0);
         end
         fprintf(file_id2,'%s ',';');
      elseif 17 <tfan <24
         for i=1:17
            fprintf(file_id2, '%s%d ',',',1);
         end
         for i=18:tfan
            fprintf(file_id2, '%s%d ',',',0);
         end
         for i=(tfan+1):24
            fprintf(file_id2, '%s%d ',',',1);
         end
         fprintf(file_id2,'%s ',';');
      elseif tfan==24
         for i=1:17
            fprintf(file_id2, '%s%d ',',',1);
         end
         for i=18:24
            fprintf(file_id2, '%s%d ',',',0);
         end
```

```
            fprintf(file_id2,'%s ',';');
        elseif tfan ==17
            for i=1:24
                fprintf(file_id2, '%s%d ',',',1);
            end
            fprintf(file_id2,'%s ',';');
        end

        fprintf(file_id2,'\n %s ', 'DAYSCHEDULE, Zone Hi Temp Day Sch,
Temperature');
        for i=1:7
            fprintf(file_id2, '%s ',' ,');
        end
        for i=8:11
            fprintf(file_id2, '%s%f ', ',', sol(i-1));
        end
        for i=12:17
            fprintf(file_id2, '%s%f ', ',', T0);
        end
        for i=18:24
            fprintf(file_id2, '%s ',' ,');
        end
        fprintf(file_id2,'%s ',';');

        fclose(file_id1);
        fclose(file_id2);


    elseif model_index == 8 %five buildings, testing for time

        dos('copy Tibase1.idf Tivary1.idf');
        dos('copy Tibase1.idf Tivary2.idf');
        dos('copy Tibase1.idf Tivary3.idf');
        dos('copy Tibase2.idf Tivary4.idf');
        dos('copy Tibase2.idf Tivary5.idf');

        for Tik = 1:5

            if Tik==1
                estr = ['file_id = fopen(''Tivary1.idf'',''A'');'];
            elseif Tik ==2
                estr = ['file_id = fopen(''Tivary2.idf'',''A'');'];
            elseif Tik ==3
                estr = ['file_id = fopen(''Tivary3.idf'',''A'');'];
            elseif Tik == 4
                estr = ['file_id = fopen(''Tivary4.idf'',''A'');'];
            elseif Tik ==5
                estr = ['file_id = fopen(''Tivary5.idf'',''A'');'];
            end
            eval(estr);

        fprintf(file_id,'\n %s ', 'DAYSCHEDULE, Zone Hi Temp Day Sch,
Temperature');
        for i=1:7
            fprintf(file_id, '%s%s', ',', ' ' )
        end
        for i=8:12
```

```
        fprintf(file_id, '%s%f ', ',', T0);
    end
    for i=13:17
       fprintf(file_id, '%s%f ', ',', sol(i-12+(Tik-1)*5));
    end
    for i=18:24
        fprintf(file_id, '%s%s ', ',', ' ');
    end
    fprintf(file_id,'%s ',';');

    fclose(file_id);
    end

end
if model_index == 6 | model_index ==7 % two-buildings
    epEval_ctr=epEval_ctr+2;
    cd C:\EnergyPlus1.1.0
    %runeplus2 starts runeplus Tivary2 weather in the end
    dos('runeplus2bldg Tivary1 LA');
 elseif model_index ==8
    epEval_ctr=epEval_ctr+5;
    cd C:\EnergyPlus1.1.0
    dos('runeplus5 Tivary1 LA');
 else %single building
    epEval_ctr=epEval_ctr+1;
    cd C:\EnergyPlus1.1.0
    dos('runeplus Tivary LA');
 end

 cd C:\my_research\single_para\genoptfiles\
 file_out=fopen('newoutput.txt','r');
 val=fscanf(file_out, '%f');

cd C:\'Program Files'\MatlabR11\work\GA
```

Appendix B.2 GenOpt codes

B.2.1 GenOpt Command file

```
Vary
{
      Parameter
      {
            Name    = Ti2;
            Min     = 22;
            Ini     = 26;
            Max     = 28;
            Step    = 0.2;
      }
      Parameter
      {
            Name    = Ti3;
            Min     = 22;
            Ini     = 26;
            Max     = 28;
            Step    = 0.2;
      }

}

OptimizationSettings
{
      MaxIte = 100;
      MaxEqualResults = 5;
      WriteStepNumber = false;
}
Algorithm
{
        Main = HookeJeeves;   // Main = EquMesh;
        StepReduction = 0.5;
        NumberOfStepReduction = 3;
}
```

B.2.2 GenOpt initialization file

```
Simulation {
  Files {
    Template {
      File1 = xTemplate.txt;
      Path1 = C:\My_Research\single_para\genoptfiles;
    }
    Input {
      File1 = x.idf;
      Path1 = C:\EnergyPlus1.1.0\ExampleFiles;
      //SavePath = Simulation.Files.Template.Path1;
    }
    Log {
      File1 = x.err;
      Path1 = Simulation.Files.Input.Path1;
      //SavePath = Simulation.Files.Template.Path1;
    }
```

```
    Output {
      File1 = newoutput.txt;
      Path1 = Simulation.Files.Template.Path1;
      //SavePath = Simulation.Files.Template.Path1;
    }
    Configuration {
      File1 = configuration.cfg;
      Path1 = C:\Research\go_prg\cfg;
    }
  }
  CallParameter { // optional section
    Suffix = Austin;
  }

  ObjectiveFunctionLocation
  {
//    Delimiter1 = "Peak load," ;
//    Name1      = "Peak load";
      Delimiter1 = "Total load," ;
      Name1      = "Total load";
//    Delimiter3 = "day avg PPD," ;
//    Name3      = "day avg PPD";
//    Delimiter1 = "Total energy cost," ;
//    Name1      = "Total energy cost";
//    Delimiter1 = "aggr cost," ;
//    Name1      = "aggr cost";

  }
} // end of section Simulation

Optimization {
  Files {
    Command {
      File1 = Command.txt;
      Path1 = Simulation.Files.Template.Path1;
    }
```

B.2.3 GenOpt configuration file

```
// Error messages of the simulation program
SimulationError
{
    ErrorMessage = "** Severe  **";
    ErrorMessage = "**  Fatal  **";
    ErrorMessage = "** EnergyPlus Terminated--Error(s) Detected";
}
// Format of simulation input files
IO
{
    NumberFormat = Float;
}
SimulationStart
{
   Command = "cmd /c \"start /DC:\\EnergyPlus1.1.0 /WAIT /MIN RunEPlus.bat
%Simulation.Files.Input.File1% %Simulation.CallParameter.Suffix%\"" ;
   WriteInputFileExtension = false;
```

Appendix B.3
VBA code for data post processing for Matlab GA and GenOpt DS

```
Public Sub transition ()

Dim title As String
Dim page As String
Dim Tik As Integer

For Tik = 1 To 5  'can post-process 1 – n buildings

title = "C:\EnergyPlus1.1.0\ExampleFiles\Tivary" & Tik & ".csv"
page = "bldg" & Tik

Workbooks.Open Filename:=title
Cells.Select
Selection.Copy
Windows("extract.xls").Activate
Sheets.Add
Cells.Select
ActiveSheet.Paste
ActiveSheet.Name = page

'column o - avg PMV
Range("O2").Formula = "=(L2+M2+N2)/3"
Range("O2").Select
Selection.AutoFill Destination:=Range("O2:O25"), Type:=xlFillDefault
' column P - avg PPD
Range("P2").Formula = "=100-95*exp(-(0.03353*O2^4+0.2179*O2^2))"
Range("P2").Select
Selection.AutoFill Destination:=Range("P2:P25"), Type:=xlFillDefault
' working time PPD
Range("P26").Formula = "=average(P9:P18)"

' column Q - load W/m2
Range("Q2").Formula = "=(G2+H2+I2)/(3600*102)"
Range("Q2").Select
Selection.AutoFill Destination:=Range("Q2:Q25"), Type:=xlFillDefault
' peak load
'matlab GA is max, so '-' used for peak load as cost fn
Range("Q26").Formula = "=max(Q2:Q25)"
' total load
Range("Q28").Formula = "=sum(Q2:Q25)"

Next Tik

For i = 9 To 18  ' only care about the peak during 8-17

    If Sheets("bldg1").Range("P" & i).Value > 50 Or
    Sheets("bldg2").Range("P" & i).Value > 50 Or
    Sheets("bldg3").Range("P" & i).Value > 50 Or
    Sheets("bldg4").Range("P" & i).Value > 50 Or
    Sheets("bldg5").Range("P" & i).Value > 50 Then
    Sheets("price").Range("J" & i).Value = 399 'set output if PPD violates

Else
```

```
    Sheets("price").Range("J" & i).Value = Sheets("bldg1").Range("Q" &
i).Value + Sheets("bldg2").Range("Q" & i).Value + Sheets("bldg3").Range("Q" &
i).Value
+ Sheets("bldg4").Range("Q" & i).Value + Sheets("bldg5").Range("Q" & i).Value

End If

Next i

' aggregated peak load - Matlab GA is maximization
Sheets("price").Range("J26").Formula = "=max(J9:J18)"
Sheets("price").Range("J27").Formula = "=400-max(J9:J18)"

fr = FreeFile
Open "C:\My_Research\single_para\genoptfiles\newoutput.txt" For Output As #fr

'peak demand output for GenOpt Direct Search
'Print #fr, "Peak load,";
'Print #fr, Sheets("price").Range("J26").Value

'peak demand output for matlab GA
Print #fr, Sheets("price").Range("J27").Value

Close #fr

Sheets("bldg1").Activate
DeleteWorksheet
Sheets("bldg2").Activate
DeleteWorksheet
Sheets("bldg3").Activate
DeleteWorksheet
Sheets("bldg4").Activate
DeleteWorksheet
Sheets("bldg5").Activate
DeleteWorksheet


ChDir "C:\EnergyPlus1.1.0\ExampleFiles"
Workbooks("Tivary1.csv").Close
Workbooks("Tivary2.csv").Close
Workbooks("Tivary3.csv").Close
Workbooks("Tivary4.csv").Close
Workbooks("Tivary5.csv").Close


ChDir "C:\My_Research\single_para\genoptfiles"
Workbooks("extract.xls").Save
Workbooks("extract.xls").Close

End Sub

Public Sub DeleteWorksheet()
Application.DisplayAlerts = False
ActiveWindow.SelectedSheets.Delete
Application.DisplayAlerts = True
End Sub
```

Appendix C.1
Key EnergyPlus inputs for three models used in the multi-building studies

Difference in thermal mass can be found in Appendix A.2

Model 1 and 2: half mass
Model 3: 2 mass

Difference in the internal load pattern

Model 1 and 3:

```
  DAYSCHEDULE,
    BLDG Day   1,  !- Name
    Any Number,  !- ScheduleType
    0.00,0.00,0.00,0.00,0.00,0.00,0.10,0.50,1.00,1.00,1.00,1.00,
    0.50,1.00,1.00,1.00,0.50,0.10,0.00,0.00,0.00,0.00,0.00,0.00;
  DAYSCHEDULE,
    BLDG Day   5,  !- Name
    Any Number,  !- ScheduleType
    0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,1.00,1.00,1.00,1.00,
    1.00,1.00,1.00,1.00,1.00,1.00,0.00,0.00,0.00,0.00,0.00,0.00;
```

Model 2:

```
  DAYSCHEDULE,
    BLDG Day   1,  !- Name
    Any Number,  !- ScheduleType
    0.00,0.00,0.00,0.00,0.00,0.50,0.50,1.00,1.10,1.20,1.30,1.30,
    1.50,1.30,1.00,0.80,0.40,0.10,0.00,0.00,0.00,0.00,0.00,0.00;
  DAYSCHEDULE,
    BLDG Day   5,  !- Name
    Any Number,  !- ScheduleType
    0.00,0.00,0.00,0.00,0.00,0.20,0.20,0.4,1.00,1.10,1.30,1.30,
    1.5,1.10,0.90,0.70,0.55,0.50,0.30,0.30,0.00,0.00,0.00,0.00;
```

Whether or not there is a west window

Models 1 and 2 have a west window, but model 3 does not

```
      !! west window in north zone
      Surface:HeatTransfer:Sub,
      Zn003:Wall001:Win001,  !- User Supplied Surface Name
      Window,  !- Surface Type
      WIN-CON-LIGHT,  !- Construction Name of the Surface
      Zn003:Wall001,  !- Base Surface Name
      ,  !- OutsideFaceEnvironment Object
      0.5000000,  !- View Factor to Ground
      ,  !- Name of shading control
      ,  !- WindowFrameAndDivider Name
      1.0,  !- Multiplier
      4,  !- Number of Surface Vertice Groups -- Number of (X,Y,Z)
group
      0.0000000E+00,10.0,3.048000,  !- X,Y,Z ==> Vertex 1
      0.0000000E+00,10,0.0000000E+00,  !- X,Y,Z ==> Vertex 2
      0.0000000E+00,8.0,0.0000000E+00,  !- X,Y,Z ==> Vertex 3
      0.0000000E+00,8.0,3.048000;  !- X,Y,Z ==> Vertex 4
```

Appendix C.2 Matlab code to match multiple building in Enumeration

```
data1=load('bldg1_load.dat');
data2=load('bldg2_load.dat');
data3=load('bldg3_load.dat');
data4=load('bldg1_load.dat');
data5=load('bldg1_load.dat');

N1=length(data1);
N2=length(data2);
N3=length(data3);
N4=length(data4);
N5=length(data5);

ppd1=load('bldg1_ppd.dat');
ppd2=load('bldg2_ppd.dat');
ppd3=load('bldg3_ppd.dat');
ppd4=load('bldg1_ppd.dat');
ppd5=load('bldg1_ppd.dat');

sum12_base=data1(:,1)+data2(:,1);
sum13_base=data1(:,1)+data3(:,1);
sum23_base=data2(:,1)+data3(:,1);
sum14_base=data1(:,1)+data4(:,1);
sum25_base=data2(:,1)+data5(:,1);

figure
plot(data1(:,1),'g*')
hold
plot(data2(:,1),'r+')
plot(data3(:,1),'kx')
plot(data4(:,1),'mo')
plot(data5(:,1),'c-')
legend('bldg1', 'bldg2', 'bldg3','bldg4','bldg5')
plot(data1(:,1),'g')
plot(data2(:,1),'r')
plot(data3(:,1),'k')
plot(data4(:,1),'m')
plot(data5(:,1),'c')
xlabel('time (hr)')
ylabel('power consumption (W/m2)')
title('single building load profiles')


[sum12, peak12, index12_1, index12_2]=match2(data1,N1, data2, N2);
plotcomp2(sum12_base, data1(:,1), data2(:,1), sum12, data1(:,index12_1),
data2(:,index12_2));
plotppd2(ppd1, index12_1, ppd2, index12_2);
[peak12_base,peak12_base_time]=max(sum12_base)
[peak12_check, peak12_time]=max(sum12)
singlesum12 = demandp*max(data1(:,1))+sum(data1(:,1).*costp)+...
   demandp*max(data2(:,1))+sum(data2(:,1).*costp);;
[singlesum12max, singlesum12max_index]=max(singlesum12)
indvsum12 = demandp*max(data1(:,32))+sum(data1(:,32).*costp)+...
   demandp*max(data2(:,28))+sum(data2(:,28).*costp);;
[indvsum12max, indvsum12max_index]=max(indvsum12)
```

```
[sum13, peak13, index13_1, index13_2]=match2(data1,N1, data3, N3);
plotcomp2(sum13_base, data1(:,1), data3(:,1), sum13, data1(:,index13_1),
data3(:,index13_2));
plotppd2(ppd1, index13_1, ppd3, index13_2);
[peak13_base,peak13_base_time]=max(sum13_base)
[peak13_check, peak13_time]=max(sum13)
singlesum13 = demandp*max(data1(:,1))+sum(data1(:,1).*costp)+...
   demandp*max(data3(:,1))+sum(data3(:,1).*costp);;
[singlesum13max, singlesum13max_index]=max(singlesum13)
indvsum13 = demandp*max(data1(:,32))+sum(data1(:,32).*costp)+...
   demandp*max(data3(:,89))+sum(data3(:,89).*costp);;
[indvsum13max, indvsum13max_index]=max(indvsum13)

% 3-building case
sum123_base=data1(:,1)+data2(:,1)+data3(:,1);
[sum123, peak123, index123_1, index123_2, index123_3]=match3(data1,N1, data2,
N2, data3, N3);
plotcomp3(sum123_base, data1(:,1), data2(:,1), data3(:,1), sum123,
data1(:,index123_1), data2(:,index123_2), data3(:,index123_3));
plotppd3(ppd1, ppd2, ppd3, index123_1, index123_2, index123_3)
[peak123_base,peak123_base_time]=max(sum123_base)
[peak123_check, peak123_time]=max(sum123)
singlesum123 = demandp*max(data2(:,1))+sum(data2(:,1).*costp)+...
   demandp*max(data3(:,1))+sum(data3(:,1).*costp)+...
   demandp*max(data1(:,1))+sum(data1(:,1).*costp);;
[singlesum123max, singlesum123max_index]=max(singlesum123)
indvsum123 = demandp*max(data2(:,28))+sum(data2(:,28).*costp)+...
   demandp*max(data3(:,89))+sum(data3(:,89).*costp)+...
   demandp*max(data1(:,32))+sum(data1(:,32).*costp);;
[indvsum123max, indvsum123max_index]=max(indvsum123)

% 4-bldg
sum1234_base=data1(:,1)+data2(:,1)+data3(:,1)+data4(:,1);
[sum1234, peak1234, index1234_1, index1234_2, index1234_3, index1234_4]...
    = match4(data1,N1, data2, N2, data3, N3, data4, N4);
 plotcomp4(sum1234_base, data1(:,1), data2(:,1), data3(:,1), data4(:,1),...
    sum1234, data1(:,index1234_1), data2(:,index1234_2),...
    data3(:,index1234_3), data4(:,index1234_4));
plotppd4(ppd1, ppd2, ppd3, ppd4, index1234_1, index1234_2, index1234_3,
index1234_4)
[peak1234_base,peak1234_base_time]=max(sum1234_base)
[peak1234_check, peak1234_time]=max(sum1234)

 %5-bldg
sum12345_base=data1(:,1)+data2(:,1)+data3(:,1)+data4(:,1)+data5(:,1);
[sum12345, peak12345, index_1, index_2, index_3, index_4,
index_5]=match5(data1,N1, data2, N2, data3, N3, data4, N4, data5, N5);
plotcomp5(sum12345_base, data1(:,1), data2(:,1), data3(:,1),data4(:,1),
data5(:,1),...
      sum12345, data1(:,index_1), data2(:,index_2), data3(:,index_3),
data4(:,index_4), data5(:,index_5));
plotppd5(ppd1, ppd2, ppd3, ppd4, ppd5, index_1, index_2, index_3, index_4,
index_5)
[peak12345_base,peak12345_base_time]=max(sum12345_base)
[peak12345_check, peak12345_time]=max(sum12345)
```

Appendix C.3 VBA code for automated E+ simulation in Smart Enumeration

C.3.1 computational savings illustration

Load control strategy	original E+ runs	New E+ runs	savings
Bldg1 – vary thermostats	90	32	64%
Bldg3 – vary thermostats	90	75	17%
Bldg1 - night cooling	70	40	43%

C.3.2 VBA code

```
Public Sub main()

Dim N, Nt, flag As Integer
Dim Tshed, deltaT As Variant
Dim sheetname As String

Dim Ti(24) As Variant
Dim I, j As Integer
Dim k As Integer                    ' 1-hr shedding
Dim k21, k22 As Integer             ' 2-hr shedding
Dim k31, k32, k33 As Integer        ' 3-hr shedding
Dim k41, k42, k43, k44 As Integer   ' 4-hr shedding

Count = 0

' compute the original case without load shedding
Worksheets("Tisum").Activate

FileCopy "C:\My_Research\Multi_SmartEnum\bldg1_Ti.idf",
"C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf"
fr = FreeFile
Open "C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf" For Append As #fr
Print #fr, "DAYSCHEDULE, Zone Hi Temp Day Sch, Temperature,";
For i = 1 To 23
Print #fr, Cells(i, 4).Value, ",";
Next i
Print #fr, Cells(24, 4).Value, ";";
Close #fr

Shell "C:\My_Research\Multi_SmartEnum\myrunep.bat"
Sleep 15000 ' wait for energyplus
Count = Count + 1

Sheets.Add
ActiveSheet.Name = "base"
Call Readin

For i = 1 To 24
Worksheets("Tisum").Cells(i + 25, 4).Value = ActiveSheet.Cells(i + 1,
17).Value
Worksheets("Load").Cells(i, 1).Value = ActiveSheet.Cells(i + 1, 17).Value
Worksheets("PPD").Cells(i, 1).Value = ActiveSheet.Cells(i + 1, 16).Value
```

```
Next i
ChDir "C:\EnergyPlus1.1.0\ExampleFiles"
Workbooks("Tivary.csv").Close

Nt = 1

' k = 1 case -------------------------------------------
For k = 1 To 5

    Worksheets("Tisum").Activate
    Tshed = 1
    deltaT = 1
 valueloop_1:

 If Tshed < 4 Then

    Worksheets("Tisum").Activate
    For i = 1 To 24
    Ti(i) = Cells(i, 4).Value
    Next i

    Ti(12 + k) = Ti(12 + k) + Tshed

    FileCopy "C:\My_Research\Multi_SmartEnum\bldg1_Ti.idf",
"C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf"

    fr = FreeFile
    Open "C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf" For Append As #fr
    Print #fr, "DAYSCHEDULE, Zone Hi Temp Day Sch, Temperature,";
    For i = 1 To 23
    Print #fr, Ti(i), ",";
    Next i
    Print #fr, Ti(24), ";";
    Close #fr

    Shell "C:\My_Research\Multi_SmartEnum\myrunep.bat"
    Sleep 15000 ' wait for energyplus
    Count = Count + 1

    Sheets.Add
    Call Readin

    flag = 0
    For i = 9 To 18
        If Range("P" & i).Value > 50 Then
        flag = 1
        End If
    Next i

    ChDir "C:\EnergyPlus1.1.0\ExampleFiles"
    Workbooks("Tivary.csv").Close

    If flag = 0 Then
    sheetname = 12 + k & "_" & Tshed
    ActiveSheet.Name = sheetname

    For i = 1 To 24
```

```
    Worksheets("Tisum").Cells(i, Nt + 4).Value = Ti(i)
    Worksheets("Tisum").Cells(i + 25, Nt + 4).Value = Cells(i + 1, 17).Value
    Worksheets("load").Cells(i, Nt + 1).Value = Cells(i + 1, 17).Value
    Worksheets("PPD").Cells(i, Nt + 1).Value = Cells(i + 1, 16).Value
    Next i
    Worksheets("Tisum").Cells(25, Nt + 4).Value = sheetname

    Nt = Nt + 1
    Tshed = Tshed + deltaT
    GoTo valueloop_1
    Else  'ActiveSheet.Move After:=Sheets("Tisum")
    DeleteWorksheet
    End If

End If 'if Tshed < 4

Next k

 ' k = 2 case ---------------------------------------------
For k21 = 1 To 4
    For k22 = k21 + 1 To 5

    Worksheets("Tisum").Activate
    Tshed = 1
    deltaT = 1

valueloop_2:

 If Tshed < 4 Then

    Worksheets("Tisum").Activate
    For i = 1 To 24
    Ti(i) = Cells(i, 4).Value
    Next i

    Ti(12 + k21) = Ti(12 + k21) + Tshed
    Ti(12 + k22) = Ti(12 + k22) + Tshed

    FileCopy "C:\My_Research\Multi_SmartEnum\bldg1_Ti.idf",
"C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf"

    fr = FreeFile
    Open "C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf" For Append As #fr
    Print #fr, "DAYSCHEDULE, Zone Hi Temp Day Sch, Temperature,";
    For i = 1 To 23
    Print #fr, Ti(i), ",";
    Next i
    Print #fr, Ti(24), ";";
    Close #fr

    Shell "C:\My_Research\Multi_SmartEnum\myrunep.bat"
    Sleep 15000 ' wait for energyplus
    Count = Count + 1

    Sheets.Add
    Call Readin
```

```
    flag = 0
    For i = 9 To 18
        If Range("P" & i).Value > 50 Then
        flag = 1
        End If
    Next i

    ChDir "C:\EnergyPlus1.1.0\ExampleFiles"
    Workbooks("Tivary.csv").Close

    If flag = 0 Then
    sheetname = 12 + k21 & "_" & 12 + k22 & "_" & Tshed
    ActiveSheet.Name = sheetname
    Tshed = Tshed + deltaT

    For i = 1 To 24
    Worksheets("Tisum").Cells(i, Nt + 4).Value = Ti(i)
    Worksheets("Tisum").Cells(i + 25, Nt + 4).Value =Cells(i + 1, 17).Value
    Worksheets("load").Cells(i, Nt + 1).Value =Cells(i + 1, 17).Value
    Worksheets("PPD").Cells(i, Nt + 1).Value = Cells(i + 1, 16).Value
    Next i
    Worksheets("Tisum").Cells(25, Nt + 4).Value = sheetname

    Nt = Nt + 1

    GoTo valueloop_2

    Else  'ActiveSheet.Move After:=Sheets("Tisum")
    DeleteWorksheet
    End If

 End If ' Tshed <4

    Next k22
Next k21

'  k = 3 case -------------------------------------------
For k31 = 1 To 3
    For k32 = k31 + 1 To 4
        For k33 = k32 + 1 To 5

    Worksheets("Tisum").Activate

    Tshed = 1
    deltaT = 1

valueloop_3:

 If Tshed < 4 Then

    Worksheets("Tisum").Activate
    For i = 1 To 24
    Ti(i) = Cells(i, 4).Value
    Next i

    Ti(12 + k31) = Ti(12 + k31) + Tshed
    Ti(12 + k32) = Ti(12 + k32) + Tshed
```

```
    Ti(12 + k33) = Ti(12 + k33) + Tshed

    FileCopy "C:\My_Research\Multi_SmartEnum\bldg1_Ti.idf",
"C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf"

    fr = FreeFile
    Open "C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf" For Append As #fr
    Print #fr, "DAYSCHEDULE, Zone Hi Temp Day Sch, Temperature,";
    For i = 1 To 23
    Print #fr, Ti(i), ",";
    Next i
    Print #fr, Ti(24), ";";
    Close #fr

    Shell "C:\My_Research\Multi_SmartEnum\myrunep.bat"
    Sleep 15000 ' wait for energyplus
    Count = Count + 1

    Sheets.Add
    Call Readin

    flag = 0
    For i = 9 To 18
        If Range("P" & i).Value > 50 Then
        flag = 1
        End If
    Next i

    ChDir "C:\EnergyPlus1.1.0\ExampleFiles"
    Workbooks("Tivary.csv").Close

    If flag = 0 Then
    sheetname = 12 + k31 & "_" & 12 + k32 & "_" & 12 + k33 & "_" & Tshed
    ActiveSheet.Name = sheetname

    For i = 1 To 24
    Worksheets("Tisum").Cells(i, Nt + 4).Value = Ti(i)
    Worksheets("Tisum").Cells(i + 25, Nt + 4).Value = Cells(i + 1, 17).Value
    Worksheets("load").Cells(i, Nt + 1).Value = Cells(i + 1, 17).Value
    Worksheets("PPD").Cells(i, Nt + 1).Value = Cells(i + 1, 16).Value
    Next i
    Worksheets("Tisum").Cells(25, Nt + 4).Value = sheetname
    Nt = Nt + 1

    Tshed = Tshed + deltaT
    GoTo valueloop_3
    Else  'ActiveSheet.Move After:=Sheets("Tisum")
    DeleteWorksheet
    End If

 End If ' if Tshed < 4

    Next k33
  Next k32
Next k31

'  k = 3 case ---------------------------------------------
```

```
For k41 = 1 To 2
    For k42 = k41 + 1 To 3
        For k43 = k42 + 1 To 4
            For k44 = k43 + 1 To 5

    Worksheets("Tisum").Activate

    Tshed = 1
    deltaT = 1

valueloop_4:

 If Tshed < 4 Then

    Worksheets("Tisum").Activate
    For i = 1 To 24
    Ti(i) = Cells(i, 4).Value
    Next i

    Ti(12 + k41) = Ti(12 + k41) + Tshed
    Ti(12 + k42) = Ti(12 + k42) + Tshed
    Ti(12 + k43) = Ti(12 + k43) + Tshed
    Ti(12 + k44) = Ti(12 + k44) + Tshed

    FileCopy "C:\My_Research\Multi_SmartEnum\bldg1_Ti.idf",
"C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf"

    fr = FreeFile
    Open "C:\EnergyPlus1.1.0\ExampleFiles\Tivary.idf" For Append As #fr
    Print #fr, "DAYSCHEDULE, Zone Hi Temp Day Sch, Temperature,";
    For i = 1 To 23
    Print #fr, Ti(i), ",";
    Next i
    Print #fr, Ti(24), ";";
    Close #fr

    Shell "C:\My_Research\Multi_SmartEnum\myrunep.bat"
    Sleep 15000 ' wait for energyplus
    Count = Count + 1

    Sheets.Add
    Call Readin

    flag = 0
    For i = 9 To 18
        If Range("P" & i).Value > 50 Then
        flag = 1
        End If
    Next i

    ChDir "C:\EnergyPlus1.1.0\ExampleFiles"
    Workbooks("Tivary.csv").Close

    If flag = 0 Then
    sheetname = 12 + k41 & "_" & 12 + k42 & "_" & 12 + k43 & "_" & 12 + k44 &
"_" & Tshed
    ActiveSheet.Name = sheetname
```

```
    For i = 1 To 24
    Worksheets("Tisum").Cells(i, Nt + 4).Value = Ti(i)
    Worksheets("Tisum").Cells(i + 25, Nt + 4).Value = Cells(i + 1, 17).Value
    Worksheets("load").Cells(i, Nt + 1).Value = Cells(i + 1, 17).Value
    Worksheets("PPD").Cells(i, Nt + 1).Value = Cells(i + 1, 16).Value
    Next i
    Worksheets("Tisum").Cells(25, Nt + 4).Value = sheetname
    Nt = Nt + 1

    Tshed = Tshed + deltaT
    GoTo valueloop_4
    Else  'ActiveSheet.Move After:=Sheets("Tisum")
    DeleteWorksheet
    End If

 End If ' if Tshed < 4

    Next k44
  Next k43
 Next k42
Next k41

End Sub

Public Sub Readin()
    Workbooks.Open Filename:="C:\EnergyPlus1.1.0\ExampleFiles\Tivary.csv"
    Cells.Select
    Selection.Copy
    Windows("bldg1_Ti.xls").Activate
    Cells.Select
    ActiveSheet.Paste

    Range("O2").Formula = "=(L2+M2+N2)/3"
    Range("O2").Select
    Selection.AutoFill Destination:=Range("O2:O25"), Type:=xlFillDefault
    ' column P - avg PPD
    Range("P2").Formula = "=100-95*exp(-(0.03353*O2^4+0.2179*O2^2))"
    Range("P2").Select
    Selection.AutoFill Destination:=Range("P2:P25"), Type:=xlFillDefault
    ' working time PPD
    Range("P26").Formula = "=average(P9:P18)"
    ' column Q - load W/m2
    Range("Q2").Formula = "=(G2+H2+I2)/(3600*102)"
    Range("Q2").Select
    Selection.AutoFill Destination:=Range("Q2:Q25"), Type:=xlFillDefault
    ' peak load
    Range("Q26").Formula = "=max(Q2:Q25)"
    ' total load
    Range("Q28").Formula = "=sum(Q2:Q25)"

End Sub

Public Sub DeleteWorksheet()
Application.DisplayAlerts = False
ActiveWindow.SelectedSheets.Delete
Application.DisplayAlerts = True
End Sub
```

Appendix D.1 Key Splus code for training load prediction models

D.1.1 Time series model for base load profile prediction in SPlus

```
import.data(DataFrame="Xf", FileName ="_Data//_model3//Xt3.txt",
   FileType="ascii")

y <- Xf[,1]
yt1 <- Xf[,2]
yt24 <- Xf[,3]
yt25 <- Xf[,4]
Tout <- Xf[,5]
SS <- Xf[,6]
SW <- Xf[,7]
SR <- Xf[,8]
SE <- Xf[,9]
Tout1 <- Xf[,10]
SS1 <- Xf[,11]
SW1 <- Xf[,12]
SR1 <- Xf[,13]
SE1 <- Xf[,14]
Tout24 <- Xf[,15]
SS24 <- Xf[,16]
SW24 <- Xf[,17]
SR24 <- Xf[,18]
SE24 <- Xf[,19]
Tout25 <- Xf[,20]
SS25 <- Xf[,21]
SW25 <- Xf[,22]
SR25 <- Xf[,23]
SE25 <- Xf[,24]

Xglm <- glm(y ~ yt1 + yt24 + yt25 + Tout+SS+SW+SR+SE +Tout1+SS1+SW1+SR1+SE1
   +Tout24+SS24+SW24+SR24+SE24 +Tout25+SS25+SW25+SR25+SE25)
plot(Xglm)
summary(Xglm)
Xres <- residuals(Xglm)
plot(Xres)
Xresacf <- acf(Xres, lag.max=48, type="correlation")
acf.plot(Xresacf)

Xglm2 <- glm(y ~ yt1 + yt24 + yt25)
summary(Xglm2)

import.data(DataFrame="reshat", FileName ="_Data//_model3//reshat.txt",
   FileType="ascii")
reshat <- reshat[,1]
Xglm3 <- glm(y ~ yt1 + yt24 +yt25 + reshat)

import.data(DataFrame="hat", FileName ="_Data//_model3//hat.txt",
   FileType="ascii")
yt1hat <- hat[,1]
ytreshat <- hat[,2]
Xglm4 <- glm(y~ yt1hat + yt24 + yt25 + ytreshat)
summary(Xglm4)
```

D.1.2 Regression model for peak load correction in SPlus

```
import.data(DataFrame="max", FileName="_Data//_deltaP//TEmax.txt",
   FileType="ascii")
Emax <- max[,1]
Tmax <- max[,2]
SSmax <- max[,3]
SWmax <- max[,4]
SRmax <- max[,5]
SEmax <- max[,6]

Emaxfullglm <- glm(Emax ~ Tmax + SSmax + SWmax + SRmax + SEmax)
summary(Emaxfullglm)

EmaxTglm <- glm(Emax ~ Tmax)
summary(EmaxTglm)
```

D.1.3 Regression model for load reduction in SPlus

```
new.database(where="_Data//_deltaP", type="directory")
attach("_Data//_deltaP", pos=1)

import.data(DataFrame="H14", FileName="_Data//_deltaP//H14.txt",
   FileType="ascii")
H14E <- H14[,1]
H14T <- H14[,2]
H14B <- H14[,3]
H14glm <- glm(H14E ~ H14T^0.5 + H14B)
H14res <- residuals(H14glm)

import.data(DataFrame="H15", FileName="_Data//_deltaP//H15.txt",
   FileType="ascii")
H15E <- H15[,1]
H15T <- H15[,2]
H15B <- H15[,3]
H15glm <- glm(H15E ~ H15T^0.5 + H15B)
H15res <- residuals(H15glm)

import.data(DataFrame="H16", FileName="_Data//_deltaP//H16.txt",
   FileType="ascii")
H16E <- H16[,1]
H16T <- H16[,2]
H16B <- H16[,3]
H16glm <- glm(H16E ~ H16T^0.5 + H16B)
H16res <- residuals(H16glm)

import.data(DataFrame="H17", FileName="_Data//_deltaP//H17.txt",
   FileType="ascii")
H17E <- H17[,1]
H17T <- H17[,2]
H17B <- H17[,3]
H17glm <- glm(H17E ~ H17T^0.5 + H17B)
H17res <- residuals(H17glm)
```

D.1.4 Regression model for PMV increase in SPlus

```
import.data(DataFrame="Hpmv3", FileName="_Data//_PMV//Hpmv.txt",
   FileType="ascii")
pmv3 <- Hpmv3[,1]
deltaT3 <- Hpmv3[,2]
baseld3 <- Hpmv3[,3]
pmvglm3 <- glm(pmv3 ~ deltaT3 + baseld3)
summary(pmvglm3)


import.data(DataFrame="TPMV", FileName="_Data//_PMV//TPMVmax.txt",
   FileType="ascii")
PMVmax <- TPMV[,1]
Tmax <- TPMV[,2]
Emax <- TPMV[,3]
PMVmaxglm <- glm(PMVmax ~ Tmax + Emax)
summary(PMVmaxglm)
```

SupportCode 1 Power difference data visualization and analysis

```
%visualize all power data in summer
close all
V_deltaT=[0 1 2 3 4];
N=length(V_deltaT); %number of shedding case + 1
M=length(Short_powerData);
powerDiff = zeros(M,N);

for i=1:N
   powerDiff(:,i)=Short_powerData(:,5+i) - Short_powerData(:,6);
end

%plotting power difference
for i=1:10
   figure
   plot(powerDiff(14+i*2*24,:), 'c*')
hold
plot(powerDiff(14+i*2*24,:),'c')
plot(powerDiff(15+i*2*24,:), 'gx')
plot(powerDiff(15+i*2*24,:), 'g')
plot(powerDiff(16+i*2*24,:), 'ro')
plot(powerDiff(16+i*2*24,:), 'r')
plot(powerDiff(17+i*2*24,:), 'k-')
plot(powerDiff(17+i*2*24,:), 'k')
title('power diff 14-17pm, 6/7')
end

Np = N-1; % number of data point categories
%graphically review for each hour the power change varying with temp
increases
Mp=M/24;
hrly_pdiff = zeros(12, Mp, Np);
hrly_load = zeros(12, Mp);

for i=1:12 % hours 7 -18 examined
   for j=1:Mp %Mp=66 with weekends taken out
```

```
      hrly_load(i,j) = Short_powerData((j-1)*24+(i+6), 6);
      for k=1:Np %Np=4
         hrly_pdiff(i,j,k)= powerDiff( (j-1)*24+(i+6), k+1);
      end
   end
end

% powerDiff for each hour varies with days and Temp change
for i=1:3
   figure
   for j=1:4
      subplot(2,2,j)
      hold
      plot(hrly_pdiff((i-1)*4+j, :, 1), 'c')
      plot(hrly_pdiff((i-1)*4+j, :, 2), 'g')
      plot(hrly_pdiff((i-1)*4+j, :, 3), 'r')
      plot(hrly_pdiff((i-1)*4+j, :, 4), 'k')
      axis([0 70 -10 5])
      xlabel('time (days)')
      ylabel('hourly power reduction (W/m2)')
   end
end

% powerDiff for each hour varies with original load and Temp change
for i=1:3
   figure
   for j=1:4
      subplot(2,2,j)
      plot(hrly_load((i-1)*4+j,:), hrly_pdiff((i-1)*4+j, :, 1), 'c.')
      hold
      plot(hrly_load((i-1)*4+j,:), hrly_pdiff((i-1)*4+j, :, 2), 'g.')
      plot(hrly_load((i-1)*4+j,:), hrly_pdiff((i-1)*4+j, :, 3), 'r.')
      plot(hrly_load((i-1)*4+j,:), hrly_pdiff((i-1)*4+j, :, 4), 'k.')
      axis([0 70 -10 5])
      xlabel('base load (W/m2)')
      ylabel('hourly power reduction (W/m2)')
   end
end
```

SupportCode 2 Power difference predication check in testing data in Matlab

```
function [lb, ub]=hrbound(nlhr, hr)
close all

nl = nlhr; %[12.558, -8.217, -0.161]' for hr17 for example

for i=1:44
   for j=1:4
      hsim(i,j) = hrly_pdiff(hr-6,(i+22),j);
      hpred(i,j) = nl(1)+nl(2)*sqrt(V_deltaT(j+1))+nl(3)*hrly_load(hr-
6,i+22);
   end
end
hres = hsim - hpred;

figure
```

```
for i=1:4
   subplot(2,2,i)
   plot(hsim(:,i), hpred(:,i),'.')
   axis([-20 3 -20 3])
end
xlabel('simulated power reduction (W/m2)')
ylabel('predication (W/m2)')


figure
for i=1:4
   subplot(2,2,i)
   plot(hres(:,i),'.')
end

std_h_res=std(reshape(hres,44*4,1));
mean_h_res=mean(reshape(hres,44*4,1));

lb = mean_h_res - 2*std_h_res;
ub = mean_h_res + 2*std_h_res;
```

Appendix D.2 AMPL code for nonlinear optimization

D.2.1 Model file

We trained two models for building 1 and building 3. the statistical results are embedded in the code. The code presented here is a 10-building case with five type-1s and five type-3s. The mix and the total number can both be adjusted easily. The data file needs to be adjusted accordingly.

```
reset;

set T := 1..24;
# total number of buildings
set bn := 1..10;
# number of building 1
set bn1 := 1..5;
set bn3 := 6..10;
param n1 := 5;

param Tmax  >= 0;
param SSmax >= 0;
param SEmax >= 0;
param SRmax >= 0;
param Tmaxpre >= 0;

param Epre {bn, T} >= 0;
param Emaxpre {bn} >= 0;
param PMVmaxpre {bn} >=0;
param corr {i in bn} := if i <= n1 then (1.468*Tmax + 0.032*SSmax +
0.015*SEmax + 0.0*SRmax)/Emaxpre[i] else 1 ;

param PMVgoal >= 0;

data 10bldg.dat;

var Ebase {bn, T} >= 0;
var Eadj {bn, T} >= 0;
var dT {bn, 1..4} >= 0;
var sqdT {bn, 1..4} >= 0;

var z >= 0;

option solver loqo;
#option solver minos;

minimize peak: z;
subject to peakcon {i in T}: sum {k in bn} Eadj[k,i] <= z;

#minimize totalcon : sum{k in bn, i in T} Eadj[k,i];


#bldg type 1
subject to initial {k in bn}: Ebase[k,1]=0;
subject to Earma1 {k in bn1, i in 2..24}: Ebase[k,i]= 0.874*Ebase[k,i-
1]+1.007*Epre[k,i]-0.88*Epre[k,i-1];
```

```
subject to Earma3 {k in bn3, i in 2..24}: Ebase[k,i]= 0.873*Ebase[k,i-
1]+1.01*Epre[k,i]-0.88*Epre[k,i-1];
subject to dTrange {k in bn, i in 1..4}: dT[k,i]<= 4;
subject to sq {k in bn,i in 1..4}: dT[k,i] = sqdT[k,i] * sqdT[k,i];

subject to Tadjusta1 {k in bn1, i in 1..13}: Eadj[k,i] = Ebase[k,i]*corr[k];
subject to Tadjust141 {k in bn1, i in 14..14}: Eadj[k,i]=Ebase[k,i]*corr[k] +
17.31 - 7.35*sqdT[k,i-13] - 0.31*Eadj[k,i];
subject to Tadjust151 {k in bn1, i in 15..15}: Eadj[k,i]=Ebase[k,i]*corr[k] +
16.53 - 8.01*sqdT[k,i-13] - 0.26*Eadj[k,i];
subject to Tadjust161 {k in bn1, i in 16..16}: Eadj[k,i]=Ebase[k,i]*corr[k] +
14.39 - 8.43*sqdT[k,i-13] - 0.20*Eadj[k,i];
subject to Tadjust171 {k in bn1, i in 17..17}: Eadj[k,i]=Ebase[k,i]*corr[k] +
12.56 - 8.22*sqdT[k,i-13] - 0.16*Eadj[k,i];
subject to Tadjustb1 {k in bn1, i in 18..24}: Eadj[k,i]= Ebase[k,i]*corr[k];
subject to PMV1 {k in bn1, i in 14..17}: 0.005*Eadj[k,i] + 0.124*dT[k,i-13] -
0.199 <= (if(PMVgoal-PMVmaxpre[k]*Tmax/Tmaxpre)> 0.15 then PMVgoal-
PMVmaxpre[k]*Tmax/Tmaxpre else 0.15);

subject to Tadjusta3 {k in bn3, i in 1..13}: Eadj[k,i] = Ebase[k,i]*corr[k];
subject to Tadjust143 {k in bn3, i in 14..14}: Eadj[k,i]=Ebase[k,i]*corr[k] +
15.62 - 4.4* sqdT[k,i-13] - 0.38*Eadj[k,i];
subject to Tadjust153 {k in bn3, i in 15..15}: Eadj[k,i]=Ebase[k,i]*corr[k] +
15.68 - 4.74*sqdT[k,i-13] - 0.363*Eadj[k,i];
subject to Tadjust163 {k in bn3, i in 16..16}: Eadj[k,i]=Ebase[k,i]*corr[k] +
15.61 - 5.08*sqdT[k,i-13] - 0.347*Eadj[k,i];
subject to Tadjust173 {k in bn3, i in 17..17}: Eadj[k,i]=Ebase[k,i]*corr[k] +
15.17 - 4.98*sqdT[k,i-13] - 0.343*Eadj[k,i];
subject to Tadjustb3 {k in bn3, i in 18..24}: Eadj[k,i]= Ebase[k,i]*corr[k];
subject to PMV3 {k in bn3, i in 14..17}: 0.00886*Eadj[k,i] + 0.0981*dT[k,i-
13] - 0.2885 <= (if(PMVgoal-PMVmaxpre[k]*Tmax/Tmaxpre)> 0.15 then PMVgoal-
PMVmaxpre[k]*Tmax/Tmaxpre else 0.15);

solve;
```

D.2.2 Data file

```
param PMVgoal := 1.5;


 param Epre :=
 :   1     2     3     4     5     6     7     8     9     10    11    12
 1   0     0     0     0     0     0     0     29.1  39.4  44.2  49    53.2
 2   0     0     0     0     0     0     0     29.1  39.4  44.2  49    53.2
 3   0     0     0     0     0     0     0     29.1  39.4  44.2  49    53.2
 4   0     0     0     0     0     0     0     29.1  39.4  44.2  49    53.2
 5   0     0     0     0     0     0     0     29.1  39.4  44.2  49    53.2
 6   0     0     0     0     0     0     0     30.2  39.4  41.1  42.8  44.6
 7   0     0     0     0     0     0     0     30.2  39.4  41.1  42.8  44.6
 8   0     0     0     0     0     0     0     30.2  39.4  41.1  42.8  44.6
 9   0     0     0     0     0     0     0     30.2  39.4  41.1  42.8  44.6
 10  0     0     0     0     0     0     0     30.2  39.4  41.1  42.8  44.6
     13    14    15    16    17    18    19    20    21    22    23    24    :=
     54.9  58.6  60.2  60.8  59.6  0     0     0     0     0     0     0
     54.9  58.6  60.2  60.8  59.6  0     0     0     0     0     0     0
     54.9  58.6  60.2  60.8  59.6  0     0     0     0     0     0     0
     54.9  58.6  60.2  60.8  59.6  0     0     0     0     0     0     0
     54.9  58.6  60.2  60.8  59.6  0     0     0     0     0     0     0
     45.1  47.4  48.9  50    49.6  0     0     0     0     0     0     0
     45.1  47.4  48.9  50    49.6  0     0     0     0     0     0     0
     45.1  47.4  48.9  50    49.6  0     0     0     0     0     0     0
     45.1  47.4  48.9  50    49.6  0     0     0     0     0     0     0
     45.1  47.4  48.9  50    49.6  0     0     0     0     0     0     0     ;


param Emaxpre :=
1  60.8
2  60.8
3  60.8
4  60.8
5  60.8
6  50.0
7  50.0
8  50.0
9  50.0
10 50.0;

param PMVmaxpre :=
1  1.21
2  1.21
3  1.21
4  1.21
5  1.21
6  1.00
7  1.00
8  1.00
9  1.00
10 1.00;
```

```
param Tmax := 28.0;
param Tmaxpre := 25.6;
param SSmax := 442;
param SEmax := 595;
param SRmax := 970;
```