DEVELOPMENT OF AN INTEGRATED BUILDING ENERGY SIMULATION WITH OPTIMAL CENTRAL PLANT CONTROL

BY

RUSSELL DEREK TAYLOR

B.S., The Pennsylvania State University, 1985
M.S., The Pennsylvania State University, 1988

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Aeronautical and Astronautical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1996

Urbana, Illinois

UMI Number: 9702682

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

MAY 1996

WE HEREBY RECOMMEND THAT THE THESIS BY

RUSSELL DEREK TAYLOR

ENTITLED DEVELOPMENT OF AN INTEGRATED BUILDING ENERGY SIMULATION WITH OPTIMAL CENTRAL PLANT CONTROL

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

C. O. Pedersen 1/9/96
Director of Thesis Research

W. C. Solomon 4/22/96
Head of Department

Committee on Final Examination†

W. C. Solomon
Chairperson

Judette L. Lieberman

Signature illegible

E. Tate

† Required for doctor's degree but not for master's.

O-517

Abstract

The purpose of computer-based building energy analysis programs is to assist heating, ventilation, and air conditioning (HVAC) engineers in the design process and to help researchers develop more effective and efficient methods of controlling a building's environment. One such program, BLAST (Building Loads Analysis and System Thermodynamics), is heat balance based and can model a building's thermal performance for an entire year. However, the component simulations: zones, fan systems, and central plants, are simulated separately and no feedback occurs. In this work a computational engine was developed to model building thermal processes so that all the elements of the simulation interacted. This new program IBLAST will, as a result, be able to simulate many of the fan systems and building thermal interactions that can not be modeled by BLAST.

An additional development of this work was the implementation of methods to optimally schedule both conventional and thermal storage central plant equipment in IBLAST. Optimal schedules, in this case, had minimum energy cost when variations in cost by: energy type, time of day, and peak energy demand, were taken into account. A limited search of the possible equipment operating schedules was used that both ensures a global minimum, and reduces the number of schedules calculated to a computationally feasible number. Results are presented for optimization problems involving cool thermal storage systems and various combinations of conventional chilling equipment. As energy cost parameters were modified, the optimal equipment operating schedule changed accordingly. In the case of thermal storage, increases in on-peak energy cost resulted in greater use of storage to meet the cooling loads. In conventional systems, high demand charges resulted in fewer changes in equipment usage from one time step to the next. If the conventional plant used more than one type of energy, relative changes in the costs of each energy source resulted in greater utilization of equipment powered by the cheapest energy. Results obtained from these methods are valuable in making cost-based comparisons between several plant equipment design options or evaluating the operating strategy of an in-service plant.

Acknowledgments

Completion of this thesis, while immensely satisfying on a personal level, is made all the more so because of the love and support given by my family and friends that helped see it through. Of course, my greatest thanks and love goes to my parents, Alan and Nancy Taylor, who have been there from the beginning with kind support and gentle words of encouragement. My advisor, Curt Pedersen, I thank for his willing guidance and helpful advice that has always steered me in the right direction. I could not have asked for a better mentor. I also want to acknowledge my friends in the BLAST Support Office who have helped me keep both my optimism and my sense of humour. However, I especially wish to thank: Rich Liesen for being an understanding friend when I really needed one, Dan Fisher for being a positive influence on my life, Rick Strand for introducing me to the thrills of the Autobahn, Kavon Pontius for her eternal patience, and Maria Pflaum for being there to fix my e-mail.

In addition, I would like to thank the members of my examining committee: Professors Wayne Solomon, Eric Loth, and Petros Voulgaris, from the Department of Aeronautical and Astronautical Engineering, and Professor Judith Liebman, from the Department of Mechanical and Industrial Engineering. I am truly greatful for the time and resources they have dedicated to this project, and the helpful comments they have provided.

I could not conclude without recognizing the love and fellowship I have received as a member of Emmanuel Memorial Episcopal Church, Champaign, Illinois, especially from: rector Fr. Alan Herbst, choir director Kirin Nielsen, and all the other members of the choir. They have shown me a love of God that has lifted my spirit and will sustain me each and every day of my life. There are many other people who have touched, and continue to touch, my life. To them I would also like to say thank you: Sandra A. Jaska; Ward and Kris Christman; Jim and Mary Ellen Trainer; Fred and Amy Williams; Neville Jopson; all my relations in England; my friends at Hobbico; and, finally, George Casey and Robert Boyer for sharing with me the joy and freedom of flight.

Table of Contents

Chapter 1: Introduction 1

1.1: Historical Perspective 2

1.2: Research Objectives 5

1.3: Benefits of BLAST Integration 6

1.4: Requirements for Optimal Control 7

1.5: Features of the Optimal Controller 8

Chapter 2: Literature Review 10

2.1: Simulation Based Approach to Optimization 10

2.2: Optimization of Thermal Storage Systems 13

2.3: Load Prediction for Thermal Storage 14

2.4: Applied Optimization Methods 15

Chapter 3: Development of the Integrated Building Simulation 17

3.1: Overview of BLAST Simulation 17

3.1.1: Zone Simulation 17

3.1.2: System Simulation 20

3.1.2.1: Overview 20

3.1.2.2: Simulation Methods 21

3.1.2.3: BLAST System Types 21

3.1.3: Plant Simulation 22

3.1.3.1: Overview 22

3.1.3.2: Simulation Methods 22

3.1.3.3: BLAST Plant Types 22

3.2: Integration of Zone and System Interactions 23

3.2.1: Zone Thermal Control 24

3.2.2: Zone-System Feedback in the Integrated Simulation .. 24

3.2.3: Calculation of Zone Surface Conduction Loads 27

3.2.4: System Control 30

3.3: Integration of System and Plant 36

3.3.1: Detailed Coil Models 36

3.3.1.1:Background 36

3.3.1.2: Heat Transfer and Energy Balance 37

3.3.1.3: Underlying Correlations, Properties, and Assumptions. 40

3.3.1.4: Solution Method of Model 43

3.3.1.5: Application of Cooling Coil Model to Heating Coils 43

3.3.2: Plant Models 44

3.3.2.1: Chiller Models 45
3.3.2.2: Boiler Models 46
3.3.2.3: Other Plant Equipment Models 47
3.3.3: System-Plant Energy Balance 48
3.3.3.1: Data Structure 48
3.3.3.2: Plant/System Water Loop 49
3.3.3.3: System and Plant Controls 50
3.4: Zone Update Method 52
3.4.1: Variable Time Step 53
3.4.2: Simultaneous Solution of Plant/System Water Loop .. 54
Chapter 4: Thermal Storage System Simulation 56
4.1: Types of Thermal Storage Systems 56
4.1.1: Direct Ice Storage 56
4.1.2: Indirect Ice Storage 58
4.2: Thermal Storage Model Implementation 60
4.3: Requirements for Optimal Control 62
Chapter 5: Optimal Building Control 66
5.1: Overall Criteria for Optimization of Building Systems 66
5.1.1: Energy Consumption 66
5.1.2: Energy Cost 67
5.2: Optimal Load Distribution 68
5.2.1: System-Plant Energy Balance 68
5.2.1.1 Heating Plant Governing Equations 69
5.2.1.2: Cooling Plant Governing Equations 72
5.2.1.3: System-Plant Interface Equations 76
5.2.2: Objective Function for Optimization of Load Distribution 79
5.2.3: Definition of Control Variables 80
5.2.3.1: Heating Plant Control Variables 81
5.2.3.2: Cooling Plant Control Variables 81
5.2.4: Variable Constraints 82
5.2.4.1: Constraints on the Heating Plant 82
5.2.4.2: Constraints on the Chilled Water Plant 83
5.2.5: Implementation of Plant Equipment Optimization in IBLAST 85
5.3: Optimal Scheduling of Conventional Plant Components 87

5.3.1: Objective Function for Optimal Scheduling 88
5.3.2: Minimization by Finding Least Cost Path 89
5.3.2.1: Algorithm Details 93
5.3.2.2: Implementation in IBLAST 99
5.4: Optimization of Thermal Storage Operating Schedule 101
5.4.1: Optimization Method 101
5.4.2: Implementation in IBLAST 104
Chapter 6: Testing and Validation 106
6.1: Objectives of Testing and Validation 106
6.2: Design of Test Cases 106
6.2.1: Building Envelope 107
6.2.2: Air Handling System 109
6.2.3: Central Plants 109
6.2.3.1: Multiple Equipment of the Same Size and Type 110
6.2.3.2: Multiple Equipment of the Same Type Using Different Sizes 111
6.2.3.3: Multiple Equipment of Different Types 111
6.2.3.4: Indirect Ice Storage Systems 112
Chapter 7: Results 113
7.1: General Observations 113
7.2: Multiple Equipment of the Same Size and Type 116
7.3: Multiple Equipment of the Same Type Using Different Sizes ..119
7.4: Multiple Equipment of Different Types 121
7.5: Indirect Ice Storage Systems 128
Chapter 8: Conclusions and Recommendations 135
8.1: General Conclusions 135
8.2: Implications for Building Simulation 138
8.3: Implications for Building Design and Operation 139
8.4: Recommendations for Future Work 139
Bibliography 141
Appendices 146
Appendix A: PLTCOMB.FTN: Listing of Subroutines Necessary for Central Plant Optimization 146
Appendix B: Subroutines Required for Ice Storage Optimization ..190
Appendix C: Sample IBLAST Input File 202
Vita 211

Chapter 1: Introduction

The purpose of this research was to develop a method that integrates all of a building's heat transfer processes to give an overall description of its thermal performance. These heat transfer processes are dependent on or influence one of three building components: the building envelope, the air handling systems, and the central plant. The envelope describes the physical spaces within the building in terms of the number and type of surfaces that make up the space and the way in which it should be controlled thermally. The air handling systems distribute hot and cold air to maintain thermal conditions with in the spaces between acceptable limits. Finally, the central plants provide sources of heated and chilled water to the air handling systems to produce hot and cold air supplies for the building spaces. The heat transfer mechanisms between these three building components are complex, and until now it has only been possible to model them approximately when considering how they affect an entire building. However, approximating these interactions places significant limitations on the types of building thermal control and systems that may be modeled. In addition, the level of detail that may be incorporated in the models and the realism of results obtained from the central plant models is also limited by such approximations. The integrated simulation approach used in this research makes available far more information about the way in which the zones, systems and plants react to the applied thermal loads. That information will, in turn, be of use in developing operating strategies for the installed equipment.

In this project, an integrated approach to building energy analysis using new simulation techniques has been demonstrated. The resulting simulation, based on this integrated method, is a powerful building energy analysis tool. Most significantly, it provides researchers and designers with a wider array of simulation choices and more comprehensive and useful simulation results than preceding and current software packages for buoilding energy analysis. For these reasons, the integrated simulation method has been selected as the building simulation engine for the National Energy Analysis Program (NEAP) project. While currently in its infancy, the purpose of this project is to merge the two major energy analysis software packages funded by the U.S. government to create a single program that would be the standard for evaluating compliance of a building design with existing building codes pertaining to energy consumption and conservation.

A final accomplishment of this research was to use the integrated energy analysis method to simulate optimal control of a building's central plant equipment. This is important in the context of building design because the choice of building equipment is a significant factor in determining the life-cycle cost of a central plant design. Two methods for optimal central plant control were developed: a general method that was applied to several types of conventional central plant and a method specifically for optimizing the operation of central plants using thermal storage to supply cooling. The result of the optimization, in each case, was an hourly schedule of equipment operation that minimized total energy cost taking into account both the hourly variations in energy cost and those assessed according to the maximum plant energy consumption.

1.1: Historical Perspective

In the mid 1970's the U.S. Army Corps of Engineers Construction Engineering Research Laboratory (USACERL) developed the Building Loads Analysis and System Thermodynamics (BLAST) program. The design of BLAST enabled it to perform a complete thermal simulation of a building by modeling its zones, air handling systems, and central plant. These features made the program an ideal tool for engineers to properly size fan system and central plant equipment for buildings of all types and uses. In addition, it allowed researchers to study and develop more energy efficient ways of controlling a building's thermal environment. The program has since been maintained and developed by the BLAST Support Office (BSO) at the University of Illinois. This affiliation has allowed BLAST to keep pace with the latest developments in building thermal control systems and simulation techniques. Contributing to the proven accuracy of BLAST is the use of an hourly energy balance on each zone to simultaneously solve for and update the conditions within the zone. This model accounts for all significant thermal interactions between the zone and its surroundings, including: external solar radiation, radiative exchange between zone surfaces, transient conduction through zone surfaces, convective transfer from internal surfaces to the zone air, mixing of air across zone boundaries, etc.

Hardware limitations at the time BLAST was developed required BLAST to model the zones, air handlers, and central plants separately and sequentially. First, the program simulates the zones which determines the amount of energy and the peak power consumption required to maintain the desired conditions. These conditions, determining the overall comfort of the zone occupants, are measured as a combination of air and radiant temperatures. The zones are controlled by using a piece wise linear "control profile" that approximates the response of the system to changes in zone temperature. Each control profile usually consists of several segments that mimic the throttling of the zone supply air flow rate and temperature to heat or cool the zone. After the cooling and heating loads on each zone are calculated for each hour of the simulation, they are stored . At the end of the zone simulation, the fan system simulation is performed using the calculated zone loads as inputs. The fan system simulation determines if the specified air handling unit can meet the zones' heating and cooling requirements and calculates each system's energy consumption. Finally, the loads from both the zone and fan system simulations are passed to the central plant. The central plant simulation determines if the specified equipment has sufficient capacity to meet all the loads and calculates the energy required. Information flows one way only in the BLAST simulation scheme, meaning that the effect of an improperly sized air handling system or central plant on the zone conditions is not directly calculated. It can only be deduced from the output.

Using three separate, sequential simulations also means that certain types of systems can not be simulated accurately because the control profile is only a function of zone temperature. This works well when the air handling system response is strongly dependent on the zone temperature, such as in a variable volume (VAV) system. But, when the system output is only a weak function of the zone temperature, or when it depends on other variables, such as the outdoor air temperature, the control profile does not approximate the system response well. An example is the case of a ventilation system that forces outside air into the zone. In this case, the system heating or cooling capacity depends on the difference between the outside air and zone air temperatures. Consequently, the control profile methodology fails badly. Another circumstance in which control profiles are inadequate is a system that uses an evaporative cooler to maintain the supply air temperature. Since the performance of the evaporative cooler unit depends on the outside air temperature, humidity ratio, and supply air flow rate and is not affected by zone temperature, control profiles are again inadequate.

The problems associated with the use of control profiles were not a great concern when BLAST was first developed. The systems with which these problems arose were not widely used. However, spurred on by rising energy costs, there is an increasing desire in the HVAC community to make more and better use of systems that provide "free cooling," such as the evaporative cooler. As a result, a new approach is needed to allow BLAST to explore all the possibilities afforded by these new system designs

Another limitation of BLAST is the reporting of so-called "unmet loads." When the air handling system is too small or improperly scheduled and the heating or cooling demands passed from the zone simulation are not met, the BLAST simulation reports an unmet load. Unmet loads measure the difference between the amount of conditioning required by the zones and the amount provided by the system. From this information it is possible to infer whether the system is improperly sized and by how much, or whether it is incorrectly scheduled. However, it is not possible to determine the deviation in zone temperature from the desired value because of the lack of feedback between the systems and the zones. In addition, the effect of the unmet load on zone comfort is difficult to determine.

The BLAST central plants simulation that models the building's chillers, boilers, generators, condensers and other components can also generate unmet loads that would indirectly affect zone conditions by reducing the available system output. However, plant sizing to meet air handling system demands is generally not as complicated as sizing fan system components. Individual pieces of plant equipment are simulated by specifying the nominal capacity and the part load performance as determined from curve fits of representative equipment data. As in actual plants, several sizes of the same type of equipment may serve the building systems. BLAST makes an additional assumption that all plant components of the same type have identical part load performance characteristics. This made it straightforward to distribute loads optimally to the plant components but, in practice, part load efficiency is a function of equipment size.

By the mid 1980's it was clear that a new program was needed to overcome the limitations inherent to the BLAST sequential simulation technique. The objectives of the new program were clear: allow feedback between the zones, systems, and plants; develop more realistic and intuitive models of zone and system controls; and incorporate more detail in the simulation as a whole. The first objective required the simultaneous solution of the zones, fan systems, and central plants. However, to do this, implementation of the second and third objectives was necessary. In BLAST, the system and central plant simulations "know" the loads they are required to meet and no direct controls are required, or feasible, between the simulations.

In an integrated simulation these loads are not known but result from the interactions of the three components. In real buildings, the zones and mechanical systems interact via controllers that regulate system and plant operation to maintain desired conditions in the building. These systems are usually very complex because, in addition to zone air temperature, they must control other system parameters: system airflow rate and pressure at various locations in the building duct network, zone supply air temperature and humidity, and chilled and hot water flow rates and temperatures in the coils. However, there is no need or desire to duplicate all the functions of an actual control system for energy analysis purposes. The principal requirement for incorporation of the desired feedback between zone, system, and plant simulations was to solve the energy and mass balances at the zone-system and system-plant interfaces. A detailed model of the heat transfer that occurs at the system-plant interface through the heating and cooling coils is also necessary to perform the system-plant heat and mass balances. Finally, elimination of the assumption of identical part load performance for different sizes of the same plant type, thereby allowing more realistic plant simulation and optimization, is an important aspect of any enhancement to the BLAST program.

1.2: Research Objectives

The primary goal of this research was the development of a numerical method to simulate concurrently: a building's zones, systems and central plant, by accurately modeling the necessary control and feedback mechanisms. The resulting integrated building simulation method would subsequently be implemented in a simulation tool for evaluation. The BLAST program was an ideal starting point because of the fundamental and proven soundness of the zone energy balance method it uses. This has been demonstrated over nearly two decades of application to building design and research problems in addition to substantial verification (Andersson et al., 1980; Bauman et al., 1981; and Yuill and Phillips, 1981). Clearly, having an accurate zone model is critical since it drives, to a large extent, the responses of the other simulation components. As an application of the integrated simulation, a secondary goal of this project was to develop and implement a generalized plant optimization scheme. This would first determine the optimal distribution of heating and cooling loads among the available central plant equipment at each time step. Second, at the end of each 24 hour period the minimum cost hourly schedule of the plant equipment components would be calculated.

Building thermal simulation programs help designers and researcher find ways to minimize energy use and energy cost. By incorporating the generalized optimization scheme in the integrated simulation, direct and useful comparisons of plant operating costs will be possible between central plant alternative designs. This is important in design practice because it can justify the choice of equipment for a building's central plant. Specific choices might include: the number and sizes of chillers or boilers to give a specified total capacity, the mixing of equipment with different energy sources such as: electric and diesel powered chillers or fuel-fired and electric boilers. Also important is the simulation of optimally controlled non-traditional plant components such as, chilled water or ice storage tanks. Implementation of such a simulation would help to provide a basis for comparison between the costs associated with thermal storage operation compared to more conventional equipment.

The operating strategy of a thermal storage system is to minimize the cost of providing cooling or heating to a building by shifting energy demand to take advantage of prevailing utility rate structures. Cool storage systems accomplish this by producing chilled water or ice during the off-peak hours when the building occupancy and energy consumption is low. The stored chilled water or ice is subsequently used to meet on-peak zone cooling loads passed from the fan systems. This set up has three benefits: 1) the bulk of the building cooling load is shifted to a period of the day when energy is at its cheapest, 2) the peak energy demand is reduced since the only energy expended to cool the building during on-peak hours is that required to circulate chilled water and air, and 3) the size of the chiller plant can be reduced. The chiller is also easier to size since this is based on the ability of the chiller plant to recharge the storage system with ice or chilled water during the off-peak hours.

General criteria for sizing storage tanks and controlling thermal storage systems are still a matter of considerable debate within the thermal storage community. Thermal storage systems, and ice storage systems in particular, operate most efficiently when the amount of the stored medium produced is just enough to meet the next day's load. In BLAST, the system cooling load is already known by the time the plant is simulated so that optimal control, in this sense, is easy to simulate. In an integrated simulation, the simplest way to achieve the same effect is to run through the same 24 hour period several times updating the load on the plant each time. However, an additional optimization problem should be considered. Frequently, thermal storage systems use a combination of storage and chiller capacity to meet on-peak loads. The relative fraction of ice storage to chiller use that also produces the lowest energy cost for the building depends on the load profile and also the utility rate structure. Finding a way to determine the best combination of storage and mechanical cooling will be an additional focus of this project.

An integrated building simulation based on BLAST can address many of the optimization issues raised above because the complex interactions between the zones, the air handling system, and the plant will be taken into account. Several previous and proposed studies, particularly regarding the use of thermal storage systems, have been based on using building cooling load profiles, such as could be generated with BLAST, to study optimal design and control. While beneficial to a degree, such studies do not take into account the effect interactions between the thermal storage system, the zones, and the zone temperature controllers have on overall performance and global optimization. In particular, the scheduling aspect of plant optimization will be emphasized since optimization of system components individually does not generally give overall optimum performance (Olson, 1987, 1990, 1993, and 1994).

1.3: Benefits of BLAST Integration

In addition to simulation of the thermal interactions between the major elements of a building, an integrated building simulation offers several advantages over sequential simulations, such as BLAST and DOE-2, currently in use. An integrated simulation can accurately model systems whose output does not depend strongly on the zone air temperature. This includes systems depending on ventilation of outside air, since the heating or cooling capacity of the system is a function of the difference between the outside air and zone temperatures. So far, BLAST control profiles have been unable to model such systems effectively. In addition, the input to specify zone and system controls becomes far more intuitive. Finally, realistic simulation of controls on the air handling system and central plant are possible because air and water loop models are included in the simulation. For example, in BLAST the coil leaving air temperature is an input parameter that changes artificially depending on the size of the coil load and the outside air conditions.

Central plant controls are nonexistent since the water loops between the chiller or boiler and the coils are not modeled. In an integrated simulation, the air handling systems respond to zone temperature setpoints similar to the response of a real system to a room thermostat setting. Coil leaving air temperatures and chilled or hot water temperatures can be controlled realistically by throttling the loop water flow rates and the chiller or boiler outputs. A further deficiency of BLAST is that all central plant equipment of the same type uses the same part load performance curves regardless of size. There is also a predetermined hierarchy of equipment usage according to energy type that does not account for energy cost or utility rate structures. For example, electric chillers are always scheduled to be used before diesel powered chillers. This has made it easy to optimize plant operation in the simulation, however it does not represent realistic plant operation. Therefore, an additional objective of an integrated simulation is to provide a method that allows different sizes of the same equipment type to be specified and simulated with appropriate part load performance curves. Finally, an optimization scheme that provides the flexibility to look at plant optimization in terms of energy cost and consumption is another essential component of the final integrated building simulation program.

1.4: Requirements for Optimal Control

Implementation of system and plant optimization routines is an important element in the development of an integrated building simulation program. In most cases, building equipment is designed to minimize energy cost while providing the necessary heating and cooling to the building zones. However, the kind of optimal control being described should be made clear. It is appropriate to think of a building's air handling system and central plant as having two levels of control. First, there is a "microscopic" control system, that adjusts valve and damper positions, to maintain: zone supply air temperatures, chilled or hot water temperature setpoints, minimum ventilation rates, duct pressures, etc. Actual buildings contain a myriad of sensors that measure these quantities and send appropriate signals to one or more controllers. These adjust a valve or damper position to maintain the physical parameter being controlled within specified limits. In simpler terms, the "microscopic" controls adjust the system and plant performance to maintain a set of desired setpoints. This level of control operates on a very short time scale, on the order of a second or less. However, such time scales are infeasible for whole building simulations with the current generation of computing equipment.

A second, macroscopic, level of control varies the system setpoints, such as the zone supply air temperature and coil water entering temperature, and adjusts the fraction of the load being met by each central plant component. This occurs on a time scale typically on the order of minutes. By varying the setpoints and equipment part load fractions, efficient operation of the air handling system and central plant can be achieved. How the microscopic control system maintains the desired setpoints by adjusting valves and dampers is largely irrelevant when considering the overall performance of an air handling system. Furthermore, simulation of the system and plant at such a refined level of detail would, in most cases be computationally prohibitive. Many modern buildings have the kind of macroscopic control of setpoints and equipment operation described here. Typically these functions are handled by an Energy Management and Control System (EMCS). The intention of this research is to model this level of control in conjunction with an integrated simulation of the elements of the BLAST program. The resulting program is called IBLAST (Integrated Building Loads Analysis and System Thermodynamics).

With sufficient building instrumentation to provide input data, IBLAST could be used as an EMCS in a building to provide optimal control of the air handling system and central plants. It is not the intention of this research to demonstrate the feasibility of this approach to building thermal control other than to consider the requirements for using the program in such a fashion. The beneficiaries of this research are building designers and researchers studying the interactions of zone controls and the systems and plants that provide the conditioning response. The simulation can provide baseline optimal energy consumption figures for a building that would be useful for comparison with conventional control schemes or actual building performance.

1.5: Features of the Optimal Controller

The wide range of options that may be exercised when developing the input for a building required that the optimization scheme developed in this work be general enough to work for all the air handling systems and central plant equipment types available in IBLAST. The simplest way to perform the optimization would be to minimize energy use or energy cost while ensuring that the temperature of each zone lies within the range of comfort appropriate to the use of the zone. Minimization of total energy use is a reasonable strategy for residential units and other small facilities not subject to time-of-day rates and demand charges. When the cost of energy is constant, minimizing total energy use is equivalent to minimizing energy cost. In contrast, industrial and commercial users of electricity are usually subject to utility rates that vary according to the time-of-use, increasing energy costs during daytime on-peak periods. In addition, demand charges put a per kilowatt premium on peak power consumption. This type of rate structure requires a control strategy that minimizes energy cost. But, the optimization problem is complicated because there is no direct relationship between steady state energy consumption and peak demand. An equipment operating schedule that results in minimum energy consumption could also have a high peak demand due to equipment start-up transients that cause spikes in power consumption . In general, the optimal solution trades increased total energy consumption for reductions in peak demand. Equipment is operated at less-than-optimal part load efficiency to eliminate changes in equipment usage and their associated transient power consumption peaks. In practice, this requires forecasting the loads on the systems to determine the appropriate schedule of plant control decisions that minimizes the peak energy consumption. Demand charges and time-of-day rates are the primary reason for the appeal of thermal storage systems since the chiller, normally accounting for the largest fraction of the energy required to cool a building, is instead used to charge a storage tank during the off-peak periods. This takes advantage of lower electricity rates in effect during the off-peak hours and greatly reduces peak electricity demand, especially if the chiller is only used to fill a storage tank and not also to provide supplemental cooling during the day.

Chapter 2: Literature Review

2.1: Simulation Based Approach to Optimization

In a building HVAC system, a goal of the system controllers should be to maintain occupant comfort with, usually, minimum energy cost. Clearly, the system and plant design have a major impact on the operating strategy that meets this goal. A greatly oversized system that operates at low part load ratios all the time will never compete, on an energy consumption or cost, basis, with a system that operates close to its optimum part load ratio most of the time. Simulating the proposed system and plant before equipment installation begins is thus an important step in building design to reduce or eliminate gross inefficiencies. A second question is how to operate an existing system to minimize its energy consumption or cost. When the equipment components are already specified, the optimization problem becomes one of determining: when to turn equipment on and off, what setpoint temperatures to use, and what flow rates are appropriate. These control decisions are usually made by the building's energy management system. Furthermore, this kind of optimization assumes a "black box" type of approach to modeling each component of the building's systems. For example, a chiller would be represented by a set of performance curves that return the minimum energy consumption for a specified load. How such curves are generated is the goal of simulations that look at building system and central plant equipment in detail.

Thus far, three types of simulation have been discussed: system design, overall system optimization, and system component optimization. These are not mutually exclusive, but can, and should, complement each other. Highly detailed models of a specific building system, such as a chiller or boiler, that consider all the components and variables associated with that system can generate performance data for large scale building simulations. Typically, such models are not appropriate for complete building simulations because they require a great deal of input, that is frequently difficult or impossible to obtain, and because they tend to be computationally intensive. The overall building simulation gives information on the way a specified set of components performs. By comparing the results obtained using different sets of system components, equipment selection can be made to most efficiently or most cheaply meet the building loads.

Even though implementation of detailed component based models was not considered appropriate for a full building simulation because of the level of detail required, a review of the literature in this area was important. Primarily, computational techniques were emphasized in this review, but how the results of the models might be used in an overall building simulation was also investigated. Component based research has emphasized the optimization of chiller operation since this represents a major component of energy usage in many buildings and because the efficient operation of chiller systems is a complex problem. Olson (1987, 1990, 1993, and 1994), Braun et al. (1989) and Hackner et al. (1985) have all developed detailed models of chiller operation to obtain optimization results. This approach provided valuable information on how to operate individual chiller components: pumps, fans, and cooling towers, and adjust temperature setpoints according to the cooling load on the chiller plant. As a consequence, general guidelines for equipment operation were developed. The control logic in a building's energy management system could use these guidelines, supplanting the need to incorporate a detailed and computationally expensive model of the chiller plant in the controller to predict and optimize performance.

However, in the above cases, the chiller plant was considered in isolation from the rest of the building. This was accomplished by generating a building cooling load profile using a separate program. Experimental data obtained directly from an existing building and system also could be used. A load profile represents the variation in the total amount of chilled water that a chiller must provide, as a function of time, to meet the cooling load. Olson's cooling load profiles used a one hour time interval since they were obtained from BLAST, which uses a one hour simulation time step. The goal was to find the combination of chiller operating parameters that provided the required cooling with the least amount of energy consumption. Dynamic interactions between the chiller and building simulations were not included or possible because of the use of load profiles. Therefore, no effect on the building simulation would be observed if the chiller could not meet the load. Such interactions would not be significant for a properly sized chiller, but these simulations provide no means of determining when this is the case. Indeed it was necessary to know, a priori, that there would be sufficient chiller capacity throughout the simulation.

Olson also looked in detail at the problem of chiller sequencing when energy cost is variable and there is a demand charge. Demand charges are assessed per kilowatt of peak power consumption for each day of the billing cycle. In the solution, Olson was able to determine an optimal trajectory of control decisions over the simulation period that minimized energy cost. The controls, in this case, simply determined the chillers to turn on or off at each time step based on a BLAST generated load profile. This was accomplished by setting up a matrix of feasible chiller operating modes, optimized for minimum energy consumption using Powell's (1983, 1984) Sequential Quadratic Programming (SQP) method, for each hour of the simulation. The optimal path through the matrix was then selected; however, optimal path selection was complicated by the startup penalty associated with a change in chiller plant operating mode. These start-up transients are short in duration, and so they do not consume much energy compared to the total consumption of the plant. But, peaks in consumption are generated that could be larger than the peak consumption occurring during steady state plant operation. These start-up peaks are significant if the change in chiller operation occurs when on-peak rates are in effect and peak consumption is being monitored for the purpose of assessing demand charges. Olson concluded that every possible path would require consideration to determine the global minimum energy cost. Therefore, his method was to use heuristic arguments in sequencing chiller operation that, although providing an improved solution, did not guarantee a global minimum energy cost. The method could be readily applied to other types of system or plant components.

Braun et al. (1987) used a quadratic fit approximation to find the minimum energy consumption in a component based chiller model. This worked because all functions tend to exhibit quadratic properties in the vicinity of an extremum. The sequencing problem was not considered in detail, but general guidelines were developed for minimizing instantaneous chiller energy consumption. These guidelines were of considerable benefit in analyzing the performance of the BLAST based optimization proposed in this research. Klein et al.(1988) have looked in detail at the performance of a steam turbine driven chiller through the use of a TRNSYS based model. TRNSYS is a building energy analysis program that performs a similar function to BLAST. However the code used to accomplish the analysis is made up of many subroutines each one of which describes a single component of the building or its systems. TRNSYS is a very flexible program, and can be adapted to solve many problems by piecing the component subroutines together as desired. In this research, operating rules for a specific chiller installation were developed from optimized performance maps based on the chilled water load, the outside air wet bulb temperature, and the cooling tower fan setting. These performance maps required many simulations, however, an improved control logic for the chiller system was obtained. The drawback of this kind of brute force approach is that the number of simulations required to cover all possible operating conditions grows rapidly as the number of control variables increases. Very quickly, the method becomes computationally impractical.

Although simulations to optimize complete building systems have been accomplished, such methods have typically used simplified models of the load (i.e. the zones) and the system components. Nizet et al. (1984) used an electrical network model to describe the zones and the performance of the VAV air handling system and plant components. The objective function, the integrated energy consumption over a 24 hour period, was minimized by using the conjugate gradient method. Comfort was maintained within the zone by including a penalty function in the optimization problem. The penalty function caused the objective function to increase rapidly as zone conditions deviated outside the desired range. The method proved to be effective and did not require a great deal of computational effort. However, the lack of detail in the simulation may limit the usefulness of results obtained from this method to developing general guidelines for system and plant control. Sud (1984) has also developed a simplified optimization procedure using a load model based on ASHRAE transfer functions. Transfer functions are a set of weighting factors to predict the current load based on past loads and the current environmental conditions. The loads were calculated hourly and the system energy consumption minimized at each time step based on system operation flowcharts that were highly specific to the system being simulated. Sud did not look at the problem of finding the global minimum energy consumption over the entire simulation period but did develop a set of specific control guidelines.

The control trajectory problem was tackled by Rink and Zaheer-Uddin (1993) who looked at the optimization of a multizone cooling system over a 24 hour period. In this example, again the load model was simplified and periodic boundary conditions were assumed to facilitate finding the optimal trajectory. In other words, environment and building load variables were assumed to vary on a 24 hour cycle. Zaheer-Uddin et al. (1990) also looked at the problem of finding optimal control trajectories for boilers being used to supply space heating and domestic hot water (DHW). The approach used assumed that the equipment considered was operated with "on-off" type controls.

2.2: Optimization of Thermal Storage Systems

The primary motivation behind the implementation and use of thermal storage systems has been to shift energy use from on-peak to off-peak periods of the day to take advantage of lower utility rates for electricity at night, and avoid demand charges. The systems available are primarily for cold storage and provide a supply of chilled water to the cooling coils. Use of thermal storage has the effect of leveling out the building electricity consumption profile over a daily cycle. This eliminates large peaks that occur during on-peak hours when equipment is started to meet increased building loads . The resulting reduction of peak demand saves on energy costs due to per kilowatt utility electric demand charges assessed over the billing period. Furthermore, off-peak use of electricity is cheaper than on-peak usage.

The optimal control strategy for minimizing the energy cost of thermal storage systems must look at an entire 24 hour period. The cycle begins in the early evening when utilities switch from on-peak to cheaper off-peak rates, and peak demand is no longer monitored. Off-peak rates are usually in effect until mid-morning of the following day. During this period, the chiller plant is used to charge the storage container with ice or chilled water and also to meet any cooling loads that may be incurred. When the on-peak period begins, cold storage can be used to meet all, or a fraction of the building's cooling demands. The reason the entire cycle must be considered is that the next day's cooling requirements must be predicted in order to determine how much ice or chilled water to store overnight. In addition, the controller must be able to optimize the operation and scheduling of the thermal storage system accounting for: the type of storage system, the cost of stored energy versus direct cooling, the utility rate structure, the magnitude and profile of the building loads, and the conditions in the building's external environment.

Spethmann (1989, 1993) looked at the effect of rate structure in determining an optimal control strategy and concluded that, in some cases, it does not make financial sense to meet the entire on-peak cooling load using storage since there is a break-even point where the cost of additional stored energy is larger than the savings made by reducing demand charges. Spethmann's basis for the next day's load prediction was the forecast high and low temperatures, and historical factors based on previous building thermal performance under similar weather conditions. Gray et al. (1988) looked at energy use for a building using simulations to compare the performance of conventional chilled water and thermal storage cooling systems. They concluded that thermal storage systems are most beneficial when the electric utility places a high penalty on peak power consumption through demand charges. The two factors that contributed to this conclusion were the high installation costs of thermal storage systems and that, per kilowatt-hour of actual cooling provided to the coils, thermal storage consumes more energy than conventional chiller systems. However, a recent proposal by Calmac to supply much colder air to the zones to reduce fan energy requirements and duct sizes should mitigate these arguments to a degree. The main problem with using very cold (less than 45°F) supply air and conventional diffusers in the zones is that it does not mix effectively with room air. The room therefore has "cold corners" that detract from occupant comfort. The Calmac system uses much smaller diffusers than normal that, correspondingly, result in a higher supply air jet velocity. The result is better mixing and a more uniform zone temperature distribution. Finally, Smith (1993), and Fiorino (1993), have looked at specific ice storage system installations and applied "intuitive" optimization strategies to the system controller in order to get the least cost performance. This type of "cut-and-try" procedure works well in an experimental setting but is less effective in enabling generalized control strategies to be developed. Furthermore, it would be very difficult to implement in a simulation program.

2.3: Load Prediction for Thermal Storage

Prediction of building cooling loads is an important component of the optimal control of thermal storage since it allows the system to determine how much ice or chilled water must be generated during the off-peak hours to meet on-peak loads. Thus, an essential part of thermal storage optimization in IBLAST is to relate on-peak loads to off-peak production of storage using load prediction or an alternative method. The methods used in actual thermal storage control systems range from the simple to the very complex. As indicated in a paper by Smith and Hittle (1993) one of the best simple control schemes assumed that the next day's load would be the same as the current day. This appeared to be most unsatisfactory however, the generation of an insufficient or excessive amount of ice was an infrequent occurrence when appropriate safety factors were used. Furthermore this method required minimal computational effort, and certainly less than load predictions based on the next day's weather forecast.

Other more complicated methods have been developed including an auto-regressive, AR, model used by Braun et al (1987) used to predict future cooling loads. Using the auto regression technique, a time series of coefficients is generated that, when multiplied by past inputs, allows the current output to be generated. Sufficient data must be available, initially, to calculate a set of time series coefficients to start the simulation. Subsequently, the coefficients can be updated to improve the model as the simulation progresses. As the time series was used to make predictions further forward in time from the current time, the predicted output became less reliable. An improvement over the AR method called auto regressive moving average ARMA is detailed by MacArthur et al.(1989, 1993)

One additional method of load prediction, that has received a lot of recent attention and has been applied to many different types of problems, uses artificial neural networks, ANNs. ANNs consist of a layer or layers of parallel nodes or neurons that are highly interconnected. Associated with each neuron is a set of parameters, usually termed "weights," that describe the relationship between the inputs to the neuron and its outputs. These weights can be "trained" by a series of control inputs and desired outputs so that the neural network as a whole can be used to model complex physical phenomena without use of the equations that govern those phenomena. Mistry and Nair (1993) have demonstrated the ability of a neural network to replicate the psychometric chart for moist air. In other words, given as input the dry bulb temperature and relative humidity, the neural network returns the values of the dew point temperature, the humidity ratio, and the enthalpy without the use of any thermodynamic relations. Anstett et al. (1993) and Curtiss et al. (1993) further demonstrated the use of neural networks in the prediction of building energy consumption. However, as was noted neural networks must be "trained" before they can be applied to a specific problem. An integrated building simulation program with optimization routines provides an effective and cheap alternative to the actual building for training the neural network.

2.4: Applied Optimization Methods

Applied optimization methods aim to improve the performance of HVAC systems using common sense or intuitive methods of optimization. These methods often use simulations of the plant components in isolation from the building. Therefore, predicted improvements in system and plant performance can overlook important thermal interactions. Using lower than normal chilled water and supply air temperature setpoints is a good example of a trial strategy. In this case, the pump and fan energy consumption would be reduced due to the lower flow rates required to provide the same amount of cooling. But, the coefficient of performance of the chiller plant would be reduced because it takes more energy to produce chilled water at the lower temperature. The increase in chiller power consumption could more than offset the reduction in pump and fan power consumption. Clearly, there are trade-off mechanisms at work that may not always be obvious and might cause problems if this trial strategy was applied indiscriminately .

A simplified model of a zone cooling system including the coil and chiller plant was employed by Braun et al. (1989) to develop some general optimizing criteria that were then applied to a real system. Clearly this approach had value in eliminating gross system efficiencies but could not be used for minimization of total energy consumption or cost over a specified period. Hackner et al. (1985) also looked at several different control strategies and evaluated their effects on minimizing chiller energy consumption. Finally, Austin (1993) and Lau et al. (1985) used a combination of simple models and experiments to develop optimizing strategies for a chiller plant.

In a more radical approach, Cumali (1988) applied the DOE2 energy analysis program to the problem of optimizing building energy consumption. DOE2 was used as the simulation kernel of a global optimization and control scheme, but few additional details were provided. The optimization scheme was reported to work well for the building to which it was applied. The zone loads were computed using weighting factors that were trained in a manner similar to that used for neural networks. The remaining simulation elements were modeled by solving the heat and mass balance problem between the coils and chiller components.

Chapter 3: Development of the Integrated Building Simulation

3.1: Overview of BLAST Simulation

In BLAST, the building zones, air handling systems, and central plant equipment are simulated sequentially with no feedback. A zone heat balance is used to update the zone conditions and the heating and cooling loads at each time step . This information is, subsequently, fed to the air handling simulation to determine the system response; but, that response does not affect zone conditions. This simulation technique works well when the system response is a well-defined function of the air temperature of the conditioned space. However, in situations where the system is dependent on outside conditions and/or other parameters of the conditioned space, the lack of feedback from the system to the building can lead to nonphysical results. For example, if the system provides too much cooling to a conditioned space the excess is reported by BLAST as an unmet load, in this case "overcooling". Other categories of unmet loads exist and are similarly reported by the program. While this kind of reporting enables the affected system or plant components to be properly sized, the system designer would, in most cases, prefer to see how far outside the desired range temperature the zone actually went.

The same types of situations can exist between the system and plant simulations when the plant components are incorrectly sized compared to the system demands. Again, the effects of the incorrect sizing are not felt by the systems and zones and BLAST reports "unmet loads". However, the BLAST method is not without advantages. Using separate simulations allows for very short program execution times, even for large and complicated buildings. In addition, the method simplifies the process of performing parametric studies to evaluate design alternatives. Once the building zones have been simulated, the results may be saved and used as input data for many different systems. This can eliminate many identical repetitions of the building simulation.

3.1.1: Zone Simulation

BLAST is structured to simulate the building, the fan systems, and the plants independently. Information is passed unidirectionally from the loads simulation to the fan system simulation to the central plants simulation. The loads simulation is performed first by computing an hourly energy balance for each zone using weather, scheduled loads (lights, people, etc.) and desired zone conditions. The zone energy balance equation is given by:

$$\sum_{i=1}^{N_{sl}} \dot{Q}_i + \sum_{i=1}^{N_{surfaces}} h_i A_i \left(T_{si} - T_z\right) + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p \left(T_{zi} - T_z\right) + \dot{m}_{inf} C_p \left(T_\infty - T_z\right) + \dot{Q}_{sys} = 0 \qquad (3\text{-}1)$$

where:

$$\sum_{i=1}^{N_{sl}} \dot{Q}_i = \text{sum of the convective internal loads}$$

$$\sum_{i=1}^{N_{surfaces}} h_i A_i \left(T_{si} - T_z\right) = \text{convective heat transfer from the zone surfaces}$$

$$\dot{m}_{inf} C_p \left(T_\infty - T_z\right) = \text{heat transfer due to infiltration of outside air}$$

$$\sum_{i=1}^{N_{zones}} \dot{m}_i C_p \left(T_{zi} - T_z\right) = \text{heat transfer due to interzone air mixing}$$

$$\dot{Q}_{sys} = \text{system output.}$$

Internal loads are due to: lights, electrical equipment, people, etc. However, it is not obvious from this formulation that the radiant component of these loads is not directly transmitted to the zone air. First it must be absorbed by the zone surfaces where it is accounted for in the surface radiation energy balance. Internal loads can represent variations in: zone occupancy, equipment usage, etc., by adjusting their peak value with hourly and daily schedules. In general, internal loads are not affected by the zone temperature. Heat transfer through zone surfaces is computed from the surface convection coefficient h_i and the surface temperature T_{si}, where each surface (wall, floor, roof, etc.) or sub-surface (window, door, etc.) is assumed to be isothermal. The surface temperatures are computed by performing heat balances on the inside and outside surfaces, and using conduction transfer functions to relate conditions across the surface. The conduction transfer function (CTF) method used in current versions of BLAST and its development is described in detail by Hittle (1980) along with requirements and limitations for its use. Infiltration represents heating or cooling due to air exchange between the zone and the outside environment. Sources of infiltration are doors, open windows, cracks in walls, etc. In BLAST, infiltration rates are calculated from the input maximum infiltration rate for each zone, modified by a set of coefficients that account for environment variables such as wind speed and direction. This allows BLAST to more closely match the actual infiltration characteristics of the building. The mixing term represents movement or exchange of air between zones. Peak mixing rates are input parameters and are modified by an hourly schedule and the temperature difference between the participating zones. A more detailed description of the loads computation used in BLAST can be found in the BLAST Users Manual (1993).

Since the air handling system is not simulated in the zone loads simulation, the system output $\dot{Q}_{sys}$ is estimated using a control profile. A control profile is a piece wise linear approximation of the system output as a function of zone mean air temperature T_Z:

$$\dot{Q}_{sys} = mT_z + b \tag{3-2}$$

where m is the slope of one linear segment, representing the change in system output with zone temperature. The segment endpoint, b, is the system output at T_Z=0. Multiple segments having different slopes may be pieced together, end-to-end, to manipulate the shape of the control profile. In this way, the heating and cooling responses of many different system types may be approximated. A generic single segment control profile is shown in Figure 3-1.

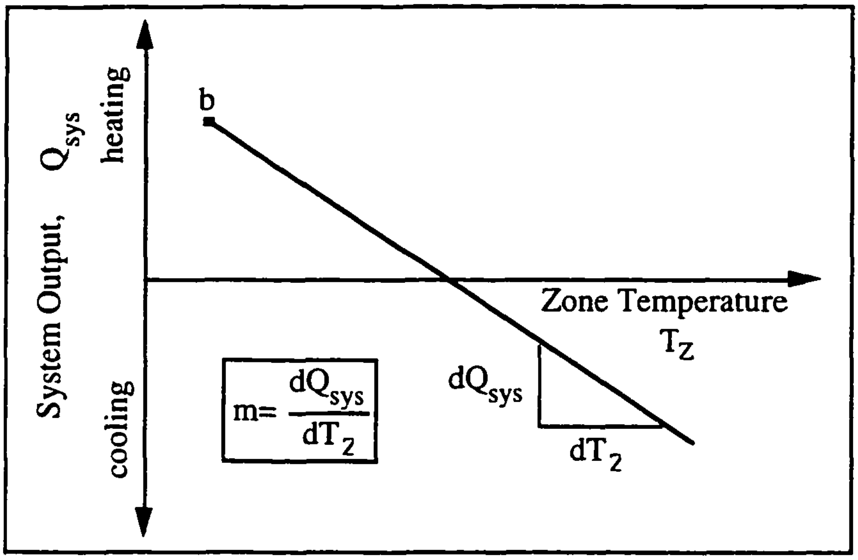


Figure 3-1: Generic single segment BLAST control profile

Equation 3-2 allows direct solution of the zone energy balance Equation 3-1 for T_Z, as shown in Equation 3-3:

$$T_z = \frac{\sum_{i=1}^{N_{sl}} \dot{Q}_i + \sum_{i=1}^{N_{surfaces}} h_i A_i T_{si} + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p T_{zi} + \dot{m}_{inf} C_p T_\infty + b}{\sum_{i=1}^{N_{surfaces}} h_i A_i + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p + \dot{m}_{inf} C_p - m} \tag{3-3}$$

BLAST obtains a solution by assuming that the zone temperature will be in a range corresponding to one of the control profile segments. Since the wall convection coefficients, the infiltration rates, and the mixing rates are also functionally dependent on the zone temperature, BLAST iterates on this equation until the change in T_Z is less than some tolerance value. When this occurs, the simulation stores the zone conditions and the system output, as calculated from the control profile, then the simulation moves on to the next hour. Because each control profile segment is only valid over a limited range of T_Z, it is possible for convergence to occur outside this range. When this happens, a different segment must be selected and the time step simulated again. This procedure is performed for each successive time step until the loads simulation is complete. The fan systems simulation then attempts to match the required $\dot{Q}_{sys}$ every hour based on zone and outside conditions, and the actual fan system performance. When the system output is strongly dependent on T_Z, the control profile closely matches the actual system performance and this technique works very well. However, in cases where the system output is also a function of the outside conditions or zone variables other than the zone temperature, a single control profile based on T_Z is no longer adequate to represent system-zone interactions. Once the fan systems loads have been calculated for the simulation period, they are used as input for the central plant simulation that attempts to supply the correct amount of energy for each time step of the simulation period.

3.1.2: System Simulation

3.1.2.1: Overview

Air handling systems perform the task of supplying hot or cold air to each building zone to meet that zone's heating or cooling loads. The range of fan system designs and control strategies available is large and each variation has advantages and disadvantages. However, fan systems currently in use have several common features. The most important of these are the heat exchangers, or coils, used to heat or cool the zone supply air. The coils are the primary interface between the system's air loop and the central plant chilled and hot water supplies, so it is essential to model them correctly is critical in a combined building simulation. In BLAST, however, no information is available about conditions on the coil water side because the plant is not simulated until after the system simulation is complete. BLAST overcomes this problem by first calculating the capacity of the coil based on nominal entering air and water conditions. Subsequently, during the simulation, the heating or cooling the coil must provide is calculated from the required air enthalpy change across the coil. This quantity is compared to the nominal coil capacity to determine if the load can be met, followed by calculation of the heat extracted or rejected to the water side which is the plant load.

A second important feature that most air handling systems have in common is a mixed air box. This device contains sets of dampers that regulate the amount of outside air being allowed to enter the system and mix with air from the return air plenum. Of course, return air must be exhausted to conserve mass in the mixing box. Outside air is brought in to ensure a sufficient supply of fresh air to the zones to keep building air contaminants such as carbon dioxide at or below acceptable levels. In addition, when conditions are favorable, outside air can be used to supplement the cooling coil through an economy cycle. BLAST simulates three economy cycles: temperature, enthalpy, and return air. These all allow the system to increase the amount of outside air brought into the building when it can enhance cooling.

Finally, the BLAST system models include the effect of air circulating fans, of which there are three types: supply, return, and exhaust. However, since the ductwork is not modeled in BLAST a fixed pressure rise across each fan is assumed. This, in conjunction with the fan part load ratio, allows calculation of the air temperature rise across the fan and the fan power consumption.

3.1.2.2: Simulation Methods

In BLAST systems, for a given supply air temperature, the required supply air flow rate for each zone can be calculated directly because the zone load is already known. Therefore, system variables are determined by starting at the zone and calculating backwards through the system air loop to the mixing box outlet. It is not possible to describe this procedure in general because the location of each system component in the air loop depends on the system configuration. However, it should be clear that: the system flow rate is known everywhere in the system air loop, the zone supply and return temperatures are known, and the outlet temperature of the mixing box can be specified from the type of economy cycle being used. This means that the system loop can be solved without the need for iteration.

3.1.2.3: BLAST System Types

The BLAST program incorporates a number of fan systems that can represent the operation of many of the systems commonly available to building designers. These systems, such as: the single zone draw through, variable volume, and dual duct, may be solved as described above. However, systems like the two and four pipe fan coil have additional components; the solution procedure , therefore, requires additional calculations but otherwise follows the same format. The BLAST User's Manual (Blast Support Office, 1993) should be consulted for a complete listing of these systems and their attributes.

3.1.3: Plant Simulation

3.1.3.1: Overview

The central plants simulation in BLAST models the components that supply hot and chilled water to the coils, usually chillers and boilers respectively, and generate electricity to supply system and zone demands. The loads imposed on the plant components are generated by the fan system simulation. The central plants simulation occurs last in the BLAST sequence so plant component performance has no effect on the system or building. However, as was the case with the system and building loads, inadequate plant capacity results in unmet loads being reported in the program output.

3.1.3.2: Simulation Methods

The BLAST central plants models are based on performance curves that, typically, are generated from manufacturer's data. Although each type of plant equipment has its own specific set of curves for which parameters must be supplied, BLAST supplies reasonable default values. In the case of boilers and chillers, the performance curves depend on the entering and leaving water conditions. But, since the water loop to the heating and cooling coils is not simulated in BLAST, nominal conditions are used. The primary goal is to meet the load without necessarily ensuring that inlet and outlet conditions are consistent with the coil outlet and inlet conditions respectively. BLAST allows up to three types of chillers, boilers, generators, and condensers on each building and there may be up to six sizes of each equipment type. In practice, this allows the part load ratio of operating plant components to be adjusted to provide the most efficient operation. However, BLAST assumes that all equipment of the same type uses the same part load performance curves. This simplified the problem of developing optimization procedures for the central plants simulation, which was accomplished by applying simple rules of thumb. Furthermore, when different types of equipment are available, BLAST assigns load to the available equipment according to a predetermined order that does not consider the relative cost of the energy types required for each equipment type. The effects of utility rate structures are ignored in determining the most cost effective equipment to operate. Exceptions to this are thermal storage plants that consider whether they are operating on- or off-peak to determine whether or not to use stored ice for cooling.

3.1.3.3: BLAST Plant Types

The central plant equipment types included in the BLAST program cover the majority of conventional chiller, boiler, generator and condenser components. In addition, models are available that simulate the performance of direct and indirect ice storage and chilled water storage. These latter systems require computational techniques different from conventional plant components because their performance is time dependent. In other words, the amount of chilled water or ice present in the storage tank changes the capacity of these systems with time. Finally, the different categories of plant can interact with each other because BLAST keeps track of the waste heat generated by some types of plant equipment. This waste heat is monitored by amount and grade and can be used to meet loads that require an equivalent or lower grade of heat input. An example is the steam turbine generator, the exhaust steam from which can be used in an absorption chiller or to produce domestic hot water. Additional detail on the BLAST central plant simulation methods and the plant types available may be found in the BLAST User's Manual (BLAST Support Office, 1993).

3.2: Integration of Zone and System Interactions

The BLAST building simulation, while being relatively fast computationally and allowing for comparisons between many types of equipment using a single zone loads calculation, is seriously limited by the lack of direct feedback between the three simulation elements of the program. This lack of feedback has been compensated for by reporting unmet loads. However these are not always easy to interpret. If the zone conditions were a true reflection of the system and plant performance, a greater sense of the actual capabilities of the systems and plants to control the building environment would be obtained. The logical solution to this problem was an integrated simulation in which the zones, fan systems, and plants interact with each other. Witte et al.(1989) and Taylor et al.(1990, 1991) have demonstrated several schemes in which the building and system portions of the BLAST simulation have been combined. Witte's approach used techniques that iterated on the zone and system simulations, such as the Newton-Raphson method, and retained the hourly nature of the simulation. Results obtained with this method were mixed; often the iterations would become unstable and the computational penalty was significant. As an alternative, Taylor et al.(1990, 1991) have concentrated on shortening the simulation time step, typically to between 0.1 and 0.25 hours, and using time-marching methods. This eliminates iteration because the system response is based on zone conditions lagged by one time step. However, the error associated with this approach depends significantly on the time step. The smaller the step size the less error, but the longer the computation time required to perform the simulation. Although requiring substantially more time to execute than BLAST, the improved realism of the results justified the use of the time marching "lagging with zone capacitance" method. This method was fully implemented in the IBLAST (Integrated Building Loads Analysis and System Thermodynamics) program and is described in detail below.

3.2.1: Zone Thermal Control

In BLAST, the temperature in each zone is controlled by specifying a control profile. The control profile specifies the heating or cooling that will be supplied to the zone as a function of zone temperature alone and, consequently, tends to maintain the zone temperature within the desired range as long as the capacity is sufficient and the slope of the profile is sufficiently steep. In fact, under these conditions, the zone temperature typically varies from the nominal design temperature only slightly. In actual buildings, a thermostat in the zone senses the temperature and sends a signal to the system to provide hot or cold air depending on the relationship of the thermostat setpoint to the zone temperature. In order to maintain a stable feedback loop and prevent rapid cycling of the system on and off, the zone temperature is allowed to oscillate around the setpoint. In the heating mode, the zone temperature rises above the setpoint before the thermostat shuts off the system. Then the zone cools below the setpoint before the system turns back on. However, such dynamic behavior was difficult to mimic in IBLAST without using a prohibitively small time step, i.e. on the order of seconds. But, the concept of a simulated zone thermostat, using the difference between the actual and desired zone temperatures to control the system, was found to be a useful one; but the normal zone dynamics associated with thermostat control had to be overlooked to preserve the stability of the simulation.

3.2.2: Zone-System Feedback in the Integrated Simulation

The method of lagging with zone capacitance uses information from previous time steps to predict system response and update the zone temperature at the current time. It can be thought of as a time marching method in which the value of a variable at the current time can be calculated directly from the results of calculations at one or more previous times without iteration. However, this method was prone to instability as the time step increased, and the overall error increased as well. In BLAST, the zone conditions at each time step are solved using a steady state iterative calculation that does not depend explicitly on the zone conditions at previous time steps. Hence the stability of the simulation is not a great concern and the size of the time step used is largely irrelevant. One hour is used in BLAST because it is convenient for record keeping purposes and it keeps computation time reasonable. But dynamic processes in the zone air can occur on a much shorter time scale than one hour; the time constant τ is on the order of:

$$\tau \approx \frac{\rho V C_p}{\left|\dot{Q}_{load} + \dot{Q}_{sys}\right|} \tag{3-4}$$

where the numerator is the zone air heat capacitance and the denominator is the net rate of heat energy input. Clearly, the value of τ can vary because the zone load and system output change throughout the simulation. Therefore, a variable adaptive time step much shorter than one hour was ideal for updating the zone conditions. Thus, it was necessary to derive an equation for the zone temperature that included the unsteady zone capacitance term and to identify methods for determining the zone conditions and system response at successive time steps.

Most systems simulated by BLAST provide hot or cold air to the zones to meet heating or cooling loads. The system energy provided to the zone, $\dot{Q}_{sys}$, can thus be formulated from the difference between the supply air enthalpy and the enthalpy of the air leaving the zone as in Equation 3-5:

$$\dot{Q}_{sys} = \dot{m}_{sys} C_p \left(T_{supply} - T_z \right) \quad (3\text{-}5)$$

This equation assumes that the zone supply air mass flow rate is exactly equal to the sum of the air flow rates leaving the zone through the system return air plenum and being exhausted directly from the zone. Both air streams exit the zone at the zone mean air temperature. The result of substituting Equation 3-5 for $\dot{Q}_{sys}$ in the heat balance Equation 3-1 and reformulating to include the effects of zone capacitance is shown in Equation 3-6:

$$C_z \frac{dT_z}{dt} = \sum_{i=1}^{N_{sl}} \dot{Q}_i + \sum_{i=1}^{N_{surfaces}} h_i A_i (T_{si} - T_z) + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p (T_{zi} - T_z) + \dot{m}_{inf} C_p (T_\infty - T_z) + \dot{m}_{sys} C_p \left(T_{supply} - T_z \right) \quad (3\text{-}6)$$

The sum of zone loads and system output now equals the change in energy stored in the zone. Typically, the capacitance C_Z would be that of the zone air only. However, thermal masses assumed to be in equilibrium with the zone air could be included in this term. In order to calculate the derivative term a finite difference approximation may be used, such as:

$$\left. \frac{dT_z}{dt} \right|_t \approx (\delta t)^{-1} \left(T_z^t - T_z^{t-\delta t} \right) + O(\delta t) \quad (3\text{-}7)$$

The use of finite differencing in a long time simulation is a cause for some concern due to the potential build-up of truncation error over many time steps. In this case, the finite difference approximation is of low order which further aggravates the problem. However, the cyclic nature of building energy simulations should cause truncation errors to cancel over each daily cycle so that no net accumulation of error occurs, even over many days of simulation (Walton, 1990). The Euler formula, Equation 3-7 was employed in Equation 3-6 to replace the derivative term. All the terms containing the zone mean air temperature were then grouped on the left hand side of the equation. Since the remaining terms are not known at the current time, they were lagged by one time step and collected on the right hand side. This manipulation resulted in Equation 3-8, the formula for updating the zone mean air temperature:

$$C_z \frac{T_z^t - T_z^{t-\delta t}}{dt} + T_z^t \left(\sum_{i=1}^{N_{surfaces}} h_i A_i + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p + \dot{m}_{inf} C_p + \dot{m}_{sys} C_p \right) =$$
$$\sum_{i=1}^{N_{sl}} \dot{Q}_i^t + \dot{m}_{sys} C_p T_{supply}^t + \left(\sum_{i=1}^{N_{surfaces}} h_i A_i T_{si} + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p T_{zi} + \dot{m}_{inf} C_p T_\infty \right)^{t-\delta t} \qquad (3\text{-}8)$$

One final rearrangement was to move the lagged temperature in the derivative approximation to the right side of the equation. The explicit appearance of the zone air temperature was thus eliminated from one side of the equation. An energy balance equation that includes the effects of zone capacitance was then obtained by dividing both sides by the coefficient of T_Z:

$$T_z^t = \frac{\sum_{i=1}^{N_{sl}} \dot{Q}_i^t + \dot{m}_{sys} C_p T_{supply}^t + \left(C_z \frac{T_z}{\delta t} + \sum_{i=1}^{N_{surfaces}} h_i A_i T_{si} + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p T_{zi} + \dot{m}_{inf} C_p T_\infty \right)^{t-\delta t}}{\frac{C_z}{\delta t} + T_z^t \left(\sum_{i=1}^{N_{surfaces}} h_i A_i + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p + \dot{m}_{inf} C_p + \dot{m}_{sys} C_p \right)} \qquad (3\text{-}9)$$

In addition to Equation 3-7, resulting in the formulation given by Equation 3-9, there are other finite difference expressions for the first derivative of the zone temperature with respect to time. By using several Taylor series expansions, higher order expressions for the first derivative, with corresponding higher order truncation errors, were developed. The goal of this approach was to allow for the use of larger time steps in the simulation than would be possible using the first order Euler form, without experiencing instabilities. Approximations from second through fifth order were tried as reported by Taylor et al. (1990) with the conclusion that the third order finite difference approximation gave the best results:

$$\left. \frac{dT_z}{dt} \right|_t \approx (\delta t)^{-1} \left(\frac{11}{6} T_z^t - 3 T_z^{t-\delta t} + \frac{3}{2} T_z^{t-2\delta t} - \frac{1}{3} T_z^{t-3\delta t} \right) + O(\delta t^3) \qquad 3\text{-}10$$

Furthermore, Taylor et al. (1990) showed that in the integrated zone and system simulation, time steps of 0.1 to 0.25 hours were adequate to maintain stability when the system response was well behaved. Well behaved, in this case, meaning that the system output varies continuously with zone temperature. Some real systems, such as the single zone draw through, are operated with on/off controls and the zone temperature inevitably oscillates about the setpoint. Modeling this behavior accurately required much shorter time steps than 0.1 hours and was therefore considered to be infeasible computationally. Therefore, stability of the simulation was determined as much by the method used to model the system response to zone temperature changes as the method used to update the zone temperature.

3.2.3: Calculation of Zone Surface Conduction Loads

In a program such as BLAST, conduction transfer functions are an efficient method to compute surface heat fluxes because they eliminate the need to know temperatures and fluxes within the surface. However, conduction transfer function series become progressively more unstable as the time step decreases. This became a problem as investigations into short time step computational methods for the zone/system interactions progressed because, eventually, this instability caused the entire simulation to diverge. This phenomenon was most apparent for thermally massive constructions with long characteristic times and, correspondingly, requiring a large number of terms in the CTF series. This indicates that the problem is related to round-off and truncation error and is in no way an indictment of the CTF method itself. Methods that develop CTF series from finite difference approximations to the heat conduction equation (Meyers, 1980; Seem, 1987) were considered to address this problem. Seem's method did give better accuracy and stability at short time steps than the current BLAST technique but, the method still had difficulty computing stable CTF series for time steps of less than 1/4 hour for the heaviest constructions in the BLAST library.

The zone load in IBLAST comprises contributions from specified internal heat gains, between zones, air exchange with the outside environment, and convective heat transfer from the zone surfaces. Of these, the surface convection load requires the most complicated calculations because a detailed energy balance is required at the inside and outside surface of each wall, floor, and roof. In addition, the transient heat conduction in the material between the surfaces must be solved. This solution gives the inside and outside temperatures and heat fluxes that must be known in order to calculate the convection component to the zone load for each zone surface. BLAST uses a conduction transfer function CTF method attributed to Hittle (1980) to solve the transient conduction problem for each surface. This results in a time series of weighting factors that, when multiplied by previous values of the surface temperatures and fluxes and the current inside and outside surface temperatures, gives the current inside and outside heat flux. The method is easily applied to multilayered constructions for which analytical solutions are unavailable. In addition, determining the series of CTF coefficients is a one time calculation, making the method much faster than finite difference calculations.

A problem with CTF methods is that the series time step is fixed; that is, a CTF series computed for a one hour time step takes information at t-1 hours, t-2 hours, etc. and computes conditions at the current time t. As time advances the oldest term in the input series is dropped and the data moved back one time step to allow the newest value to be added to the series. For convenience, the time step used to determine the CTF series should be the same as the time step used to update the zone mean air temperature in the zone energy balance. But, as the time step used to calculate the CTF series gets shorter, the number of terms in the series grows. Eventually, with enough terms, the series becomes unstable due to truncation and round-off error. Heavy constructions, such as slab-on-grade floors (12" heavyweight concrete over 18" dirt), have accuracy and stability problems at time steps as large as 0.5 hours when modeled by Hittle's CTF method. In an attempt to overcome this problem, Hittle's method was replaced by Seem's method (1987) in IBLAST. This resulted in some improvement in stability at shorter time steps, but not enough to allow IBLAST to run at a 0.1 hour time step without restricting the types of surfaces that could be used.

Even though CTF methods require that values of the surface temperatures and fluxes be stored for only a few specific times before the current time, the temperature and flux histories are, actually, continuous functions between those discrete points. However, there is no way to calculate information at these intermediate times once a series has been initialized. The terms in the temperature and flux histories are out of phase with these points. However, they can be calculated by shifting the phase of the temperature and flux histories by only a fraction of a time step. This procedure would allow a CTF series computed for a time step Δt, to be used to compute information at times t+Δt/2, t+Δt/3, t+Δt/4, or any other arbitrary fraction of the time step, so long as the surface temperatures and flux values were still Δt apart. Several ways of doing this are described below.

The method shown in Figure 3-2 maintains two sets of histories out of phase with each other. The figure shows how this would work for two sets of histories out of phase by one half of a time step. More sets of temperature and flux histories could be used, allowing the simulation time step to take on values: 1/3, 1/4, 1/5, etc., of the minimum time step allowed for the CTF calculations. The time step between inputs to the CTF series would be the smallest convenient interval at which the CTF series is stable. This scenario is illustrated in Figure 3-2 for two separate sets of temperature and flux histories. Cycling through each history, in order, allowed calculations of the zone energy balance to be performed with updated surface information at a shorter time step than one CTF history series would otherwise allow. This method required no interpolation between the series once each set of histories was initialized. However, if the smallest time step for a stable CTF series was large compared to the zone temperature update time step, significant memory was required to store all the sets of histories.

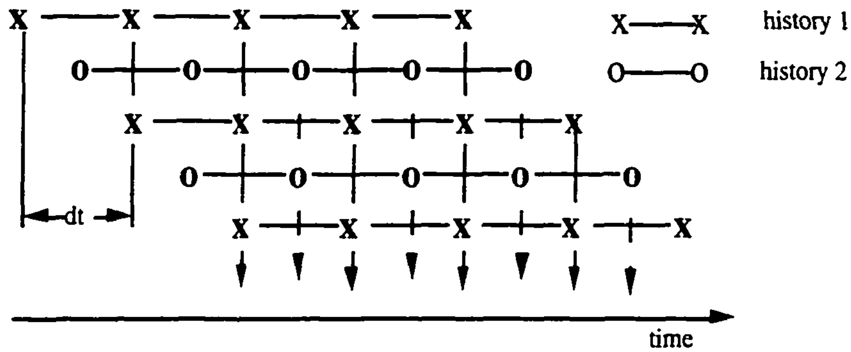


Figure 3-2: Multiple, staggered time history scheme

Another method is shown in Figure 3-3 that uses successive interpolations to determine the next set of temperature and flux histories. The current history is interpolated directly from the previous history set using the required time phase shift between the two. This method required permanent storage for only one set of temperature and flux histories at a time, but smoothed out temperature and flux data as more interpolations were performed. As a result, at concurrent simulation times current values of history terms were different form previous "in phase" history terms. This was unacceptable from, a physical point of view, because it allowed current information to change data from a previous time.

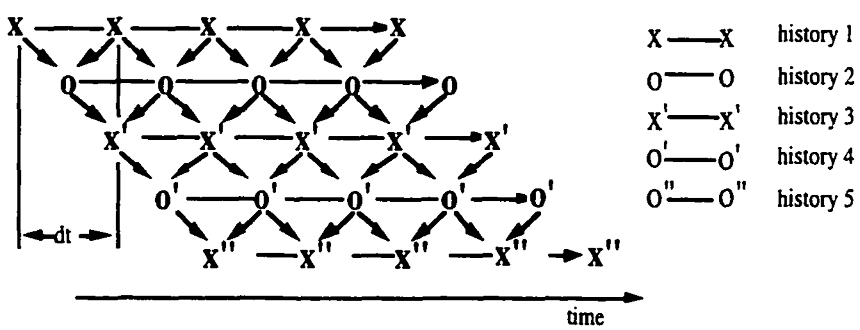


Figure 3-3: Sequential interpolation of new histories

A final method, shown in Figure 3-4, was something of a hybrid of the previous two methods. One "master" history set was maintained and updated for all time; this solved the problem of current events propagating information backwards in time. When surface fluxes needed to be calculated at times out of phase with this master history a new, temporary history was interpolated from the master values. This method proved to be the best of the three options described because it eliminated propagation of information backwards in time and only required concurrent storage of two sets of temperature and flux histories. This method was subsequently incorporated into the IBLAST program in conjunction with Seem's procedure for calculating the coefficients of the CTF series.

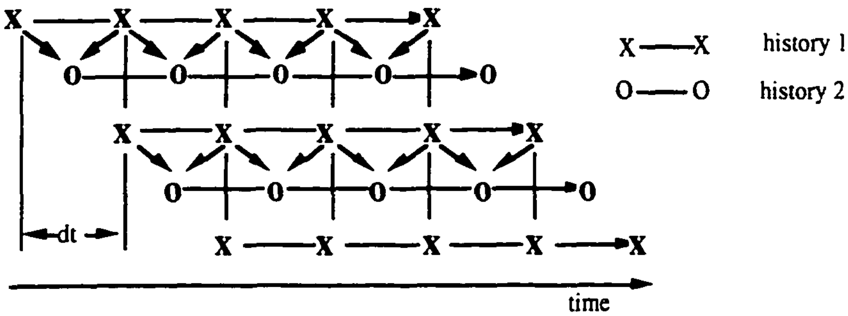


Figure 3-4: Master history with interpolation

3.2.4: System Control

Previously, the formulation of a new heat balance equation with an unsteady zone capacitance term was discussed (Equation 3-5). In this equation the updated zone temperature was calculated by removing its explicit dependence from the right hand side and lagging, by one time step, the unknown terms on that side. However, the right hand side still contains implicit dependencies on the zone temperature through the system control logic; the need for heating or cooling in the zones, is based on zone temperature. In real buildings the control system consists of one or more sensing units in the zone, such as a wall thermostat that samples the air temperature and sends signals to a control unit. The controller looks at the difference between the actual zone temperature and the desired temperature to ascertain if heating or cooling is required and then sends appropriate signals to the system components to drive the zone temperature closer to the desired value.

Although many control systems use only the zone temperature to control the system, most modern energy management systems consider many other variables, such as outside environment conditions. Simulating such controllers would seem to be relatively straightforward in a simulation especially since some of the more complex control problems, such as managing duct pressures and flow rates, are not modeled. However, real controllers have an advantage because they can sample zone conditions, and thus update system response, on a time scale much shorter than any characteristic time of the system or zone. Thus the feedback between zone and system usually results in steady or, at worst, slowly oscillating zone conditions and system operation unless the system is grossly oversized. On the other hand, the numerical model is only able to sample zone conditions at discrete time intervals. In the interest of minimizing computation time, these intervals need to be as long as possible. Frequently, they are of the order of, or longer than, the characteristic times of the system and zones, except in the case of small system capacity in relation to zone capacitance. This situation has the potential for unstable feedback between zone and system, resulting in an oscillatory or diverging solution.

Prior to implementing the new heat balance method in IBLAST, several system control strategies were considered. The primary objective was selection of a control method that would: be numerically stable over a reasonable range of conditions, realistic from the standpoint of looking and operating like an actual system controller, and flexible enough to be applied to all current and projected systems available in BLAST. The method actually implemented in IBLAST took advantage of the computational model's "knowledge" of how much energy enters or leaves the zone as a function of zone temperature i.e., the zone load. The real controller, on the other hand, does not have this information. The net zone load is given by Equation 3-11:

$$\dot{Q}_{load} = \sum_{i=1}^{N_{sl}} \dot{Q}_i + \sum_{i=1}^{N_{surfaces}} h_i A_i \left(T_{si} - T_z\right) + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p \left(T_{zi} - T_z\right) + \dot{m}_{inf} C_p \left(T_\infty - T_z\right) \qquad (3\text{-}11)$$

This is Equation 3-5 without the term due to the system. In addition, T_Z is now the *desired* zone temperature as defined by the control system setpoints that must be specified for each zone. An assumption was made that if the system has sufficient capacity (based on the desired zone temperature) to meet the zone conditioning requirements (i.e. $\dot{Q}_{sys} = \dot{Q}_{load}$) at the desired zone temperature then those requirements will be met. On the other hand, if the system can not provide enough conditioning to the zone to maintain the desired temperature, then the system provides its maximum output to the zone and the zone temperature is allowed to "float." Equation 3-11 was used to calculate the system output required to maintain the desired zone temperature; the actual zone temperature update was accomplished using Equation 3-9. This method was called *predictive system energy balance*. It has many characteristics of a predictor-corrector method since the system response is first approximated based on a predicted zone temperature and then the actual change in zone temperature is determined from that system response. The predictive system energy balance method required that the system controls on air mass flow rate, supply air temperature, etc., be formulated as a function of the zone temperature. However, this was not a serious drawback. The first example considered was a single zone draw through system. Typically, such systems have a cooling coil and heating coil in series, and constant air volume flow rate. Single zone draw through systems run at maximum capacity when turned on so the only way to regulate net system output and keep the zone temperature within the desired range is to turn the system on and off. A simplified schematic of this system type is shown in Figure 3-5.

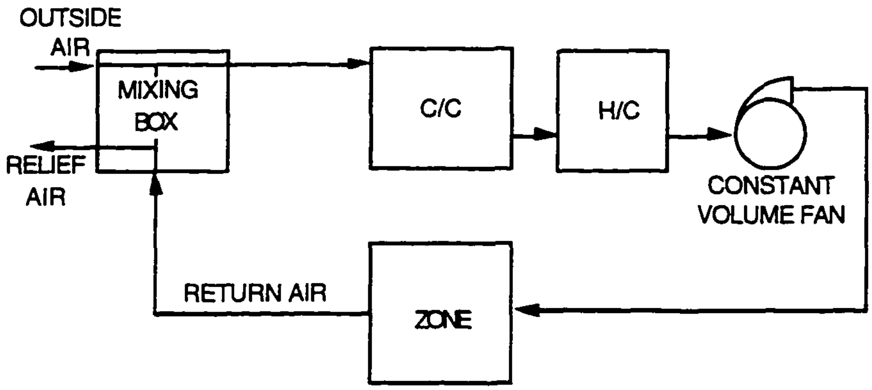


Figure 3-5: Simplified single zone draw through system

The amount of heating or cooling provided by the system in relation to the desired zone temperature is given by:

$$\dot{Q}_{sys} = \dot{m}_{sys} C_p \eta \left(T_{sup} - T_{z,desired} \right) \qquad (3\text{-}12)$$

where η is the fraction of the time step that the system is turned on and varies between 0 and 1. The supply air temperature is also implicitly limited by the effectiveness of the coils and the operating parameters of the central plant components. These interactions are considered later.

A far more complex, though again simplified, system is the variable air volume (VAV) system, shown in Figure 3-6. In VAV systems, the supply air temperature as well as the supply air volume are continuous functions of zone temperature. As shown in Figure 3-7, when the zone temperature is between T_{cl} and T_{cu}, cooling is required and the system varies the supply air flow rate while maintaining a constant supply air temperature. When the zone temperature is between T_{hl} and T_{hu}, heating is required and air is supplied at a constant minimum flow rate while the

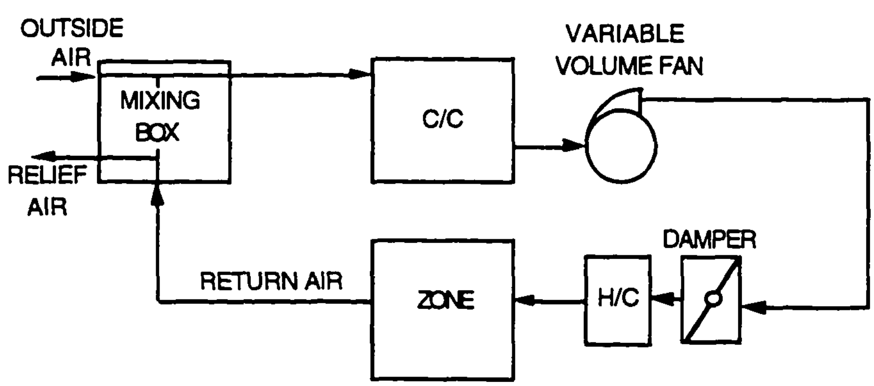


Figure 3-6: Simplified variable volume system

supply air temperature is varied. Figure 3-7 shows idealized behavior of a VAV system; in practice, the air flow rate and temperature are not exact linear functions of zone temperature.

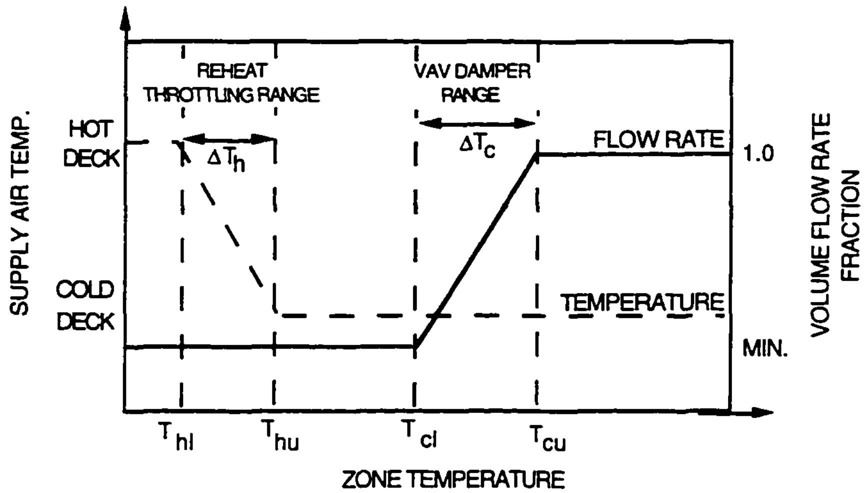


Figure 3-7: Idealized variable volume system operation

As long as a VAV system has sufficient capacity, the zone temperatures can be expected to vary within the limits defining the range of operation of the air damper, when cooling, or the throttling range of the reheat coil, when the system is heating. This means that the desired zone temperature, used to predict the system response, is variable and must be calculated in order to determine the system output. For the purposes of this calculation, the following definitions were found useful:

$$\dot{Q}_0 = \sum_{i=1}^{N_{sl}} \dot{Q}_i + \sum_{i=1}^{N_{surfaces}} h_i A_i T_{si} + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p T_{zi} + \dot{m}_{inf} C_p T_\infty \tag{3-13}$$

$$\dot{Q}_{slope} = \sum_{i=1}^{N_{surfaces}} h_i A_i + \sum_{i=1}^{N_{zones}} \dot{m}_i C_p + \dot{m}_{inf} C_p \tag{3-14}$$

Equations 3-13 and 3-14 are derived, respectively, from the numerator and denominator of Equation 3-3 but with the system related terms omitted. Also excluded from these expressions are the effects of zone capacitance.

When a zone requires cooling, the VAV system is designed to provide air to that zone at a constant supply air temperature. The amount of cooling is matched to the load by dampers in the supply air duct that vary the air volume flow rate of being supplied to the zone. Assuming that the volume flow rate varies linearly with zone temperature, the volume flow rate of supply air normalized to the maximum flow rate, or supply air fraction, is given by:

$$\eta_c = \eta_{c,min} + \left(1 - \eta_{c,min}\right)\left(\frac{T_z - T_{c,lower}}{T_{c,upper} - T_{c,lower}}\right); \eta_{c,min} \le \eta_c \le 1.0 \tag{3-15}$$

Normally, the minimum supply air fraction $\eta_{c,min}$ must be greater than zero to ensure a supply of fresh air sufficient to eliminate contaminants from the zone.

Conversely, when heating is required in a zone, the VAV system becomes a constant volume flow rate system with a variable supply air temperature. The dampers are set to provide air to the zone at the minimum supply air fraction. The supply air temperature is modulated by throttling the hot water supply to the reheat coil which effectively alters the coil's heating capacity. Again assuming the heat energy output varies linearly with zone temperature and normalizing with respect to the maximum coil output gives the following result:

$$\eta_h = \left(\frac{T_{h,upper} - T_z}{T_{h,upper} - T_{h,lower}}\right); 0 \le \eta_h \le 1.0 \tag{3-16}$$

Observe that when η_h is equal to zero, the zone is supplied with air at the cooling coil outlet temperature at the minimum air fraction. Because the control strategies of the VAV system are different whether the system is heating or cooling, two equations are necessary to describe the system output in terms of η_h and η_c. These expressions are as shown in Equations 3-17 and 3-18:

$$\dot{Q}_{sys,h} = \eta_h \dot{Q}_{h/c,max} + C_p \rho \dot{V}_{min}\left(T_{c/c} - T_{z,pred,heat}\right) \tag{3-17}$$

$$\dot{Q}_{sys,c} = C_p \rho\left(\eta_c \dot{V}_{max}\right)\left(T_{c/c} - T_{z,pred,cool}\right) \tag{3-18}$$

Equation 3-17 is valid for zone temperatures below $T_{h,upper}$, while Equation 3-18 is valid for all temperatures above this value. Equating the system output to the zone load, as given by Equation 3-11, the definitions of η_c and η_h were then used to develop expressions for the predicted zone temperature in the cases of heating and cooling:

$$T_{z,pred,heat} = \frac{\dot{Q}_{h/c,max} T_{h,upper}}{T_{h,upper} - T_{h,lower}} + \dot{Q}_0 + \frac{C_p \rho \dot{V}_{min} T_{c/c}}{\dfrac{\dot{Q}_{h/c,max}}{T_{h,upper} - T_{h,lower}} + C_p \rho \dot{V}_{min} + \dot{Q}_{slope}} \tag{3-19}$$

$$T_{z,pred,cool} = \frac{B_1 + \sqrt{B_1^2 + B_2}}{2} \tag{3-20}$$

where,

$$B_1 = T_{c/c} + T_{c,lower} - \frac{\eta_{c,min} - C_2}{C_1} \tag{3-21a}$$

$$B_2 = 4\left(\frac{C_3}{C_1} + T_{c/c}\left(\frac{\eta_{c,min}}{C_1} - T_{c,lower}\right)\right) \tag{3-21b}$$

and,

$$C_1 = \frac{1 - \eta_{c,min}}{T_{c,upper} - T_{c,lower}} \tag{3-22a}$$

$$C_2 = \frac{\dot{Q}_{slope}}{C_p \rho \dot{V}_{max}} \tag{3-22b}$$

$$C_3 = \frac{\dot{Q}_0}{C_p \rho \dot{V}_{max}} \tag{3-22c}$$

Once the predicted zone temperature has been calculated from Equations 3-19 and 3-20, the system response may be determined. When a zone requires cooling the system supply air temperature is constant at the cooling coil outlet temperature and the volume flow rate is given by:

$$\dot{V}_{supply} = \eta_c \dot{V}_{max} \tag{3-23}$$

where the supply air fraction η_c is computed from Equation 3-15. When heating is required by the zone the system provides air at the minimum volume flow rate and at a temperature given by:

$$T_{supply} = T_{c/c} + \frac{\eta_h \dot{Q}_{h/c,max}}{C_p \rho \dot{V}_{min}} \tag{3-24}$$

The reheat coil capacity fraction η_h is determined by using Equation 3-16. Once Equation 3-23 or 3-24, has been used, the supply air flow rate and temperature are known. These values are then used in Equation 3-9 to calculate the updated zone temperature. The equations describing VAV system operation may be solved without iteration if the cooling coil outlet temperature is constant, i.e. if the coil has infinite capacity, and if the reheat coil capacity varies linearly with zone temperature. This is not the case, either in practice or in simulations, when realistic coil models are used, as in IBLAST. Therefore, an iteration scheme was developed that solved these equations simultaneously with the coil performance models. Furthermore, linear throttling of heating coil output is not directly possible. In IBLAST the effect was simulated by throttling the coil hot water flow rate linearly with zone temperature.

3.3: Integration of System and Plant

In BLAST, the processes and equipment that allow feedback between the zones, fan systems and central plants are not modeled. In order to integrate the air handling system simulation with the zones simulation in IBLAST, methods were developed to model the system air loop and its interactions with the zones due to temperature controls and the relative difference between the zone and supply air temperatures. A similar situation was encountered when integrating the central plants simulation in IBLAST. Typically, the central plant interacts with the systems via a fluid loop between the plant components and the system heating and cooling coils. In BLAST, neither the coil nor the loop models were sufficiently detailed to be useful in the integrated simulation. The system simulation merely generated coil loads that were summed as appropriate and sent to the central plant which then determined whether or not it could meet the loads. The BLAST plant models are mostly based on curve fits generated from plant performance data. However, these models were detailed enough to generate water outlet temperatures given an inlet water temperature and flow rate. Although, more detailed models may eventually be desirable, the BLAST models were sufficient to demonstrate the feasibility of an integrated zone-system-plant scheme.

However, the BLAST coil models were not satisfactory. Since there is no water loop in BLAST, the coil models cannot make use of water side information. The coil capacities are initialized using nominal input conditions and these values are held throughout the simulation regardless of the plant capacity. Therefore, the BLAST coil models were replaced by models that obtain the coil outlet conditions by solving the heat balance between the air and water streams given the flow entering conditions and the coil geometry. Thus, the plant outlet conditions, coil inlet conditions, coil outlet conditions, and plant inlet conditions could all be related using energy and mass balances on the appropriate fluid streams.

3.3.1: Detailed Coil Models

3.3.1.1:Background

Heat transfer between the chilled water supplied by the central plant, and the air supply to the building zones occurs in heat exchangers, commonly referred to as cooling coils. In BLAST, the chilled water supply flow rates and temperatures to the coil are not needed because only a cooling load is passed to the central plant. The plant is assumed to provide chilled water at nominal design conditions to meet the cooling load. In IBLAST the performance of the systems and plant are interdependent because the simulations are combined. The plant outputs must match the system inputs and vice versa. That is, the temperature of the chilled water leaving the plant must equal the temperature of the water entering the coils, and the chilled water flow rate must satisfy mass continuity. In addition, coil controls are usually necessary to ensure that the values of chilled water flow variables entering and leaving the coil remain in a reasonable range. The specific controls vary from application to application but two common possibilities are: maintaining a constant coil leaving air temperature or limiting the water temperature rise across the coil.

In order to provide this simulation capability, a detailed coil model that predicted changes in air and water flow variables across the coil based on the coil geometry was required. A greatly simplified schematic of a counterflow cooling/dehumidifying coil is shown in Figure 3-8. In addition, the variables required to model a cooling coil and their definitions are extensively listed in Table 3-1. The input required to model the coil includes a complete geometric description that, in most cases, should be derivable from specific manufacturer's data. The coil simulation model is essentially the one presented by Elmahdy and Mitalas (1977) and implemented in MODSIM (Clark, 1985), a modular program also designed for energy analysis of building systems. The model solves the equations for the dry and wet sections of the coil using log mean temperature and log mean enthalpy differences between the liquid and the air streams. Elmahdy and Mitalas state that crossflow counterflow coils with at four rows or more are approximated well by this model. This does not constitute a major limitation since cooling and dehumidifying coils typically have more than four rows.

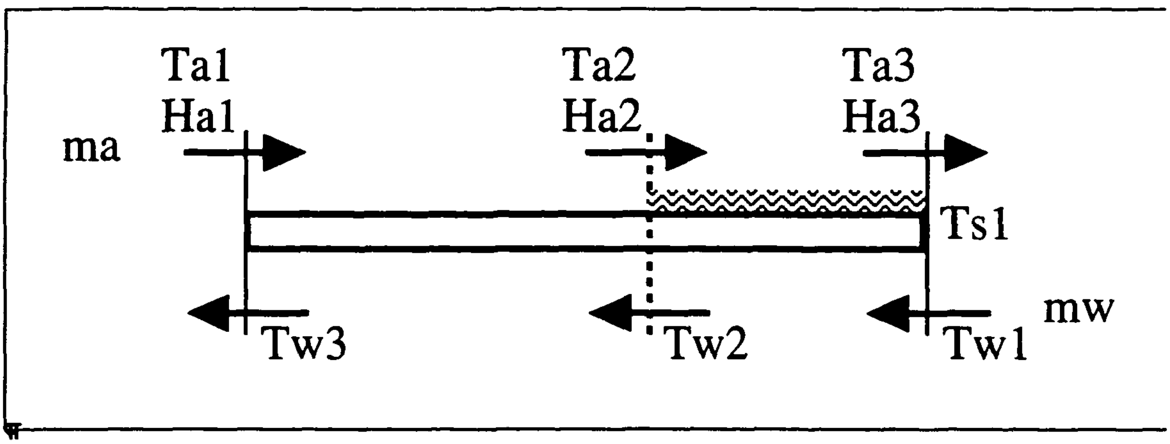


Figure 3-8: Simplified Schematic of Enthalpy and Temperature Conditions in a Counterflow Cooling/Dehumidifying Coil

3.3.1.2: Heat Transfer and Energy Balance

The cooling coil may be completely dry, completely wet with condensation, or it may have wet and dry sections. The actual condition of the coil surface depends on the humidity and temperature of the air passing over the coil and the coil surface temperature. The partly wet-partly dry case represents the most general scenario for the coil surface conditions. The all dry and all wet cases can be considered as limiting solutions of the wet or dry areas respectively going to zero. In the general case, equations are written for both the dry and wet regions of the coil. For each region the heat transfer rate from air to water may be defined by the rate of enthalpy change in the air and in the water. The rates must balance between each medium for energy to be conserved. Equations 3-25 and 3-26 and 3-27 and 3-28 express the energy balance between the water and the air for the case of dry and wet coils respectively. Equations 3-29 and 3-30 represent the heat transfer rate between water and air based on the actual performance of the coil. The UA parameter can be calculated from the parameters in Table 3-1. Equations 3-25 through 3-30 represent two

Table 3-1: Coil geometry and flow variables required to model cooling/dehumidifying coils

A	area	LMHD	log mean enthalpy difference
a	air, air side	LMTD	log mean temperature difference
aa, bb	coeff. in enthalpy approximation	$\dot{m}$	mass flow rate
C1, C2	coeff. in air side film coeff.	mf	metal and fouling
Cp	specific heat	μ	viscosity
D	diameter, effective diameter	o	outside (air side)
Dhdr	hydraulic diameter on air side	Pr	Prandtl number
d	dry region	$\dot{Q}$	heat transfer rate
δ	thickness	R	overall thermal resistance
Δ	spacing	Re	Reynolds number
f	heat transfer film coefficient	ρ	ratio of diameters
fai	variable in fin eff. calculation	s	surface, outside of metal
fin, fins	air side fin geometry	St	Stanton number
H	enthalpy	T	temperature
η	efficiency	tube	water tube
I0()	mod Bessel fn, 1st kind, ord 0	UAdry	dry heat xfer coeff. * dry area
I1()	mod Bessel fn, 1st kind, ord 1	UcAw	wet heat xfer coeff. * wet area
K0()	mod Bessel fn, 2nd kind, ord 0	ub, ue	variables in fin eff. calculation
K1()	mod Bessel fn, 2nd kind, ord 1	V	average velocity
i	inside (water side)	w	water, water side, or wet region
K1	variable in sol'n form of eq.	wa	humidity ratio
k	thermal conductivity	Z	variables in sol'n form of eq.
L	length	1, 2, 3	positions (see diagram)

sets of three equations with 7 unknowns: $\dot{Q}_d$, $T_{a,1}$, $T_{a,2}$, $T_{w,2}$, $T_{w,3}$, $\dot{m}_a$, $\dot{m}_w$. However, normally at least four of these variables are specified, for example: inlet water temperature, outlet air temperature, water flow rate, air flow rate, so that the system of equations is effectively closed.

$$\dot{Q}_d = m_a Cp_a (T_{a,1} - T_{a,2}) \tag{3-25}$$

$$\dot{Q}_d = m_w Cp_w (T_{w,3} - T_{w,2}) \tag{3-26}$$

$$\dot{Q}_d = (UA_{dry})(LMTD) \tag{3-27}$$

$$\dot{Q}_w = m_a (H_{a,2} - H_{a,3}) \tag{3-28}$$

$$\dot{Q}_w = m_w Cp_w (T_{w,2} - T_{w,1}) \tag{3-29}$$

$$\dot{Q}_w = (U_c A_w)(LMHD) \tag{3-30}$$

In order to manipulate these equations, the log mean temperature and enthalpy differences are expanded as shown in Equations 3-31 and 3-32. Finally, a linear approximation of the enthalpy of saturated air over the range of surface temperature is made using Equation 3-33. Note that in Equation 3-32 H_W refers to the enthalpy of saturated air at the water temperature.

$$LMTD = \frac{(T_{a,1} - T_{w,3}) - (T_{a,2} - T_{w,2})}{\ln \frac{T_{a,1} - T_{w,3}}{T_{a,2} - T_{w,2}}} \tag{3-31}$$

$$LMHD = \frac{(H_{a,2} - H_{w,2}) - (H_{a,3} - H_{w,1})}{\ln \frac{H_{a,2} - H_{w,2}}{H_{a,3} - H_{w,1}}} \tag{3-32}$$

$$H_w = aa + bb \ T_w \tag{3-33}$$

Equation 3-34 is derived from the above equations and is used to solve for the coil conditions when all of the inlet conditions are given as input. Operating in this manner, the coil does not have a controlled outlet air temperature.

$$T_{w,2} = \frac{(1-Z)(H_{a,1} - aa - K1\ Cp_a\ T_{a,1}) + Z\ T_{w,1}\left(bb - \frac{m_w Cp_w}{m_a}\right)}{bb - Z\frac{m_w Cp_w}{m_a} - (1-Z)\ K1\ Cp_a} \tag{3-34}$$

An alternative solution method is to define the coil leaving air temperature as an input with a variable water flow rate. In this case Equations 3-35 and 3-36 are more convenient. Equations 3-37a-c define terms that are used to simplify Equations 3-34, 3-35 and 3-36.

$$T_{w,2} = \frac{(1-Z)(H_{a,3} - aa) + T_{w,1}\left(\frac{m_w Cp_w}{m_a} - bb\ Z\right)}{\frac{m_w Cp_w}{m_a} - bb} \tag{3-35}$$

$$T_{w,2} = \frac{(Z_d - 1)T_{a1}Cp_a + T_{w,3}\left(Cp_a - Z_d\frac{m_w Cp_w}{m_a}\right)}{Z_d\left(Cp_a - \frac{m_w Cp_w}{m_a}\right)} \tag{3-36}$$

$$Z = \exp\left(U_c A_w\left(\frac{1}{m_a} - \frac{bb}{m_w Cp_w}\right)\right) \tag{3-37a}$$

$$K1 = \frac{Z_d - 1}{Z_d - \frac{m_a Cp_a}{m_w Cp_w}} \tag{3-37b}$$

$$Z_d = \exp\left(U_c A_{dry}\left(\frac{1}{m_a Cp_a} - \frac{1}{m_w Cp_w}\right)\right) \tag{3-37c}$$

3.3.1.3: Underlying Correlations, Properties, and Assumptions

Overall heat transfer coefficients are calculated from the specified coil geometry and by using empirical correlations from fluid mechanics and heat transfer. For the water side, Equation 3-38 gives the film heat transfer coefficient in SI units:

$$f_i = 1.429(1 + 0.0146\ T_w)V_w^{0.8}D_i^{-0.2} \tag{3-38}$$

This is valid for Reynolds numbers greater than 3100 based on water flow velocity and pipe inside diameter and is given in Elmahdy and Mitalas (1977) as recommended in the standard issued by the Air-Conditioning and Refrigeration Institute (1972) for air-cooling coils. The definition of overall inside thermal resistance follows directly as shown in Equation 3-39.

$$R_i = \frac{1}{f_i A_i} \tag{3-39}$$

Equation 3-40a gives the film coefficient for the air side. Another form of the same equation is Equation 3-40b, which is familiar from the data presented in Kays and London (1984). For coil sections that have a wet surface due to condensation, the air side film coefficient is modified according to Equation 3-41. The correction term, a function of air Reynolds number, is valid for Reynolds numbers between 400 and 1500. The coefficients in Equation 3-40 are calculated by Equations 3-42 and 3-43 that are functions of the coil geometry. Elmahdy (1977) explains the modifier for the wet surface and coefficients for the film coefficient. Equations 3-44(a-d) show definitions and values of common parameters and properties.

$$f_o = C_1 Re_a^{C_2} \frac{m_a}{A_{a\_min\_flow}} Cp_a Pr_a^{2/3} \tag{3-40a}$$

$$C_1 Re_a^{C_2} = St_a Pr_a^{2/3} \tag{3-40b}$$

$$f_{o.w} = f_o\left(1.425 - 5.1\times10^{-4} Re_a + 2.63\times10^{-7} Re_a^2\right) \tag{3-41}$$

$$C_1 = 0.159\left(\frac{\delta_{fin}}{D_{hdr}}\right)^{-0.065}\left(\frac{\delta_{fin}}{L_{fin}}\right)^{0.141} \tag{3-42}$$

$$C_2 = -0.323\left(\frac{\Delta_{fins}}{L_{fin}}\right)^{0.049}\left(\frac{D_{fin}}{\Delta_{tube\_rows}}\right)^{0.549}\left(\frac{\delta_{fin}}{\Delta_{fins}}\right)^{-0.028} \tag{3-43}$$

$$D_{hdr} = \frac{4A_{a\_min\_flow}\delta_{coil}}{A_{s\_total}} \tag{3-44a}$$

$$Re_a = \frac{4\delta_{coil}(1+w_a)m_a}{A_{s\_total}\mu_a} \tag{3-44b}$$

$$Pr_a = 0.733 \tag{3-44c}$$

$$\mu_a = 1.846\times10^{-5} \tag{3-44d}$$

The film coefficients above act on the extended surface of the air side, that is the area of the fins and the tubes. Therefore the fin efficiency must also be considered in calculating the overall thermal resistance on the outside. Gardner (1945) gives the derivation of Equation 3-45, used as a curve fit to find the fin efficiency as a function of film coefficient. This equation is based on circular fins of constant thickness. To model a coil with flat fins, an effective diameter, that of circular fins with the same fin area, is used. Equations 3-46(a-d) define variables used in

Equation 3-45. The overall efficiency of the surface is shown by Equation 3-47. Note that the efficiency is found by the same equations for the wet surface using the wet surface film coefficient.

$$\eta_{fin} = \frac{-2\rho}{fai(1+\rho)}\left[\frac{I_1(u_b)K_1(u_e) - K_1(u_b)I_1(u_e)}{I_0(u_b)K_1(u_e) + K_0(u_b)I_1(u_e)}\right] \tag{3-45}$$

$$fai = \frac{(D_{fin} - D_{tube})}{2}\sqrt{\frac{2f_o}{k_{fin}\delta_{fin}}} \tag{3-46a}$$

$$\rho = \frac{D_{tube}}{D_{fin}} \tag{3-46b}$$

$$u_e = \frac{fai}{1-\rho} \tag{3-46c}$$

$$u_b = u_e\rho \tag{3-46d}$$

$$\eta_o = 1 - (1 - \eta_{fin})\frac{A_{fins}}{A_{s\_total}} \tag{3-47}$$

The definition of overall outside thermal resistance is given in Equation 3-48 as a function of fin efficiency and film coefficient. For a wet coil surface the resistance must be defined differently because the heat transfer equations are based on enthalpy rather than temperature differences, as shown in Equation 3-49.

$$R_o = \frac{1}{f_o\eta_o A_{s,total}} \tag{3-48}$$

$$R_{o,w} = \frac{Cp_a/bb}{f_{o,w}\eta_{o,w}A_{s,total}} \tag{3-49}$$

Equation 3-50 gives the last two overall components of thermal resistance. They represent the metal tube wall and internal fouling. The fouling factor, due to deposits of dirt and corrosion of the tube inside surfaces, is assumed to be 5×10^{-5} $m^2\cdot K/W$. All components of thermal resistance are added in series to produce the overall heat transfer coefficients shown in Equations 3-51a and 3-51b.

$$R_{mf} = \frac{\delta_{tube}}{k_{tube}A_i} + \frac{Fl}{A_i} \tag{3-50}$$

$$UA_{dry} = \frac{A_{dry}}{A_{s,total}}\left[\frac{1}{R_i + R_{mf} + R_o}\right] \tag{3-51a}$$

$$U_cA_w = \frac{A_w}{A_{s,total}}\left[\frac{1/bb}{R_i + R_{mf} + R_{o,w}}\right] \tag{3-51b}$$

3.3.1.4: Solution Method of Model

The complicated equations derived above were implemented in a successive substitution solution procedure to calculate the coil performance based on the input parameters. The MODSIM implementation of a cooling coil, the TYPE12 subroutine, was the motivation for this approach; the method used there has been retained with modifications for the uncontrolled coil model. Clark (1985) contains notes about the MODSIM routine.

In the general case, the cooling coil is only partially wet. For an uncontrolled coil, Equation 3-34 is used to find the water temperature at the boundary. Several simple equations in the loop adjust the boundary point until the dry surface temperature at the boundary is equal to the dew point of the inlet air. For the controlled coil, Equations 3-35 and 3-36 give two calculations of the boundary temperature, and the water flow rate and boundary position are adjusted until the two equations agree.

Special cases occur when the coil is all wet or all dry. The coil is solved as if it were all wet before the general case is attempted. If the wet surface temperatures at the coil inlet and outlet are both below the dew point, no further solution is required. However, to ensure a continuous solution as flow variables are changed, when the surface is all dry or when it is wet with only the dry surface equations yielding a surface temperature below the dew point at the water outlet, the general solution is used to calculate the unknowns. In the solution of the controlled coil the outlet air enthalpy, given some resulting dehumidification, must correspond to the enthalpy at the specified outlet air temperature.

3.3.1.5: Application of Cooling Coil Model to Heating Coils

The implementation of detailed heating coil models in IBLAST was another important aspect of the system/plant integration. The same kind of loops exist to provide hot water to the heating coils from the boilers as exist to supply the cooling coils with chilled water from the chillers. Some simplifications can be made, however, since the enthalpy change of the air flowing over a heating coil is entirely sensible. There is no condensation in a heating coil. In order to allow heating and cooling coils to be specified using the same geometric parameters, a heating coil simulation was developed from the cooling coil model described above by eliminating the wet surface analysis.

In addition, it was concluded that, since much simpler and less computationally expensive heating coil simulations are possible, an option was provided in IBLAST for a heating coil design using only the UA value of the coil, the product of heat transfer coefficient and coil area. This model was largely based on the TYPE10 subroutine implemented in MODSIM. The equations used to model the performance of the TYPE10 heating coil are as follows:

$$T_{a,out} = T_{a,in} + \left(T_{w,in} - T_{ain}\right)\varepsilon\left(\frac{\min\left(C_{p,a}\dot{m}_a, C_{p,w}\dot{m}_w\right)}{C_{p,a}\dot{m}_a}\right)$$
$$T_{w,out} = T_{w,in} - \left(T_{a,out} - T_{ain}\right)\left(\frac{C_{p,a}\dot{m}_a}{C_{p,w}\dot{m}_w}\right) \tag{3-52}$$

where the coil effectiveness is given by:

$$\varepsilon = 1 - \exp\left(\frac{\left\{\exp\left[-\left(\frac{\min\left\{C_{p,a}\dot{m}_a, C_{p,w}\dot{m}_w\right\}}{\max\left\{C_{p,a}\dot{m}_a, C_{p,w}\dot{m}_w\right\}}\right)\{NTU\}^{0.78}\right] - 1\right\}}{\left(\frac{\min\left\{C_{p,a}\dot{m}_a, C_{p,w}\dot{m}_w\right\}}{\max\left\{C_{p,a}\dot{m}_a, C_{p,w}\dot{m}_w\right\}}\right)\{NTU\}^{-.22}}\right) \tag{3-53}$$

The parameter NTU is the number of transfer units and is defined as a function of the UA value of the coil as follows:

$$NTU = \frac{UA}{\min\left(C_{p,a}\dot{m}_a, C_{p,w}\dot{m}_w\right)} \tag{3-54}$$

3.3.2: Plant Models

The heating and cooling coil models provided IBLAST with a method to link the system air loop, that provides hot or cold air to condition the building zones, to the chilled and hot water loops served by the central plant. The next step in the integration process was to add the plant components to the simulation so that the performance of these components would affect the conditions in the loop. These components are, typically: chillers, that supply chilled water to the cooling coils; and boilers, that supply hot water or steam to the heating coils. Even though simplified, the BLAST central plants models were suitable for implementation in IBLAST. Furthermore, this was achieved with very little modification to the individual plant component simulations. The major changes required were in the routines controlling the chiller and boiler simulations. These now had to provide input to the plant components based on the results of the coil simulations instead of the nominal values obtained from the input file.

In addition to chillers and boilers, BLAST also simulates generators that can be used to meet the building's electric load. However, feedback between the generator and the building would not normally be an important consideration unless the generator were the building's only source of electrical power. In most cases, insufficient generating capacity would be supplemented by electricity purchased from the local utility. But, the strategies for reducing load on an overloaded generator are not well defined and would probably be different for each occurrence of insufficiency. That being the case, modeling the feedback effects between an overloaded generator and the building and developing rules for shedding generator load was considered beyond the scope of this work, especially since such occurrences should be rare. Scheduling and efficiently utilizing boiler and chiller capacity is, however, an important concern in any building where more than one of either component is present. Therefore the main emphasis of this work was accurately modeling the chillers and boilers and their interactions with the air handling systems to enable implementation of an optimization scheme.

3.3.2.1: Chiller Models

A typical chilled water plant may have several chillers operating in parallel. This arrangement allows the chillers to be sequenced to operate near their optimal part load ratio for a much larger fraction of the time the plant is operating than if one large chiller were used. The chiller models in IBLAST were adapted directly from those in BLAST level 65+ and are described in detail in the BLAST Users Manual. Additional information may be found in the ASHRAE Equipment Handbook. The following chiller types were implemented in IBLAST by modifying the respective BLAST models: air cooled, diesel driven, direct cooling tower, double bundle, free cooling, gas turbine driven, open, and reciprocating.

The operating characteristics of each chiller are simulated using equipment performance parameters obtained from manufacturer's data. All the models use the same basic set of four equations to simulate chiller operation and determine power consumption. The distinct operating characteristics of each chiller type are obtained from these equations by adjusting four sets of performance parameters The first set of performance parameters (ADJT) is used to calculate the chiller equivalent temperature difference ΔT:

$$\Delta T = \left[\frac{\left(T_{cond} - ADJT(1)\right)}{ADJT(2)}\right] - \left[T_{cw} - ADJT(3)\right] \tag{3-55}$$

In BLAST, the values of T_{cond} the condenser water leaving temperature and $T_{chiller,\,out}$ the leaving chilled water temperature are specified by the user and do not change during the simulation. In IBLAST, the leaving chilled water temperature must be the same as the coil entering water temperature. Next the ratio of chiller available to nominal capacity ANCR is determined using the calculated ΔT and a second set of performance parameters RCAV:

$$ANCR = [RCAV(1)] + [RCAV(2)]\Delta T + [RCAV(3)](\Delta T)^2 \tag{3-56}$$

The ADJE performance parameter set is used to compute the full load power ratio FLPR. The full load power ratio is given by the power consumption at the available capacity divided by the available chiller capacity. The ratio of FLPR to the nominal full load power ratio is then given by:

$$\frac{FLPR}{NFLPR} = [ADJE(1)] + [ADJE(2)](ANCR) + [ADJE(3)](ANCR)^2 \tag{3-57}$$

Finally, the fraction of full load power FFL is the ratio of chiller actual to full load power consumption calculated from the RPWR parameter set and the chiller part load ratio PLR:

$$FFL = [RPWR(1)] + [RPWR(2)](PLR) + [RPWR(3)](PLR)^2 \tag{3-58}$$

where PLR is the cooling load divided by the actual capacity of the chiller. The calculation of the chiller actual power consumption varies according to the type of chiller being simulated. Additional details regarding this calculation and the determination of the four performance parameter sets from manufacturer data is detailed in the BLAST User Reference. Once the amount of cooling provided by the chiller to the coil water supply has been calculated the result must be checked for consistency with the enthalpy change of the water across the chiller. That is, the result must satisfy Equation 3-59:

$$OCAP + \dot{m}_{cw} C_{p,w} (T_{chiller,out} - T_{chiller,in}) = 0 \tag{3-59}$$

where OCAP is the actual cooling provided by the chiller. This equation must be solved simultaneously with the coil performance equations to obtain chiller entering and leaving water temperatures that match the cooling coil entering and leaving water temperatures.

3.3.2.2: Boiler Models

Two types of boiler models are available in the IBLAST simulation: a fossil fuel fired boiler and an electric boiler. As with the chiller models. the boiler simulations are based on curve fits to manufacturer performance data that can be tailored to match specific equipment. In the case of the fuel boiler the performance is specified by the RFUELB data set that is used to compute the ratio of theoretical fuel consumption to actual fuel consumption. The theoretical fuel consumption is computed from:

$$\dot{Q}_{fuel,theoretical} = \frac{\dot{Q}_{load}}{\left[0.87 - 1.25\left(\frac{STRATB}{HFUELB}\left[T_{leave} - T_{air}\right]C_{p,exhaust}\right)\right]} \tag{3-60}$$

where STRATB is the fuel air ratio in lb/lb, HFUELB is the heating value of fuel in Btu/lb, T_{leave} is the boiler stack leaving temperature in °F, T_{air} is the ambient air temperature in °F, and $C_{p,exhaust}$ is the specific heat of the exhaust gas and is assumed to be a constant value of 0.24 Btu/lb-°F. The theoretical fuel consumption is then used to compute the actual fuel consumption of the boiler from:

$$\dot{Q}_{fuel,actual} = \frac{\dot{Q}_{fuel,theoretical}}{\left[RFUELB(1) + \{RFUELB(2)\}PLR + \{RFUELB(3)\}(PLR)^2\right]} \tag{3-61}$$

With this model the performance of the boiler is independent of the entering and leaving water temperatures so it is only necessary to ensure that:

$$\dot{Q}_{load} + \dot{m}_{hw}C_{p,w}(T_{boiler,in} - T_{boiler,out}) = 0 \tag{3-62}$$

This equation must be satisfied in conjunction with the solution of the heating coil performance equations. The electric boiler is somewhat simpler to model and details of the parameters required to simulate such a boiler can be found in the BLAST User Reference.

3.3.2.3: Other Plant Equipment Models

The boilers and chillers, the plant equipment types that have been discussed so far, both have a direct effect on the operation of the building's air handling systems. This is because they have a finite capacity limiting the maximum water flow rate that can be heated or cooled to a specified temperature. However, two other types of plant equipment exist that have much less effect on the operation of the fan systems and the building in general: these are condensers and generators.

The IBLAST condenser model computes the power input required by the condenser to meet the load passed to it by the chiller regardless of the size of the load. Effectively, this means that condensers never run out of capacity. This methodology allows correct sizing of condenser equipment but not observation of the effects of the condenser running out of capacity. More detailed models of both chillers and condensers would be required to accurately model feedback effects. This is analogous to the need for detailed coil models to simulate the air handling systems and the central plant. The types of condenser currently available in IBLAST are: cooling tower, evaporative condenser, and well water condenser. Their operation is simulated using curve fits of actual performance data in the same way as the chillers and boilers.

The BLAST generator models have not yet been integrated into the IBLAST program. Three generator types are available in BLAST: diesel, gas turbine, and steam turbine driven generators. These can meet all or part of a building's electric power demands, however, it is not yet clear how IBLAST would deal with a situation where the generator runs out of capacity. Additional modeling work needs to be accomplished in this regard before feedback can be allowed between the building and systems and the generators in IBLAST.

3.3.3: System-Plant Energy Balance

Sections 3.3.1 and 3.3.2 have described in some detail how new models were needed before the task of integrating the fan system simulation with the central plant could be undertaken. So far, the techniques used to ensure consistency between the plant and system inputs and outputs have not been discussed. In addition, the effect that the integrated simulation method might have on future program usage has not been considered. Therefore it is useful to describe the function and features of the plant models in BLAST by way of contrast to the IBLAST methods. Then the methods used to obtain a self-consistent solution of system and plant parameters and the rationale behind them will be discussed.

3.3.3.1: Data Structure

Significant preparatory work had to be accomplished before attempting to combine the system and plant models in IBLAST. In the BLAST scheme the central plants were designed to be simulated one at time after both the zones simulation and the system simulation had been performed. That is, the plant components are simulated after generation of zone and system loads for the simulation period. Therefore, feedback can not occur between the systems and plants. One plant, representing the collection of chillers, boilers, generators, thermal storage systems, condensers, and other auxiliary equipment serving a set of the fan systems on the building, is simulated at a time. Therefore, only one set of plant equipment performance parameters was required to be available at a given time. Parameters for other plants are stored on disk until needed. In IBLAST, all the plants serving the building are simulated concurrently so their performance parameters cannot be stored on disk until needed without incurring serious penalties on program execution time. Concurrent storage of all the equipment parameters for each plant required significant changes in the original data structure. By contrast, multiple plants may be specified in

BLAST, each of which may serve some or all of the building's systems. This capability allows users to see the effect of different combinations of plant equipment on energy consumption without repeating the building and system simulations and is one of the major strengths of the BLAST simulation method. The tradeoff between BLAST and IBLAST is between having the flexibility to examine many options or the ability to examine one case in detail.

The BLAST method and the manner in which the program stores input and simulation data is primarily a result of the limited memory of the computers available when the program was developed in the late 1970's. Much effort went into designing ways to efficiently make use of disk storage to save input parameters and intermediate results. At the same time, the simulation retained sufficiently rigorous physical models to provide useful results to the designer. In BLAST, parameters are read from the input file then immediately stored in a scratch file until the plant that they define is simulated. When the plant is simulated, the input parameters are read back from the file along with system loads information that is stored in a separate file. The results of each plant simulation is written to yet another scratch file until needed by the reporting routines. This method puts a much greater emphasis on the use of disk space than random access memory (RAM), reflecting the more limited computational capabilities available when the BLAST program was originally developed.

As building technology has advanced, however, the interactions that occur between the system and plant components have become more complex and modeling them correctly is important if accurate results are to be obtained. In order to simulate multiple plants concurrently, a number of basic changes were made to the central plant simulation data structure and the processing of: input data, output data, and simulation results. Since all the input parameters required for each central plant must be available throughout the simulation, in IBLAST most of the intermediate files have been eliminated. Instead equipment parameters are read from the input file directly into the variables that are needed for the plants simulation. Additionally, these variables now must be arrays. Therefore, IBLAST requires considerably more RAM than BLAST, but intermediate output files are largely eliminated. This is a significant advantage for reducing computational time.

3.3.3.2: Plant/System Water Loop

In addition to the changes in data structure, new models were implemented to allow for realistic interactions between the system and plant components. These models simulate the loops supplying hot and chilled water to the heating and cooling coils, the heat exchange that occurs in those coils, and the control of the water and air temperatures and flow rates through the coils. The key elements to providing this simulation capability were the detailed heating and cooling coil models and the improved central plant component models described previously.

In actual buildings, the interactions between the fan systems and the central plant occur through the water loops that connect the plant components to the coils. A change in the temperature of the water leaving the plant affects the capacity of the coils to heat or cool the air flowing over them. Likewise, changes in the water temperature leaving the coils influences chiller and boiler performance and the energy required to produce a desired outlet water temperature. However, BLAST only passes energy consumption rates from the system to the plant; no attempt is made to simulate the details of the system plant water loops. When variables representing flow conditions in the water loop are required, to simulate a coil or component of plant equipment, nominal design conditions are used. The development of cooling and heating coil models allows interactions between water and air side variables to occur. But, the coil models alone are not sufficient because they provide no way of determining coil air and water inlet conditions.

Coil air inlet conditions are relatively easy to specify since they are dependent on the ambient outside air and zone return air conditions. The water side inlet conditions are a function of the capacity of the chillers or boilers and their ability to provide a certain flow rate of water at a specified temperature. Conservation of mass and energy between the coils and the plant allow the following relationships to be defined:

$$\sum_{\substack{\text{cooling}\\\text{coils}}} \dot{m}_{w,in} = \sum_{\substack{\text{cooling}\\\text{coils}}} \dot{m}_{w,out} = \dot{m}_{chiller,in} = \dot{m}_{chiller,out} \quad (3\text{-}63)$$

$$T_{cooling\ coil,in} = T_{chiller,out} \quad (3\text{-}64)$$

$$\sum_{i=1}^{\#\substack{\text{cooling}\\\text{coils}}} \dot{m}_{i,w,out} T_{i,w,out} = \dot{m}_{chiller,in} T_{chiller,in} \quad (3\text{-}65)$$

Obviously, these equations do not represent a closed system since, as has already been observed, the coil outlet conditions are determined by the coil performance and the coil inlet conditions are determined by the operation of the central plant components. These equations merely represent the link between the coil and plant simulations.

3.3.3.3: System and Plant Controls

In the previous two sections, the interactions between the systems and plant have been discussed in detail. The heat exchange occurring between the air and water in the coils and the relationship of the coil water side inputs and outputs to the plant inputs and outputs have also been defined. In addition it has been noted that models of the specific plant equipment types are required to relate the plant inputs to the outputs. However, even with such models the system of equations is not closed because for a given load on the air handling system there are an infinite number of combinations of supply air temperature and flow rate that can meet that load. The same can be said of the hot or chilled water flow to the coils. The parameters that must be explicitly specified in order to close the set of equations and obtain a solution are those that duplicate the functions of system and plant controls.

Controls are the devices used by the system and plant to regulate the quantity and temperature of air flowing out of the coils so that the correct conditioning is provided to the building zones. This can be expressed mathematically in terms of the steady state zone energy balance as follows:

$$\dot{Q}_{load} + \dot{m}_{air,in} C_{p,air} \left(T_{\substack{supply \\ air}} - T_{zone} \right) = 0 \qquad (3\text{-}66)$$

where $\dot{Q}_{load}$ is the rate of energy transfer to the zone due to external and internal loads: radiation, conduction through the walls, infiltration, people, electrical equipment, etc. Since it is impossible to control $\dot{Q}_{load}$ in any practical way, the goal of the building air conditioning system is to adjust $T_{supply\ air}$ and $\dot{m}_{air,in}$ so that Equation 3-66 is satisfied and T_{zone} is maintained within a range consistent with the comfort requirements of the buildings occupants.

Implementation of controls in IBLAST was accomplished using the two forms of coil model described previously. In one form, the coil leaving air temperature is specified along with the air inlet conditions and the inlet water temperature. The air flow rate through the coil is set to satisfy Equation 3-66 and the chilled water flow rate is then an output of the coil model determined by the energy balance in the coil. Setting a fixed coil leaving air temperature or having a fixed relationship between the leaving air temperature and another parameter, for example the outside air dry bulb temperature, is a common practice in actual HVAC installations.

A second common option is to run a completely uncontrolled coil, frequently called a "wild coil." The temperature of the air leaving the coil is determined by the system air flow rate and the plant operating capacity. In this strategy, the system air flow rate is manipulated to maintain the zone energy balance. Feedback between the system and plant change the plant operating conditions to meet the load most efficiently so that the supply water temperatures and flow rates may also vary as the load changes. Thus, the supply air temperature is whatever value results from the current combination of air flow rate, plant operating status, and the heat and mass transfer processes occurring in the coil itself. Although many other control strategies could be defined, such as controlling to maintain a fixed water temperature change across the coil, the two described are most frequently found in practice.

3.4: Zone Update Method

The procedure used to integrate the IBLAST simulation was dictated to a large extent by the models describing the building, systems, and plants, and the connections linking these models to each another. An IBLAST building can consist of up to one hundred zones. A zone is not necessarily a single room but is usually defined as a region of the building or a collection of rooms subject to the same type of thermal control and having similar internal load profiles that, subsequently, can be grouped together. Zones can interact with each other thermally through adjacent surfaces and by intermixing of zone air. In BLAST an iterative procedure is used to solve for all the zone temperatures and surface conditions simultaneously. In IBLAST, the conditions in each zone are updated by finite differencing (Equation 3-9) which uses previously calculated values of the zone conditions. This means that IBLAST does not have to iterate to find a self consistent solution of the updated zone conditions. However, because heat transfer through each zone's surfaces and interzone mixing of air still occur, the new space temperatures must be computed at the same simulation time and on the same time step in all zones, even though conditions in one zone may be changing much more rapidly than conditions in the other zones. Two papers by Taylor et al (1990, 1991) have previously documented the method used to update the zone temperature at each time step.

In order to control the temperature in each zone, an air handling system and a sequence of zone temperature setpoints that define when the system heats and cools the zone must be specified. In the early versions of IBLAST, the system simulation was operated using an on/off control strategy. This meant that the system responded only to changes in the zone air temperature, much as a real system controlled by a thermostat would. However, a system sized poorly in relation to the conditioning demand usually caused instabilities in the simulation and gave inaccurate results. In reality, such instabilities, in which the system continually cycles on and off, are not uncommon in systems that are significantly oversized. The instability problem was countered by using the zone load to help determine the system response. This allowed the simulation to run at a much longer time step than would otherwise be possible. As in BLAST, the zone load is the heating or cooling required by the zone to offset the net cooling or heating, respectively, from external and internal influences other than the air conditioning system. In effect, a zone temperature based on the steady state energy balance was first computed. This temperature was used to determine the actual system response, that was then input to the transient zone energy balance to update the zone temperature for the time step. In practice, this procedure worked well and kept the simulation stable except in the case of grossly oversized systems.

3.4.1: Variable Time Step

Prior to the integration of the central plant simulation in IBLAST, a time step Δt for the zone temperature update of 0.25 hours (15 minutes) was found to give stable results without a large increase in computation time. The first step in integrating the plants was to implement the detailed coil models and coil control strategies without actually adding the plant models themselves. This meant that the user had to specify the coil water inlet temperature and the maximum coil inlet water flow rate to run the simulation. The real life analogy would be a chiller of very large, though not infinite, capacity. The coil capacity was controlled by adjusting the water flow rate but the effect of the plant on the chilled water temperature was eliminated. After implementation of this step, experience with the program showed that updating the zone temperatures on a fixed time step frequently resulted in instabilities unless a very short time step was used. However, as the time step got shorter the time required to execute the program got prohibitively high.

Clearly, an adaptive time step was required. This would shorten the time step to maintain stability of the zone air energy balance calculation when zone conditions were changing rapidly and expand it to speed computation when zone conditions were relatively unchanging. But, the adaptive time step could not be applied easily to the surface heat transfer calculations, even using interpolation methods to determine new temperature and flux histories. The problem of updating the zone temperature was resolved by using a two time step approach in which the zone air temperature is updated using an adaptive time step that ensures stability. In this two time level scheme, the contributions to the zone loads from the surfaces, infiltration, mixing, and user specified internal loads are updated at the default or user specified time step which is constant. A second variable time step is used to update the system response and the zone mean air temperature. This time step is selected by first calculating the system response and updating the zone temperature using the user specified time step Δt. The maximum temperature change experienced by a zone is then evaluated on a system by system basis. If the maximum zone temperature change is more than a preset maximum of 1°C the system and zone updates are performed using a new time step δt. This adaptive time step is initially set to $\Delta t/2$ and is successively halved until the maximum zone temperature change is less than the allowable maximum change. This approach can be justified because the internal loads, surface temperatures, infiltration and mixing vary on a different and longer time scale than the system response and the zone air temperature. The zone temperature update was made using Equation 3-67 for each adaptive time step, which is just a different form of Equation 3-9:

$$T_z^t = \frac{\left(\sum \dot{Q}_c + \sum_{i=1}^{\#surf} h_i A_i T_{si} + \sum_{j=1}^{\#zones} \dot{m}_j C_p T_{zj} + \dot{m}_{inf} C_p T_\infty\right)^{(t-\Delta t)} + \left(\frac{C_z}{\delta t} T_z + \dot{m}_{sys} C_p T_{supply}\right)^{(t-\delta t)}}{\left(\sum_{i=1}^{\#surf} h_i A_i + \sum_{j=1}^{\#zones} \dot{m}_j C_p + \dot{m}_{inf} C_p\right)^{(t-\Delta t)} + \left(\frac{C_z}{\delta t} + \dot{m}_{sys} C_p\right)^{(t-\delta t)}} \tag{3-67}$$

In Equation 3-67, Δt is the user specified time step and δt is the adaptive time step that is always less than or equal to Δt.

3.4.2: Simultaneous Solution of Plant/System Water Loop

Simultaneous solution of the system and plant operating parameters required that the temperature of the water entering the coils must be the same as the temperature leaving the chillers or boilers. In addition, the temperature of the return water from the coils must be equal to the chiller or boiler entering water temperature. In practice so long as the plant is not out of capacity the leaving water temperature from chillers and boilers is constant and equal to the design value. No iteration was required to match system and plant boundary conditions. However, if either the chiller or boiler plant was overloaded then the temperature of the water leaving the plant was not equal to the design value and the maximum output of the plant could change because of the off-design conditions. An iterative scheme using the secant method to predict successive updates to the plant leaving water conditions was therefore employed to solve for the water loop conditions with the plant operating at its maximum capacity.

The convergence criteria for the secant method are: the change in enthalpy of the fluid passing through all the systems served by the plant must be equal to the heating or cooling provided by the plant, the coil entering temperatures must equal the plant leaving temperature, and the system return temperature must equal the plant entering temperature. The secant method computes the change in the system coil load as a function of a finite deviation in the system entering temperature. The system entering temperature, at which the convergence criteria are satisfied, can then be extrapolated from this derivative approximation to the slope of the coil load curve as a function of supply temperature. The updated temperatures and coil loads are fed back into the plant simulation and the process repeated until the change in the plant leaving temperature becomes small.

Quasi-steady state conditions are assumed in the water loop between the plant and the coils because the thermal capacitance of the water loop is typically small compared to the cooling or heating capacity of the plant. Thus the corresponding time constant is short in comparison to the time step used for the simulation. This would not be such a good approximation for systems such as the water loop heat pump or for thermal storage systems that are close to exhausting their capacity, e.g. an ice storage tank where all the ice is melted and only sensible cooling capacity remains, because then the supply water temperature could change appreciably in a time step. Fortunately, the water loop heat pump model can be implemented as a combined system and plant model eliminating the need to iterate between the system and plant. In the case of the ice storage tank whose latent capacity is expended, any errors introduced are likely to be small since the tank sensible cooling capacity, when all the ice is melted, is small compared to the capacity of the fully charged tank.

Chapter 4: Thermal Storage System Simulation

4.1: Types of Thermal Storage Systems

The ice storage models developed for the BLAST program (Strand, 1992) were used as the starting point for a similar simulation in IBLAST. Although the models were operated on the one hour time step used by BLAST, the time step variable was parameterized making it very easy to change to other values as was necessary when installed in IBLAST. The BLAST ice storage models assume quasi steady conditions during a time step; i.e. they assumed that the temperatures and flow rates entering and leaving the ice storage unit are constant during the time step even though the amount of ice in the tank is increasing or decreasing. Assuming quasi-steady conditions in the ice tank is valid so long as the change in the amount of ice stored in the tank is small compared to the tank capacity. This simulation method fits well with the IBLAST program since in general the time step over which the systems and plants are simulated is much shorter than one hour. Because shorter time steps would result in smaller changes in the amount of ice stored or melted, the quasi steady assumption should be even more accurate for IBLAST than BLAST. The models simulate system performance by using of curve fits relating the performance of the tank to the cooling load. Given storage tank inlet brine conditions, temperature and flow rate, the model returns the outlet brine temperature and the rate of change of ice to water in the tank normalized to the tank capacity.

4.1.1: Direct Ice Storage

In BLAST, direct ice storage systems which may be simulated are ice-on-coil and ice harvester, as shown in Figure 4-1 and Figure 4-2 respectively. They are referred to as direct storage systems because the ice forms directly on the plates of the evaporator. In the ice-on-coil system the evaporator is a coiled tube inside a storage tank filled with water. During charging, ice grows radially on the surface of the coil. A second coil in the tank circulates brine, usually a mixture of water and ethylene glycol between the storage tank and the cooling coils. By contrast, the ice harvestor system forms ice on the evaporator plates until it reaches a certain thickness. The refrigeration cycle is then reversed causing the ice adjacent to the evaporator plates to melt and the remaining ice to fall into the storage tank. Again, there is a brine filled coil in the tank that rejects heat from the cooling coils to melt the ice in the tank .

Direct ice storage systems may be operated in 4 distinct modes: full storage, optimal control, compressor-aided, and demand limiting. In the full storage mode of operation, stored ice

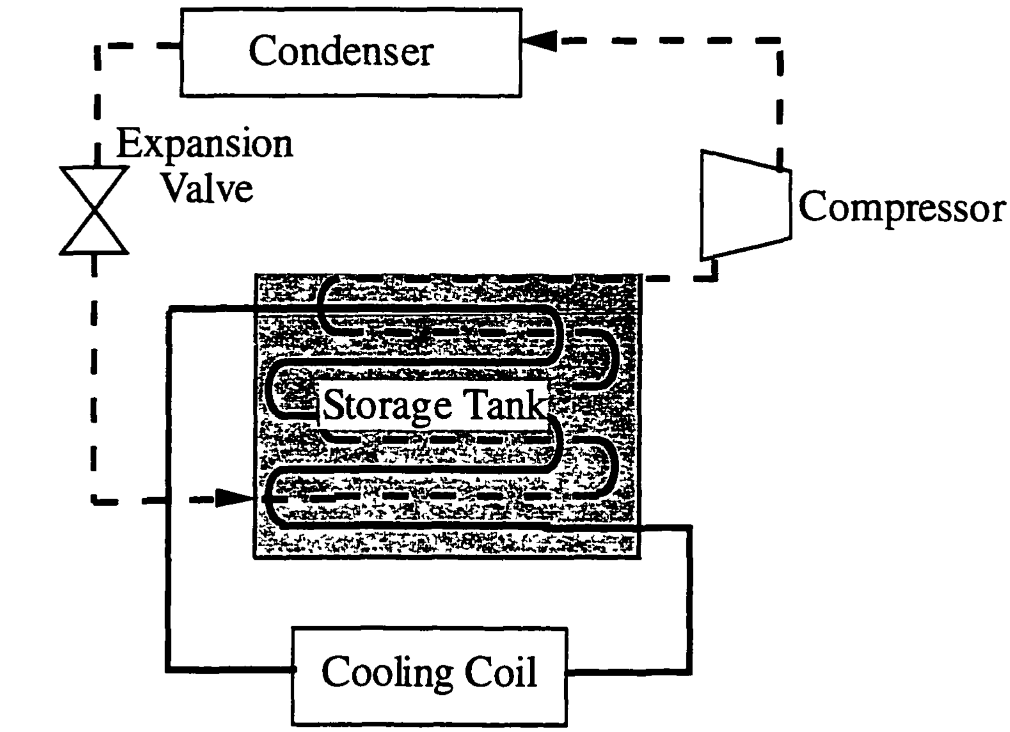


Figure 4-1: Simplified schematic of ice on coil storage system

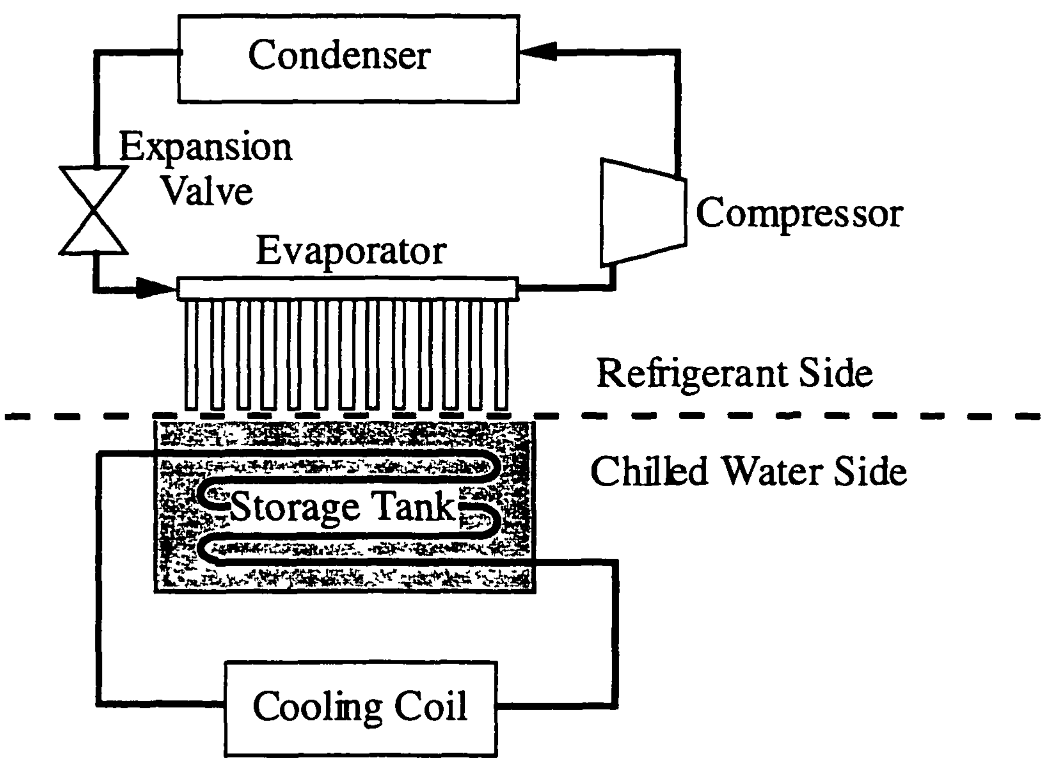


Figure 4-2: Simplified schematic of ice harvester direct storage system

is used to meet one hundred percent of the on-peak cooling loads. During the off-peak hours the chiller is used to meet system cooling loads and to recharge the storage system. In this mode of operation the chiller will attempt to charge the storage tank to full capacity. However, with the ice on coil storage system this is not always the best strategy because, as ice is built on the coil, lower and lower refrigerant temperatures are required to maintain a reasonably rapid rate of ice formation. This, in turn, lowers the coefficient of performance COP of the chiller plant and causes it to consume more energy. The optimal strategy for this type of ice storage system is to build only as much ice as will be used the following day because this allows the most efficient charging of the storage tank. In fact, this precisely describes how the optimal control strategy works in BLAST. Since BLAST already has the system loads when the ice storage plant is simulated, this was an easy option to implement; load prediction was not required. While the ice harvestor system does not suffer from increasingly lower COP as ice is formed, the fact that the refrigeration cycle must be reversed periodically to allow ice to be deposited in the storage tank means that it also benefits from the optimal control strategy. Making only enough ice is made to meet the next day's cooling load means that a minimum number of reverse cycles occur.

The compressor-aided and demand-limiting options are similar in that they use ice and the chiller plant simultaneously to meet cooling loads. The compressor-aided strategy uses ice up to a specified fraction of the cooling load with the remainder being met by the chiller. In contrast, the demand limiting strategy uses the chiller to meet cooling loads until the load exceeds a certain threshold. The fraction of the load above that preset value is met using ice.

4.1.2: Indirect Ice Storage

The BLAST indirect ice storage systems are the ice container and the ice tank. Indirect systems are so-called because a chiller is used to cool a secondary brine loop that circulates in the storage device where the ice is formed. The same brine flow may also be diverted through the coils when cooling loads exist and the chiller and stored ice may be used simultaneously. The ice container system, shown in Figure 4-3, circulates brine from the chiller to a tank filled with many water filled containers. The containers may be any shape, though spheres about 4 to 6 inches in diameter are a popular and space efficient design. When the system is charging, the brine is used to freeze the water in the ice containers. During discharge, ice is melted as warm brine from the cooling coils is passed over the ice containers. The ice tank system shown in Figure 4-4 has a brine filled spiral wound coil inside a tank filled with water. The tank is charged by circulating refrigerated brine from the chiller through the coil to freeze the water in the tank. When there is a cooling load on the plant, the circulation in the brine loop is reversed and the chilled brine is passed

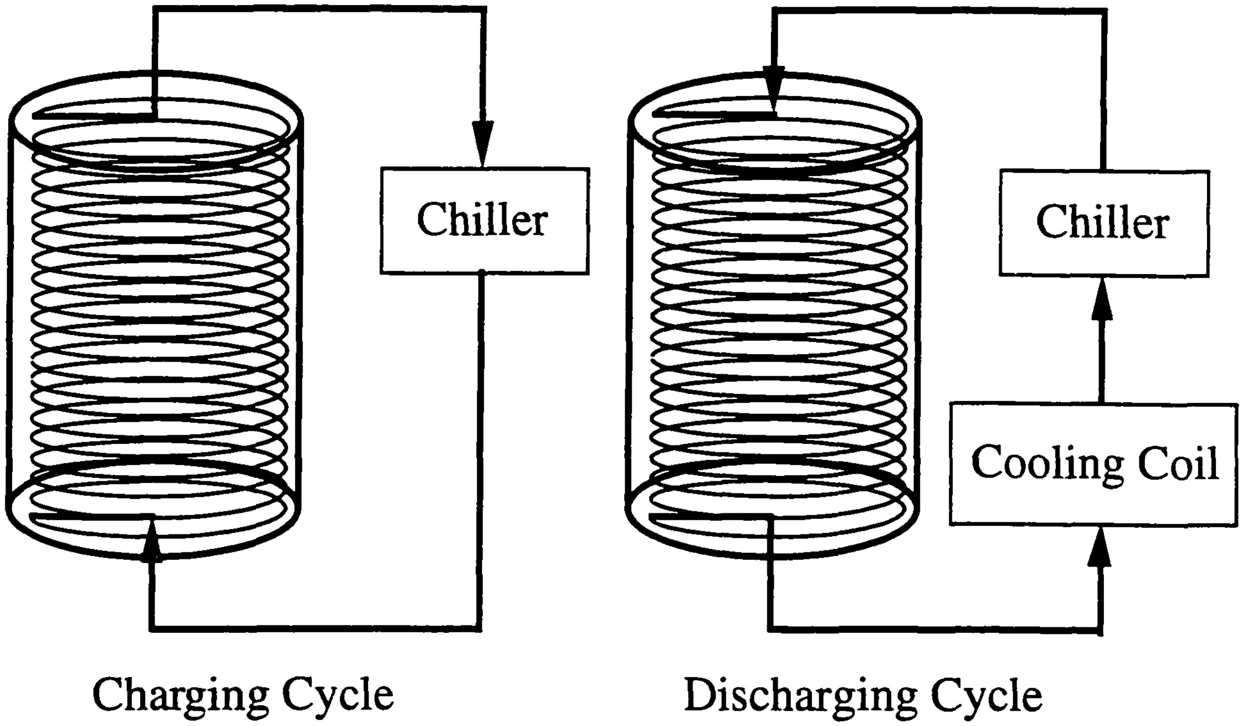


Figure 4-3: Simplified schematic of ice tank indirect storage system

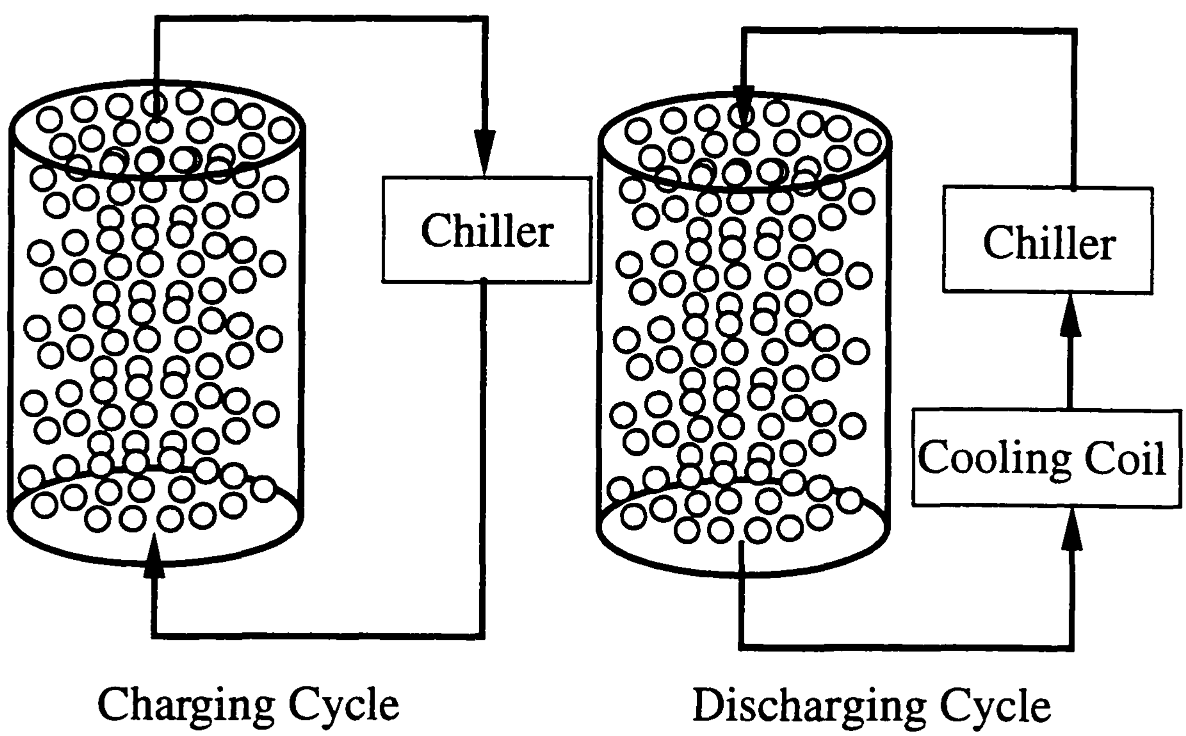


Figure 4-4: Simplified schematic of ice container ice storage system

through the cooling coils. Relatively warm brine returns from the coils, entering the tank and causing the ice to melt.

The indirect ice storage control strategies are: full storage, ice priority and chiller priority. Full storage is defined the same way for indirect storage as for direct storage; all on-peak loads are met using ice. The ice priority control strategy is similar to compressor aided control in direct storage systems; ice is used until the load reaches a specified threshold with additional load being met by the chiller. Likewise the chiller priority strategy is similar to demand limiting. The chiller is used to meet cooling loads up to a preset level then additional load is met with ice. However, one difference is that indirect storage systems do not suffer the same performance penalties for the overproduction of ice as direct storage systems. The chiller operates with a constant COP when charging the storage system regardless of how little or how much ice is in the tank. Therefore, the normal method of operation for indirect storage systems is to always charge them to full capacity.

4.2: Thermal Storage Model Implementation

One goal of this research was to optimize thermal storage systems by determining the instances when full storage or a combination of storage and mechanical cooling should be used. When a combination is appropriate, it is also necessary to know how to distribute the cooling load to each component of the system. Therefore, to demonstrate the ability of IBLAST to simulate and optimize thermal storage, the indirect ice storage systems were implemented in IBLAST. Although the so-called optimal control strategy applicable to direct storage systems is not used in practice with indirect storage, this strategy was implemented to ensure that the methods used would be completely compatible with direct storage systems.

The performance of the indirect ice storage models was simulated using curve fits that related the normalized outlet brine temperature, the normalized inlet brine temperature, and the normalized load on the storage tank $\dot{Q}^*$. Separate curve fits were required for each type of storage system and also to model the charging and discharging cycles for each type of storage device. The discharge cycle of the ice container system used a curve fit that related the normalized load on the system to the log mean temperature difference between the inlet and outlet brine temperatures:

$$\dot{Q}^* = f_1\left(C_1, P_c, \Delta T_{lm}^*\right) \tag{4-1}$$

In Equation 4-1, C_1 represents the coefficients of the polynomial curve fit and these are calculated from actual performance data. In the same equation, P_c is the ratio of the remaining storage unit capacity to its maximum capacity, and ΔT^{*}_{lm} is the log mean temperature difference that is calculated from:

$$\Delta T^{*}_{lm} = \frac{\dfrac{T_{brine,in} - T_{brine,out}}{\ln\left(\dfrac{T_{brine,in} - T_{freeze}}{T_{brine,out} - T_{freeze}}\right)}}{\Delta T_{nominal}} \tag{4-2}$$

The charging cycle of the ice container system was modeled using the same equation, but with a different set of coefficients that matched the actual performance of an ice container storage unit.

The simulation of the ice tank storage system followed a similar procedure. However for this system the curve fit of the performance data generates the normalized outlet temperature T^{*}_{o} according to a curve fit function of the form:

$$T^{*}_{o} = f_2\left(C_2, Pc, T^{*}_{i}, \dot{Q}^{*}\right) \tag{4-3}$$

where C_2 again represents the coefficients of a polynomial curve fit to the performance data. The normalized inlet and outlet temperatures required in this equation are calculated as shown in Equations 4-4 (a and b):

$$T^{*}_{o} = \frac{T_{o}}{T_{freeze}} \tag{4-4a}$$

$$T^{*}_{i} = \frac{T_{i}}{T_{freeze}} \tag{4-4b}$$

The charging cycle of the ice tank requires the use of a second polynomial curve fit that returns the log mean temperature difference across the tank when the percentage of tank charged, normalized load, brine mass flow rate $\dot{m}$, and coefficients C_3 are specified as shown in Equation 4-5:

$$\Delta T^{*}_{lm} = f_3\left(C_3, P_c, \dot{Q}^{*}, \dot{m}^{*}\right) \tag{4-5}$$

Additional details on the development of the BLAST ice storage models and methods for determining the sets of polynomial coefficients from performance data can be found in Strand (1992) and the BLAST Users Manual (1993).

Once the ice storage models had been inserted into the IBLAST code the correct links were made to preserve the system-plant energy balance. This was accomplished by determining the correct inlet and outlet temperatures and brine flow rate to pass to the ice storage system. In BLAST these values are determined from user inputs. The temperatures used depended on the location of the chiller with respect to the storage tank. Three possible locations exist: upstream, downstream, and parallel. During the discharge cycle with an upstream chiller, brine enters the chiller at the system return temperature. If the chiller is operating, the brine may be cooled before exiting the chiller and passing through the storage tank. The target temperature for the brine leaving the ice storage unit is the desired system entering temperature. Since the chiller and the storage unit are in series, the mass flow rate is the same in both. The downstream chiller option reverses the order of chiller and storage unit so that brine enters the storage unit at the system return temperature. The ice storage leaving temperature is equal to the chiller inlet temperature, and the temperature of the brine at the chiller outlet must then be equal to the system inlet temperature. Finally, when a parallel chiller is specified, the temperature of brine leaving the system must equal the chiller and storage unit entering temperatures; and the blended temperature of the brine from the chiller and storage outlet streams must equal the system inlet temperature.

During the charging cycle the location of the chiller with respect to the ice storage unit is unimportant since the coldest possible brine temperature is desired for making ice. Therefore, the storage unit always receives brine at the chiller outlet temperature. The chiller inlet temperature is thus given by:

$$T_{chiller,in} = T_{chiller,out} + \frac{\dot{Q}_{absorbed,ice} + \dot{Q}_{absorbed,system}}{\dot{m}_{chiller} C_{brine}} \tag{4-6}$$

where: $\dot{Q}_{absorbed,ice}$ is the rate of energy absorption by the brine stream from the ice storage unit, and $\dot{Q}_{absorbed,system}$ is the rate of energy absorption from the cooling coils.

4.3: Requirements for Optimal Control

The principal requirement for optimizing the use of thermal storage systems is to control the system so that it produces just enough ice during the off-peak hours to meet the following day's cooling loads. This is especially important for ice-on-coil systems since the production of ice reduces the performance of the refrigeration plant. However, since storage tanks are not perfectly insulated, all ice storage systems experience some heat leakage into the storage tank from the environment. Additional ice must be produced to compensate for this heat leakage. Therefore, the best control strategy is to make the ice in the off-peak hours closest to the time when it will be used. This minimizes the amount of ice melted due to heat leakage through the walls of the storage tank. Figure 4-5 shows the expected variation in tank capacity for an optimally controlled storage

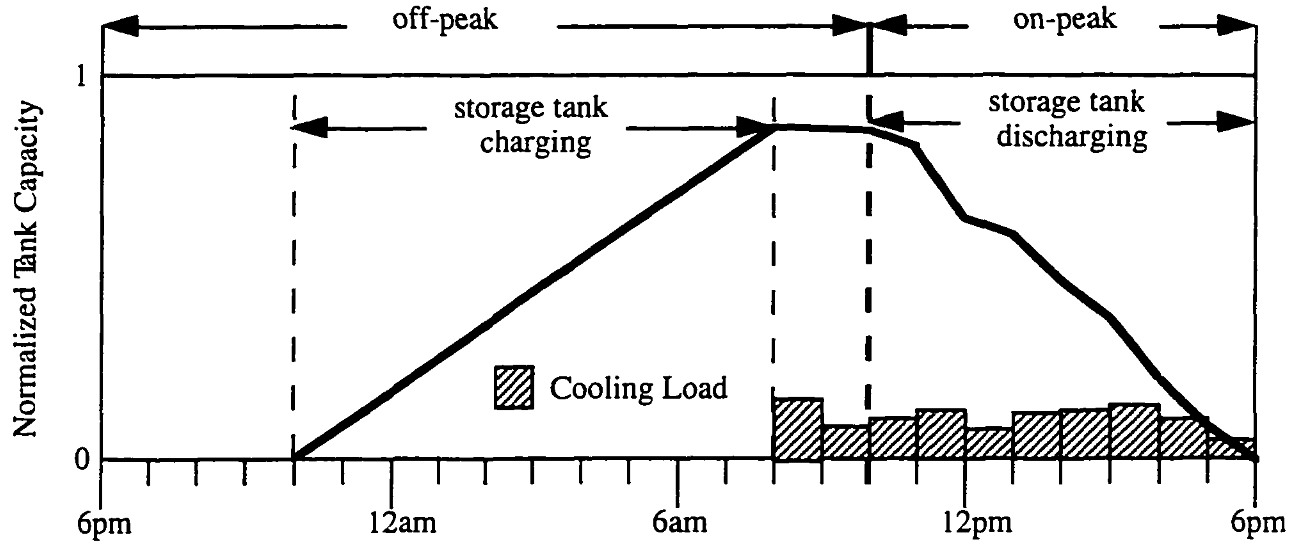


Figure 4-5: Charging and discharging of an optimally scheduled storage tank

system. The off-peak hours are from 6pm until 8am the following day. The hours from 8am to 10am are off-peak based on the utility rate and are not counted for the purpose of determining demand charges. In this example, the chiller operates to meet cooling loads during these hours instead of using ice from storage. The decrease in tank capacity between 8am and 10am is due to heat leakage into the storage tank. From 10am to 6pm, all cooling loads are met using stored ice; this is an example of a storage system using the full storage control strategy. The cooling load is also the fraction of tank capacity used during the indicated hour so that the total amount of cooling required by the system equals the initial tank capacity.

Figure 4-6 represents a storage system that is not scheduled optimally though it does not overproduce ice during the charging cycle. That is, at 6pm the storage tank has zero cooling capacity left. Clearly, this strategy consumes more energy than the one shown in Figure 4-5 since additional charging is required to "top-off" the storage tank due to the loss of ice that occurs between 4am and 8am because of heat leakage into the tank. The discharge process is, of course, identical to the previous case.

From Figure 4-7, it is clear that an optimally scheduled storage tank that is always charged to full capacity will consume more energy than a system only making enough ice to meet the next day's load. Again, additional chiller use is required to offset the loss of storage due to heat leakage through the tank walls. However, as has been previously mentioned the efficiency of the charging process can vary as a function of the tank storage fraction, the ratio of the actual cooling capacity of the ice in the tank to the full storage capacity, especially in ice-on-coil ice storage systems. This

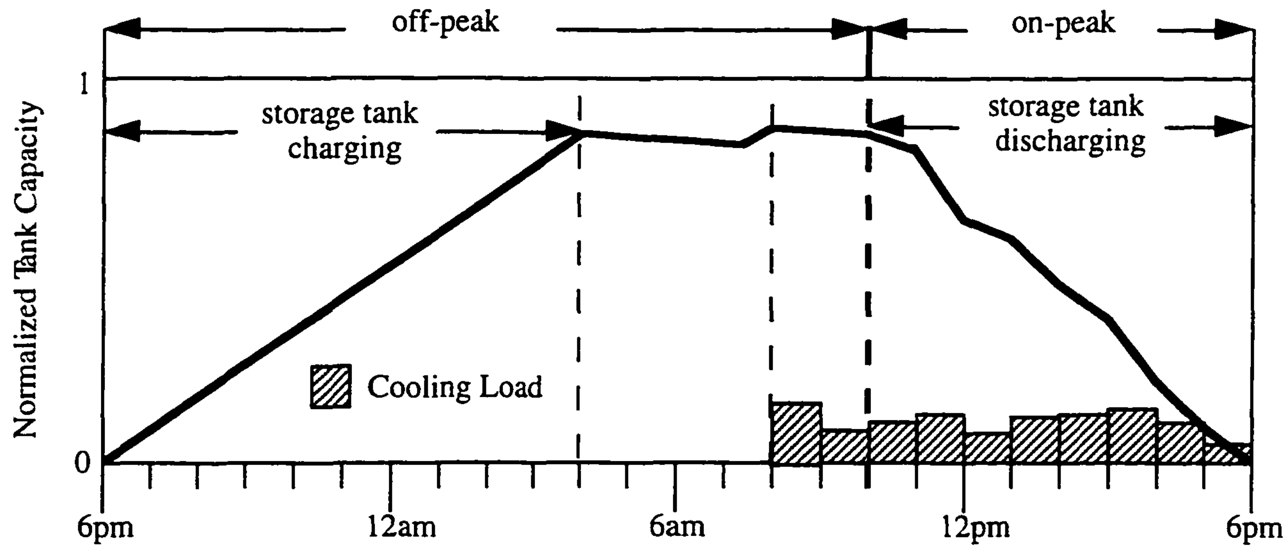


Figure 4-6: Charging and discharging of a non-optimally scheduled storage tank

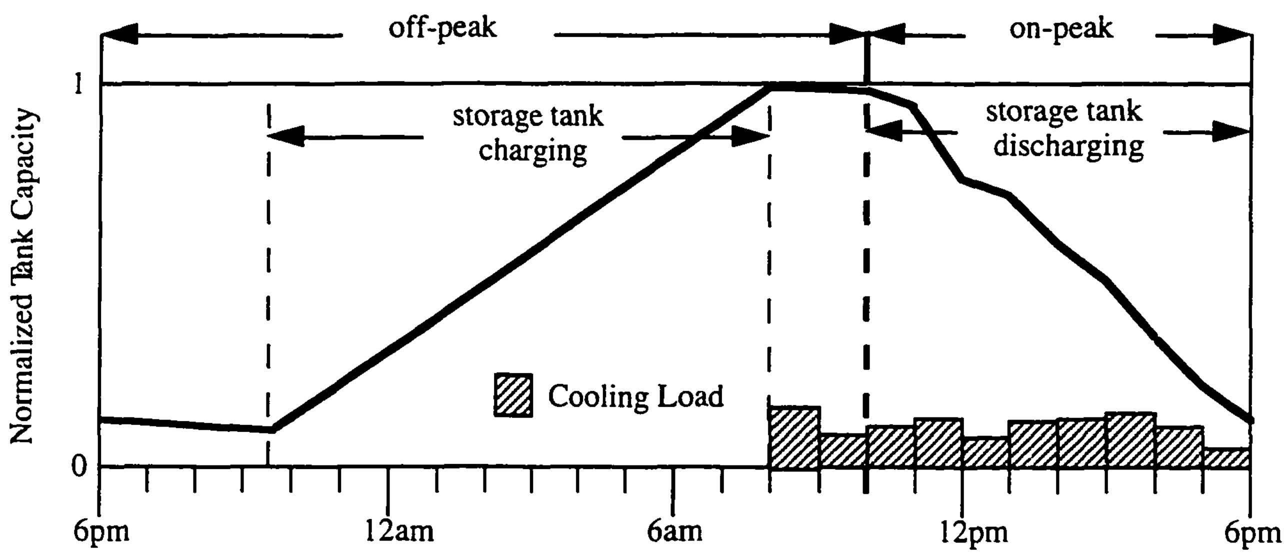


Figure 4-7: Operation of an optimally scheduled storage tank charged to full capacity each cycle

effect is not shown on these figures and would cause even more energy consumption to occur than in the optimal case of Figure 4-5.

Finally, Figure 4-8 is the non-optimally scheduled system of Figure 4-6 that is charged to capacity each cycle. Unless the storage system performance is a function of the tank storage fraction, fully charging the tank each cycle does not incur any additional energy loss penalty due to leakage.

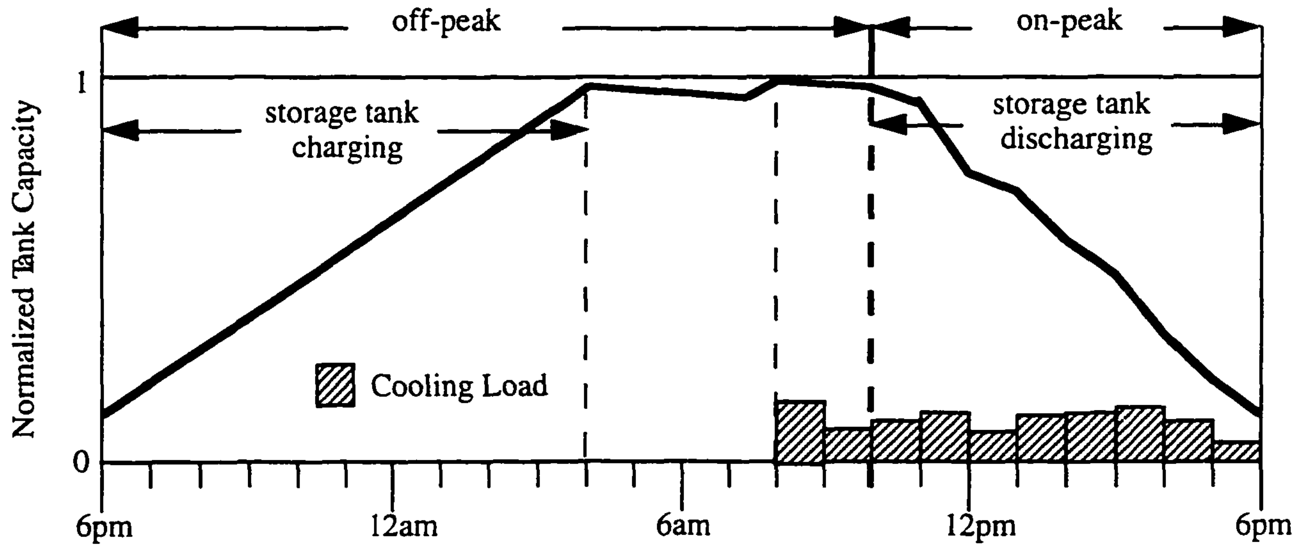


Figure 4-8: Operation of a non-optimally scheduled storage tank charged to full capacity each cycle

Figures 4-5 through 4-8 were used to describe how the performance of an ice storage system might be affected by variations in when charging of the ice tank begins and whether the tank is charged to its full capacity or to a level that is just enough to meet the next day's cooling load. Ice alone was assumed to meet the on-peak cooling loads leading to a second consideration for the optimization of thermal storage: what fraction of the on-peak load should be provided by storage? Obviously the answer is related to the difference between the cost of a unit of cooling when provided by storage and the cost of the same unit of cooling provided by direct cooling. Even taking this into account, the resulting solution will have all the cooling load being met from storage or from direct cooling depending on which is cheaper per unit of cooling. But, the on-peak hours of the day are usually also hours when utilities keep track of a customer's peak energy consumption for the purposes of assessing demand charges. The demand charge structure is what eventually leads to an optimal solution that uses a combination of direct cooling and storage. However, optimizing the scheduling of storage tank charging is unaffected by what fraction of the cooling load is met by using storage or direct cooling. It is always optimal to produce only enough ice to meet the next day's ice demand and have the storage tank fully discharged at the end of the on-peak period, i.e. 6pm in the examples.

Chapter 5: Optimal Building Control

5.1: Overall Criteria for Optimization of Building Systems

Optimization of a building HVAC system requires the minimization of the life-cycle cost of that system. Contributions to life cycle cost that building designers can manipulate include: initial costs to purchase and install the equipment, scheduled and unexpected maintenance costs required by the equipment over its design life span, decommissioning costs, as well as system operating costs. However, many of the decisions that must be made when selecting system and plant components for a new building or retrofit project depend on such intangibles as: the building's construction materials and its intended use. Such variables are difficult to quantify in a meaningful way and in most cases the system designer must rely on careful reasoning based on experience and judgment to select appropriate equipment.

Once the type of equipment to be installed in the building has been selected, operation of that system to minimize its energy cost becomes a quantifiable problem. The results of this optimization may subsequently be combined with the initial investment and other costs in a life-cycle cost calculation to validate the criteria used to make the original equipment selection decisions. One objective of this research is to determine the minimum operating cost of a selected set of equipment in conjunction with a building. This is accomplished by considering two separate optimization problems: minimizing the energy consumption of the operating plant equipment at each instant of the simulation, and minimizing the total cost of the energy consumed by the system over a daily cycle. The distinction between these two problems will become clear in the following discussion.

5.1.1: Energy Consumption

When the cost of energy is constant throughout the simulation period, minimizing the total energy consumption of the system will also minimize energy cost. However, a common practice among electric utilities is to manipulate the cost of power over daily and annual cycles to encourage electricity consumers to smooth out hourly variations in their power consumption. This practice has the effect of leveling the load on the generating plant so that a large excess of capacity need not be maintained to meet short duration peak loads. Notwithstanding the effects of non constant energy costs, the optimization of instantaneous energy consumption by the building's system and plant remains important to the cost minimization problem.

Most large buildings with conventional heating and cooling equipment, e.g. chillers and boilers, normally have several small units of each type of equipment connected in parallel rather than one large unit of equivalent capacity. The rationale behind this is obvious; using a number of smaller sized units allows the operating capacity of the plant to be adjusted. Variable plant capacity means that when the heating or cooling load is less than the maximum plant capacity, equipment may be turned on or off so that the equipment can run closer to its optimum part load ratio than if there were only one large chiller or boiler sized to meet the entire peak load. The associated optimization problem is to assign a fraction of the plant load to each of the plant components so that energy consumption is minimized. Depending on the size of the load in relation to the capacity of the plant, at any given time one or more of the chillers or boilers need not be assigned any of the load.

The BLAST program incorporates a simplified optimization scheme involving some heuristic methods to assign load to each plant component. The BLAST scheme assigns load to the chiller plant using a fixed hierarchy depending on the type of energy used to power the chiller, for example: electricity, diesel fuel, steam, etc. Furthermore the BLAST method assumes that all equipment of the same type has the same part load operating characteristics. This simplifies the optimization procedure because the part load ratio Y of each operating plant component must be the same and equal to:

$$Y = \frac{\dot{Q}_{load}}{\dot{Q}_{oper\,cap}} = \frac{\dot{Q}_{load}}{\sum_{i=1}^{\#\,oper\,equip} \dot{Q}_{cap,i}} = \frac{\dot{Q}_{load,i}}{\dot{Q}_{cap,i}} \qquad (5\text{-}1)$$

where: $\dot{Q}_{load}$ is the total load on the plant, $\dot{Q}_{cap,i}$ is the maximum operating capacity of the ith plant component that equals $\dot{Q}_{oper\,cap}$ when summed over all the operating equipment of that type i.e. $\sum_{i=1}^{\#\,oper\,equip} \dot{Q}_{cap,i}$, and $\dot{Q}_{load,i}$ is the load assigned to the ith plant component. This is easy to simulate, but is not representative of real systems in which the equipment part load performance curves usually vary considerably with the capacity of the equipment. On the other hand, the optimization scheme developed for IBLAST does not require any assumptions about the part load performance of identical types of plant equipment and the fraction of the load assigned to each operating plant component need not be the same.

5.1.2: Energy Cost

Once the optimal load distribution among the operating plant components is found for an instant of time, the next step is to minimize the energy cost over an interval of time. This requires not only that the equipment combination change to minimize total energy consumption, but also that the peak energy consumption be factored into the problem when demand charges are in effect.

As an example, consider a cooling plant with two identical chillers. During the morning hours the load on the plant is relatively low and one chiller can easily meet the cooling load. In the afternoon, peak rates and demand charges go into effect so that turning on the second chiller during this period to meet loads greater than the capacity of a single chiller may not be advisable because of start up transients. Depending on the rate structure, the optimal schedule may require turning the second chiller on before the on-peak period begins, or using only one chiller if its capacity will not be exceeded.

The example shows that the optimization problem, when demand charges are being considered, is one of scheduling the available equipment so that it meets the loads as efficiently as possible at each instant of time, while avoiding spikes in energy consumption due to start up transients. Consideration of demand charges in selecting an optimal path through all the possible equipment combinations complicates the problem substantially as discussed by Olson (1987, 1990, 1993, and 1994). When demand charges are not in effect, the optimized solution merely requires selection of an equipment combination that consumes the least energy at each time step.

5.2: Optimal Load Distribution

The solution to the optimal load distribution problem required finding the load on each of the operating plant components to minimize the power consumption at the instant the optimization was performed. The problem first required formulation of the governing equations for the heating and cooling plant. From these equations, several different types of variables were defined according to their function: control variables, that are adjusted by the optimizer to improve plant performance; parameters, constant values obtained from input; and state variables, that describe other aspects of plant performance but are functionally dependent on the controls and the input parameters. This procedure was carried out for the heating and cooling plant separately, however the objective function contained both heating and cooling plant energy consumption information because interactions between the two types of equipment are common. Next the constraints on the control and state variables necessary to ensure physically reasonable solutions were defined. Finally, the objective function, the quantity being minimized, was formulated.

5.2.1: System-Plant Energy Balance

In this section, the variables that define the performance of the heating and cooling plant components and those system variables applicable to the coils are defined. Furthermore, the governing equations that describe system-plant energy and mass balances are derived. These equations enable a self consistent solution to be obtained among the system and plant components. This solution was used to calculate the energy consumption of all the components of each system and plant in terms of the control and state variables, and the fixed parameters. In addition, all possible interactions between the system and plant components could be accounted for.

5.2.1.1 Heating Plant Governing Equations

A schematic of a simplified heating plant that has several boilers in parallel is shown in Figure 5.1 to illustrate the scope of the problem and the variables required. The boilers need not be of the same type, as defined by the energy they consume. In series with each boiler is a pump to circulate hot water through the system. However, there could also be just one pump in the water loop on either the feedwater or system supply side of the boilers. In order to reduce the load on the heating plant, waste heat from other sources may preheat the feedwater before it enters the boilers. This is denoted by the $\dot{Q}_{waste}$ term in the Figure 5-1. Finally, there is a bypass line that can send hot water directly from the supply side of the system to the boiler feedwater side. Hot water is diverted through the bypass when the load on the plant is less than the operating capacity of the boilers or the hot water flow rate required by the system is less than the minimum that can be provided by the pumps. Finally, the "Control Inputs" block defines the plant control variables that the optimizer adjusts to modify the performance of the boilers and pumps.

In Figure 5-1, the main boiler feedwater stream, which is also the return line from the heating coils, splits to allow control over the flow of water to each boiler. Conversely, the hot water streams exiting each boiler are collected into one main system supply line. However, some of this supply water may be diverted through the bypass. The water mass flow rate in each branch is calculated by going around the heating system in the direction of the water flow and applying conservation of mass at each flow intersection. Where the system bypass line intersects the main boiler feedwater supply, labeled A on the figure, the sum of the bypass water flow rate and the system return water flow rate equals the flow rate of water going to the boilers:

$$\dot{m}_{hwr} = \dot{m}_{hpr} + \dot{m}_{hwb} \tag{5-2}$$

The main boiler supply water flow splits at point C to provide feedwater to each boiler. The fraction of the total flow going to each boiler is $X_{bo,i}$. Summation of the individual boiler water flow rates yields both the return flow to the boilers $\dot{m}_{hwr}$ and the supply flow from the boilers to the system $\dot{m}_{hws}$ as given in Equation 5-3

$$\dot{m}_{hwr} = \dot{m}_{hws} = \sum_{i=1}^{N_{boilers}} X_{bo,i}\dot{m}_{hwr} \tag{5-3}$$

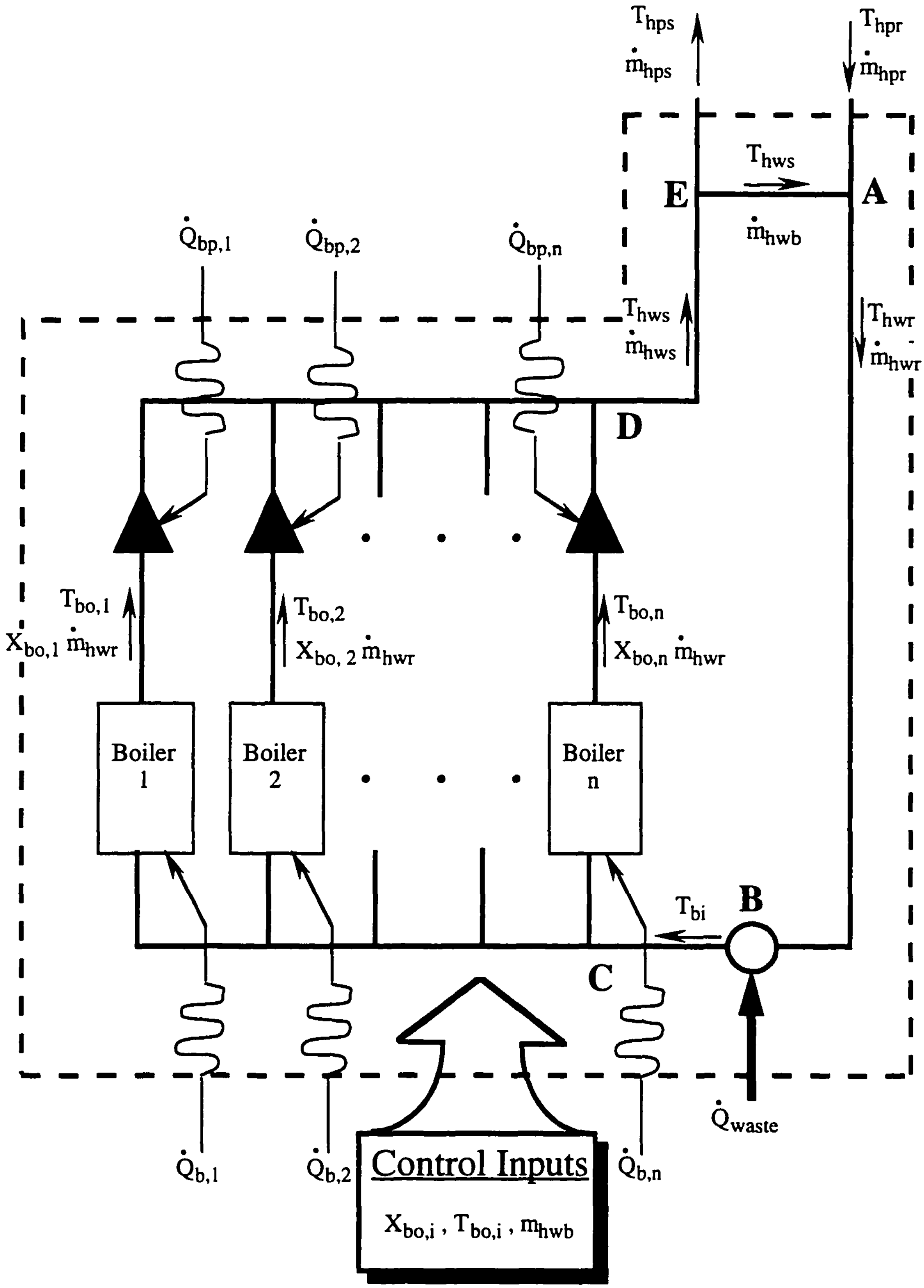


Figure 5-1: Schematic of heating plant showing boilers and possible pump locations

The actual hot water flow rate to the heating coils $\dot{m}_{hps}$ is determined by performing a mass balance at the intersection of the boiler hot water supply line with the bypass line, point E, as given in Equation 5-4:

$$\dot{m}_{hws} = \dot{m}_{hps} + \dot{m}_{hwb} \tag{5-4}$$

Next conservation of energy at flow intersections must be used to determine the temperature of the water on each side of the intersection. At point A, this yields the following expression:

$$T_{hwr}\dot{m}_{hwr} = T_{hwb}\dot{m}_{hwb} + T_{hpr}\dot{m}_{hpr} \tag{5-5}$$

Note that the specific heat is constant for all the water streams and is therefore, dropped from the equation. Conservation of energy at point B, where waste heat is introduced, is given by Equation 5-6:

$$T_{bi}C_{p,H_2O}\dot{m}_{hwr} = T_{hwr}C_{p,H_2O}\dot{m}_{hwr} + \dot{Q}_{waste} \tag{5-6}$$

Since there is no heat gain or loss at point C, the temperature of the supply water to each individual boiler is equal to T_{bi}. The water outlet temperature from a specific boiler is determined by the amount of heat added by the boiler to the supply water flow. Ideally, boilers attempt to control the water outlet temperature to a specified value $T_{bo,i}$ and deviation from this only occurs when the boiler runs out of capacity. Therefore, in the simulation, the boiler is initially assumed to be able to produce water at the desired temperature and the boiler energy consumption is given by Equation 5-7:

$$\dot{Q}_{b,i} = f_{q,i}(T_{bi}, X_{bo_m,i}\dot{m}_{hwr}, Y_i, T_{bo,i}) \tag{5-7}$$

where Y_i represents input parameters describing boiler performance. In the event that the maximum boiler capacity is reached, the boiler outlet temperature is calculated from an energy balance across the boiler using the inlet flow conditions and the maximum boiler heating capacity based on those conditions:

$$X_{bo,i}\dot{m}_{hwr}C_{p,H_2O}T_{bo,i} = X_{bo,i}\dot{m}_{hwr}C_{p,H_2O}T_{bi} + \dot{Q}_{b,max,i}\left(T_{bi}, X_{bo,i}\dot{m}_{hwr}, Y_i, T_{bo,i}\right) \tag{5-8}$$

Since the boiler capacity can be a function of the outlet water temperature, iteration is typically required to solve Equation 5-8. Next, conservation of energy at the convergence of the individual boiler outlet streams, point D, is considered to find the temperature of the water being supplied to the system:

$$T_{hws}\dot{m}_{hws} = \sum_{i=1}^{N_{boilers}} X_{bo_{m},i} T_{bo,i}\dot{m}_{hwr} \quad (5\text{-}9)$$

Again the specific heat is dropped in Equation 5-9. Finally, the energy balance at point E gives the result that the temperature of the water in the coil supply line and the bypass line is equal to the boiler outlet temperature T_{hws}:

$$T_{hws} = T_{hwb} = T_{hps} \quad (5\text{-}10)$$

The total energy consumption of the heating plant can be computed once the load on each boiler and the flow rate through each pump is known. The energy consumption of a pump can generally be given as a function of: the flow rate through the pump; the parameters defining the performance characteristics of the pump $Y_{bp,i}$; and the pump speed $S_{bp,i}$, in the case of a variable speed pump. This functional dependence is given by Equation 5-11;

$$\dot{Q}_{bp,i} = f_{q_{bp},i}(X_{bo_{m},i}\dot{m}_{hwr}, Y_{bp,i}, S_{bp,i}) \quad (5\text{-}11)$$

In addition, there is usually parasitic electric consumption, incurred whenever the boiler is turned on, that is used to run the boiler's control systems. Parasitic electric is assumed to be a constant value independent of the load. Thus, total energy consumption of the heating plant may be calculated by summing the boiler, pump and parasitic electric consumption for each boiler. However, this is not always appropriate because different types of energy, with different costs associated with them, may be involved.

5.2.1.2: Cooling Plant Governing Equations

Figure 5-2 shows a simplified schematic of a typical chilled water plant with several chillers and a thermal storage system arranged in parallel. However, in practice, a building with thermal storage would probably not have supplemental chillers other than those required to charge the ice or chilled water storage tanks. The chilled water plant is similar to the heating plant in that: the chillers need not be of the same type nor of the same size and there could be pumps for each chiller or a single pump in either the chilled water supply or return line. However, unlike the heating plant which can utilize waste heat, there is no "waste cooling" in a chilling plant Throughout this discussion, the fluid flowing through the chillers is referred to as water, but, when an ice storage system is specified, the cooling fluid is referred to as brine and is usually a mixture of water and ethylene glycol.

As with the heating plant, the first step in solving for the flow parameters in this system is to perform a mass balance at each of the indicated locations: A, B, C, and D. At the point where the return stream from the coils mixes with the bypass water, the total flow rate to the chilled water plant $\dot{m}_{cwr}$ is found from Equation 5-12:

$$\dot{m}_{cwr} = \dot{m}_{cpr} + \dot{m}_{cwb} \tag{5-12}$$

Applying conservation of mass at points B and C gives the result that the sum of the water flow rates through all the chillers and the water flowing through the thermal storage unit equals the total return water flow rate at point B $\dot{m}_{cwr}$ and also equals the total chilled water supply flow rate as measured at point C $\dot{m}_{cws}$. Equation 5-13 expresses this result:

$$\dot{m}_{cwr} = \dot{m}_{cws} = \sum_{i=1}^{N_{chillers}} X_{co,i}\dot{m}_{cwr} + X_{tso}\dot{m}_{cwr} \tag{5-13}$$

Finally, Equation 5-14 represents conservation of mass at point D and allows the chilled water flow rate to the coils $\dot{m}_{cps}$ to be calculated:

$$\dot{m}_{cws} = \dot{m}_{cps} + \dot{m}_{cwb} \tag{5-14}$$

Again following the method used for the heating plant, conservation of energy is imposed around the cooling plant water loop at the indicated points. Equation 5-15 describes conservation of energy at point A and allows the chiller entering temperature T_{cwr} to be determined from the temperatures and flow rates in the bypass and coil return lines.

$$T_{cwr}\dot{m}_{cwr} = T_{cwb}\dot{m}_{cwb} + T_{cpr}\dot{m}_{cpr} \tag{5-15}$$

The chilled water outlet temperatures from the chillers or thermal storage unit are also assumed to be fixed parameters unless the chiller capacity is exceeded. The chiller energy consumption based on the inlet and nominal outlet conditions is therefore given by Equation 5-16:

$$\dot{Q}_{ch,i} = f_{Q_{ch},i}(T_{cwr}, X_{co_m,i}\dot{m}_{cwr}, Z_i, T_{co,i}) \tag{5-16}$$

where Z_i represents the input chiller performance parameters. When chiller capacity is exceeded, the chiller outlet temperature is calculated from the inlet conditions and the full load chiller capacity at those inlet conditions, as in Equation 5-17:

$$X_{co,i}\dot{m}_{cwr}C_{p,H_2O}T_{co,i} = X_{co,i}\dot{m}_{cwr}C_{p,H_2O}T_{cwr} + \dot{Q}_{ch,max,i}\left(T_{cwr}, X_{co,i}\dot{m}_{cwr}, Z_i, T_{co,i}\right) \tag{5-17}$$

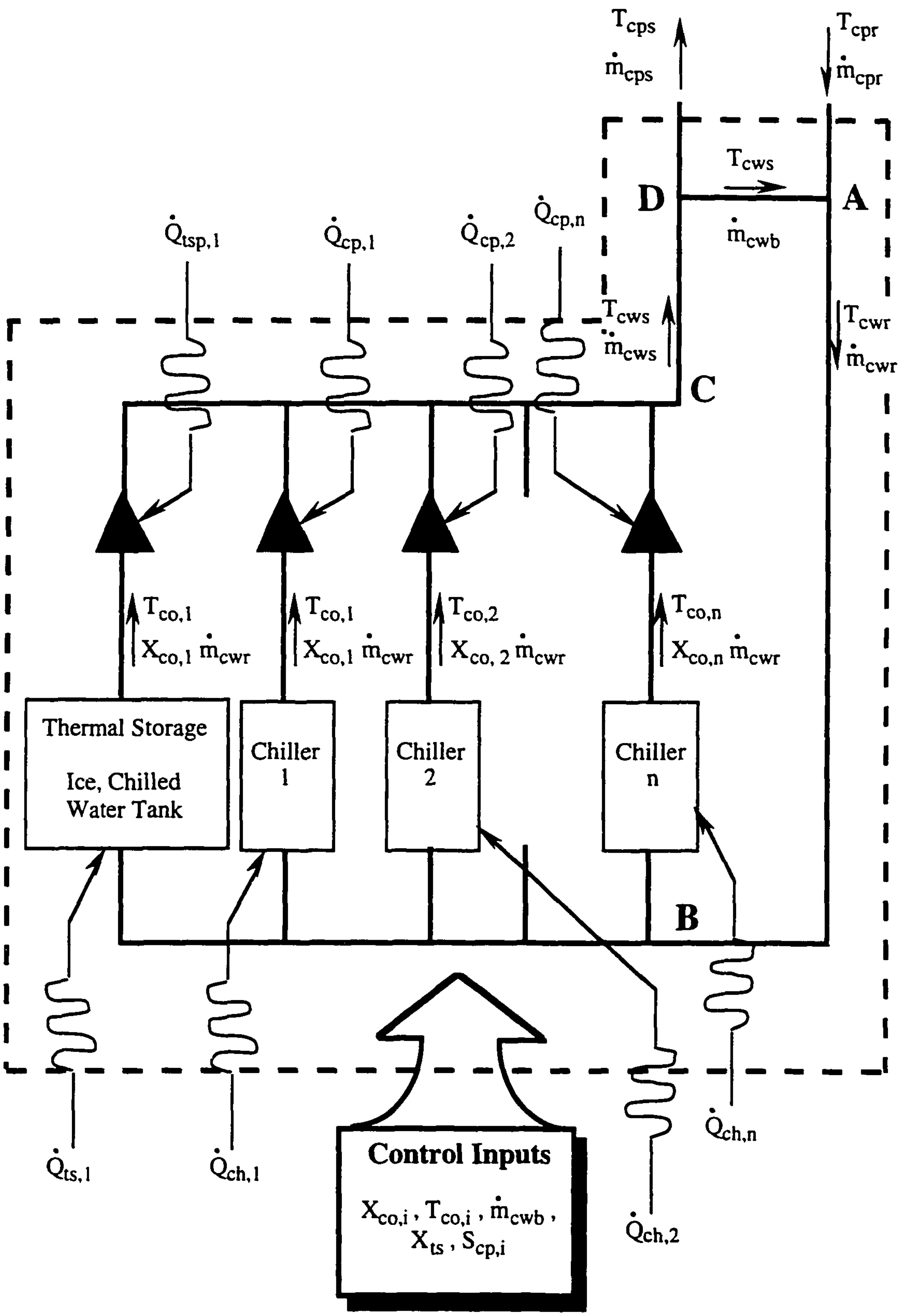


Figure 5-2: Schematic of chilled water plant showing chillers, thermal storage, and possible pump locations

Equation 5-16 requires modification to describe a thermal storage system and this is shown in Equation 5-18. This equation allows the rate of consumption of stored ice or chilled water in the storage tank in terms of the load and the performance of the storage system to be calculated:

$$\dot{Q}_{ts,lost} = f_{Q_{ts}}(T_{cwr}, X_{tso}\dot{m}_{cwr}, Z_{ts}, T_{tso}, \frac{Q_{ts,rem}}{Q_{ts,max}}, X_{shave}\dot{Q}_{tsc,max}) \tag{5-18}$$

From Equation 5-18, gives the amount of storage lost from the tank as a function of the input performance parameters Z_{ts}, the normalized fraction of storage capacity remaining in the tank $\frac{Q_{ts,rem}}{Q_{ts,max}}$, and the fraction of the total load on the storage system being assigned to the storage tank $X_{shave}\dot{Q}_{tsc,max}$. Clearly, if only part of the load is met using storage the rest must come from the storage system chiller. The energy consumption of this chiller may be calculated from an equation of the same form as Equation 5-16. Note that the capacity lost from the storage tank was generated by energy consumption at a different time of the day. Energy consumption due to the chiller and thermal storage system pumps is given by Equations 5-19 and 5-20 respectively that have the same form as for the boiler pumps but with different performance parameters $Z_{cp,i}$ and Z_{tsp}:

$$\dot{Q}_{cp,i} = f_{Q_{cp},i}(X_{co,i}\dot{m}_{cwr}, Z_{cp,i}, S_{cp,i}) \tag{5-19}$$

$$\dot{Q}_{tsp,i} = f_{Q_{tsp},i}(X_{tso,i}\dot{m}_{cwr}, Z_{tsp}, S_{tsp}) \tag{5-20}$$

Applying conservation of energy at point C, where the chiller and thermal storage outlet streams mix allows the temperature of the supply water to the coils to be calculated from Equation 5-21:

$$T_{cws}\dot{m}_{cws} = \sum_{i=1}^{N_{chillers}} X_{co,i}T_{co,i}\dot{m}_{cwr} + X_{tso}T_{tso}\dot{m}_{cwr} \tag{5-21}$$

Finally, at point D, the water temperature in the bypass line and cooling coil supply line is equal to the temperature of the water coming from the chilled water plant:

$$T_{cws} = T_{cwb} = T_{cps} \tag{5-22}$$

Completing the analogy with the heating plant, parasitic electric consumption must also be added into the total energy consumption of the plant in order to develop an accurate picture of how the plant is operating.

5.2.1.3: System-Plant Interface Equations

In Sections 5.2.1.1 and 5.2.1.2, the equations that govern changes in the water loop variables in the plant simulation were discussed. However, these equations are not a closed set. The other half of the water loop, the system side, must also be described to obtain a mathematically consistent set of equations. In actuality, the division made here between the system and plant is quite artificial. The primary benefits gained by making this distinction are in programming, variable record keeping, and simulation input.

Figure 5-3 is representative of the chilled water flow path in a typical air conditioning system with one cooling coil per system. Chilled water enters at T_{cps} and is distributed via parallel circuits to each of the cooling coils. The chilled water absorbs heat from the air entering each cooling coil and exits the coil at T_{wo}. The outlet streams from each coil are collected into one main line that returns the water to the plant. Since no mass is gained or lost in the coils the return mass flow rate to the plant must be equal to the supply mass flow rate from the plant. Proceeding as before, the first consideration in describing this system is to perform a mass balance on the supply and return sides of the system. This gives the result that the sum of the individual cooling coil mass flow rates equals the total flow rate to and from the plant as shown in Equation 5-23:

$$\dot{m}_{cps} = \sum_{i=1}^{N_{systems}} \dot{m}_{w,cc,i} = \dot{m}_{cpr} \tag{5-23}$$

Equation 5-24 simply states that the water supply temperature is equal to the temperature of the main water stream from the chiller plant:

$$T^{in}_{w,cc,i} = T_{cps} \tag{5-24}$$

Performing an energy balance on the return water streams from all the cooling coils enables calculation of the plant return water temperature T_{cpr} from Equation 5-25:

$$\sum_{i=1}^{N_{systems}} \left[T^{out}_{w,cc,i} \left(\frac{\dot{m}^{in}_{w,cc,i}}{\dot{m}_{cps}} \right) \right] = T_{cpr} \tag{5-25}$$

Calculation of the water temperature change across the coils was discussed in detail in Section 3.3.1. Therefore, here it is sufficient to reiterate that this temperature change is a function of: the coil air inlet conditions; the coil size, geometry, and construction; the water inlet conditions;

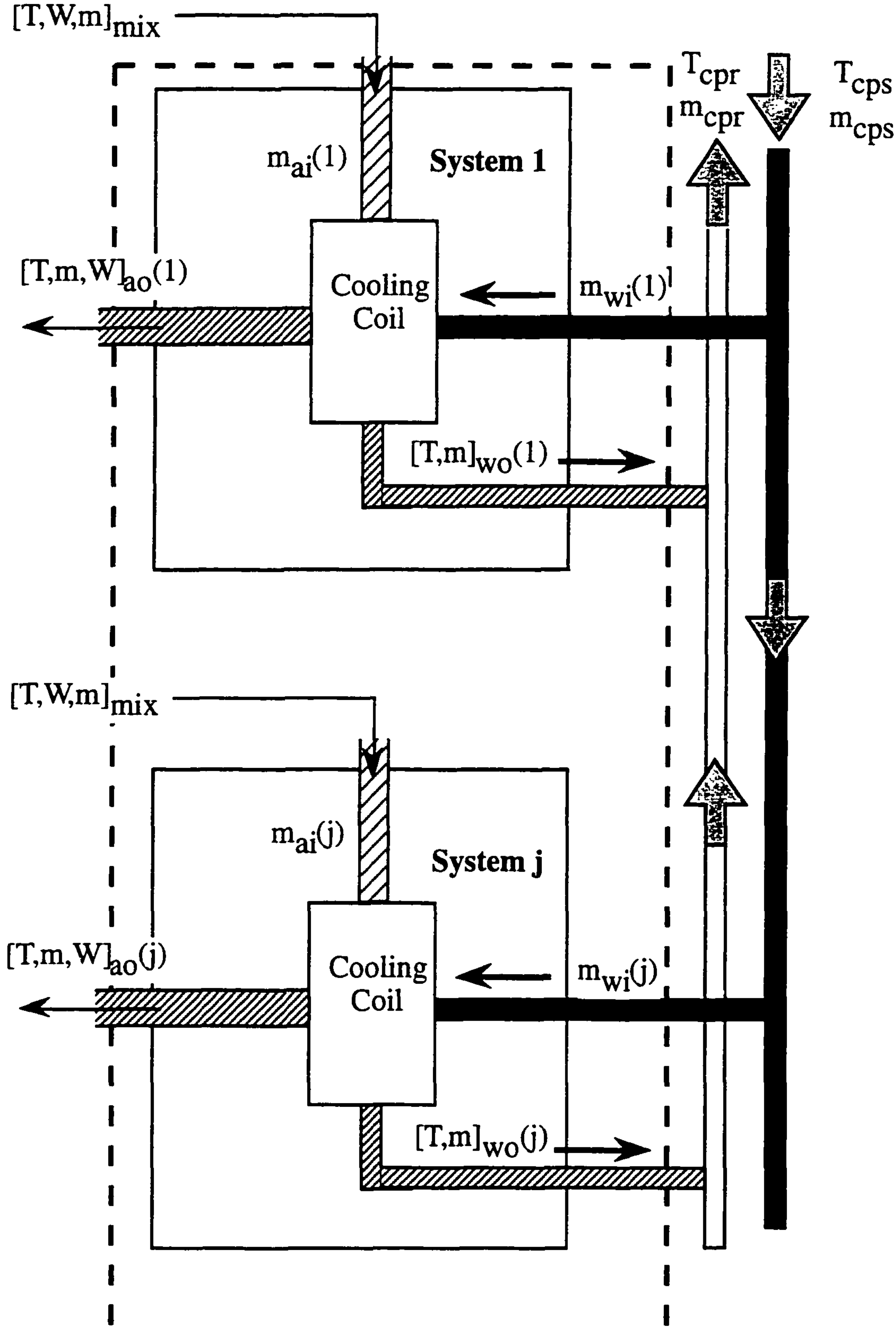


Figure 5-3: System chilled water loop and flow variables

and the desired air outlet temperature. The coil air and water outlet conditions are found from the simultaneous solution of Equations 5-26, 5-27, and 5-28:

$$T^{out}_{cc,i} = f_{cc,i}\left(\dot{m}_{mix,i}, T^{in}_{cc,i}, W_{mix,i}, W^{out}_{cc,i}, Y_{cc,i}, \dot{m}^{in}_{w,cc,i}, T^{in}_{w,cc,i}, T^{out}_{w,cc,i}\right) \tag{5-26}$$

$$T^{out}_{w,cc,i} = f_{w,cc,i}\left(\dot{m}_{mix,i}, T^{in}_{cc,i}, W_{mix,i}, W^{out}_{cc,i}, Y_{cc,i}, \dot{m}^{in}_{w,cc,i}, T^{in}_{w,cc,i}, T^{out}_{cc,i}\right) \tag{5-27}$$

$$W^{out}_{cc,i} = f_{W,cc,i}\left(\dot{m}_{mix,i}, T^{in}_{cc,icc_{m},i}, W_{mix,i}, Y_{cc,i}, \dot{m}^{in}_{w,cc,i}, T^{in}_{w,cc,i}, T^{out}_{cc,i}, T^{out}_{w,cc,i}\right) \tag{5-28}$$

However, the form these equations take depends upon the assumptions made to describe the heat transfer processes in the cooling coil between the air and water. Except for the most simple derivations, it is usually not possible to write explicit forms of Equations 5-26 through 5-28.

The procedure for describing the heating and reheat coils is much the same as that used for the cooling coils. The one major difference is that there may be several reheat coils per system, such as in a variable air volume system serving several zones. Conservation of mass on the hot water streams in this system is given by Equation 5-29:

$$\sum_{i=1}^{N_{systems}}\left[\sum_{j=1}^{N_{zones(i)}} \dot{m}^{out}_{w,rhc,j,i} + \dot{m}^{out}_{w,hc,i}\right] = \dot{m}_{hpr} \tag{5-29}$$

Because reheat coils are associated with a zone as well as a system, the reheat coil water mass flow rate is summed over each zone on each system attached to the plant. Conservation of energy on the inlet side of the heating coils requires the coil water entering temperature to equal the supply temperature from the plant as expressed by Equation 5-30:

$$T^{in}_{w,hc,i} = T^{in}_{w,rhc,j,i} = T_{hps} \tag{5-30}$$

On the return side of the heating coils, conservation of energy gives the return water temperature to the plant T_{hr} shown in Equation 5-31:

$$\sum_{i=1}^{N_{systems}}\left[\sum_{j=1}^{N_{zones(i)}} T^{out}_{w,rhc,j,i}\left(\frac{\dot{m}^{in}_{w,rhc,j,i}}{\dot{m}_{hps}}\right) + T^{out}_{w,hc,i}\left(\frac{\dot{m}^{in}_{w,hc,i}}{\dot{m}_{hps}}\right)\right] = T_{hpr} \tag{5-31}$$

A set of equations similar to 5-26, 5-27, and 5-28 is needed to describe how the air and water temperatures change across the coil. But, because all heat exchange is sensible, the humidity ratio is constant across a heating coil, that is:

$$W^{out}_{hc,i} = W^{in}_{hc,i}, W^{out}_{rhc,j,i} = W^{in}_{rhc,j,i} \tag{5-32}$$

Therefore, simultaneous solution of Equations 5-33 and 5-34 yields the two unknown outlet variables, the air outlet and water outlet temperatures:

$$T^{out}_{hc,i} = f_{hc,i}\left(\dot{m}_{mix,i}, T^{in}_{hc,i}, Y_{hc,i}, \dot{m}^{in}_{hc,i}, T^{in}_{w,hc,i}, T^{out}_{w,hc,i}\right) \tag{5-33}$$

$$T^{out}_{w,hc,i} = f_{w,hc,i}\left(\dot{m}_{mix,i}, T^{in}_{hc,i}, Y_{hc,i}, \dot{m}^{in}_{hc,i}, T^{in}_{w,hc,i}, T^{out}_{hc,i}\right) \tag{5-34}$$

As with the equations for the cooling coil outlet conditions, usually explicit forms of Equations 5-33 and 5-34 can not be written without oversimplifying the problem.

5.2.2: Objective Function for Optimization of Load Distribution

The problem being considered is the minimization of the total energy consumption of all the feasible combinations of plant equipment at each time step of the simulation. Feasible combinations are those that can supply the necessary amount heating or cooling to the systems served by the plant. This optimization is a necessary first step to determine the plant equipment operating schedule that minimizes the total cost of the energy used over a daily cycle of operation. For each equipment combination, the quantity that must be minimized is calculated by summing the power consumption of each plant and system component. Equation 5-35 represents this summation:

$$J = \sum_{i=1}^{N_{chillers}} \dot{Q}_{c,i} + \sum_{i=1}^{N_{chiller\,pumps}} \dot{Q}_{cp,i} + \sum_{i=1}^{N_{boilers}} \dot{Q}_{b,i} + \sum_{i=1}^{N_{boiler\,pumps}} \dot{Q}_{bp,i} + \sum_{i=1}^{N_{systems}} \dot{Q}_{system,i} + \dot{Q}_{ts} + \dot{Q}_{tsp} \tag{5-35}$$

The first term on the right hand side represents the power consumption of each chiller and can be broken down further into a load dependent component and a load independent, or parasitic, component that runs the chiller's control system, and any other equipment for which power consumption is not a function of the load on the chiller. This is given by Equation 5-36:

$$\dot{Q}_{c,i} = \dot{Q}_{c,load,i} + \dot{Q}_{c,parasite,i} \tag{5-36}$$

Similarly the power consumption of each boiler has load dependent and independent parts as given by Equation 5-37:

$$\dot{Q}_{b,i} = \dot{Q}_{b,load,i} + \dot{Q}_{b,parasite,i} \tag{5-37}$$

Both the chiller and boiler plants require pump energy to circulate the chilled or hot water between the system and plant. This energy consumption is $\dot{Q}_{cp,i}$ for the chiller pumps and $\dot{Q}_{bp,i}$ for the boiler pumps. There need not be the same number of pumps as chillers or boilers depending on the specific plant design.

Although none of the system variables are being directly controlled by the optimization process, the energy consumption rate of the system is affected by the performance of the central plant. This is especially true if the system can mix heated and cooled air streams as in the VAV and Dual Duct (Multizone) systems. Therefore, the system energy consumption should be included in the objective function because it is, at least implicitly, dependent on what the plant is doing. The components of the total system energy consumption are given in Equation 5-38:

$$\dot{Q}_{system,i} = \left(\sum_{j=1}^{N_{fans}} \dot{Q}_{fan,j,i} + \dot{Q}_{control,i} + \dot{Q}_{other\ equip,i} \right) \tag{5-38}$$

The first component of the system energy consumption is due to the fans that circulate air throughout the system. There is a fan in the supply duct and there may also be fans in each zone to exhaust air directly from the building. Finally, fans in the return air duct are used to provide positive return air pressure to the mixing box but are not present in all. As with the chillers and boilers, system control equipment energy consumption is relatively load insensitive. Additionally, there may be equipment associated with the system, such as electric heating coils, for which energy consumption is load dependent but the energy is not provided by the chillers or boilers.

The last two terms on the right hand side of Equation 5-35 represent the energy consumption of the thermal storage system. The first of these terms $\dot{Q}_{ts}$ is the power consumption of the storage system chiller and does not include the consumption of ice from the storage tank. This is because the rate of ice consumption is determined from the equipment schedule optimization. The second term $\dot{Q}_{tsp}$ is the power consumption of the pumps used to circulate brine through the thermal storage system .

5.2.3: Definition of Control Variables

In optimization problems, controls are variables the optimizer adjusts to improve the solution and drive the objective function to a minimum. At the same time, the optimizer can only change these variables in a way that also satisfies the implicit and explicit constraints of the system. Control variables generally represent physical parameters of the problem, meaning that sensors can measure their value and they can be manipulated in order to obtain the calculated performance. In this problem, the primary controls are on plant flow variables such as mass flow rates and temperatures. To clarify the discussion, the plant control variables will be considered separately according to whether they affect the heating or cooling plant equipment.

5.2.3.1: Heating Plant Control Variables

The heating plant consists of one or more boilers, possibly of different types, connected in parallel. The load applied to each boiler can therefore be varied independently by changing the water mass flow rate to each boiler and the water temperature change across the boiler. However, rather than control the actual mass flow rate, use of a normalized flow rate can help avoid scaling problems in the optimization routine because it only varies from 0 to 1. The normalized mass flow rate defines the flow rate to each boiler in terms of the heating plant total flow rate and the normalized flow rate as in Equation 5-39:

$$\dot{m}_{w,b,i} = X_{bo,i}\dot{m}_{hpr} \tag{5-39}$$

Equation 5-40 follows from Equation 5-39 based on the definition of the normalized flow rate:

$$\sum_{i=1}^{N_{boilers}} X_{bo,i} = 1 \tag{5-40}$$

The performance of each boiler can further be manipulated by adjusting the water outlet temperature $T_{bo,i}$. However, the water outlet temperature cannot be adjusted independently of both the plant heating load and the plant water flow rate unless excess energy can bypass the system. The final necessary control variable is the mass flow rate through the heating plant bypass line $\dot{m}_{hwb}$. Adjustments are made, as necessary, to ensure that the net plant heating output matches the system hot water demands.

5.2.3.2: Cooling Plant Control Variables

Other than the fact that the cooling plant produces chilled instead of hot water, the cooling plant layout differs little from that of the heating plant. The cooling plant equivalent of Equation 5-39 is given by Equation 5-41:

$$\dot{m}_{w,c,i} = X_{co,i}\dot{m}_{cpr} \tag{5-41}$$

If there is a thermal storage unit on the plant an additional expression is required to define the flow rate through the storage system in terms of the overall flow rate:

$$\dot{m}_{w,ts} = X_{tso}\dot{m}_{cpr} \tag{5-42}$$

Combining Equations 5-41 and 5-42 gives the cooling plant equivalent to Equation 5-40, stating that the sum of all the chiller mass flow rates and the thermal storage flow rate will be equal to the total plant chilled water flow rate:

$$\sum_{i=1}^{N_{chillers}} X_{co,i} + X_{tso} = 1 \tag{5-43}$$

As with the boiler water outlet temperatures, the chiller and thermal storage system outlet water temperatures may be adjusted to improve the efficiency of each unit. This requires controlling the flow of chilled water through the bypass line to maintain the energy balance between the system and plant interface as was the case with the hot water plant.

5.2.4: Variable Constraints

Constraint equations can have several purposes and they can be in the form of equalities or inequalities. One function is to keep simulation variable, or quantities based on a combination of several variables, within specified limits or equal to a desired value. This is usually to ensure that the solution generated by the optimizer is physically reasonable. However, it is also possible to incorporate the state equations, the governing equations of the system, as equality constraints. In this problem, these equations would be conservation of mass and energy for the plant and fan system simulations and the equations governing the performance of the chillers, boilers and pumps. However, this would vastly increase the number of control variables that the optimizer would have to adjust in order to satisfy all the constraints. This would be a major drawback to this approach because the number of computations for each iteration of the optimizing routine is proportional to the square of the number of control variables.

5.2.4.1: Constraints on the Heating Plant

The constraints for the heating plant will be considered first. Equation 5-44, normalized by the total hot water flow rate, ensures that conservation of mass is satisfied in the heating system by requiring the sum of the individual boiler hot water flows to equal the total hot water flow:

$$\sum_{i=1}^{N_{boilers}} X_{bo,i} - 1 = 0 \tag{5-44}$$

However, each boiler may have a maximum allowable flow rate due to finite pump capacity. Equation 5-45 prevents this flow rate from being exceeded:

$$\dot{m}_{bo,i}^{max} - X_{bo,i}\dot{m}_{hwr} \geq 0 \tag{5-45}$$

The flow rate through each boiler must also be positive to ensure a physically realistic solution. This constraint is enforced by Equation 5-46:

$$X_{bo,i} \geq 0 \tag{5-46}$$

The remaining flow constraints are on the flow through the bypass line. Clearly, the flow rate in this line must be greater than or equal to zero and less than or equal to the flow rate in the hot water supply line from the boilers, as shown in Equations 5-47 and 5-48 respectively:

$$\dot{m}_{hwb} \geq 0 \tag{5-47}$$

$$\dot{m}_{hwb} - \dot{m}_{hws} \leq 0 \tag{5-48}$$

Finally, constraints on the maximum and minimum boiler outlet water temperature must be considered. Regardless boiler capacity, there will be a water outlet temperature that cannot be exceeded. However, because the boiler performance is simulating using curve fits and not a detailed model of the heat transfer processes occurring within the boiler a constraint is necessary to ensure physically realistic results:

$$T_{bo,i} - T_{bo,i}^{max} \leq 0 \tag{5-49}$$

Equation 5-50 simply states that the outlet temperature of the boiler cannot be less than the inlet water temperature:

$$T_{bo,i} - T_{hwr} \geq 0 \tag{5-50}$$

5.2.4.2: Constraints on the Chilled Water Plant

The chilled water plant has generally the same constraints as the heating plant with the exception that constraints must be placed on the control variables associated with the thermal storage system. The chilled water plant equivalent of Equation 5-44 ensures that the sum of the flows through each chiller and the thermal storage tank is equal to the total flow rate in the chilled water system, as given by Equation 5-51:

$$\sum_{i=1}^{N_{chillers}} X_{co,i} + X_{tso} - 1 = 0 \tag{5-51}$$

The flow rate through any one chiller or the thermal storage system must also be less than or equal to the maximum allowable chilled water flow rate through that component of the system. This is represented by Equations 5-52 and 5-53 respectively:

$$\dot{m}_{co,i}^{max} - X_{co,i}\dot{m}_{cwr} \geq 0 \tag{5-52}$$

$$\dot{m}_{tso}^{max} - X_{tso}\dot{m}_{cwr} \geq 0 \tag{5-53}$$

The minimum flow rate through each chiller and the thermal storage system must also be greater than or equal to zero to obtain physically reasonable results, as indicated in Equations 5-54 and 5-55:

$$X_{co,i} \geq 0 \tag{5-54}$$

$$X_{tso} \geq 0 \tag{5-55}$$

Again, the flow in the bypass line must be controlled so that it does not exceed the flow rate in the system or be less than zero. These constraints are satisfied by enforcing Equations 5-56 and 5-57 respectively:

$$\dot{m}_{cwb} \geq 0 \tag{5-56}$$

$$\dot{m}_{cwb} - \dot{m}_{cws} \leq 0 \tag{5-57}$$

As with the heating plant, the outlet temperature from the chillers and thermal storage system must be constrained because of the use of curve fit performance models to simulate them. Equations 5-58 and 5-59 limit the outlet temperature from the chillers and thermal storage tank to be greater than or equal to a specified minimum value:

$$T_{co,i} - T_{co,i}^{min} \geq 0 \tag{5-58}$$

$$T_{tso} - T_{tso}^{min} \geq 0 \tag{5-59}$$

The outlet temperature must also be less than or equal to the chiller or storage system inlet temperature as given by Equations 5-60 and 5-61:

$$T_{co,i} - T_{cwr} \leq 0 \tag{5-60}$$

$$T_{tso} - T_{cwr} \leq 0 \tag{5-61}$$

Equations 5-51 through 5-61 thus represent all the constraints necessary to control the cooling plant components and produce the physical results.

5.2.5: Implementation of Plant Equipment Optimization in IBLAST

The purpose of implementing the optimization of the central plant equipment components as described above was to provide, at each time step of the simulation, all the feasible optimized states from which an optimal schedule of equipment operation could be developed. Therefore, the variables of greatest interest were the individual part load ratios of each piece of plant equipment. These indicated directly how the loads were being apportioned to the various chilling and heating units. As a result, the control variables used to optimize the central plant components were the part load ratios for each piece of equipment. In implementing the optimization using this simplified method, instead of as described in previous sections, it was crucial to ensure: that energy was conserved between the systems and plant, that the system leaving water temperature matched the plant entering temperature, and that the plant leaving temperature matched the system entering temperature. The main simplifying assumption, differentiating this method from the one outlined previously, is that the leaving water temperature is the same from each plant component. Provided that a particular chiller or boiler is not operating at the limit of its design capacity this assumption is not unreasonable. However, if all the chillers or boilers on the plant were operating at capacity then there would not be any degrees of freedom left to optimize since the part load ratio for each chiller or boiler, depending on whether the heating plant or cooling plant was out of capacity, would be set to meet as much of the load as possible. Furthermore, the plant leaving temperatures would reflect the out of capacity condition of the plant.

The objective function that was actually minimized, for each plant combination at every time step, was the total cost of the energy consumed during each simulation time step. That is, the power consumption of each energy type multiplied by the appropriate cost factor as shown in Equation 5-62:

$$J = C_{elec}\dot{Q}_{elec} + C_{gas}\dot{Q}_{gas} + C_{\substack{boiler\\fuel}}\dot{Q}_{\substack{boiler\\fuel}} + C_{\substack{diesel\\fuel}}\dot{Q}_{\substack{diesel\\fuel}} \tag{5-62}$$

where J is the objective function, C represents the cost factor for each type of energy, and $\dot{Q}$ is the energy consumption rate of each energy type. Since $\dot{Q}$ is constant over a time step there was no need to convert from power consumption to the energy consumption by multiplying by the time step. The constraint that was used for the chiller plant was conservation of energy. This was defined so that the cooling load minus the contribution from each individual chiller was equal to zero when the constraint was satisfied. This is shown in Equation 5-63:

$$\dot{Q}_{c,load} - \sum_{i=1}^{n_{chillers}} Q_{i,cooling} = 0 \tag{5-63}$$

Equation 5-64 is the heating plant's counterpart to Equation 5-63. Note that this equation is coupled to the cooling plant via waste heat produced by fossil fuel powered chilling equipment. Implicit in this equation is the heat energy required to operate equipment such as, absorption chillers, further linking the optimization of the boiler plant to the chiller plant.

$$\dot{Q}_{h,load} - \sum_{i=1}^{n_{chillers}} Q_{i,waste} - \sum_{i=1}^{n_{boilers}} Q_{i,heating} = 0 \tag{5-64}$$

The control variables that were varied to minimize the energy cost of the plant were the part load ratios for each plant component as defined by:

$$\dot{Q}_i = X_{i,plr} \dot{Q}_{i,capacity} \tag{5-65}$$

where the part load ratio $X_{i,plr}$ modifies the equipment capacity to give the actual heating or cooling provided. The part load ratio, $X_{i,plr}$, was constrained to values between the maximum and minimum allowable operating part load ratios for each piece of equipment.

In order to accomplish the actual optimization and determine the best equipment operating part load ratios each time step, a solver was required. A generalized reduced gradient (GRG) routine (GRG User's Guide for UNIX, 1986) was readily available and the problem described above was modified for solution by this method. Implementation of the GRG solver required the development of two subroutines: one to control the GRG solver OPTIMISE and the other to evaluate the objective function GCOMP. The source code for these subroutines is listed in Appendix A. In addition to defining all the variables necessary for running GRG, most of which were left with their default values, the OPTIMISE routine also performed an initializing simulation of the operating plant components to obtain a feasible starting point for the solver to work with.

Performing the plant equipment optimization in this manner results in a combined equipment energy consumption that is not, in general, a global minimum. The GRG solver varies the equipment part load ratios of each feasible equipment combination at every time step resulting in an approximate global minimum. Other parameters affecting plant operation, such as the plant leaving water temperature, would need to be included in the GRG solver procedure to obtain a true global minimum. However, the governing equations and objective function are nonlinear in these parameters, increasing the likelihood of the GRG solver finding local instead of global minima. In the approach used for this problem, these additional parameters were set outside the GRG routine eliminating the possibility of local minima because the coefficient of performance of a chiller is a quadratic function of part load ratio and the efficiency of a boiler is at most a quadratic function of part load ratio. The objective function for combinations of several chillers and boilers is still, at most, a quadratic in the control variables because it is, simply, the sum of several quadratic equations. Thus, there could be only one feasible minimum when optimizing on the basis of part load ratio alone and local minima should not exist.

5.3: Optimal Scheduling of Conventional Plant Components

Scheduling of central plant components implies changing the operating plant equipment over a period of time. In practice, this requires physically turning chillers, boilers, pumps, or other equipment on or off as the load changes to have the plant components operate as efficiently as possible. For the purpose of this work, optimal scheduling of a building's central plant components implies selection of plant equipment operation to minimize the cost of the energy consumed by the plant. This is the logical approach from the building management standpoint since the bottom line is typically measured in dollars not kilowatt-hours. If the cost of energy is constant over one simulation period, then minimizing total energy consumption is equivalent to minimizing cost. Fuels such as heating oil and coal for boilers and diesel fuel for generators and chillers are generally bought in bulk quantities and stored at the site where they are used. Therefore, the cost per unit of energy released can be considered a constant over a one day cycle. However, the cost of electricity provided by a utility varies over a daily cycle, usually the rate structure is such that the cost of energy is high when consumption is high, such as during the late morning and early afternoon hours, and lower when consumption is low. In addition, many utilities impose demand charges on their customers who consume large amounts of energy. Typically demand charges are determined by the peak electric power consumption during the utility's billing cycle and is assessed per day of that cycle. For example, a simple formula for calculating the demand charge is given by Equation 5-66:

$$(\text{demand charge}) = \left(\frac{\$}{\text{day} - \text{kW}}\right)(\#\,\text{days in cycle})(\text{peak kW}) \tag{5-66}$$

In practice, the demand charge structure may be considerably more complex and varies between utility companies. However, the procedures to solve the optimization problem with a different rate structure should not require radical changes from those presented here.

Demand charges penalize peak energy consumption rates independent of the amount of energy actually consumed. In order to avoid or at least minimize these charges, consumers must operate their equipment so that the situations that cause large peaks in power consumption do not occur during the period when peak power consumption is being tracked. Spikes in power consumption generally occur when equipment is turned on because, during the start-up sequence, operating conditions are far from nominal and additional power is required to accelerate: pumps, fans, motors, etc., up to their design operating speeds. Typically, the chiller start-up spike has a short duration of about 5 minutes, and causes the equipment to draw 20% more power than at steady state (Olson, 1987, 1990, 1993, and 1994). It is reasonable to assume that similar effects are observed with other types of equipment though the duration and percentage increase in consumption may be different. Therefore, in order to avoid a large demand charge but maintain sufficient capacity to meet the building loads, equipment may be turned on before the on-peak period begins. However, this will usually increase the total amount of power consumed because the equipment will be operating further away from the best part load ratio. In addition, this method requires prediction of the peak loads during the on-peak period so that the available equipment meets the peak load with little or no excess capacity. This presents a dilemma because adding too much capacity means the equipment runs at an inefficient part load ratio and consumes much energy. But, not adding enough capacity means the loads cannot be met and conditions in the building are likely to become uncomfortable for the occupants

Clearly, optimizing the plant equipment schedule involves a trade-off between increased total energy consumption, because excess plant capacity is maintained to avoid equipment start-up transients, and operating the equipment at an optimal part load ratio but incurring many transients as a result of changes in equipment operation. The incentive to pursue the former strategy increases as the demand charge grows relative to the other operating costs. The latter strategy would be the best option when there is no demand charge since it guarantees the lowest possible power consumption during any and every time interval. The purpose of Sections 5.3.1 and 5.3.2 is to discuss methods of quantifying these trade-offs by defining an objective function and implementing an appropriate method to minimize it.

5.3.1: Objective Function for Optimal Scheduling

Minimization of the total energy cost incurred over a specified period of time requires the quantity being minimized to be the sum of all energy costs over the simulation period accounting for different types of fuel, hourly variations in utility rates and demand charges. This quantity, minimized by the optimal equipment schedule, is calculated from Equation 5-67:

$$C = \sum_{i=1}^{N_{fuel\,types}} c_i Q_{i,tot} + \int_0^T c_{util\,elec}(t)\dot{Q}_{elec}(t)dt + c_{demand}\left(\dot{Q}_{elec}^{peak} - \dot{Q}_{elec}^{base}\right) \tag{5-67}$$

The first term on the right hand side of this equation represents the total cost over the simulation period for energy that has a fixed cost per unit of consumption. That is, the cost of the energy does not vary with time of day. Therefore, the cost may be calculated by computing a total for each type of energy over the entire simulation period, and then multiplying by the cost per unit of energy consumption. The energy use of fossil fuel powered equipment such as diesel generators and natural gas fired boilers would be included in this term. The second term on the right hand side of Equation 5-67 is used to calculate the total cost of energy when that cost is time dependent. However, utility electricity cost normally varies in discreet amounts, not continuously, during the day. Therefore, the second term of Equation 5-67 can be written as a summation over all the time steps of the simulation period as in Equation 5-68:

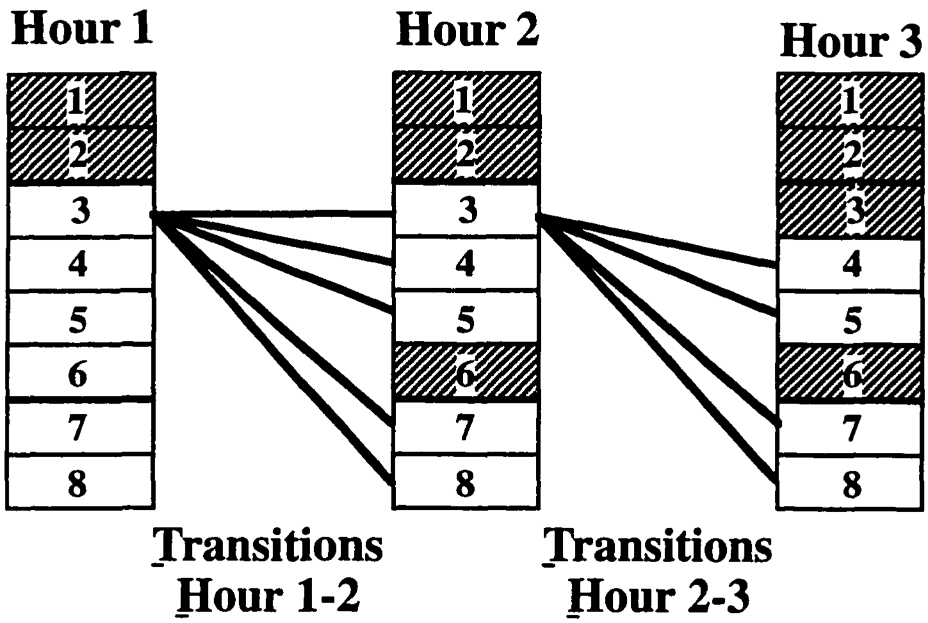
$$\int_0^T c_{util\ elec}(t)\dot{Q}_{elec}(t)dt = \sum_{i=1}^{N_{time\ steps}} c_{util\ elec}(t)\dot{Q}_{elec}(t)\Delta t_i \qquad (5\text{-}68)$$

The last term on the right hand side of Equation 5-67 represents the increase in energy cost due to utility demand charges. These are assessed per unit power consumption at the peak rate, but the peak may also be required to be above a certain threshold level $\dot{Q}_{elec}^{base}$ before demand charges begin to be assessed.

5.3.2: Minimization by Finding Least Cost Path

The objective function given by Equations 5-67 and 5-68 gives a formula for calculating the total energy cost incurred over a specified simulation period. However, these equations provide no information which leads directly to a method of minimization. A clue is the discreet nature of the simulation in terms of both plant equipment choices and the finite time step used to update zone, system, and plant conditions. This means that, in principle, the data may be tabulated and the least cost path computed at the end of the simulation period.

An overview of how a tabular approach to optimization may work for conventional central plant scheduling is shown in Figures 5-1,through Figure 5-4. In Figure 5-1, one column represents the all the plant combinations possible during an hour of the simulation period. Each block in a column defines the total energy consumption, the part load ratios X_i of each piece of equipment, and peak, initial, and final power consumptions associated with one specific combination of plant equipment. Since the simulation time step is shorter than the one hour record keeping time step, the total energy consumption for a combination is computed by summation of the energy consumed during each time step. The equipment part load ratios may vary with each time step during the hour so an average value is used to represent the combination for a given hour. The peak power consumption is the maximum energy consumption rate experienced by the combination during the hour. Lastly, the initial and final power consumptions are, respectively, the energy consumption rates at the beginning and end of the hour. All equipment combinations may not be feasible during a specified hour because of insufficient capacity to meet the heating or cooling loads. For example, in hour 2 combinations 1, 2, and 6 are represented as infeasible. The lines from combinations at one hour to combinations at the next hour represent changes in equipment usage from one hour to the next. These will, subsequently, be referred to as *transitions*, each of which has a power consumption rate associated with it to account for the base equipment configuration of the current hour, plus the effects of equipment start-up to obtain an equipment combination in the subsequent hour. Equipment transitions are only allowed to occur at the end of an hour, not during the hour itself.



3-3	4-3	••••	8-3
3-4	4-4	••••	8-4
3-5	4-5	••••	8-5
3-7	4-7	••••	8-7
3-8	4-8	••••	8-8

3-4	4-4	••••	8-4
3-5	4-5	••••	8-5
3-7	4-7	••••	8-7
3-8	4-8	••••	8-8

Figure 5-1: Evaluate All Feasible Equipment Combinations and Transitions

Figure 5-2 shows the minimum energy consumption path based on the performance of 8 possible equipment combinations. This is determined by selecting, at each hour, the equipment combination with the minimum energy consumption. Repeating this procedure for every hour gives a sequence of equipment operation which results in the minimum total energy consumption for the optimization period. Peak, initial, and final power consumption is, initially, ignored in computing the minimum energy schedule. However, these power consumption values are needed to determine the demand charge for the minimum energy path and ultimately allow the total energy cost to be computed. It is not possible, *a priori*, to determine whether the peak power consumption results from an equipment transition or the load on the plant equipment. The minimum energy consumption path gives a baseline energy cost and peak power consumption which are compared to the feasible alternative plant equipment schedules, i.e. sequences of plant combinations which differ from the minimum energy plant equipment schedule. In order to determine the actual minimum cost path all the remaining feasible paths must be searched since the peak power consumption may result from an equipment transition between one hour and the next. However, for a path to have a lower total cost than the minimum energy path, it must have a lower peak power consumption because its total energy consumption will, of necessity, be higher. As a result, equipment combinations and transitions that have a higher peak power consumption than the minimum energy path may be eliminated, reducing the total number of search paths. In addition, the peak power consumption associated with each equipment transition may be calculated to identify and eliminate transitions with higher peak power consumption than the minimum energy path. Figure 5-3 shows how the equipment combination and transition eliminations can substantially reduce the number of search paths required to find the minimum cost path. Finally, the remaining allowable paths are searched to determine the minimum cost path as shown in Figure 5-4. The procedure outlined above and its implementation into the IBLAST program are detailed further in Sections 5.3.2.1 and 5.3.2.2 respectively.

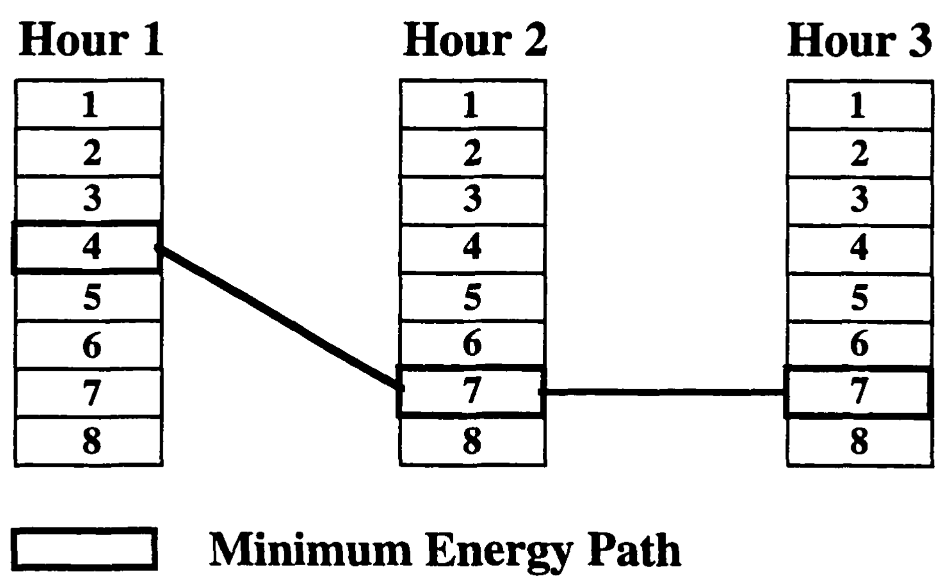


Figure 5-2: Calculate Minimum Energy Path Based on Hourly Energy Consumption

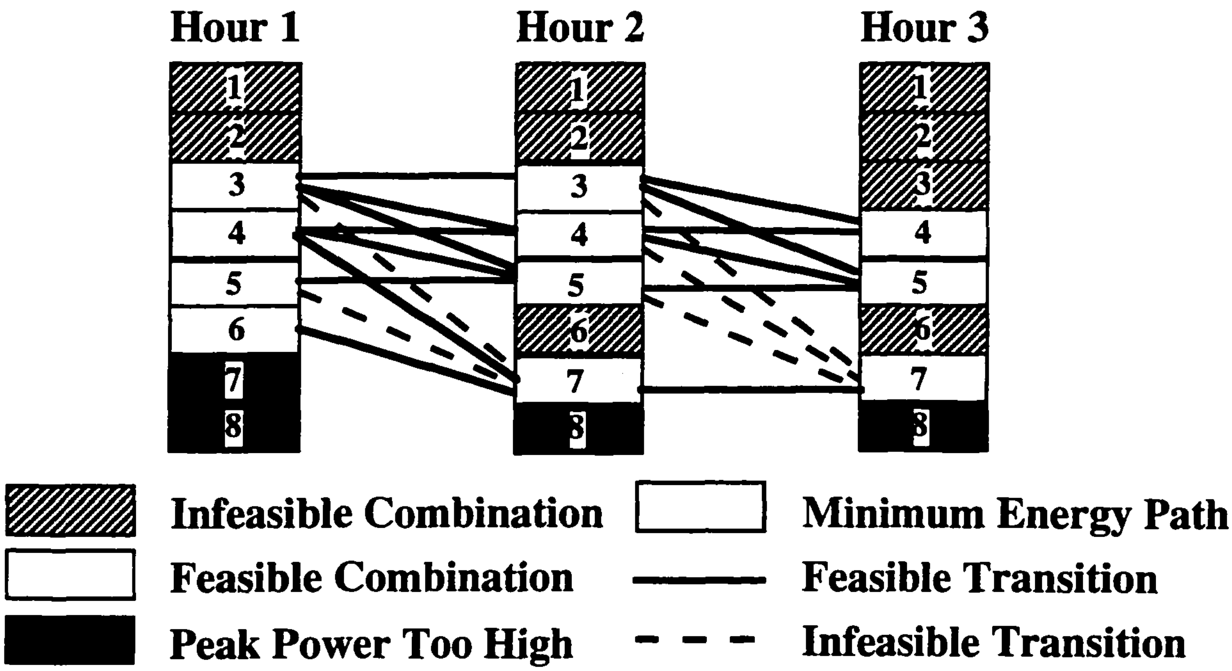


Figure 5-3: Determine Feasible Paths to Search by Eliminating Combinations and Transitions

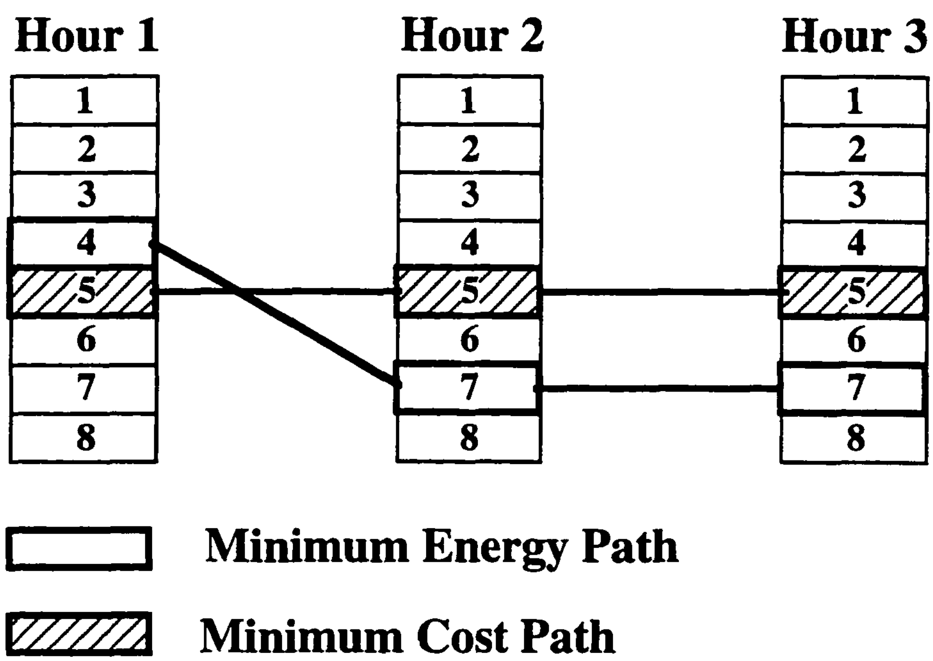


Figure 5-4: Determine Minimum Cost Path by Comparing Feasible Paths to Minimum Energy Path

5.3.2.1: Algorithm Details

One possible method for minimizing Equation 5-67 would be to evaluate all the possible paths, based on the feasible equipment combinations at each time step, from the initial time to the final time. As an example, consider a building with two identical chillers and one boiler on the central plant. At each time step of the simulation the plant has 6 possible operating configurations as indicated in Table 5-1. A path is generated by setting the equipment configuration at the current time and using this to determine the building state at the end of the time step. At the end of the first time step there would be 6 possible end states, each consisting of the variables required to define the conditions in all the zones, the systems, and central plants. Advancing the simulation another time step results in 36 possible end states. If the simulation period is 24 hours and a one hour time step is used then the total number of possible paths, or, more correctly, the number of possible simulation end states, is: $6^{24}=4.74\text{x}10^{18}$. By any standard, evaluating all these possible paths would be computationally infeasible, even for quite simple systems and very capable computers. Clearly, an alternative to this method, preferably one that reduces or eliminates the path dependency of the end states, is required.

A key to reducing the number of paths is realizing that, in general, system and plant operation is not path dependent. That is, the plant operating configuration at one time can not be

Table 5-1: Possible Equipment Operating Configurations for Plant with 2 Chillers and 1 Boiler

Configuration # Cooling Capacity / Heating Capacity	# Chillers	# Boilers
1 0 kW / 0 kW	0	0
2 100 kW / 0 kW	1	0
3 200 kW / 0 kW	2	0
4 0 kW / 150 kW	0	1
5 100 kW / 150 kW	1	1
6 200 kW / 150 kW	2	1

used to deduce the plant operating configuration at a previous time because there are normally several ways that the plant can meet the loads imposed on it. On the other hand, the conditions in the zone simulation are path dependent because of transient conduction through the walls and the zone air capacitance. However, zone conditions will track one path only, regardless of the plant equipment configuration operating at each time step, so long as the system and plant provide the same heating or cooling to each zone. This reduces the problem to one of calculating the energy consumption of each equipment configuration at each time step based on a fixed load to the system.

A simple example will now be considered to illustrate how an optimal path may be determined based on the plant equipment of Table 5-1 and a set of representative plant loads. Once all the equipment configurations have been considered for each time step, Table 5-2 is generated that represents typical equipment energy consumption based on the plant equipment configurations of Table 5-1.

Table 5-2 gives total and peak energy consumption data for all the equipment configurations over a block of time from 12 noon to 4pm. The bracketed numbers indicate that insufficient plant capacity was available to meet the zone loads. Equipment configurations 4 through 6 mirror configurations 1 through 3 except that the boiler is turned on. The numbers assume that, when turned on, both the chillers and boiler will consume 5kW of power to run control systems plus whatever power is required to meet the load. Therefore, configurations 4

Table 5-2: Representative hourly total and peak energy consumption information for the plant configurations defined by Table 5-1

	Time of Day / Cooling Load			
Configuration #	12-1 pm / 92kWh	1-2 pm / 130kWh	2-3 pm / 150kWh	3-4 pm / 165kWh
1	[0 / 0]	[0 / 0]	[0 / 0]	[0 / 0]
2	28kWh/30kW	[30kWh/30kW]	[30kWh/30kW]	[30kWh/30kW]
3	33kWh/35kW	43kWh/46kW	48kWh/50kW	51kWh/54kW
4	[5kWh/5kW]	[5kWh/5kW]	[5kWh/5kW]	[5kWh/5kW]
5	33kWh/35kW	[35kWh/35kW]	[35kWh/35kW]	[35kWh/35kW]
6	38kWh/40kW	48kWh/51kW	53kWh/55kW	56kWh/59kW

through 6 have a higher power consumption than 1 through 3, even though there is no heating load. Configuration 1 is always infeasible because there is a cooling load in each hour. Based on total power consumption alone, the optimal path operates the plant in configuration 2 between 12pm and 1pm, then in configuration 3 from 1pm to 4pm. The total consumption in this case would be 170kWh and peak consumption would be 54 kW. However, the start up of the second chiller at 1 pm has not been accounted for in determining the 54kW peak. Typically, the peak chiller power consumption during startup is 20% higher than its steady state value. Assuming that the chillers share the load equally when the second chiller comes on line, the peak consumption should be about 44kW based on a steady state load of 40kW at the beginning of the hour. Therefore, the decision to switch from configuration 2 to 3 could be made without affecting demand charges. However, if the transition between these configurations resulted in a startup peak consumption of more than 54kW, for example 57kW; finding the optimal path would require cost calculations for the two possible paths. Path 1 is as previously described and would have an associated cost of:

$$\text{Path 1 Cost} = 170\text{kWh} \times C_1(\$/\text{kWh}) + 57\text{kW} \times C_2(\$/\text{kW}) \qquad (5\text{-}69)$$

The cost of path 2, operating the plant in configuration 3 from noon to 4pm, would be:

$$\text{Path 2 Cost} = 175\text{kWh} \times C_1(\$/\text{kWh}) + 54\text{kW} \times C_2(\$/\text{kW}) \qquad (5\text{-}70)$$

The outcome of these calculations is dependent on the cost factors, C_1 and C_2, as shown in Table 5-3. As the demand charge C_2 increases the optimal path tends increasingly towards Path 2.

While the above example is instructive it does not incorporate the complexities occurring in larger plants that would result in the existence of many feasible paths. Olson concluded that an exhaustive search of all the possible paths would be required to find a global minimum because the peak energy consumption of a path is a function of the path itself. This conclusion led Olson to use heuristic methods to find the optimal path. However, the resulting solution was never guaranteed to be a global minimum.

Table 5-3: Path 1 and 2 Energy Cost Variations with Increasing Demand Charge

C_1 (\$/kWh) / C_2 (\$/kW)	Path 1 Cost (\$)	Path 2 Cost (\$)
0.10 / 0.00	17.00	17.50
0.10 / 0.10	22.70	22.90
0.10 / 0.17	26.50	26.50
0.10 / 0.20	28.40	28.30
0.10 / 0.30	34.10	33.70

An alternative to Olson's heuristic approach was developed that does result in a global minimum energy cost without requiring an exhaustive search. The key to this approach was that an arbitrary path has a cost associated with total energy consumption and a second cost for peak consumption. The next consideration is how to improve on the cost of this initial path. For example, the initial path could be the one consuming the minimum total amount of energy. As has been shown, this is found by selecting the plant configuration that consumes the least energy at each time step, ignoring the effect of demand charges. However, with this path there will also be an associated peak energy consumption. Therefore, the total cost associated with this path is:

$$C_{ref} = C_1 Q_{tot}^{min} + C_2 \dot{Q}_{peak}^{min} \tag{5-71}$$

where the min superscript implies that the total energy and peak power consumptions are on the path that consumes the least amount of energy. If the demand charge C_2 is zero then this is the least cost path. However, if C_2 is non zero then criteria may be defined that another path must meet to have a lower total cost. The total cost of any other arbitrary but feasible path is given by Equation 5-72:

$$C_i = C_1 Q_{tot}^{i} + C_2 \dot{Q}_{peak}^{i} \tag{5-72}$$

But, for any other path, the total energy consumed must be greater than the energy consumed by the least cost path. This is given by Equation 5-73:

$$Q_{tot}^{i} > Q_{tot}^{min} \tag{5-73}$$

Therefore, for the total cost of path C_i to be less than C_{ref}, the following must be true:

$$\dot{Q}_{peak}^{i} < \dot{Q}_{peak}^{min} \tag{5-74}$$

This means that any path segment, or transition, between equipment configurations having a higher peak consumption than the reference path can be eliminated from further consideration. The global minimum may thus be found by searching on a much reduced subset of the feasible paths.

A simple hypothetical example is illustrative in showing differences in the minimum energy and minimum cost paths and finding the minimum cost path without an exhaustive search. This example, in Table 5-4, represents typical chiller consumption trends over a four hour period, during which time the load on the chillers increases. The path that consumes the least energy from interval 1 through interval 4 uses the following sequence of options 1-1-2-3. The total consumption is 440 units with a peak of 148 units/time caused by the 2-3 transition at the end of hour 3. Thus, any option or transition with a peak consumption of more than 148 units/time can be eliminated further consideration. Since the load is increasing, transitions from high plant capacity to low capacity may also be disallowed. The five additional paths for which the total and peak consumption must be calculated to obtain a global minimum cost are given in Table 5-5, along with the minimum total consumption case that is indicated in boldface type.

Table 5-4: Typical Trends in Consumption and Peak Values for a Hypothetical Chiller Plant

Interval	1		2		3		4	
Option	Total	Peak	Total	Peak	Total	Peak	Total	Peak
1	87	95	100	103	--	--	--	--
2	95	102	108	112	121	125	138	138
3	110	115	120	124	130	133	132	135
4	136	138	142	145	150	153	152	154
	Transition Peaks							
	From	To						
	1	1	0.0		--		--	
	1	2	120		136		--	
	1	3	135		151		--	
	1	4	150		163		--	
	2	1	119		--		--	
	2	2	0.0		0.0		0.0	
	2	3	134		145		148	
	2	4	156		167		175	
	3	1	119		--		--	
	3	2	123		135		142	
	3	3	0.0		0.0		0.0	
	3	4	165		170		178	
	4	1	119		--		--	
	4	2	123		135		148	
	4	3	135		144		151	
	4	4	0.0		0.0		0.0	

Final determination of the minimum cost path from the paths in Table 5-5 requires that the costs C_1 and C_2 be specified. Figure 5-5 illustrates how the value of C_2 affects the total cost and, correspondingly, the minimum cost path. In order to more clearly illustrate this, the difference between the cost of the minimum energy path (path1) and the other paths from Table 5-5 are plotted against increasing demand charge C_2. Initially, the minimum energy path is the minimum cost path, but when C_2 increases above 0.14 (cost per unit peak consumption) path 2 becomes the optimal solution. Although, the example given is in many ways contrived it does highlight the steps necessary to accomplish the energy cost minimization.

Table 5-5: Total and Peak Energy Consumption for Paths Required to Obtain Global Minimum

Path	Total Consumption (Units)	Peak Consumption (Units/Time)
Path1 (1-1-2-3)	**440**	**148**
Path2 (1-2-2-2)	454	138
Path3 (1-2-3-3)	457	145
Path4 (2-2-2-2)	462	138
Path5 (2-3-3-3)	477	135
Path5 (3-3-3-3)	492	135

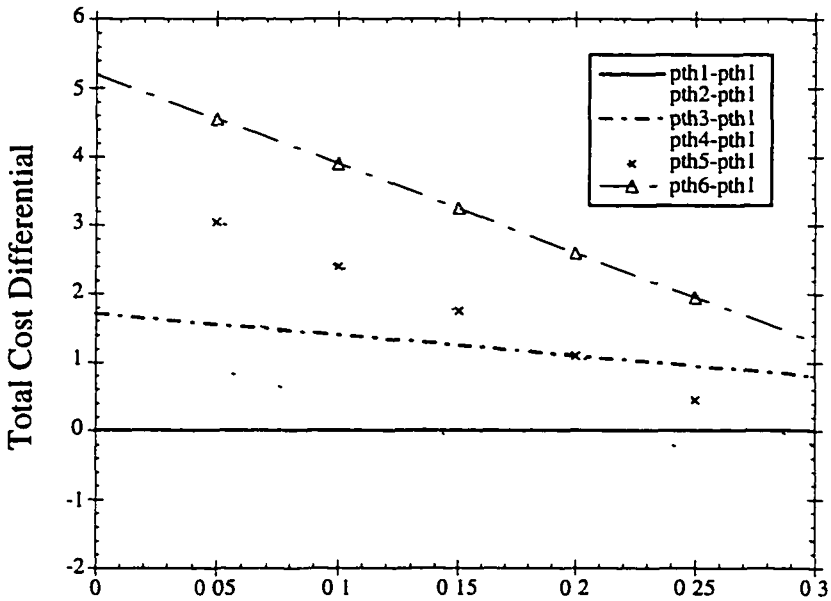


Figure 5-5: Effect of C_2 on Cost Differential Between Minimum Energy Consumption Path and Feasible Paths. C_1=0.1/Unit Total Consumption

5.3.2.2: Implementation in IBLAST

The above algorithm was implemented in IBLAST essentially as outlined. However, some procedural modifications and simplifications were found to be necessary and advantageous for effective implementation. The actual code required to implement the algorithms is contained in a FORTRAN source code file "pltcomb.ftn," and is listed in Appendix A. The order of the individual subroutines in this file is more or less the same order that they are called by the IBLAST program.

The first step in the optimization procedure is to initialize the simulation by evaluating all the possible central plant equipment combinations. This is accomplished by the EQCOMB subroutine that creates an array from which each combination of central plant equipment may be simulated. Each row of the array represents one combination, while each element of a row indicates how many of a specified equipment size and type will be allowed to operate. EQCOMB determines all the possible equipment combinations and does not exclude combinations of incompatible equipment, though this is certainly a feasible enhancement.

The SETEQUIP subroutine, following EQCOMB in Appendix A, selects the central plant equipment combination to be simulated, calls the main air handling system and plant simulation routine, and records information necessary to optimize the plant schedule. This information includes energy consumption stored by energy type and plant equipment part load fractions stored by equipment size and type. A complication was the need to simulate only the feasible equipment combinations, i.e. those that had enough heating and cooling capacity to meet the loads required to maintain the proper zone temperatures. This strategy can be justified because building space temperatures are not ordinarily allowed to exceed their control limits when there is additional central plant capacity available to meet the loads, assuming that plant capacity is the limiting factor. Normally in this situation, unutilized equipment would be turned on to maintain the proper zone temperature. Therefore, the equipment combination in which all the plant equipment is operating was simulated first in every time step. This generated baseline heating and cooling loads that were used for comparison with the heating and cooling capacities of subsequent combinations. Combinations with insufficient capacity were ignored. An additional procedure was required for the heating plant because waste heat can meet some or all of the heating load. However, this waste heat might not be available in every chiller plant combination resulting in a need for boiler energy even though none was required in the base case with everything turned on. Therefore, all the feasible chiller combinations were simulated with all the boiler plant combinations, whether or not a heating load was present, to avoid this situation.

The OPTAVRG and OPTAVRGH subroutines perform similar functions: averaging variables and searching for feasible combinations. OPTAVRG is called at the end of every user-defined time step, that determines when to update the surface temperatures in each zone. Energy consumption and equipment part load fractions are averaged over this time step, and equipment combinations that are not feasible for the whole time step are discarded. OPTAVRGH repeats the same procedure at the end of every hour. The result for each hour of the simulation is the average and peak energy consumption by energy type and the average and initial equipment part load fractions for each feasible combination.

Finally, the PLTOPSHCD routine computes the minimum cost hourly operating schedule for conventional (non-thermal storage) central plant equipment. Step one in this procedure was computing what will be referred to as the minimum energy path. However, this is actually a minimum cost path computed without considering the effects of demand charge. When the central plant equipment uses only one form of energy, this schedule is a true minimum energy path. However when combinations of energy types with different costs are used the resulting schedule is not generally a minimum energy path.

The next step determines the equipment combinations having a higher peak energy consumption than the minimum energy path at each hour . Such combinations are discarded since the minimum cost equipment schedule must have a lower peak energy consumption than the minimum energy path. The third step determines all the ways to get from each feasible equipment combination at one hour to each feasible combination at the next hour. The peak energy consumption associated with each transition accounts for changes in equipment usage from one hour to the next. Subsequently, transitions with a higher peak consumption than the minimum energy path are eliminated, along with transitions that do not follow plant load trends. For example, a transition that results in a reduction in chiller capacity is not allowed if the cooling load is increasing. Finally, the last major section of PLTOPSCHD computes the cost associated with each plant equipment schedule based on the remaining feasible plant equipment transitions. For each hour, equipment combinations are linked together so that if combination A were the optimal combination at the last off-peak hour, then the only transitions considered between that hour and the next would be transitions from combination A. Transitions between subsequent hours build on the previous hour's feasible combinations. The result of this procedure is a set of feasible paths and their associated costs, including the effect of peak demand. The least cost equipment schedule was then determined by comparing the path costs with each other and the minimum energy path.

5.4: Optimization of Thermal Storage Operating Schedule

5.4.1: Optimization Method

Initially, the thermal storage systems were to be optimized with the same procedure used for conventional central plant equipment. However, thermal storage systems have a characteristic that made direct implementation of this method impossible; the quantity of ice in the storage tank is time dependent. That is, the amount of ice in the tank depends on how much ice was used in previous hours and also affects storage tank performance. Therefore, optimizing the ice storage plant is not a question of piecing together a series of independent combinations to get an optimal schedule. Each prospective optimal path must be simulated from beginning to end.

Another question was the appropriate variable or variables to optimize the use of ice from storage compared to the use of direct cooling from the compressor or chiller plant. The answer was to use the compressor part load fraction for each on-peak hour of the simulation. However, the compressor part load fraction is continuously variable and can take on any value between 0 and 1. This means that infinitely many paths exist from the beginning of the optimization period to the end. The total number of paths was reduced by discretizing the variation in compressor part load fraction each time step to allow only a finite number of values. However, the number of paths that must be evaluated grows rapidly as more compressor part load fractions are considered each hour. If there are six on-peak hours and two possible compressor part load fractions, the total number of paths is $2^6 = 64$; certainly a reasonable number. However, if the compressor part load fraction is allowed to take on 3 or 4 values then the total number of paths that must be calculated is 729 or 4096 respectively. The number of paths rapidly becomes unmanageable. Figure 5-6 illustrates the growth in the number of possible chiller operating schedules when two chiller fractions are allowed each hour. Each path has a different associated cost and all paths must be searched to find the minimum cost. In the figure, the path resulting in COST 6 is shown as the optimal path.

On the other hand, the increase in the number of paths is accompanied by only a small a increase in accuracy of the optimal compressor schedule. Moreover, doubling the number of compressor operating states each hour increases the number of computations by a factor of 64, but only halves the uncertainty in the optimal path. In this context, accuracy implies the difference between the approximate optimal schedule calculated by this method and the true optimal schedule.Uncertainty, refers to the difference between the trial compressor part load fractions which is initially 0.34 when two part load ratios are allowed and 0.2 when four part load ratios are allowed. After one iteration, the uncertainties drop to 0.17 and 0.08, respectively.

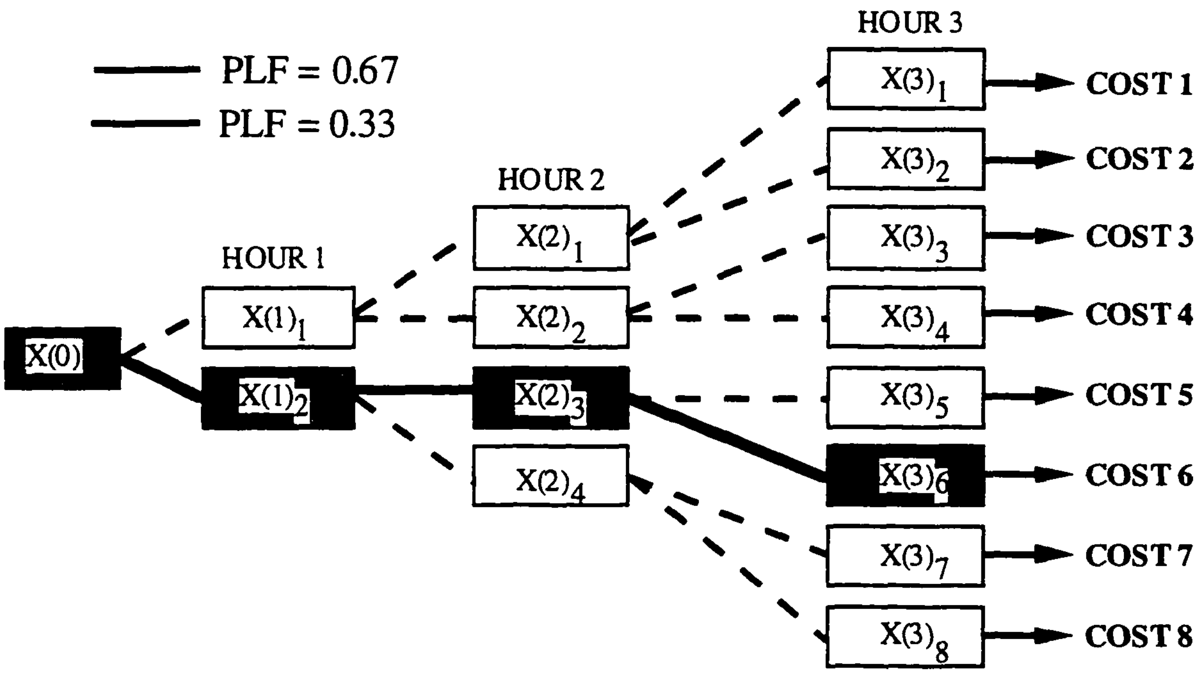


Figure 5-6: Growth of Storage Tank States With Time for Two Compressor Part Load Fractions

An alternative to using many compressor operating states each hour to improve accuracy is to use as few states as possible and repeat the optimization procedure several times to improve the accuracy of the solution. In each subsequent iteration, the interval between the allowable compressor part load fractions was narrowed while at the same time being centered on the previous iteration's optimal part load fraction for each hour.

This method yields an approximation to the global minimum energy cost path. An infinite number of compressor fractions, hence an infinite number of paths, would have to be considered for each time step of the simulation to obtain a true global minimum. However, the sensitivity of the solution to the number of compressor fractions allowed each hour should indicate whether the results obtained using 2 compressor part load fractions are reasonable. Therefore, the effects of using 2, 3, and 4 compressor operating states at each time step was evaluated assuming a four hour on-peak period to keep the total number of paths evaluated low. This resulted in 16, 81, and 256 paths respectively being generated. The compressor part load fractions used to initialize each test were: 0.33 and 0.67 for two allowable part load fractions per hour; 0.25, 0.5, and 0.75 for three; and finally, 0.2, 0.4, 0.6, and 0.8 for four. Figure 5-7 shows the results of these test cases after several days of iteration to ensure a converged solution i.e. one with a small uncertainty. In this example, the approximate optimal paths, defined by hourly variations in chiller part load fraction, were within 5% of each other during each hour of the 4 hour on-peak period. Correspondingly, Figure 5-8 shows good agreement in the hourly change in storage tank capacity for each of the three cases. In conclusion, using only 2 paths for each hour and repeating the optimization results in a reasonably good approximation to the global minimum cost path.

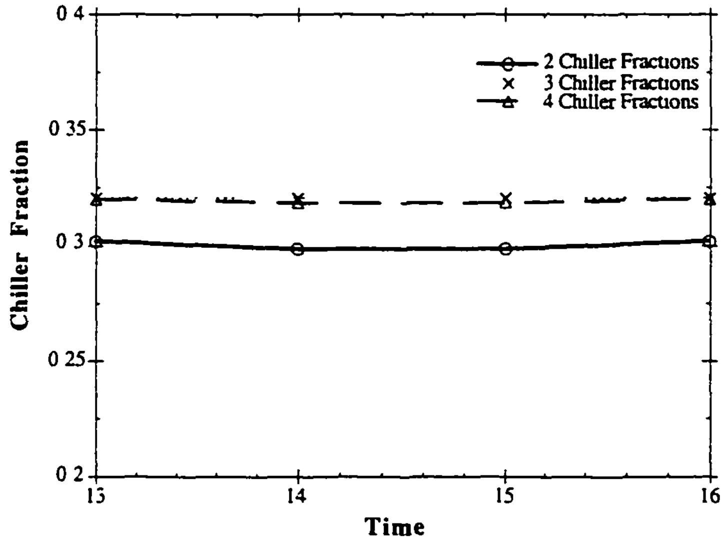


Figure 5-7: Comparison of Optimized Chiller Operating Fractions for Simulations Using 2, 3,, and 4 Chiller Fractions Per Time Step

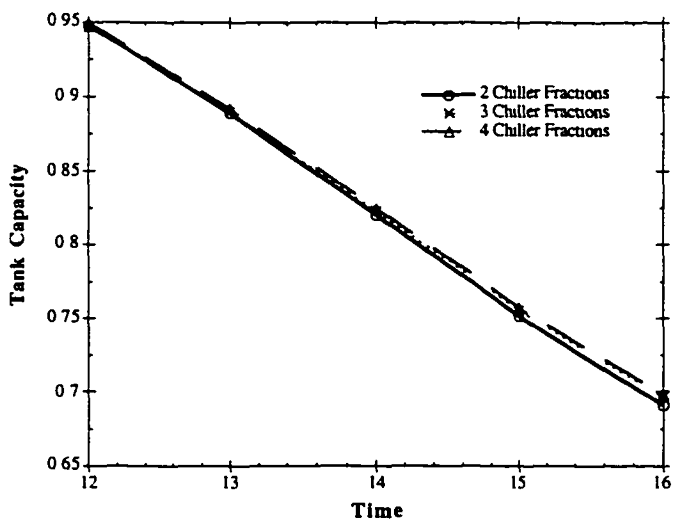


Figure 5-8: Comparison of Normalized Ice Storage Tank Capacity for Simulations Using 2, 3, and 4 Chiller Fractions Per Time Step

Finding the optimal compressor operating schedule through the on-peak hours required calculation of the total energy cost for that period of time. Selection of the optimal path was accomplished by direct comparison of the costs of each path. However, there are no direct costs associated with the consumption of ice from the storage tank because the energy required to build the ice was consumed at some prior time when, presumably, the cost of energy was cheaper. Yet a way of including the cost of ice used to compute the total cost of each path was required since the ice is clearly not free. The average cost of ice per kilowatt-hour used was calculated by determining the amount of energy required to build ice off-peak and dividing the resultant cost of that energy by the total amount of ice built. This yields a direct cost for on-peak ice consumption that can be added to the cost of compressor electricity consumption, when ice is consumed, to generate a total energy cost for each path. Finally. the peak energy consumption for each path is determined in order to determine the correct demand charge to be added to the total path cost.

5.4.2: Implementation in IBLAST

The FORTRAN subroutines required to implement the optimization of the ice storage compressor part load schedule in IBLAST are listed in Appendix B. The routines were designed to be compatible with the optimization routines for conventional central plant equipment. However, since conventional chillers can not be specified at the same time as thermal storage equipment this meant that, if necessary, the operation of the heating plant equipment could be optimized along with the thermal storage tank.

The purpose of the ICESIM subroutine was to manage the actual simulation of the thermal storage plant since it would now have to be called several times per time step. Warm-up days were handled differently than the actual days of the simulation since these are primarily used to smooth out the effects of initial conditions. A compressor part load fraction of 0.5 was used for warm-up days to ensure the building of some ice before the first simulation day and allow the calculation of a per kilowatt-hour cost for the use of ice.

The ICEAVRGH subroutine determined the amount of ice and energy consumed each hour of the simulation. In addition, ICEAVRGH picked the correct final tank capacity from the previous time step and fed it into the ice storage simulation as the initial tank capacity for the current time step. The number of possible storage tank states i.e., the quantity of ice in the tank, increased by a factor of two each hour because for each initial tank state there were two possible final states.

The purpose of INICEOPT1 and INICEOPT2 was to initialize and reset variables on a daily or hourly basis as required. INICEOPT3, called at the end of each day of the simulation, determined the compressor operating part load fractions for each hour of the next day's simulation. This was accomplished by reducing the interval between the previous two possible part load fractions by a fixed ratio. The new trial part load fractions for each hour were equidistant and on either side of the optimal part load ratio from the previous iteration. This guaranteed that the difference between the two trial compressor part load ratios would decrease from one iteration to the next until a converged solution was obtained.

Finally, ICESCHDOPT was the routine that determined the optimal compressor part load schedule for each hour from the simulation data. The first step was to calculate the per kilowatt-hour cost of ice depleted from storage. This required calculation of the total amount of ice built during the off-peak hours and the total amount of energy consumed by the compressor. These values allowed calculation of the cost of the compressor energy and the overnight change in the storage tank capacity. Off-peak cooling loads were added to the change in tank capacity because they are a load on the compressor and do not directly contribute to the cost of the stored ice. Finally, the per kilowatt-hour of cooling cost of stored ice was the result of dividing the compressor energy cost by the change in tank capacity plus off-peak loads. The second part of ICESCHDOPT calculated the cost of each compressor part load fraction schedule and saved the information pertinent to the schedule with the lowest overall cost accounting for the effects of ice consumption and peak demand. Finally, these results were reported.

Chapter 6: Testing and Validation

6.1: Objectives of Testing and Validation

Previous publications have validated the methods used in IBLAST: to update the zone conditions using a third order finite difference method (Taylor et al.,1990); to integrate realistic system models with the new zone energy balance calculations (Taylor et al., 1991); and to simulate the system-plant water loop with a realistic interface between water and air side conditions (Taylor et al., 1994). This research builds on these integration methods and adds a technique to simulate optimal control and scheduling of the central plant equipment. Therefore, testing and validation focused on verifying the results obtained from this method. Verification implies that realistic results were obtained over a considerable range of input parameters and different equipment selection such as would occur in practical applications. Comparisons of program output between the new IBLAST program with optimization and the existing IBLAST program, that retained the hardwired BLAST optimization method, help to ensure that the results of the optimization are reasonable.

In addition, it was desirable to show that optimal schedules obtained with the methods employed here satisfied the criteria for a global minimum energy cost. However, properly implemented this method should always result in the lowest cost path because the path is generated by searching on a subset of the feasible paths that, by definition, must contain the optimal path. Any cases in which no improvement over the standard IBLAST optimization algorithm is observed would be important to document and explain. In this context, improvement implies that the new optimization method is able to find a lower cost path to meet the building loads than optimizing based on the instantaneous performance of the plant equipment compared to the load. As has been noted, this type of optimization produces a lowest cost path only when demand charges for peak consumption are an insignificant fraction of the total energy cost.

6.2: Design of Test Cases

In order to evaluate the simulation techniques implemented in IBLAST, test cases were developed that exercised the plant optimization algorithms over a wide range of conditions and input parameters. This demonstrated the method's robustness as well as its usefulness for many of the situations encountered in building equipment design problems. All of the input files used for testing included the same building and fan system combination because the IBLAST simulation of the zone-air handler interactions has been substantially verified in earlier works (Taylor et al.,1990, Taylor et al., 1991, Taylor, Pedersen, and Lawrie, 1994). Instead, the central plant equipment and its performance was varied to illustrate how plant operation was affected by the optimization procedure, heating and cooling loads, and energy costs. A complete sample IBLAST input file is provided in Appendix C.

6.2.1: Building Envelope

The building envelope model used in the test series was a section of the Fort Monmouth Education Center at Fort Monmouth, New Jersey. The thermal performance of this building has been extensively documented through many cycles of testing with the BLAST program. The expected zone heating and cooling load trends were therefore well known and could be used for comparison with the output from IBLAST. The wing of the building modeled was divided into two zones as shown in the plan view in Figure 6-1.

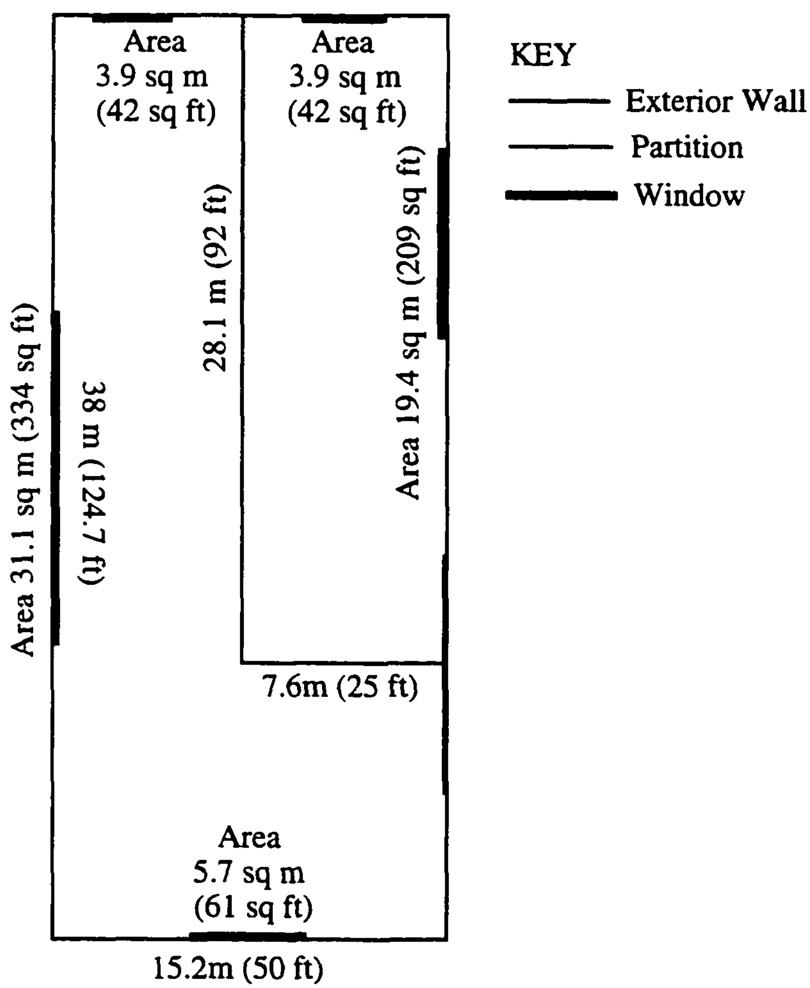


Figure 6-1: Plan view of Fort Monmouth building

The floor plan of the building wing is a rectangle with a floor area of 579 m^2 (6235 ft^2), and a wall height of 3.05 m (10 ft). The exterior walls of the building, those exposed to the outdoor environment are composed, from outside to inside, of the BLAST library elements A2, C8, IN3 and PL4. These are, respectively: 4 inches of dense face brick, 8 inches of heavy weight concrete, 6 inches of fibrous mineral fiber, and a 5/8 inch layer of plaster. Interior walls, or partitions, and internal masses consist of PL4, B3, and PL4, where B3 is 2 inches of insulation. The roof of each zone is made up of BLAST library elements RF4, IN71, and BB17, that are: 3/8 inches of built-up roofing, preformed roof insulation, and 3/4 inches of acoustic tile. Finally, the zone floors are 1/16 inches of tile flooring FF5 laid on 4 inches of concrete CO17, on top of 12 inches of earth, DIRT 12 IN. Both building zones have significant areas of double pane glazing, as indicated in the figure, and some surfaces also have hollow wood doors, though their location is not indicated. Internal loads, such as: people, lights, other electrical equipment, and infiltration, were used to provide additional load variations for each zone . These loads, shown in Figure 6-2, are the result of electrical equipment and people, who generate a sensible and a latent load, within each zone. The scheduled loads profile is intended to represent internal loads for a typical office building.

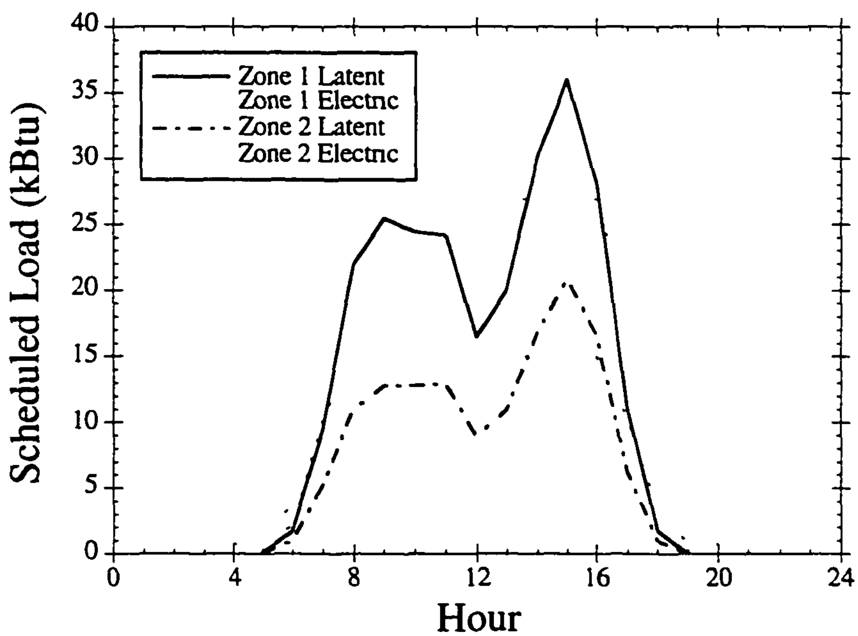


Figure 6-2: Zone 1 and 2 Scheduled Loads

The control statement in each zone description is used to set the temperature setpoints that the system tries to maintain when the zone is heated or cooled. The selected control profile, NWS2 (night and weekend setback 2) provides heating or cooling as necessary during the day and allows the zone temperature to float over a much wider temperature band at night. This is fairly typical for a building that is occupied during day time hours and has minimum occupancy the remainder of the time. When cooling is required between 8am and 6pm, the NWS2 schedule sets the desired zone temperature at 25.6°C (78°F). Outside of this time frame, the zones are in setback mode allowing their air temperatures to float. In heating mode, the desired zone temperature is 20°C (68°F) but the system output is controlled over a range from 19.4°C (67°F) to 20.6°C (69°F). In setback mode, the control schedule is set so that the system controls the zones between 15.6°C (60°F) and 16.7°C (62°F).

6.2.2: Air Handling System

The two zones were each served by a single zone draw through system. This system was selected because it is typical of the systems found in many buildings, especially residential units and older commercial buildings. Newer buildings typically use variable volume systems that, when operated correctly, can be far more efficient. However, the single zone draw through system in IBLAST provides precise zone temperature control by adjusting the amount of time that the system is allowed to run to meet the heating or cooling loads. Furthermore, each single zone draw through system is an independent unit so, depending on the building loads, a plant serving multiple systems could be required to provide both hot and chilled water simultaneously. This circumstance is quite common in commercial buildings in which the interior zones are dominated by internal loads and the exterior zones are dominated by energy exchange with the building's outside environment. Frequently, in such buildings, cooling is required year round in the interior zones while in exterior zones cooling is required in the summer and heating in the winter as would be expected.

In the sample input file, the system was sized to supply a maximum of 1.79 m^3/sec (3800 cfm) to zone 1 and 1.18 m^3/sec (2500 cfm) to zone 2. The heating coils were specified using the "simple" input syntax that required only an overall UA value and a desired inlet water temperature. The system cooling coil geometry was specified using the default input parameters. These were found to be adequate to provide the desired coil outlet temperature at the maximum system air flow rate The desired coil outlet air temperature was fixed by the coil control strategy to 12.8°C (55°F). However, this temperature could only be maintained as long as the plant capacity was sufficient to meet system demands.

6.2.3: Central Plants

The primary focus of testing was to evaluate how the algorithms for simulation of optimally controlled and scheduled central plant equipment work under a variety of realistic situations.

Ideally, these situations would be representative of building design practice if not based on an actual building. Each test case was selected to exercise a different feature of the optimization scheme and, consequently, a case by case evaluation of the results was required. However, several issues common to all the test cases were addressed to evaluate the optimization procedure for use as a generalized method in building simulation. The following were global concerns on which all test cases were evaluated: computation time, computational effort, accuracy and consistency of results.

Computation time was a significant concern since there is a trade-off between the additional time required to perform a calculation and the benefits that will offset that additional time compared to a more basic computation. Computational effort, in this instance, refers to the hardware and software resources required to perform the optimization calculations. As has been noted, the path optimization method used was a search that depended on the optimal path being contained in a small and easily identifiable subset of all the possible paths. This was necessary because the total number of paths rapidly becomes unmanageable as the number of feasible transitions between steps and the number of steps in the path increases. In order for this method to be useful, the number of paths to be evaluated by the exhaustive search must be reasonable for common input scenarios. The remaining item to be evaluated, though certainly no less important than the previous concerns, was the validity of the optimization results. Since the central plant models remained essentially unchanged from those installed in BLAST, the results of optimizing instantaneous plant performance with the new algorithm should be consistent with results obtained from the BLAST method. Since BLAST assumes that equipment of a given type operates with the same part load performance curves regardless of equipment size the optimal solution is to run all the equipment at the same part load fraction. However, because BLAST operates plant equipment types according to fixed scheduling priorities, the IBLAST optimization scheme would not necessarily give the same results.

6.2.3.1: Multiple Equipment of the Same Size and Type

A common scenario in building central plant design is to use multiple pieces of equipment sized so that the total heating and cooling capacities of the plant can meet the peak anticipated loads. This approach results in the plant equipment being operated close to its best part load fraction for a more time than if the plant consisted of a single piece of equipment of equivalent capacity. The disadvantage of this approach, as has been noted, is that surges in power consumption occur when equipment is turned on. However, this is a concern for electric equipment only. Several test cases were developed for IBLAST using 3, 4, and 5 electric powered chillers and one boiler. The total chilling capacity of the plant was also varied for performing simulations with the plant sized close to the peak load and significantly in excess of the peak load. In this case, boilers were required to supply domestic hot water to the building since no usable waste heat was generated by the chillers. The effects of primary interest were: change in optimal equipment usage as the demand charge was varied and change in optimal equipment usage as the relative cost of on-peak electric consumption was varied.

6.2.3.2: Multiple Equipment of the Same Type Using Different Sizes

Another common scenario in central plant design practice is to use multiple pieces of equipment of the same type but different sizes. Again the goal of this approach is to schedule the individual chillers and boilers to operate near their best part load ratio for most of the time One way to operate this type of plant would be to use the largest chiller to meet the "base" load and use the additional smaller chillers to supplement the base chiller capacity as required. However, if the equipment is allowed to cycle on and off freely to most efficiently meet the load, spikes in consumption due to equipment start-up transients will be created during the on-peak hours. This is undesirable when the electric utility is keeping track of peak electric consumption for the purpose of assessing demand charges. Therefore, as in the previous cases, the effects of demand charges and on-peak electricity rates on the lowest cost chiller operating schedule were observed by varying the peak rate multiplier and the demand charge. The input file developed to test the optimization algorithm with multiple pieces of equipment of different sizes, had a combination of 2 large chillers, 3 small chillers and 1 boiler to supply domestic hot water.

6.2.3.3: Multiple Equipment of Different Types

A variation of the previous case would have a chiller or boiler using one energy type be supplemented by equipment using a different energy type. This could be beneficial when an electric chiller meets the base building cooling loads but a diesel chiller adds cooling capacity during the on-peak hours to cap peak electric consumption and eliminate equipment start-up transients. The effectiveness of this strategy depends on: the ratio of diesel fuel cost to electricity cost per kilowatt-hour of cooling provided by the chiller plant and the demand charge. The chiller plant for this test case had one large electric chiller to meet the base building cooling loads, two smaller electric chillers, and one diesel chiller. The optimal equipment schedule was determined for several variations in the cost of each energy type, the on-peak cost of electricity, and the demand charge. In this example, interactions between the heating and cooling plant can occur since waste heat from the diesel chiller provides energy to produce domestic hot water load. Therefore, the effect of using different boiler energy sources, fuel or electricity, was also considered.

6.2.3.4: Indirect Ice Storage Systems

The effect of having a storage system in a building's cooling plant is to shift part of the energy use required to cool the building to take advantage of cheaper off-peak energy rates. However, heat gains to the storage tank and the fact that a chiller is less efficient when producing brine at a temperature between -22°C and -13°C than chilled water between 4°C and 7°C, cause thermal storage systems to consume more total energy per kilowatt-hour of cooling than conventional plants. Therefore, the utility rate structure is an important factor in determining how much of the on-peak cooling load should be met by melting stored ice.

The effect of the utility rate structure was evaluated for an ice container system by varying independently both the ratio of the on-peak electric rate to the off-peak rate and the demand charge for the peak electricity consumption rate. In addition, the size of both the storage tank and the compressor was varied. The storage tank size was expected to have the smallest effect on the optimal scheduling of the compressor, whereas the size of the compressor was expected to be important to scheduling because its efficiency is a strong function of part load ratio. In order to evaluate the effect of these two quantities the compressor size was varied between being large enough to refill the storage tank overnight, but not being able to meet the peak cooling load, and being large enough to meet all the on-peak and off-peak cooling loads. The storage tank size was varied between being very oversized and being so small that the ice in the tank was almost exhausted by the end of the day.

Chapter 7: Results

7.1: General Observations

The results presented in the following sections document the use of the methods developed in this work. However, before describing these results individually, some general observations on all the results are appropriate. These relate specifically to the performance of the optimization routines and interpretation of the results. Results are presented describing the thermal performance of the building zones and systems. However, the integration of the BLAST zone, system and plant simulations, was not evaluated in great detail because this has been covered extensively in previous work by Taylor et al (1990, 1991, 1994) and Metcalf et al. (1995).

One aspect of the results that was considered in detail was how the GRG routine optimized the performance of a specified equipment combination. That is, what were the resulting optimal part load fractions for each piece of equipment in the central plant. The BLAST plant optimization algorithm assumes that when several pieces of equipment of the same type, but not necessarily the same size are operated, they will run at the same part load ratio. This characteristic was duplicated by the results obtained from the GRG solver indicating that the BLAST method is a reasonable approach when equipment part load performance is independent of size. This occurs because chiller coefficient of performance is a quadratic function of part load ratio implying maximum coefficient of performance is obtained for a unique part load fraction. Combinations of chillers behave the same way because the overall COP is still a quadratic function of part load fraction. As an example, consider two identical chillers, with an optimal part load fraction of 0.65, operating at an actual part load fraction of 0.5 . If the part load fraction of one chiller is increased to 0.65 its efficiency improves and the cost per unit cooling provided by that chiller decreases. However, to provide the same total amount of cooling the other chiller must reduce its part load ratio to 0.35 with a corresponding decrease in efficiency. The quadratic nature of chiller performance curves means that the increase in cost per unit cooling for the chiller operating at 35% of capacity than offset the decrease in cost obtained by operating the other chiller at 65% of capacity.

When equipment using different performance curves were mixed, such as chillers and diesel chillers, the same part load ratio was not obtained for each equipment type. Part load ratios were only constant within equipment type. However, because the objective function included energy cost factors, equipment was not always used in the same order of preference as it would have been in BLAST. The fraction of the load assigned to each chiller was determined by the relative costs of the different types of energy. When, in the simulation, electricity was made very expensive compared to diesel fuel, diesel chiller usage would be favored over electric powered chiller usage. The reverse was also true.

An important distinction, in this discussion, is the difference between the minimum cost and the minimum energy schedules. The equipment schedule optimization routine first calculates a minimum cost schedule ignoring demand charges. This will subsequently be referred to as the minimum energy schedule because, when the plant uses only one type of energy, this path consumes the minimum amount of energy that will allow the loads to be met. However, when different types of energy are used by the plant this path does not necessarily consume the least energy. On the other hand, the minimum cost schedule is the equipment schedule that results in the least energy cost when the demand charge is taken into account. The two schedules are the same when the optimizer is unable to find an equipment schedule with a peak power consumption low enough to offset the additional cost due to increased total energy consumption.

A second observation concerns the number of equipment schedules that had to be calculated for conventional (not thermal storage) equipment. The method adopted in this work was a limited, targeted search of the hourly equipment schedules based on the necessary criteria for an improvement in energy cost. The success of the method depended on reducing the number of equipment schedules calculated to no more than a few thousand. This requirement would allow a maximum of four feasible equipment combinations for each hour of a six hour period. However, by also considering the feasible ways to transition from one hour to the next, the total number of equipment hourly schedules calculated rarely exceeded the 5000 allowed by the program. This was true, even when many more than four feasible combinations existed at each hour. Typically, the number of paths was on the order of 10 to 100.

Finally, the results of the zones and systems loads calculation are presented as they are related to all subsequent results. These plots give the temperature and cooling load variations occurring in each zone during the simulation, and the change in system chilled water demand. Figure 7-1 shows the diurnal variation in outside dry bulb temperature and the temperature profiles for zones one and two. Each single zone draw through system controls the zone temperature effectively from hour 9 (8am to 9am) when the system was turned on, until hour 17 when the system was turned off again. Figure 7-2 shows the cooling load profile for each zone. The dramatic spike in cooling load during the first hour was due to the system having to cool each zone to the desired temperature and overcome the zone internal loads. The dip in cooling load around hour 12 is due to a reduction in internal loads at that time. Finally, in Figure 7-3, the chilled water consumption passed to the central plant by each system closely follows the trend in zone cooling load.

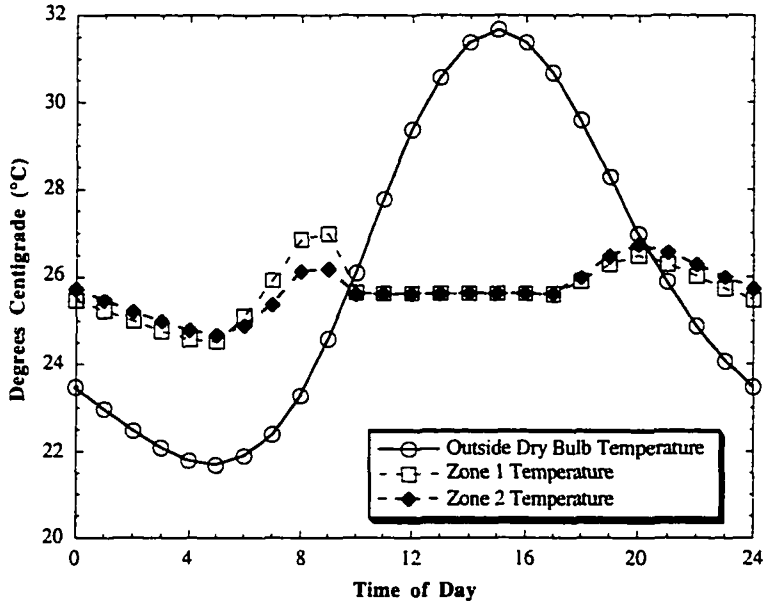


Figure 7-1: Calculated Zone Temperature Profiles For a Summer Design Day

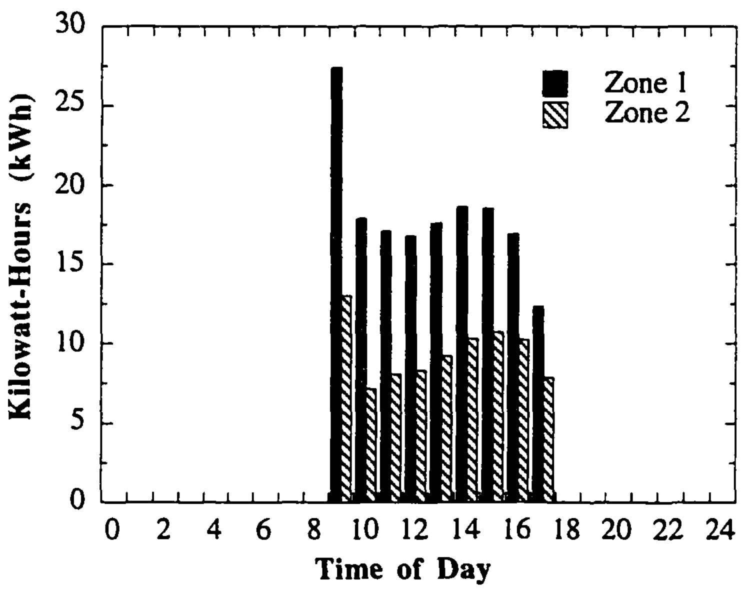


Figure 7-2: Calculated Zone Cooling Load Profiles

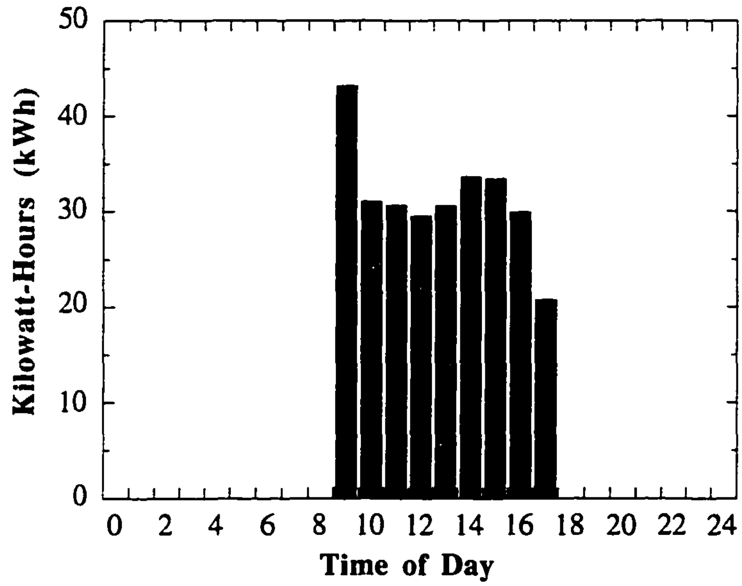


Figure 7-3: Calculated Chilled Water Demand For a Summer Design Day

7.2: Multiple Equipment of the Same Size and Type

The purpose of this series of runs was to observe how the minimum cost operating schedule of a central plant is affected when the number of equipment components and their individual and total capacities are varied. Three cases are presented for central plants having 3, 4, and 5 identical chillers. The total chiller plant capacity was greater than or equal to the anticipated chilled water demand and divided equally among the plant components. The boiler was sized at 51 kW for the maximum heating load but never operated at a very high part load fraction because the only heating load, from the domestic hot water, was small. Thus boiler scheduling was not an important factor in the plant optimal scheduling since the electric chillers specified in these plants produced no waste heat. The boiler had to turn on when a domestic hot water load, however small, was present.

Figure 7-4 represents the minimum cost equipment schedule for three 26kW chillers and shows the total chiller and boiler plant operating capacity for each hour of the day. In this case the minimum cost schedule and minimum energy schedule are identical. All three chillers operated for each hour of the day that a cooling load existed. The reason this schedule was optimal became clearer upon examination of Figure 7-5 that shows the amount of energy actually consumed by the central plant to meet the hot water demand and the system cooling loads. The peak energy

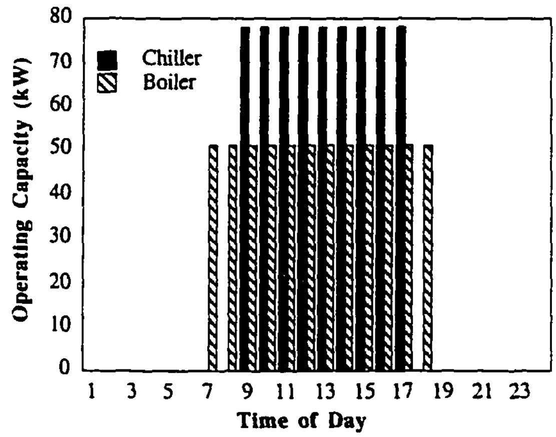


Figure 7-4: Optimal Chiller and Boiler Operating Schedule for Three 25kWChillers and One 150kW Boiler

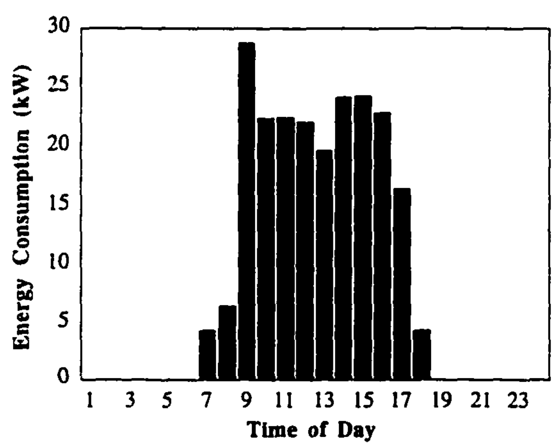


Figure 7-5: Hour-by-Hour Plant Electric Consumption for the Optimal Schedule

consumption actually occurred the first hour that cooling was required in the morning. A second lower peak occurred in the afternoon but the load variation was not large. Since the cooling load was close to the chiller plant capacity most of the time, all the chillers were necessary to meet the cooling loads.

Figure 7-5 shows the total plant electric consumption for each hour of the day. This includes the boiler's energy use that shows up explicitly during hours 7, 8, and 9 when, with no cooling load, no chillers were required to be turned on.

Figure 7-6 is equivalent to Figure 7-4 for a chiller plant consisting of four 30kW chillers. In this case, all the chillers were turned on during hour 9 when the day's peak cooling load occurs. After hour 9, and until hour 16, three out of four chillers were operated with a total cooling capacity of 89kW. Finally, during hour 17 only two chillers were required to meet the cooling loads. Given the results of the plant with three chillers, this makes sense because the actual operating capacity of each plant during the on-peak hours would be very similar. If all four chillers were allowed to operate for the entire day the peak energy consumption by the plant would increase, and the operating part load ratio of four 30kW chillers would be well below the optimum value of 0.65.

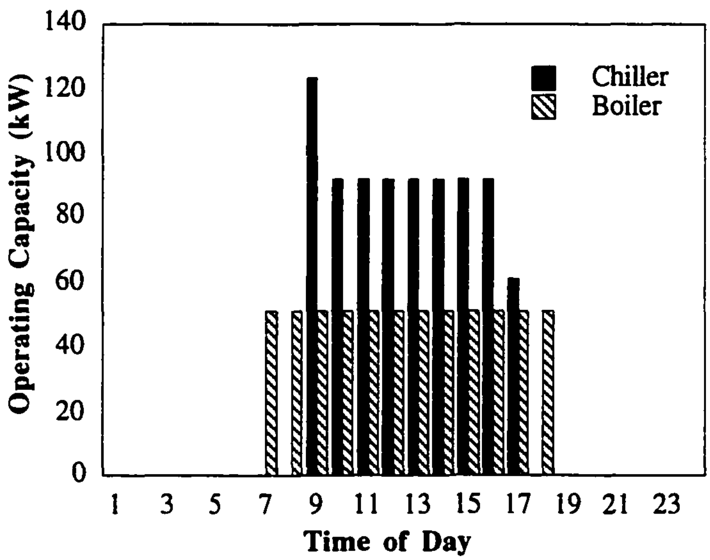


Figure 7-6: Optimal Chiller and Boiler Operating Schedule for four 30kW Chillers and 1 Boiler

Finally, a case with five 22kW chillers, giving a maximum capacity of 110 kW, was tested. The results, shown in Figure 7-7, indicate the same trends seen in Figure 7-6. All the chillers were required to be turned on for the first hour of the day that the building had a cooling load. Subsequently, one was turned off and 4 chillers operated until hour 17, when only 3 chillers were required. Interestingly, the chillers operating between hours 10 and 16 had a total capacity of 88kW so that the plant operated at essentially the same part load ratio as the three 30kW chillers in the previous case. This consistency of results makes sense since the part load performance of each chiller was identical and verified that the GRG solver was correctly finding the optimum chiller part load fractions when similar equipment was used.

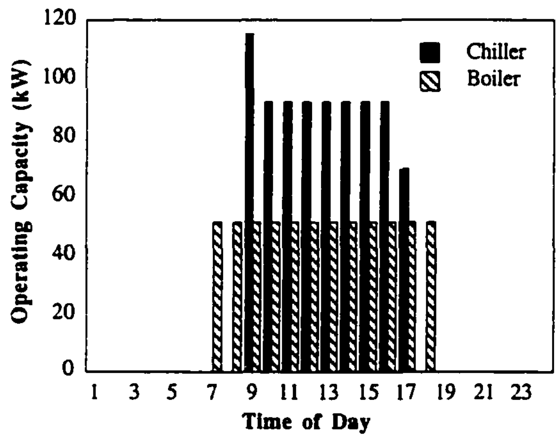


Figure 7-7: Optimal Chiller and Boiler Operating Schedule for five 22kW Chillers and 1 Boiler

7.3: Multiple Equipment of the Same Type Using Different Sizes

This section considers the results obtained from a chiller plant in which the chillers are of the same type but of several sizes. This is a fairly typical situation in practice as it allows the plant to meet base cooling loads efficiently while the additional capacity is used to meet peak cooling loads. The following results were generated by simulating a chiller plant with two 30kW chillers and three 14kW chillers. The on-peak cost of electricity and the demand charge were both varied and the effect on the least cost equipment schedule, as compared to the minimum energy schedule, noted.

In the first set of results, presented in Figures 7-8 and 7-9, the effect of varying the demand charge was observed by running the simulation for several values between 0/kW and 200/kW. The on-peak electricity rate multiplier was set to 2, doubling the cost of on-peak energy consumption compared to off-peak energy consumption. Figure 7-8 shows the difference between the minimum energy and minimum cost chiller schedules for a demand charge of 200/kW. The minimum energy path shows that, to maintain an optimal chiller part load fraction, an additional

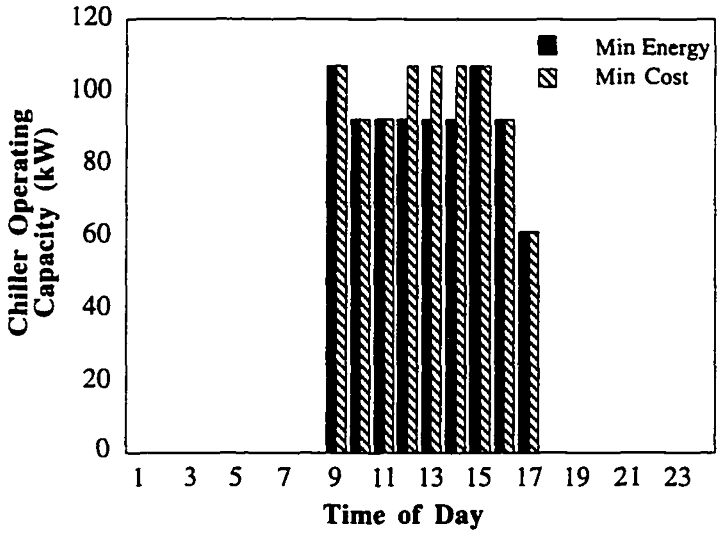


Figure 7-8: Difference Between Minimum Energy and Minimum Cost Chiller Operating Capacity for a Demand Charge = 200/kW

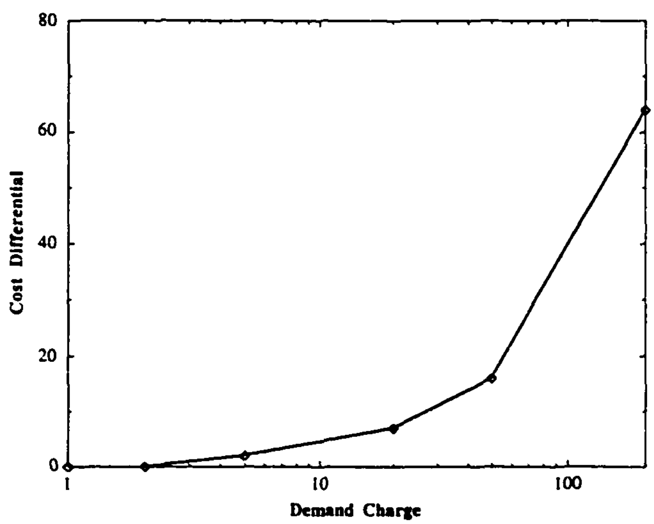


Figure 7-9: Difference Between Minimum Cost and Minimum Energy Paths as Related to the Utility Demand Charge on Peak Electricity Use

14kW chiller was required to meet the building cooling load during hour 15. However, the minimum cost path turned on the extra chiller at hour 12 to avoid an on-peak start-up transient. In the minimum energy schedule, this transient is responsible for the day's peak energy consumption and is avoided by the minimum cost schedule.

Figure 7-9 shows the difference in cost between the minimum cost and minimum energy paths as the demand charge is increased. When the demand charge is small the minimum cost schedule is also the minimum energy schedule and the cost differential is zero, as indicated. This occurs because the increased energy consumption associated with the minimum cost path, shown in Figure 7-8, results in an increase in cost over the minimum energy schedule. When the demand charge is high, the reduction in peak demand causes a reduction in demand charge costs that more than offsets the increase in cost of the additional energy consumed. But when the demand charge is low, this offset does not occur. In Figure 7-9, therefore, when the cost differential is zero the lowest cost plant schedule is the minimum energy schedule from Figure 7-8. When a cost differential exists, the minimum energy schedule is more expensive than the minimum cost path and lowest energy costs are obtained by following the minimum cost schedule shown in Figure 7-8.

A second set of simulations was run with no demand charge but with variations in the peak rate multiplier. In these cases there was no difference in cost between the minimum energy and minimum cost paths. This result was obtained because the peak rate multiplier is a charge on total, not peak, energy consumption and, since the minimum cost path in Figure 7-8 always consumed more energy than the minimum energy path, the minimum cost schedule was also the minimum energy schedule.

7.4: Multiple Equipment of Different Types

The next case considered a central plant composed of equipment of different sizes and powered by different energy sources. Specifically, the central plant consisted of two 17kW electric powered chillers, one 44kW electric powered chiller, one 23kW diesel powered chiller, and one 5kW electric boiler to supply domestic hot water. The influences of the relative cost of each type of energy and the demand charge on the least cost equipment schedule were of most interest. However, this case also demonstrated how the GRG routine optimized equipment with different part load performance curves.

In the first set of examples, the demand charge was 200/kW and the on-peak rate multiplier was 2. Diesel cost was given values of 1.5/kWh, 3/kWh, 5/kWh, and 10/kWh to observe how the minimum cost schedule changed. Figure 7-10 shows the minimum energy equipment schedule when the cost of diesel fuel is 1.5/kWh. This schedule does not utilize the diesel chiller at all and the electric boiler is used to supply domestic hot water. Figure 7-11, the minimum cost schedule, provides a significant contrast since the diesel chiller operates during hours 13 through 17 and the boiler does not run because waste heat the diesel chiller meets the domestic hot water load. However, during hour 12 the diesel chiller produced insufficient waste heat from and the boiler operated at a minimal part load fraction. During hour 14, a transition results in all the chillers operating and much more capacity is available in this case than the for the minimum energy schedule. This situation results because the peak electric consumption is a steady state peak and is not due to an equipment start-up transient. Further analysis indicated that turning on the diesel chiller shifted the peak electric consumption to hour 12, in the minimum cost schedule, from hour 14, in the minimum energy schedule. In addition, a significant reduction in peak electric consumption occurred because the diesel chiller allowed the electric powered chillers to operate at lower part load fractions. However, the method of calculating the start-up transient has a significant effect on the final optimal schedule. If the increase in consumption over the eventual steady state value were 50% instead of 20% the change in operating equipment between hour 13

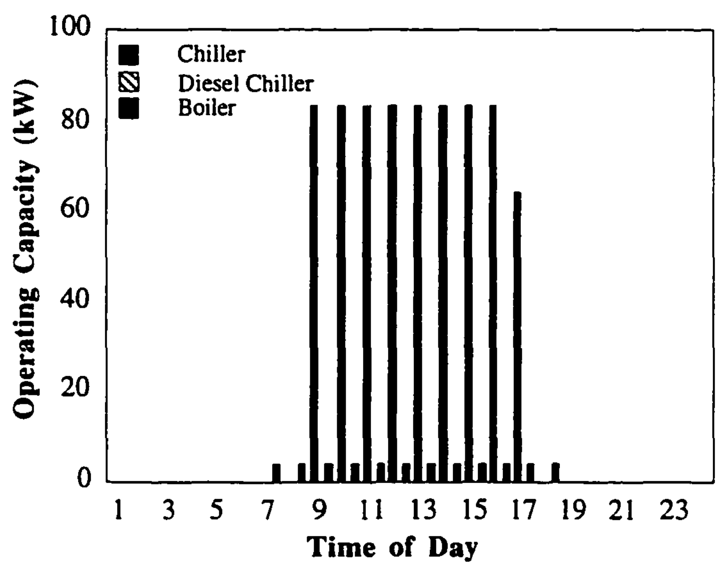


Figure 7-10: Minimum "Energy" Plant Operating Schedule for a Combination of Electric and Diesel Chillers and an Electric Boiler (Diesel Cost = 1.5/kWh)

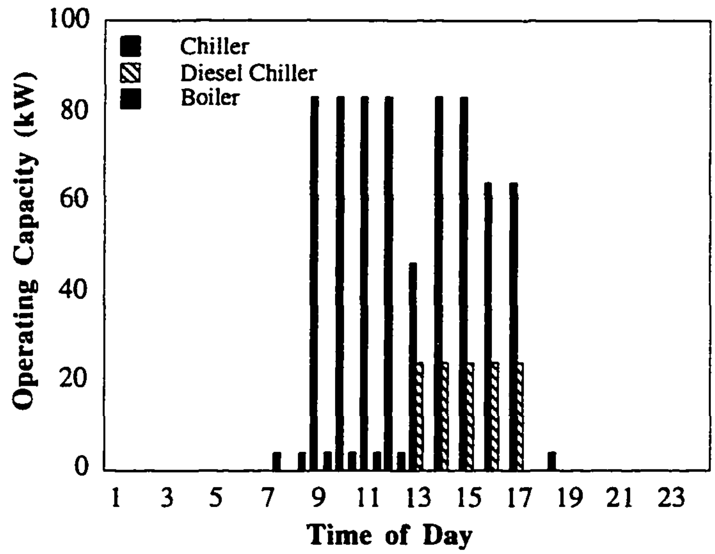


Figure 7-11: Minimum Cost Plant Operating Schedule for a Combination of Electric and Diesel Chillers and an Electric Boiler (Diesel Cost = 1.5/kWh)

and hour 14 might not have occurred. Figures 7-12 and 7-13 show, respectively, the hour-by-hour difference in total energy consumption and total energy cost for the minimum energy and minimum cost schedules represented in Figures 10 and 11. The change in the least cost schedule is a result of the demand charge, because the total cost of the energy consumed by the minimum energy path was lower than the total energy cost of the minimum cost path when peak demand was not considered.

Figure 7-14 shows the effect increasing the cost of diesel fuel has on the difference between the minimum cost and minimum energy schedules. When diesel costs less than about 5.5/kWh the minimum cost path uses the diesel chiller during the on-peak hours, as shown in Figure 7-11. However, when the cost of diesel fuel is greater than this value, the total cost, including demand charge, for using all electric chillers becomes less than for using a combination of diesel and electric chillers, even with the reduction in peak electricity demand.

In another example the electric boiler was replaced with a fuel fired boiler of the same capacity. In addition, a diesel fuel cost of 2.0/kWh and a boiler fuel cost of 0.5/kWh were specified. Figure 7-15 shows that the minimum energy equipment schedule for this case is identical to the plant schedule shown in Figure 7-10. However, the minimum cost plant schedule, shown in Figure 7-16, is significantly different from the corresponding figure of the previous example, Figure 7-11.

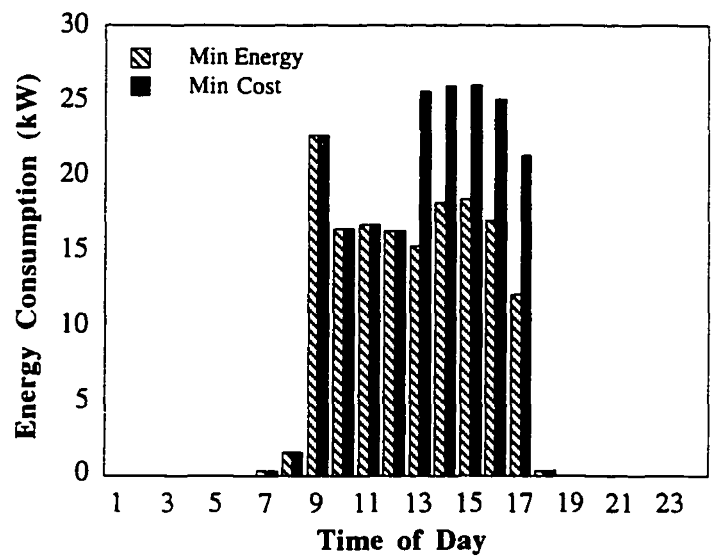


Figure 7-12: Difference in Actual Energy Consumption Between Minimum Energy and Minimum Cost Schedules (Diesel Cost = 1.5/kWh)

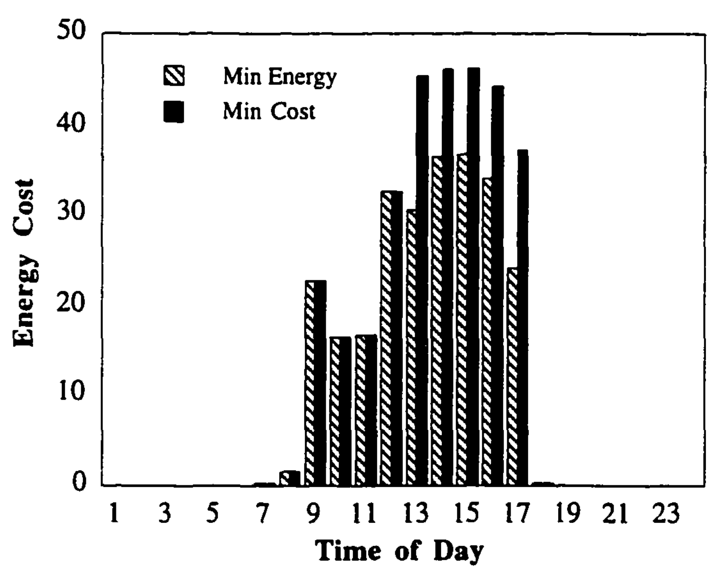


Figure 7-13: Difference in Actual Energy Cost Between Minimum Energy and Minimum Cost Schedules Neglecting Demand Charges (Diesel Cost = 1.5/kWh)

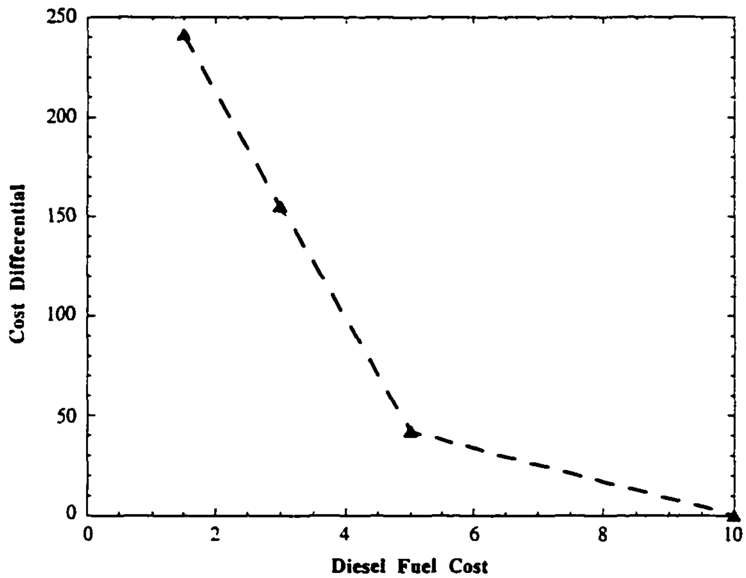


Figure 7-14: Total Cost Differential Between Minimum "Energy" and Minimum Cost Schedules as Diesel Fuel Cost is Increased

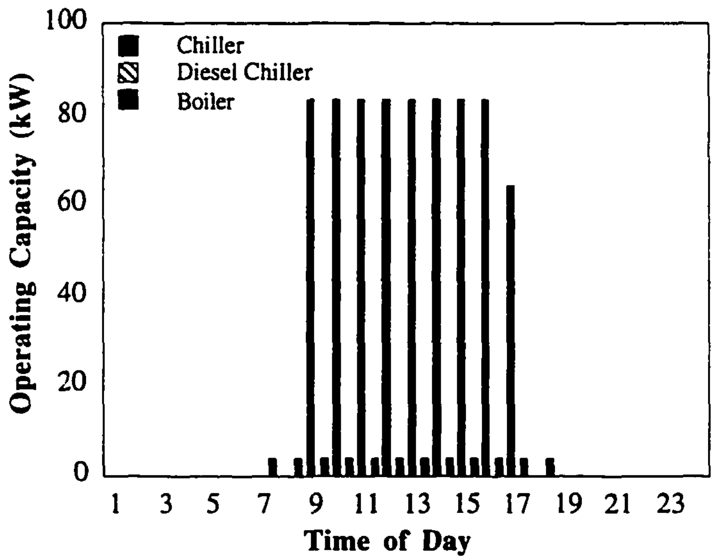


Figure 7-15: Minimum "Energy" Plant Operating Schedule for a Combination of Electric and Diesel Chillers and a Fuel Fired Boiler (Diesel Cost = 2.0/kWh)

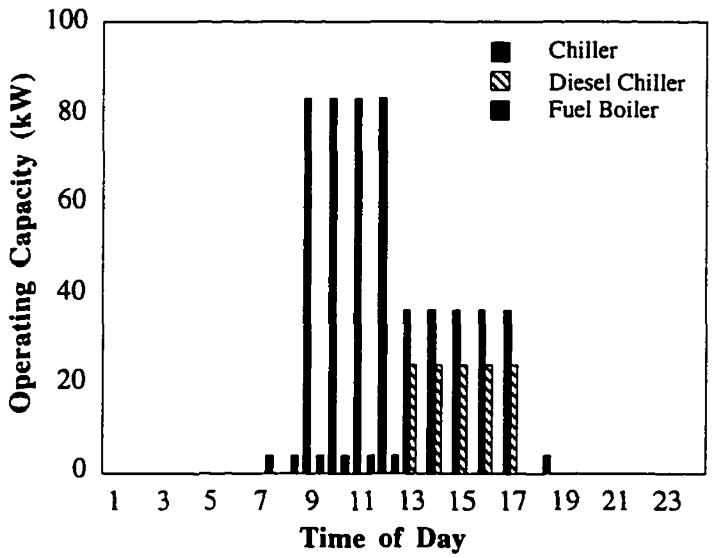


Figure 7-16: Minimum Cost Plant Operating Schedule for a Combination of Electric and Diesel Chillers and a Fuel Fired Boiler (Diesel Cost = 2.0/kWh)

The minimum cost path in this example still utilized the fuel boiler for hour 12 but the remaining electric chiller was not turned on at the end of hour 13, as in the previous case. In the previous example, the peak electric consumption occurred at hour 12 for minimum cost path and the electric boiler was turned on, contributing to the peak electricity load. In this example, the boiler is fuel fired so that, although the peak electric consumption also occurs at hour 12, it is a lower peak. Turning on the chiller at hour 14, as in the previous example, would increase peak electric consumption and move the time at which it occurs to hour 14. Not turning on the additional chiller and operating all the equipment at overall higher part load fractions to meet the cooling load avoided this peak in consumption.

The equipment from the previous example was also used to evaluate the sensitivity of the minimum energy and minimum cost equipment schedules to variations in demand charge and on-peak electricity rate multiplier. The demand charge was varied from 200/kW to 1/kW for a fixed value of the on-peak rate multiplier and four on-peak electric rate multiplier values were used: 2, 4, 6, and 10. The results are given in Figure 7-17, showing the additional cost of the minimum energy schedule compared to the minimum cost plant operating schedule.

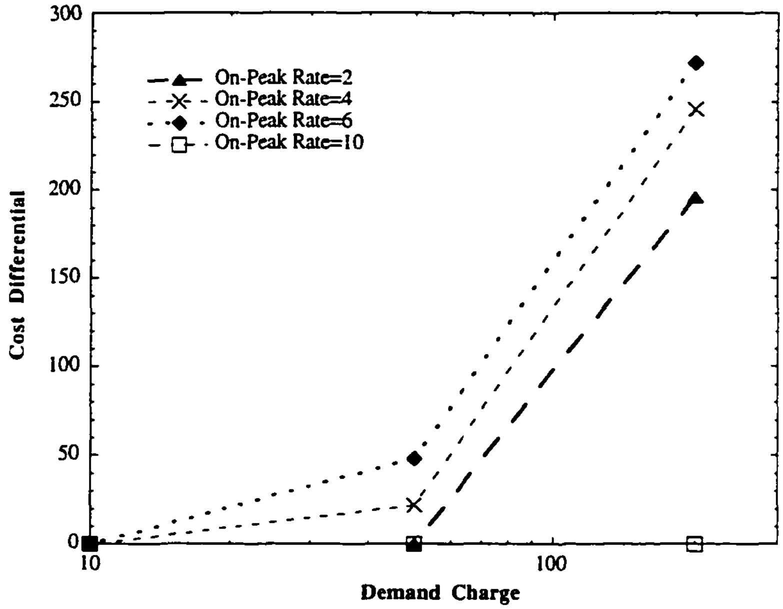


Figure 7-17: Effect of Demand Charge and On-Peak Electric Rate On the Difference Between Minimum Energy and Minimum Cost Schedules (Diesel Cost = 2.0/kWh)

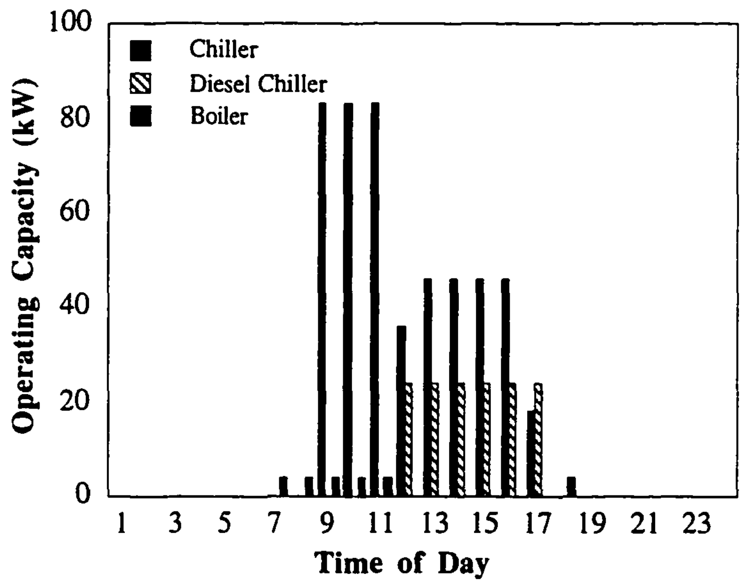


Figure 7-18: Optimal Plant Equipment Schedule when On-Peak Electricity = 10/kWh, Demand Charge = 200/kW and Diesel = 2.0/kWh

As the demand charge was increased, in all cases except for an on-peak rate multiplier of 10, the difference in cost between the two schedules increased. The minimum energy and minimum cost paths also converged when the demand charge was reduced to less than 10/kW. These results matched those obtained in previous cases except when the on-peak electric rate multiplier was 10. An example of this case is plotted in Figure 7-18 for a demand charge of 200/kW, showing the on-peak use of diesel chillers because, per unit cooling, electricity was more expensive than diesel fuel. Another effect was the raising of the diesel chiller part load ratio at hour 12 so that the boiler was not required to supply any of the domestic hot water load.

7.5: Indirect Ice Storage Systems

The examples presented in this section illustrate how the optimal use of cold thermal storage is affected by four variables: on-peak electricity cost, demand charge, chilling plant capacity, and storage tank capacity. The primary motivation behind the use of thermal storage systems in practice is to level loads on the cooling plant by shifting portions of the load to off-peak hours when electricity is cheaper and demand charges are not imposed. The following examples show the best way to operate the chilling plant compressor by optimizing its part load ratio hour-by-hour under a variety of energy cost and equipment size combinations. One immediate observation seen in all the examples, was an essentially constant compressor part load fraction over the optimization period. This result reflects the fact that the optimal way for a compressor to provide a fixed amount of cooling is for it to operate at a constant part load fraction. This is true, even if that fraction is not the compressor's optimal part load fraction, because a deviation from a constant part load ratio might improve efficiency at one hour but would reduce it at another enough to offset the initial improvement. Thus, in the majority of the cases presented below, cold thermal storage is used to level the on-peak load on the compressor.

The first example showed the effects of varying the on-peak electricity cost without a demand charge. The thermal storage system consisted of a 440kWh storage tank and a 44kW compressor to charge the tank and meet other cooling loads. The system was simulated and optimized for four values of on-peak electricity rate multiplier: 1, 2, 4, and 10. The results are presented in Figures 7-19 and 7-20. Figure 7-19 shows the hour-by-hour fractional reduction in tank capacity as the on-peak rate multiplier is increased. As might be expected, as the cost of on-peak electricity increased, more of the cooling load was shifted to the storage tank from the compressor. This is seen directly from a corresponding reduction in compressor part load fraction. However, the transition was gradual, indicating that the cost of ice per kilowatt-hour of cooling was not constant, but increased as more ice was built. In addition, the optimization algorithm seemed to find a local minimum, at small compressor part load fractions, where a decrease in the

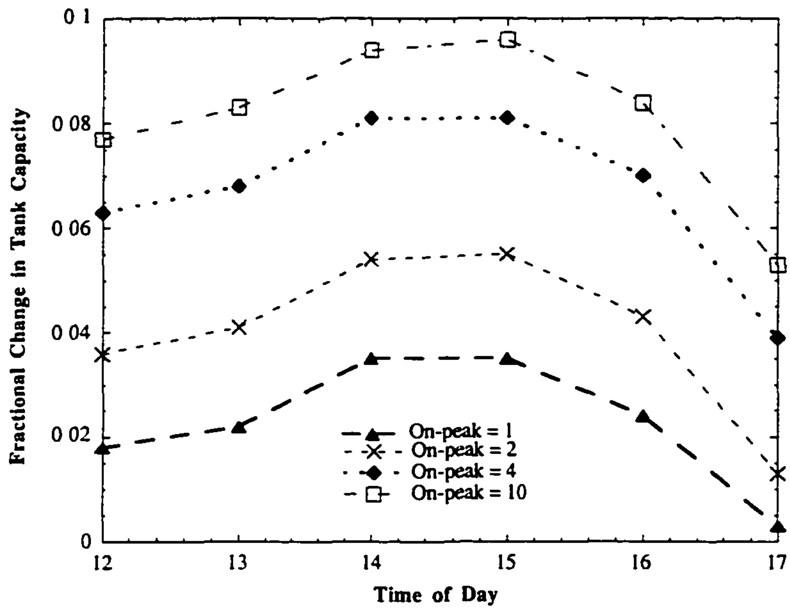


Figure 7-19: Hourly Variation in Consumption of Stored Ice for Four Values of On-Peak Rate Multiplier

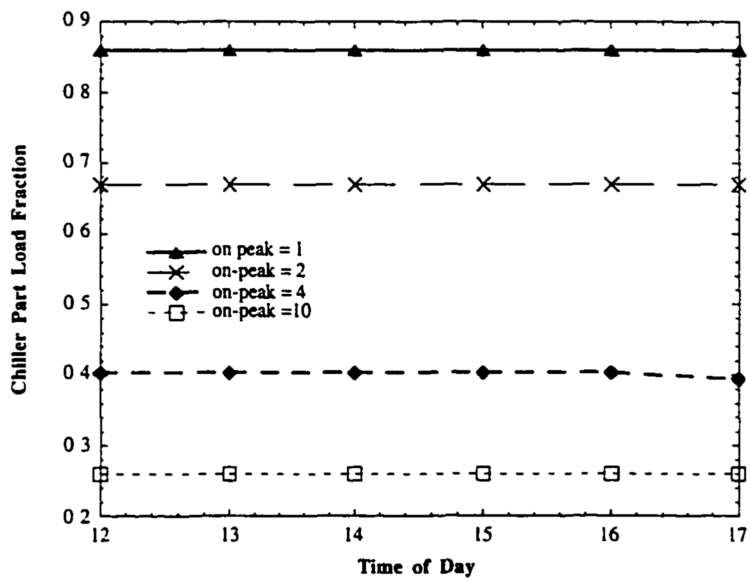


Figure 7-20: Hourly Variation in Chiller Part Load Ratio for Four Values of On-Peak Rate Multiplier

part load ratio resulted in an increase in compressor energy consumption. This, in combination with an increase in the stored energy required, drove the optimizer in the wrong direction. This problem could be overcome by simulating, as a separate case, the ice storage system using "full storage" control. Comparison of the two sets of results would determine the schedule giving the true global minimum energy cost.

Figures 7-21 and 7-22 show results obtained from the same central plant as the previous ice storage case, when the demand charge was varied and the on-peak rate multiplier was given a constant value of 2. Increasing the demand charge had similar effect to increasing the on-peak electricity rate; as demand charge increased a greater emphasis was placed on ice consumption rather than direct cooling.

A third example shows what effect a change in compressor capacity has on the optimal compressor part load fraction. Three compressor sizes, 66kW, 44kW, and 29kW, were selected, the on-peak rate multiplier was 2.0, and the demand charge was 0.0. The results are shown in Figure 7-23.

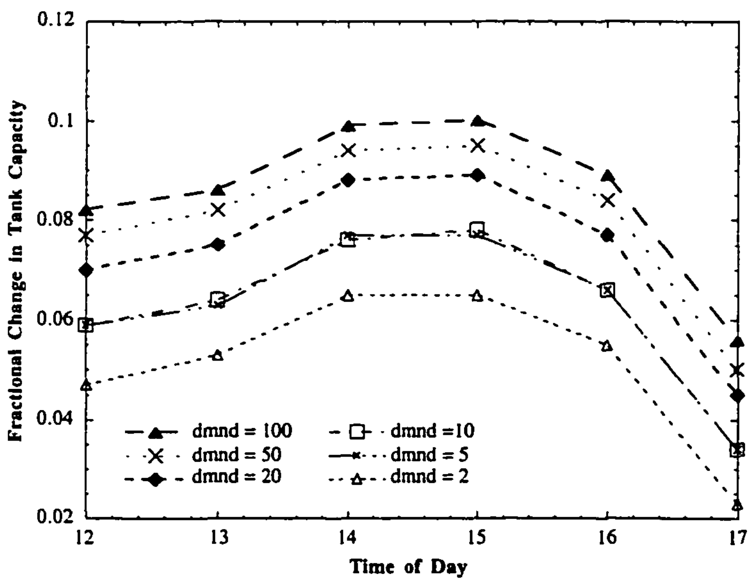


Figure 7-21: Hourly Variation in Consumption of Stored Ice for Six Values of Electricity Demand Charge

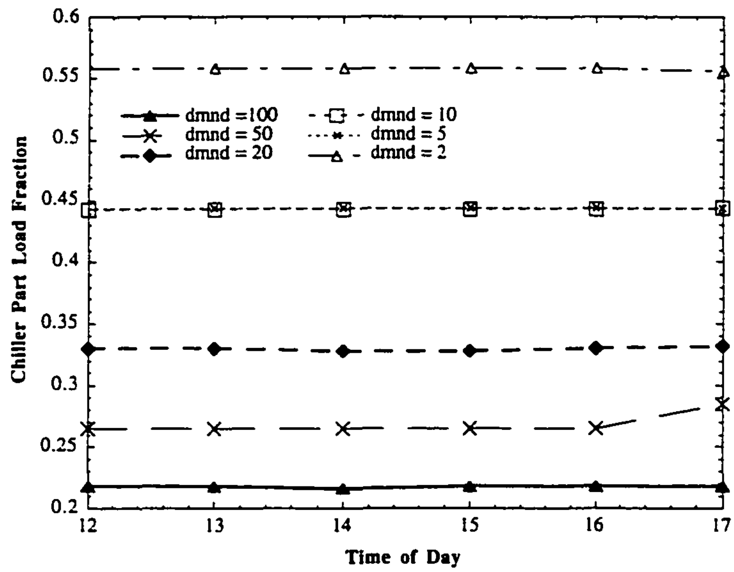


Figure 7-22: Hourly Variation in Chiller Part Load Ratio for Six Values of Electricity Demand Charge

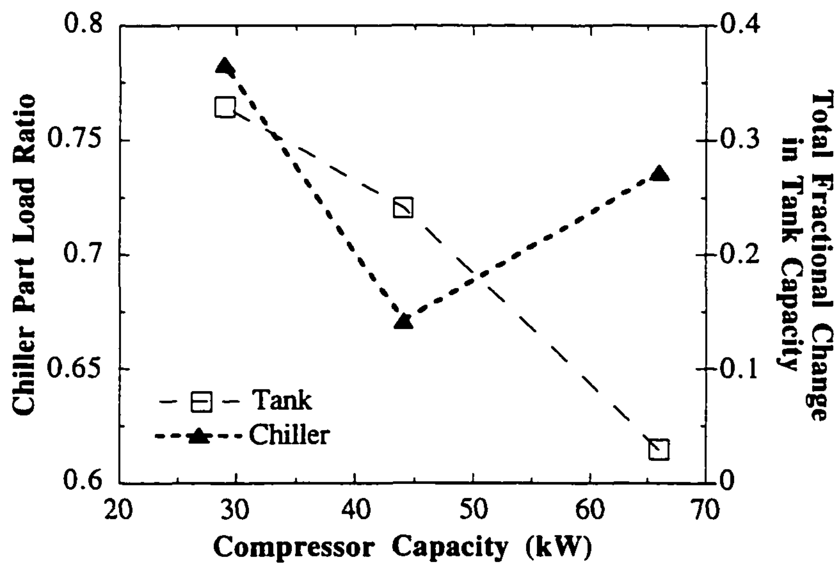


Figure 7-23: Effect of Compressor Capacity on Optimal Chiller Part Load Ratio and Fraction of Stored Ice Consumed (Tank Capacity = 440kWh, On-Peak Rate = 2.0/kWh)

Increasing the compressor capacity first results in a decrease in the optimal part load ratio, but for the 66kW compressor the optimal part load ratio is higher. Compressor energy consumption, shown in Figure 7-24 increased with each increase in compressor capacity. However, the plant with the 66kW compressor also consumed no ice. The changes in the amount of ice remaining in the tank were due to thermal gains from the outside environment into the storage tank. In this case, the compressor was large enough to meet all cooling loads so no ice was consumed. However, in the other cases, some ice had to be used off-peak as well as on-peak because the chiller was too small to meet the peak cooling loads. A final observation from this example, was that total on-peak energy cost was almost invariant for each of the three cases. Presumably, the lack of a demand charge was the principle reason for this as peak electrical consumption would be much higher for the 66kW compressor example than the other two examples, shown in Figure 7-24. This figure also shows opposite trends in compressor and storage tank energy consumption.

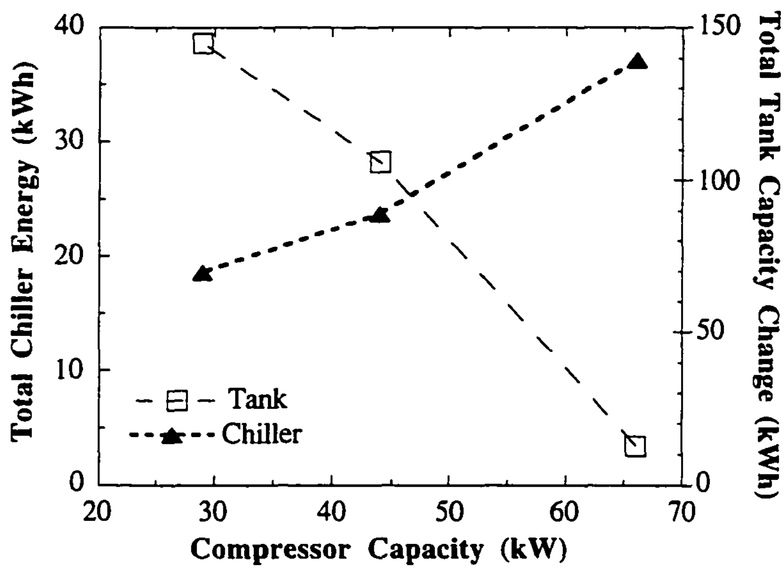


Figure 7-24 Effect of Compressor Capacity on Total Chiller Energy Consumption and Total On-Peak Consumption of Stored Ice (Tank Capacity = 440kWh, On-Peak Rate = 2.0/kWh)

In one final example, the compressor size was fixed at 44kW and the storage tank capacity varied from 307kWh to 613kWh. The on-peak rate multiplier was 1.5 and there was a demand charge of 2/kW. Figure 7-25 shows how the optimal chiller part load ratio and the fractional change in tank capacity were affected. As tank capacity was increased, the fraction of that capacity used to meet the on-peak loads decreased because the total amount of ice available in the tank was

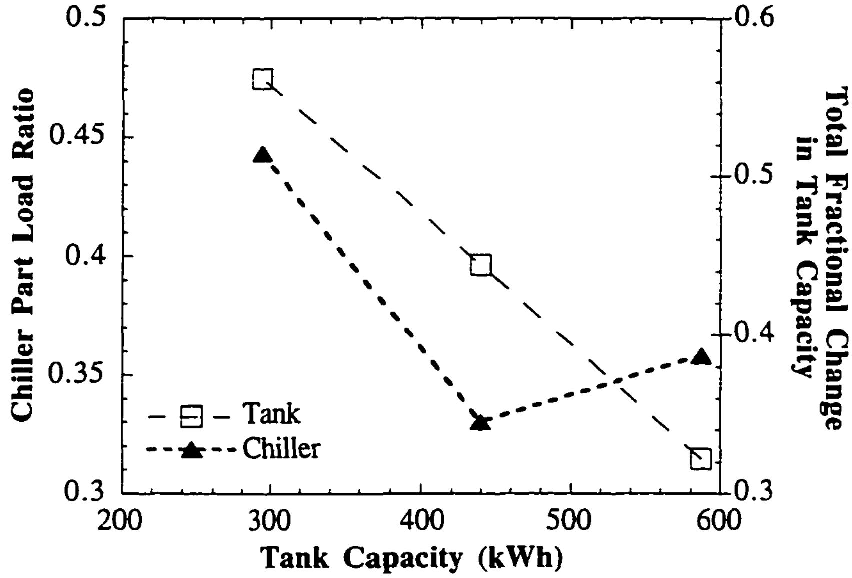


Figure 7-25: Effect of Storage Tank Capacity on Optimal Chiller Part Load Ratio and Fraction of Stored Ice Consumed (Compressor Capacity = 38kW, On-Peak Electric Rate Multiplier = 1.5, Demand Charge = 2.0/kW)

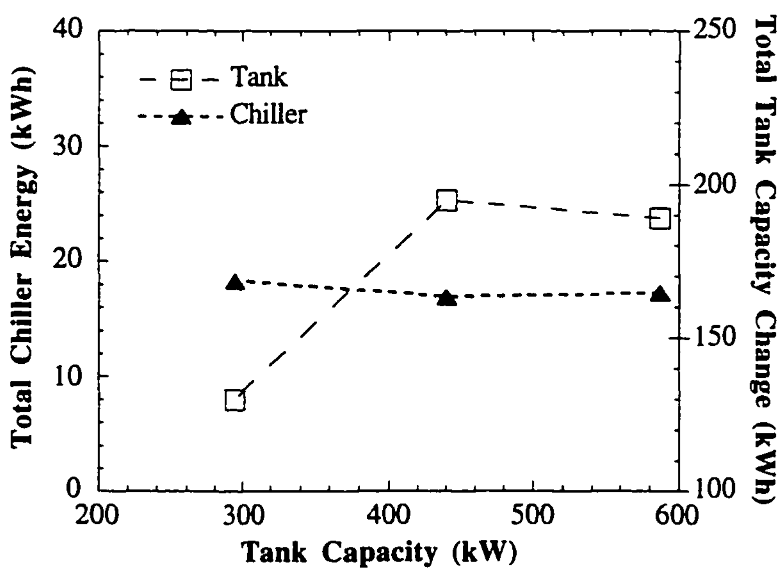


Figure 7-26: Effect of Storage Tank Capacity on Total Chiller Energy Consumption and Total On-Peak Consumption of Stored Ice (Compressor Capacity = 38kW, On-Peak Electric Rate Multiplier = 1.5, Demand Charge = 2.0/kW)

greater. The behavior of the chiller part load ratio can be explained by noting that with a capacity of 300 kWh the storage tank was almost depleted by the end of hour 17. Therefore, the compressor had to be used to supplement the storage tank so that ice would be available through the end of the on-peak hours. Figure 7-26 provides the same information as Figure 7-25 in terms of compressor and storage tank energy consumption. Chiller energy consumption is relatively constant because of the insensitivity of this variable to part load ratio at small loads. The total change in tank capacity, which was limited by the storage tank capacity for the smallest storage tank, was highest for the 440kwh tank, and then decreased slightly for the largest storage tank. The total energy cost for the largest tank was 123.6 units, 122.0 units for the medium sized tank, and 122.5 for the smallest tank. The larger tank had higher thermal losses, while the smaller tank used more chiller energy than optimal, accounting for the small variation in total on-peak energy cost.

Chapter 8: Conclusions and Recommendations

8.1: General Conclusions

Several specific contributions have resulted from this work that improve and extend the ability of researchers and designers to simulate building thermal performance. First, a single combined building simulation method was developed to model the thermal interactions between building zones, fan systems and central plants. This method, as implemented in the IBLAST program has been chosen as the basis for the National Energy Analysis Program that will become the standard for building simulation. Second, the IBLAST program was applied to the problem of modeling optimally controlled central plant equipment with or without thermal storage.

The integration of BLAST to create the IBLAST program, which provided the basis for the optimization techniques used, represents an important step forward in building thermal performance simulation. IBLAST enables researchers and designers to study the operation of building, system, and central plant working as part of a system rather than as independent components. This results in an improved assessment of the interactions of these components, and leads to increased confidence when relating the results of the simulation to the building. Furthermore, systems that BLAST previously could not realistically simulate because there was no way to describe their output as a function of zone temperature, present no special problems using the IBLAST simulation scheme. This was a result of additional detail present, in the IBLAST simulation, to allow feedback between the simulation elements. This detail, such as the water loops between central plant equipment and the air handling system, will allow a simulation to provide much more information about equipment performance, and makes possible development of models that more closely mimic real systems. In turn, models that operate like the equipment they simulate can be described more intuitively, and the results of the simulation are easier to relate to actual experience with the equipment.

In order to determine optimal plant equipment schedules, the feasible optimal equipment combinations at each time step had to be evaluated. The optimization was based on selecting the part load ratio at which each piece of available equipment was to be operated. In BLAST, all equipment of the same type is assumed to have the same part load performance characteristics. The minimum energy solution to this problem is operation of all the equipment necessary to meet the load at the same part load ratio regardless of equipment capacity. In addition, there is in BLAST a fixed hierarchy that determines how equipment of different types can be used. The effect of the relative cost of different energy types is not accounted for by this hierarchy. The use of a solver, in this case a GRG routine, to determine the equipment part load ratios that minimize energy cost has the potential for several benefits even without optimal scheduling. First, energy costs and utility rate structures are taken into account in determining the lowest cost way to operate equipment. This means that the results obtained reflect not just the specific equipment used in a building but also, given significant national and global variations in energy costs, its location. Furthermore, the cost of each energy type and the way those costs change on an hourly or daily basis has been shown to be an important factor in determining the best operating conditions for a given central plant. Therefore, an accurate after the fact energy cost analysis, based on plant operating conditions determined without including cost factors, is not possible.

Since the same limitations on equipment performance parameters existed in IBLAST, the results of solving for the optimal equipment part load ratios piece-by-piece matched those obtained from BLAST when equipment of only one type was used. However, when more than one equipment type was used, and especially when the different types were powered by different energy sources, the optimal solution was strongly dependent on the relative difference between the energy rates and, when applicable, their variations during a one day cycle. In addition. the problem was formulated in such a way that future versions of the program could easily allow for the individual performance description of each piece of central plant equipment and also allow more detailed models so that equipment size could also be a factor in the optimization.

The optimal equipment schedules described in this research were determined by piecing together, for each hour of the simulation, a series of equipment combinations whose energy cost had been minimized by selecting the appropriate part load ratios for each piece of equipment. Normally, in order to find the lowest cost schedule, a search would have to be conducted on all the feasible paths because part of the objective function included the peak utility electric demand that could not be known until after a path was completely determined. As the number of hours being optimized grows so does the number of feasible equipment states. This results in the total number of possible paths increasing until required computation time and data storage become prohibitively large. An alternative to a global search of all possible paths was used in solving this problem. This alternative recognized that only a limited subset of the possible paths need be considered as candidates for a true global minimum. This subset was made up of those paths that had lower peak energy consumption than the reference minimum energy path, which is defined in Section 7.1. In cases where the number of possible equipment combinations was large, the reduction in the total number of paths calculated was frequently several orders of magnitude, and allowed solution of some problems that would not have been feasible computationally had the number of paths considered not been reduced. Furthermore, the method finds a true global minimum energy cost under the assumption that the cost of the additional energy consumed during an equipment start-up spike is negligible. This is a reasonable assumption given that the start-up process lasts on the order of 0.1 hour and the increase in energy consumption over steady state during that period is 20%. In this case, the maximum additional energy consumption for the hour would be 2% of the total consumption, and the actual value would be less since the energy consumption would decay from its peak value to its steady state value over the duration of the start-up period.

In addition to the ability of the method described above to handle a range of problems, the optimal scheduling method provided results that follow the predicted trends. For example, as the utility demand charges on peak electric consumption increase, the simulation model predicts that fewer and less dramatic changes in equipment operating capacity should occur when all the equipment was of the same type and used the same type of energy. In addition, when equipment utilizing an alternative energy source was available this equipment was used to limit peak electricity demand. While these results intuitively make sense, they do not lead to obvious generalizations or rules-of-thumb. The wide selection in equipment types and the variations in energy costs imply that a simulation of the proposed building and its equipment is necessary to accurately determine the best way to operate equipment.

A final objective of this work was to develop a method for optimally scheduling the use of thermal storage systems so as to minimize daily energy cost. The selected method assumes that a single constant cost per unit of ice removed from storage could be used to calculate the cost of using ice compared to the cost of direct cooling. In addition, in order to determine the lowest cost schedule, defined by the compressor part load ratio for each on-peak hour, all the possible schedules had to be considered to ensure that a global minimum was found. However, this would have resulted in an unwieldy computational problem so only two compressor part load fractions were allowed in each hour of the simulation. Subsequently, an iteration was performed to improve the accuracy of the solution. This method proved quite tractable from a computational point of view, as many one day iterations could be performed for minimal additional simulation time when the number of allowable compressor states is small for each hour. Use of additional compressor states each hour did not appreciably change the optimal schedule.

The results of the optimization showed that increasing the on-peak cost of electricity, the demand charge, or a combination of the two caused the thermal storage system to favor the use of ice instead of direct cooling. Somewhat surprisingly, the optimized compressor part load fraction was constant for every hour of the interval being optimized. This indicates that the best efficiency can be obtained from mechanical cooling when it operates to meet a constant load. In that case, the melting of ice accommodates variations in the load. In other words, the compressor is base loaded at an efficient part load fraction, and ice is used to reduce the amount of on-peak energy consumption and reduce maximum on-peak demand. This makes sense since, for a fixed amount of cooling supplied by the compressor, a compressor part load schedule that was not constant would have a higher peak energy consumption and would spend periods of time operating at a less efficient setting. In future, it should only be necessary to optimize on a single compressor part load fraction which would then be applied to all the on-peak hours. A final observation from the use of this method is that obtaining the compressor part load fraction resulting in the true global minimum energy cost requires an additional simulation that assumes ice is used to meet all the on-peak loads. The results of this simulation must be compared to the optimized solution to find the best schedule. This step is necessary because, with the current compressor simulation in IBLAST and BLAST, the optimizer got stuck in a local minimum as the on-peak cost of electricity or the demand charge got large. The predicted optimal compressor part load fraction for each hour was therefore small, but non-zero, which is not correct.

8.2: Implications for Building Simulation

In the Section 8.1, the questions answered were related to the ability of the methods developed in this work to perform their intended function under a range of common building scenarios. However, there are additional questions that address the usefulness of these methods in building simulation programs of any type. The first question concerns the integration of the BLAST simulation elements to form a combined program with feedback among all the components of the building being modeled. The integrated simulation approach provides a more realistic and rigorous way of computing building thermal performance characteristics than the non-integrated BLAST method. In addition, the approach allows, in fact requires, more detail in the specification of the building. This in turn results in a more computationally intensive simulation than the separate simulation approach. Modeling a single element (zone, system, plant) of a building independently also becomes problematic because there is no way to provide the correct feedback mechanisms which drive thermal changes in the element. Furthermore, equipment sizes must be reasonably close to the correct size to obtain accurate results. However, especially in the case of a new building design, the appropriate equipment sizes are not known in advance. An additional simplified simulation method is therefore required to approximately size the system and plant components.

Many of the methods that were used in this work for optimizing central plant conventional and thermal storage equipment could be used in simpler, less integrated building simulation codes such as BLAST. However, a feature of IBLAST that proved very beneficial in this work was the short time step used to update the zone air temperature. This in turn made the systems and central plants operate on the same short time step. Therefore, information such as the peak energy consumption during an hour and the energy consumption at the start of the hour are available and are very useful for calculating energy consumption during start-up spikes. Simulations with a one hour time step cannot provide, nor can they use, this level of simulation detail because the energy balance is performed on each zone only once per hour. As a result, these simulations are fast but the amount of information available during each hour is limited. Furthermore, the plant/system water loop in IBLAST makes possible future enhancements that allow optimization based on the water flow rate through each chiller or boiler and their respective outlet temperatures instead of the equipment part load ratio. This is not a physical parameter that can easily be adjusted, and usually it is calculated from other equipment performance indicators. IBLAST also has the advantage that the zone space temperatures are based directly on the performance of system and plant. However, a further step in the simulation and optimization procedure should be added so that each day is re simulated once the optimal schedule has been calculated. This would represent a relatively small increase in computation time but would ensure that results from the optimal combination were correctly displayed in the output.

8.3: Implications for Building Design and Operation

The methods described in this work have several additional implications beyond those already discussed for building simulation programs in general. Firstly, as implemented in the IBLAST program, they will provide a sound basis to evaluate and compare the performance of different equipment types. This could be an issue in both the initial design phase of a building's central plant or when retrofitting new equipment into an existing building to meet new demand or replace old plant. In these situations, the methods could be used to evaluate the relative merits of different central plant equipment options based on performance and utility rate structures. Furthermore, the results from the optimization provide a way to predict how the central plant should be scheduled to obtain the lowest cost of operation. In the same vein, the equipment schedules generated by these methods could be determined for an existing central plant as a way to check if the equipment is operating as cost effectively as possible or if changes in the equipment schedule could yield lower operating costs.

8.4: Recommendations for Future Work

A final consideration is the way in which the simulation and optimization methods developed in this work can be improved by further development and better implementation, specifically in the IBLAST program. One improvement would be a more rigorous treatment of the plant equipment part load optimization. Instead of finding the optimal part load ratios directly, the GRG solver would control physical variables such as hot and cold water flow rates to each piece of equipment and the corresponding water outlet temperatures in order to conserve energy. While this approach is feasible, substantially more storage would be required along with additional computational effort to re simulate the air handling systems each time the plant leaving water temperature changed. Depending on the level of output detail required the additional computational effort may not be warranted. However, a simplification may perhaps be made to the optimization procedure used for thermal storage systems since the optimal compressor part load ratio need only be calculated once. This could be accomplished by a the use of a solver routine, such as GRG, to update the next guess for the optimal compressor part load ratio each day. Each day would be simulated until convergence was obtained.

An additional major enhancement would be to implement changes to IBLAST to allow the part load performance of each piece of equipment, or at least each different size to be specified. This represents, far more closely, the operating characteristics of real equipment. From the standpoint of BLAST this would not be a feasible implementation because of the hardwired optimization code in that program, but in IBLAST the use of GRG to find the optimal equipment part load ratios allows ultimate flexibility. It would also be interesting to repeat some of the examples described in Chapter 6 after this has been implemented to observe the effect on optimal equipment scheduling.

Bibliography

Andersson, B., F. Bauman, R.C. Kammerud, "Verification of BLAST by Comparison with Direct-Gain Test Cell Measurements," Lawrence Berkeley Laboratory Report LBL-10619, October 1980.

Anstett, M., and J.F. Kreider, "Application of Neural Networking Models to Predict Energy Use" preprint for ASHRAE Transactions, Part 1, 1993.

Austin S.B., "Chilled Water System Optimization," ASHRAE Journal, 1993, Jul, Vol. 35, No. 7, pp. 50-56.

Bauman, F., B. Andersson, W. Carroll, R.C. Kammerud, N.E. Friedman, "Verification of BLAST by Comparison with Measurements of a Solar Dominated Test Cell and a Thermally Massive Building," ASHRAE Journal of Solar Engineering, Vol. 105, No. 2, 1983.

BLAST Technical Reference, BLAST Support Office, University of Illinois at Urbana-Champaign, 1993.

BLAST Users Manual, BLAST Support Office, University of Illinois at Urbana-Champaign, 1993.

Braun, J.E., J.W. Mitchell, and S.A. Klein, "Performance and Cooling Characteristics of a Large Cooling System," ASHRAE Transactions, Part 1, 1987, pp. 1830-1852.

Braun, J.E., S.A. Klein, J.W. Mitchell, and W.A. Beckman, "Applications of Optimal Control to Chilled Water Systems Without Storage," ASHRAE Transactions, Part 1, 1989, pp. 663-675.

Braun, J.E., S.A. Klein, W.A. Beckman, and J.W. Mitchell, "Methodologies for Optimal Control of Chilled Water Systems Without Storage," ASHRAE Transactions, Part 1, 1989, pp. 652-662.

Clark, D.R., HVACSIM+ Building Systems and Equipment Simulation Program Reference Manual, Pub. No. NBSIR 84-2996, National Bureau of Standards, U.S. Department of Commerce, January, 1985.

Cumali, Z., "Global Optimization of HVAC System Operations in Real Time," ASHRAE Transactions, Part 1, 1988, pp. 1729-1744.

Curtiss, P.S., J.F. Kreider, and M.J. Brandemuehl, "Adaptive Control of HVAC Processes Using Predictive Neural Networks" preprint for ASHRAE Transactions, Part 1, 1993.

Elmahdy, A.H., Analytical and Experimental Multi-Row Finned-Tube Heat Exchanger Performance During Cooling and Dehumidifying Processes, Ph.D. Thesis, Carleton University, Ottawa, Canada, December, 1975.

Elmahdy, A.H., and Mitalas, G.P., "A Simple Model for Cooling and Dehumidifying Coils for Use in Calculating Energy Requirements for Buildings," ASHRAE Transactions, 1977, Vol. 83, Part 2, pp. 103-117.

Fiorino, D.P., "Energy Conservation With Chilled-Water Storage," ASHRAE Journal, 1993, May, V35, N5, pp. 22-32.

Gardner, K.A., "Efficiency of Extended Surface," Transactions ASME, 1945, Vol. 67, pp. 621-631.

Gray, B.F., C.A. Johnson, G.J. Schoneau, R.W. Besant, "Energy Consumption and Economic Evaluation of Thermal Storage and Recovery Systems for a Large Commercal Building," ASHRAE Transactions, Part 1, 1988, pp. 412-424.

GRG Users Guide for UNIX, Computing Services Office, University of Illinois at Urbana-Champaign, January 1986.

Hackner, R.J., J.W. Mitchell, and W.A. Beckman,, "HVAC System Dynamics and Energy Use in Buildings- Part II," ASHRAE Transactions, Part 1B, 1985, pp. 781-797.

Hittle, D. C., Calculating Heating and Cooling Loads Using the Frequency Response of Multilayered Slabs, Ph.D. Thesis, University of Illinois, 1980.

Kays, W.M., and London, A.L., Compact Heat Exchangers, 3rd Edition, McGraw-Hill Book Co., 1984.

Klein, S.A., D.R. Nugent, and J.W. Mitchell, "Investigation of Control Alternatives for a steam Turbine Driven Chiller," ASHRAE Transactions, Part 1, 1988, pp. 627-643.

Lau, A.S., W.A. Beckman, and J.W. Mitchell, "Development of Computerized Control Strategies for a Large Chilled Water Plant," ASHRAE Transactions, Part 1B, 1985, pp. 766-780.

MacArthur, J.W., A. Mathur, and J. Zhao, "On-Line Recursive Estimation for Load Profile Prediction," ASHRAE Transactions, Part 1, 1989, pp. 621-628.

MacArthur, J.W., and M.A. Woessner, "Receding Horizon Control: A Model Based Policy for HVAC Applications," preprint for ASHRAE Transactions, Part 1, 1993.

Metcalf R. R., R. D. Taylor, C. O. Pedersen, R. J. Liesen, and D. E. Fisher, "Incorporating a Modular System Simulation Program into Large Energy Analysis Program: The Linking of IBLAST and HVACSIM+," IBPSA Building Simulation '95: 4th International Conference Proceedings, Madison, Wisconsin, August 14-16, 1995, pp. 415-422.

Mistry, S.I., and S.S. Nair, "Nonlinear HVAC Computations Using Neural Networks" preprint for ASHRAE Transactions, Part 1, 1993.

Nizet, J.L., J. Lecomte, F.X. Litt, "Optimal Control Applied to Air Conditioning in Buildings," ASHRAE Transactions, Part 1B, 1984, pp. 587-600.

Olson, R.T., Optimal Allocation of Building Cooling Loads to Chilled Water Plant Equipment, Ph.D. Thesis University of Illinois at Urbana-Champaign, 1987.

Olson, R.T. and J.S. Liebman, "Optimization of a Chilled Water Plant Using Sequential Quadratic Programming", Engineering Optimization, 1990, Vol. 15, pp. 171-191.

Olson, R.T., C.O. Pedersen and J.S. Liebman, "A Dynamic Procedure for the Optimal Sequencing of Plant Equipment Part I: Algorithm Fundamentals", Engineering Optimization, 1993, Vol. 21(1), pp. 63-78.

Olson, R.T. and J.S. Liebman, "A Dynamic Procedure for the Optimal Sequencing of Plant Equipment Part II: Validation and Sensitivity Analysis", Engineering Optimization, 1994, Vol. 22, pp. 163-183.

Powell, M.J.D., "VMCWD: A FORTRAN Subroutine for Constrained Optimization," Report DAMTP/1982/NA4, University of Cambridge, England, 1984.

Powell, M.J.D., "ZQPCVX: A FORTRAN Subroutine for Convex Quadratic Programming," Report DAMTP/1983/NA17, University of Cambridge, England, 1983.

Rink, R.E., M. Zaheer-Uddin, "Energy-Optimal Control Of A Bilinear Multizone Cooling System With Chilled-Water Storage," Energy Conversion And Management 1993, Dec, Vol. 34, No. 12, pp. 1229-1238

Seem, J. E., Modeling of Heat Transfer in Buildings, Ph.D. Thesis, University of Wisconsin-Madison, 1987.

Smith, S.F., "Thermal Storage HVAC System Retrofit Provides Economical Air Conditioning," ASHRAE Journal, 1993, Mar, Vol. 35, No. 3, pp. 28-30.

Smith, T.R., and D.C. Hittle, "Analysis of an Ice Storage Air Conditioning System with Heat Recovery," Joint ASME/ASES Conference, April, 1993.

Spethmann, D.H., "Applications of Considerations in Optimal Control of Cool Storage," preprint for ASHRAE Transactions, Part 1, 1993.

Spethmann, D.H., "Optimal Control for Cool Storage," ASHRAE Transactions, Part 1, 1989, pp. 1189-1193.

"Standard for Forced Circulation Air-Cooling and Air-Heating Coils," Air-Conditioning and Refrigeration Institute Standard, 1972, pp. 410-472.

Strand, R.K., Indirect Ice Storage System Simulation, Master's Thesis, Department of Mechanical Engineering, University of Illinois at Urbana-Champaign, 1992.

Sud, I., "Control Strategies for Minimum Energy Usage," ASHRAE Transactions, Part 2A, 1984, pp. 247-277.

Taylor, R. D. C.O. Pedersen, and L. Lawrie, "Simultaneous Simulation of Buildings and Mechanical Systems in Heat Balance Based Energy Analysis Programs," Proceedings of the 3rd International Conference on System Simulation in Buildings, Liege, Belgium, December 3-5, 1990, pp. 87-106.

Taylor, R. D. C.O. Pedersen, and L. Lawrie, "Simulation of Thermal Storage Systems in an Integrated Building Simulation Program," CISS: First Joint Conference of International Simulation Societies Proceedings, Zurich, Switzerland, August 22-25, 1994, pp. 744-748.

Taylor, R. D., C.O. Pedersen, R.J. Liesen, D. Fisher, and L. Lawrie, "Impact of Simultaneous Simulation of Buildings and Mechanical Systems in Heat Balance Based Energy Analysis Programs on System Response and Control," IBPSA Building Simulation '91: 2nd International Conference Proceedings, Nice, France, August 20-22, 1991, pp. 227-234.

Witte, M. J., C. O. Pedersen, and J. D. Spitler, "Techniques for Simultaneous Simulation of Buildings and Mechanical Systems in Heat Balance Based Energy Analysis Programs," Conference Proceedings of IBPSA Building Simulation '89, Vancouver, B.C., June 1989.

Yuill, G.K., E.G. Phillips, "Comparison of BLAST Program Predictions with the Energy Consumptions of Two Buildings," ASHRAE Transactions, Vol. 87, Part 1, 1981, pp. 1200-1206.

Zaheer-Uddin M., R.E. Rink, and ,V.G. Gourishankar, "Heuristic Control Profiles for Integrated Boilers," ASHRAE Transactions, Part 2, 1990, pp. 205-211.

Appendices

Appendix A: PLTCOMB.FTN: Listing of Subroutines Necessary for Central Plant Optimization

```
*DECK EQCOMB
      SUBROUTINE EQCOMB
CD    TITLE:=  EQCOMB - CALCULATE ALL FEASIBLE EQUIPMENT COMBINATIONS
CD             FOR SYSTEM PLANT OPTIMISATION
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  DECEMBER 1994
CD    PURPOSE:=
CD
CD   Routine to determine all possible
CD   operating combinations of plant equipment
CD
CD    VARIABLE DICTIONARY:=
c      inputs
c     -    NCHAVAIL(MXCHSZ,MXCHTYP,MXPLTS),
c     -    NUMCHSIZ(MXCHTYP,MXPLTS),
c     -    NBLRAVAIL(MXBLRSZ,MXBLRTYP,MXPLTS),
c     -    NUMBLRSIZ(MXBLRTYP,MXPLTS)
c     -    NGNAVAIL(MXGNRSZ,MXGNRTYP,MXPLTS),
c     -    NUMGNSIZ(MXGNRTYP,MXPLTS)
c     -    NCNDRAVAIL(MXCNDSZ,MXCNDTYP,MXPLTS),
c     -    NUMCNDRSIZ(MXCNDTYP,MXPLTS)
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
c
      INCLUDE 'boilrdcl.inc'
      INCLUDE 'blsfls.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'gnrtrdcl.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'iceopt.inc'
C
C     LOCAL VARIABLES
C
      integer i,i1,j,iptr,nskip,ictr,k,k1,k2,iplt,ityp
      real   cumchcap,cumblcap,cumgncap,cumcdcap,chmult,blmult
C
c*******Main Program*******
      iptr=0
c
c    Compute total number of equip. types
c
      do 51 iplt=1,ninpl
c
c      nchtyp(iplt)=0
```

```
c      nbltyp(iplt)=0
c      ngntyp(iplt)=0
c      ncdtyp(iplt)=0
c
      nchtyp=0
      nbltyp=0
      ngntyp=0
      ncdtyp=0
c
c
      do 50 ityp=1,3
        if (numchsiz(ityp,iplt).gt.0) then
c         nchtyp(iplt)=nchtyp(iplt)+1
         nchtyp=nchtyp+1
        endif
        if (numblrsiz(ityp,iplt).gt.0) then
c         nbltyp(iplt)=nbltyp(iplt)+1
         nbltyp=nbltyp+1
        endif
        if (numgnsiz(ityp,iplt).gt.0) then
c         ngntyp(iplt)=ngntyp(iplt)+1
         ngntyp=ngntyp+1
        endif
        if (numcndrsiz(ityp,iplt).gt.0) then
c         ncdtyp(iplt)=ncdtyp(iplt)+1
         ncdtyp=ncdtyp+1
        endif
50    continue
c      neqtyp(iplt)=nchtyp(iplt)+nbltyp(iplt)+ngntyp(iplt)+ncdtyp(iplt)
      neqtyp=nchtyp+nbltyp+ngntyp+ncdtyp
c
      write(output,*)
      write(output,201)
      write(output,202)
c      write(output,101) nchtyp(iplt),nbltyp(iplt),ngntyp(iplt),
c     -                  ncdtyp(iplt)
      write(output,101) nchtyp,nbltyp,ngntyp,ncdtyp
c
c   Read in Number of Equipment Sizes of Each Type and number of
c     equipment of that size and type
c
c      if (neqtyp(iplt).eq.0) goto 900
c       if (nchtyp(iplt).gt.0) then
c        do 10 i=1,nchtyp(iplt)
      if (neqtyp.eq.0) goto 900
       if (nchtyp.gt.0) then
        do 10 i=1,nchtyp
         write(output,*)
         write(output,203) i
         write(output,207) (chnomcap(j,i,iplt),j=1,numchsiz(i,iplt))
         write(output,102) (nchavail(j,i,iplt),j=1,numchsiz(i,iplt))
         do 20 j=1,numchsiz(i,iplt)
           nequip(j+iptr,iplt)=nchavail(j,i,iplt)
           nalleqp(iplt)=nalleqp(iplt)+nchavail(j,i,iplt)
20       continue
         iptr=iptr+numchsiz(i,iplt)
10      continue
       endif
```

```
      nchts(iplt)=iptr
c      if (nbltyp(iplt).gt.0) then
c       do 11 i=1,nbltyp(iplt)
      if (nbltyp.gt.0) then
       do 11 i=1,nbltyp
        write(output,*)
        write(output,204) i
        write(output,207) (blrnomcap(j,i,iplt),j=1,numblrsiz(i,iplt))
        write(output,102) (nblravail(j,i,iplt),j=1,numblrsiz(i,iplt))
        do 21 j=1,numblrsiz(i,iplt)
          nequip(j+iptr,iplt)=nblravail(j,i,iplt)
          nalleqp(iplt)=nalleqp(iplt)+nblravail(j,i,iplt)
21      continue
        iptr=iptr+numblrsiz(i,iplt)
11     continue
      endif
      nblts(iplt)=iptr-nchts(iplt)
c      if (ngntyp(iplt).gt.0) then
c       do 12 i=1,ngntyp(iplt)
      if (ngntyp.gt.0) then
       do 12 i=1,ngntyp
        write(output,*)
        write(output,205) i
        write(output,207) (gnnomcap(j,i,iplt),j=1,numgnsiz(i,iplt))
        write(output,102) (ngnavail(j,i,iplt),j=1,numgnsiz(i,iplt))
        do 22 j=1,numgnsiz(i,iplt)
          nequip(j+iptr,iplt)=ngnavail(j,i,iplt)
22      continue
        iptr=iptr+numgnsiz(i,iplt)
12     continue
      endif
c      if (ncdtyp(iplt).gt.0) then
c       do 13 i=1,ncdtyp(iplt)
      if (ncdtyp.gt.0) then
       do 13 i=1,ncdtyp
        write(output,*)
        write(output,206) i
        write(output,207) (cndrnomcap(j,i,iplt),j=1,numcndrsiz(i,iplt))
        write(output,102) (ncndravail(j,i,iplt),j=1,numcndrsiz(i,iplt))
        do 23 j=1,numcndrsiz(i,iplt)
          nequip(j+iptr,iplt)=ncndravail(j,i,iplt)
23      continue
        iptr=iptr+numcndrsiz(i,iplt)
13     continue
      endif
      neqtot(iplt)=iptr
c
C     compute possible equipment combinations
c
c     max number of combinations
      maxcom(iplt)=1
      do 31 i=1,iptr
         maxcom(iplt)=maxcom(iplt)*(nequip(i,iplt)+1)
31    continue
c
c     compute all possible combinations
c
      nskip=maxcom(iplt)
```

```
      do 41 i=1,iptr
       j=0
       nskip=nskip/(nequip(i,iplt)+1)
       Do While (j.le.maxcom(iplt))
       Do 43 ictr=1,nequip(i,iplt)+1
        do 42 k=1,nskip
         j=j+1
         eqpcomb(j,i,iplt)=ictr-1
42      continue
43      continue
       End Do
41    Continue
      Do 55 j=1,maxcom(iplt)
         cumchcap=0.0
         cumblcap=0.0
         cumgncap=0.0
         cumcdcap=0.0
         iptr=0
c         Do 56 k1=1,nchtyp(iplt)
         Do 56 k1=1,nchtyp
            chmult=maxchfr(k1,iplt)
            Do 57 k2=1,numchsiz(k1,iplt)
               cumchcap=cumchcap+eqpcomb(j,k2+iptr,iplt)*
     -                   chnomcap(k2,k1,iplt)*chmult
57          Continue
            iptr=iptr+numchsiz(k1,iplt)
56       Continue
c         Do 66 k1=1,nbltyp(iplt)
         Do 66 k1=1,nbltyp
            blmult=maxblrfr(k1,iplt)
            Do 67 k2=1,numblrsiz(k1,iplt)
               cumblcap=cumblcap+eqpcomb(j,k2+iptr,iplt)*
     -                   blrnomcap(k2,k1,iplt)*blmult
67          Continue
            iptr=iptr+numblrsiz(k1,iplt)
66       Continue
c         Do 76 k1=1,ngntyp(iplt)
         Do 76 k1=1,ngntyp
            Do 77 k2=1,numgnsiz(k1,iplt)
               cumgncap=cumgncap+eqpcomb(j,k2+iptr,iplt)*
     -                   gnnomcap(k2,k1,iplt)
77          Continue
            iptr=iptr+numgnsiz(k1,iplt)
76       Continue
c         Do 86 k1=1,ncdtyp(iplt)
         Do 86 k1=1,ncdtyp
            Do 87 k2=1,numcndrsiz(k1,iplt)
               cumcdcap=cumcdcap+eqpcomb(j,k2+iptr,iplt)*
     -                   cndrnomcap(k2,k1,iplt)
87          Continue
            iptr=iptr+numcndrsiz(k1,iplt)
86       Continue
         eqpcomb(j,iptr+1,iplt)=cumchcap
         eqpcomb(j,iptr+2,iplt)=cumblcap
         eqpcomb(j,iptr+3,iplt)=cumgncap
         eqpcomb(j,iptr+4,iplt)=cumcdcap
55    Continue
c
```

```
c         save maximum equipment capacities
c
c          maxchlcap(iplt)=eqpcomb(maxcom(iplt),iptr+1,iplt)
c          maxblrcap(iplt)=eqpcomb(maxcom(iplt),iptr+2,iplt)
c          maxgnrcap(iplt)=eqpcomb(maxcom(iplt),iptr+3,iplt)
c          maxcdrcap(iplt)=eqpcomb(maxcom(iplt),iptr+4,iplt)
          maxchlcap=eqpcomb(maxcom(iplt),iptr+1,iplt)
          maxblrcap=eqpcomb(maxcom(iplt),iptr+2,iplt)
          maxgnrcap=eqpcomb(maxcom(iplt),iptr+3,iplt)
          maxcdrcap=eqpcomb(maxcom(iplt),iptr+4,iplt)
c
c         write out equipment combinations and capacities
c
      write (output,*)
      do 401 j=1,maxcom(iplt)
         write(output,300) j,(eqpcomb(j,i,iplt),i=1,iptr),
     -                    (eqpcomb(j,i1,iplt),i1=iptr+1,iptr+4)
401   continue
c
c         read in energy cost factors and range of on-peak
c         hour for utility electric consumption
c
      open (unit=75,file='costing.dat',status='old')
      read(75,*)
      read(75,*)
      read(75,210) (cstfac(j),j=1,4),dmndfac
      read(75,*)
      read(75,211) bgnonpk,endonpk
      read(75,*)
      read(75,*)
      read(75,212) onpkmult
      close(75)
c
c         echo cost information to output file
c
      write(output,*)
      write(output,310) cstfac(1)
      write(output,311) cstfac(2)
      write(output,312) cstfac(4)
      write(output,*)
      write(output,313) cstfac(3),onpkmult,dmndfac
      write(output,320) bgnonpk,endonpk
      write(output,*)
c
900   Continue
51    Continue
c
999   return
101   format(t7,i2,t19,i2,t30,i2,t44,i2)
102   format(t7,i2,t15,i2,t23,i2,t31,i2,t39,i2,t47,i2)
201   format(t3,'# Chiller',t15,'# Boiler',t25,'# Generator',t38,
     -       '# Condenser')
202   format(t5,'Types',t17,'Types',t28,'Types',t42,'Types')
203   format(3x,'# of Chiller Sizes of Type ',i2)
204   format(3x,'# of Boiler Sizes of Type ',i2)
205   format(3x,'# of Generator Sizes of Type ',i2)
206   format(3x,'# of Condenser Sizes of Type ',i2)
207   format(t5,f4.0,t13,f4.0,t21,f4.0,t29,f4.0,t37,f4.0,t45,f4.0)
```

```
210    format(t5,1p1e10.3,3x,1p1e10.3,3x,1p1e10.3,3x,1p1e10.3,3x,
     -        1p1e10.3)
211    format(t5,2(i3,10x),f5.1)
212    format(t5,1p1e10.3)
C207    format(t5,'Size 1',t13,'Size 2',t21,'Size 3',t29,'Size 4',t37,
C      -        'Size 5',t45,'Size 6')
300    format(t5,29(I3,1X))
310    format("Gas cost:           ",1pe10.3,"(Units/KWH)")
311    format("Diesel cost:        ",1pe10.3,"(Units/KWH)")
312    format("Boiler fuel cost: ",1pe10.3,"(Units/KWH)")
313    format("Util elec cost: ",1pe10.3,"(Units/KWH) Pk rate mult: ",
     -1p1e10.3," Dmnd mult: ",1pe10.3)
320    format("On-peak begins: ",i2," Ends: ",i2)
       end

*DECK SETEQUIP
      SUBROUTINE SETEQUIP(STINC,STPTIM)
CD    TITLE:=  SETEQUIP - SELECT PLANT EQUIPMENT COMBINATION AND
CD             PERFORM SYSTEMS AND PLANT SIMULATION FOR ALL
CD             FEASIBLE COMBINATIONS FOR FIXED ZONE LOADS
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  JANUARY 1995
CD    PURPOSE:=
CD
CD    VARIABLE DICTIONARY:=
c      outputs
c     -   NCHAVAIL(MXCHSZ,MXCHTYP,MXPLTS),
c     -   NUMCHSIZ(MXCHTYP,MXPLTS),
c     -   NBLRAVAIL(MXBLRSZ,MXBLRTYP,MXPLTS),
c     -   NUMBLRSIZ(MXBLRTYP,MXPLTS)
c     -   NGNAVAIL(MXGNRSZ,MXGNRTYP,MXPLTS),
c     -   NUMGNSIZ(MXGNRTYP,MXPLTS)
c     -   NCNDRAVAIL(MXCNDSZ,MXCNDTYP,MXPLTS),
c     -   NUMCNDRSIZ(MXCNDTYP,MXPLTS)
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
c
      INCLUDE 'absenv.inc'
      INCLUDE 'boilrdcl.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'gnrtrdcl.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
      INCLUDE 'tepdcl.inc'
      INCLUDE 'zoncon.inc'
      INCLUDE 'nigel.inc'
```

```
      INCLUDE 'iceopt.inc'
C
C     LOCAL VARIABLES
C
      integer i,j,iptr,nskip,ictr,k,iplt,ityp,icmb,ipt,ns,isys,
     -        cmbctr,iequip,chctr,oldcmb,chdiff
      real stptim,minclng,minhtng,oldchlcap
      real stinc,rhoairsa,rhoairra,rhoairfn,fanpowpfac
c*******Variables*******
c
C   i
C   j
C   iptr
C   nskip
C   ictr
C   k
C   iplt
C   ityp
C   icmb - points to location of current equip combination
C   ipt
C   ns
C   isys
C   cmbctr - keeps track of how many feasible combinations there
C            are this time step
C   stptim
C   stinc
C   rhoairsa
C   rhoairra
C   rhoairfn
C   fanpowpfac
c
C
c*******Main Program*******
C
C
C          for each plant on the building consider feasible
C          equipment combinations
C
      do 101 iplt=1,ninpl
C
C          initialise and zero necessary arrays
C
      PLTEQNO(IPLT,STPNM)=0
      ecoolf(iplt)=0.0
      abshtng(iplt)=0.0
      eboler=0.0
      cmbctr=0
      pltclng=0.0
      plthtng=0.0
      minclng=0.0
      minhtng=0.0
      oldchlcap=0.0
      oldcmb=0
      chctr=0
      callice=.true.
C
C          simulate each combination of plant equipment:  a check will be
C          made to determine if the load on the plant from the previous
```

```
c          combination is greater than the capacity (heating or cooling)
c          of the equipment for the current combination.  If so it is
c          skipped since we do not want to optimise an overloaded plant,
c          unless this occurs when all the equipment is operating.
c
      icmb=maxcom(iplt)+1
      do while ((icmb.gt.1.and..not.warmup).or.icmb.eq.maxcom(iplt)+1)
       icmb=icmb-1
      iptr=0
      numofch=0
      numofbl=0
c
c         SELECT EQUIPMENT COMBINATION TO BE OPERATED
c
c      if (neqtyp(iplt).eq.0) goto 900
c       if (nchtyp(iplt).gt.0) then
c        do 10 i=1,nchtyp(iplt)
      if (neqtyp.eq.0) goto 900
       if (nchtyp.gt.0) then
        do 10 i=1,nchtyp
         do 20 j=1,numchsiz(i,iplt)
           nchavail(j,i,iplt)=eqpcomb(icmb,j+iptr,iplt)
           if (icmb.eq.maxcom(iplt)) then
             chctr=chctr+1
           endif
20       continue
         iptr=iptr+numchsiz(i,iplt)
10      continue
       endif
c       if (nbltyp(iplt).gt.0) then
c        do 11 i=1,nbltyp(iplt)
       if (nbltyp.gt.0) then
        do 11 i=1,nbltyp
         do 21 j=1,numblrsiz(i,iplt)
           nblravail(j,i,iplt)=eqpcomb(icmb,j+iptr,iplt)
21       continue
         iptr=iptr+numblrsiz(i,iplt)
11      continue
       endif
c       if (ngntyp(iplt).gt.0) then
c        do 12 i=1,ngntyp(iplt)
       if (ngntyp.gt.0) then
        do 12 i=1,ngntyp
         do 22 j=1,numgnsiz(i,iplt)
           ngnavail(j,i,iplt)=eqpcomb(icmb,j+iptr,iplt)
22       continue
         iptr=iptr+numgnsiz(i,iplt)
12      continue
       endif
c       if (ncdtyp(iplt).gt.0) then
c        do 13 i=1,ncdtyp(iplt)
       if (ncdtyp.gt.0) then
        do 13 i=1,ncdtyp
         do 23 j=1,numcndrsiz(i,iplt)
           ncndravail(j,i,iplt)=eqpcomb(icmb,j+iptr,iplt)
23       continue
         iptr=iptr+numcndrsiz(i,iplt)
13      continue
```

```
      endif
c
c         read in nominal cumulative operating capacity for each type
c         of plant component in this combination
c
      cumchlcap=eqpcomb(icmb,iptr+1,iplt)
      cumblrcap=eqpcomb(icmb,iptr+2,iplt)
      cumgnrcap=eqpcomb(icmb,iptr+3,iplt)
      cumcndcap=eqpcomb(icmb,iptr+4,iplt)
      chdiff=0
      if (icmb.ne.maxcom(iplt)) then
        do 31 j=1,chctr
          chdiff=chdiff+iabs(eqpcomb(icmb,j,iplt)-
     -             eqpcomb(oldcmb,j,iplt))
31      continue
      endif
      if (chdiff.gt.0.and.minhtng.eq.0.0) then
        minhtng=1.0
      endif
c
c     Check to see if current capacity is less than previous load
c     If capacity is O.K. (i.e. large enough) then it's alright to
c     simulate this combination.  Since ecoolf and eboler are reset
c     each time setequip is called the first combination with all
c     plant components available will always be simulated.
c
      If(
ccc
     -  (
cc
     -  (
c
     -  (
     -      minclng.gt.0.0.and.
c    -      (cumchlcap.ge.minclng.or.cumchlcap.eq.maxchlcap(iplt))
     -     (cumchlcap.ge.minclng.or.cumchlcap.eq.maxchlcap)
     -  ).or.
c
     -     (minclng.eq.0.0.and.cumchlcap.eq.0.0)
     -  )
cc
     -      .and.
cc
     -  (
c
     -  (
     -      minhtng.gt.0.0.and.
c    -      (cumblrcap.ge.minhtng.or.cumblrcap.eq.maxblrcap(iplt))
     -     (cumblrcap.ge.minhtng.or.cumblrcap.eq.maxblrcap)
     -  ).or.
c
     -     (minhtng.eq.0.0.and.cumblrcap.eq.0.0)
c    -      (abshtng(iplt).eq.0.0.and.cumblrcap.eq.0.0)
     -  )
cc
     -  ).or.
ccc
     -      icmb.eq.maxcom(iplt)
```

```
     -    )
     -        then
c        If((cumchlcap.ge.ecoolf(iplt).and.
c     -    .not.(eboler.eq.0.0.and.cumblrcap.gt.0.0))
c     -    .and.(cumblrcap.ge.eboler.and.
c     -    .not.(ecoolf(iplt).eq.0.0.and.cumchlcap.gt.0.0))
c     -    .or.icmb.eq.maxcom(iplt))
c     -    then
          cmbctr=cmbctr+1
C
C         SIMULATE SELECTED EQUIPMENT COMBINATION
C
      CALL SYSSIM(STINC,IPLT)
c
c         set min cooling and heating loads plant must meet
c
        if (icmb.eq.maxcom(iplt)) then
          minclng=pltclng
        endif
        if (icmb.eq.maxcom(iplt).or.cumchlcap.ne.oldchlcap) then
          minhtng=plthtng
        endif
C
C         STORE THE HEATING AND COOLING PROVIDED BY THE PLANT FOR THE
C         CONFIGURATION WITH ALL EQUIPMENT OPERATING
C
C
C         UPDATE RECORD KEEPING FOR THE SELECTED COMBINATION
C
        PLTEQNO(IPLT,STPNM)=PLTEQNO(IPLT,STPNM)+1
        PLTCMBREC(CMBCTR,IPLT,1,STPNM)=ICMB
        PLTCMBREC(CMBCTR,IPLT,2,STPNM)=numofch
        PLTCMBREC(CMBCTR,IPLT,3,STPNM)=numofbl
        PLTSTP(IPLT,STPNM)=STINC
        PLTOPREC(CMBCTR,IPLT,1,STPNM)=EFUELG(IPLT)
        PLTOPREC(CMBCTR,IPLT,2,STPNM)=EFUELD(IPLT)
        PLTOPREC(CMBCTR,IPLT,3,STPNM)=EELECT(IPLT)
        PLTOPREC(CMBCTR,IPLT,4,STPNM)=EFUELB(IPLT)
        PLTOPREC(CMBCTR,IPLT,5,STPNM)=EUT(IPLT)
        do 501 iequip=1,numofch
          PLTOPDAT(CMBCTR,IPLT,iequip,STPNM,1)=optchfin(iequip)
          PLTOPDAT(CMBCTR,IPLT,iequip,STPNM,2)=opchelec(iequip)
          PLTOPDAT(CMBCTR,IPLT,iequip,STPNM,3)=opchengy(iequip)
501     continue
        do 502 iequip=1,numofbl
          PLTOPDAT(CMBCTR,IPLT,numofch+iequip,STPNM,1)=optblfin(iequip)
          PLTOPDAT(CMBCTR,IPLT,numofch+iequip,STPNM,2)=opblelec(iequip)
          PLTOPDAT(CMBCTR,IPLT,numofch+iequip,STPNM,3)=opblengy(iequip)
502     continue
      Endif
      oldchlcap=cumchlcap
      oldcmb=icmb
      callice=.false.
      Enddo
101   Continue
900   Continue
      Return
      End
```

```
C
*DECK optavrg
      SUBROUTINE optavrg
CD    TITLE:=  optavrg - average plant energy consumption
CD             for determining optimal equipment schedule
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  FEBRUARY 1995
CD    PURPOSE:=
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c      cmbctr - keeps track of how many of the total number of equipment
c               combinations have been considered
c      cmbctr2 - keeps track of how many of the total number of equipment
c               combinations have been considered
c      feascomb - counter to determine the number of feasible combinations
c                 this time step
c      i
c      icmb - the equipment combination reference number.  Enables the
c             actual equipment being used to be determined from the
c             eqpcomb array set up in EQCOMB.
c      icmb1 - the equipment combination reference number.  Enables the
c             actual equipment being used to be determined from the
c             eqpcomb array set up in EQCOMB.
c      iplt
c      istp - keep track of how many STINC time steps have elapsed
c      mtchfnd - used to determine if a combination became infeasible
c                during a TINC interval.  In SETEQUIP an infeasible (i.e.
c                insufficient capacity) system would not be simulated so
c                it is conceivable that a combination which appears at the
c                beginning of the time step would not appear at the end.
c
c   INPUTS
c
c      acttim
c      pltcmbrec(array)
c      pltoprec(array)
c      pltstp(array)
c      stpnm
c      tinc(local)
c
c
c   OUTPUTS
c
c      numfeas(TIME,PLANT#)
c      pltavop(EQUIP,TIME,PLANT#,ENERGY TYPE)
c      pltopcomb(EQUIP,TIME,PLANT#)
c      pltpkop(EQUIP,TIME,PLANT#,ENERGY TYPE)
c
c
c
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
```

```
      integer cmbctr,cmbctr2,feascomb,icmb,icmb1,istp,iplt,i
      integer numofch,numofbl
      logical mtchfnd
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
c
      INCLUDE 'absenv.inc'
      INCLUDE 'boilrdcl.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'gnrtrdcl.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
      INCLUDE 'tepdcl.inc'
      INCLUDE 'vartic.inc'
      INCLUDE 'zoncon.inc'
c
c*******Main Program*******
c
      do 101 iplt=1,ninpl
        write(*,*) nchtyp,nbltyp
            feascomb=0
c
c           go through all the combinations available at the beginning of
c           the time step
c
        do 102 cmbctr=1,plteqno(iplt,1)
c
c           begin counting feasible combinations
c
          feascomb=feascomb+1
          icmb1=pltcmbrec(cmbctr,iplt,1,1)
          pltopcmb(feascomb,curinth,1,iplt)=icmb1
          pltopcmb(feascomb,curinth,2,iplt)=pltcmbrec(cmbctr,iplt,2,1)
          pltopcmb(feascomb,curinth,3,iplt)=pltcmbrec(cmbctr,iplt,3,1)
          if (curinth.eq.1) then
          pltinelec(feascomb,stdtim)=pltoprec(cmbctr,iplt,3,1)
          do 110 i=1,pltcmbrec(cmbctr,iplt,2,1)+
     -             pltcmbrec(cmbctr,iplt,3,1)
             pltintop(feascomb,stdtim,i,1)=pltopdat(cmbctr,iplt,i,1,1)
             pltintop(feascomb,stdtim,i,2)=pltopdat(cmbctr,iplt,i,1,2)
             pltintop(feascomb,stdtim,i,3)=pltopdat(cmbctr,iplt,i,1,3)
110       continue
          endif
c
c           for all the "stinc" time steps look for the selected combination
c           and compute its total and peak energy consumption
c
c           initialisations
c
          istp=0
          mtchfnd=.true.
          do 105 i=1,25
```

```
          pltavop(feascomb,curinth,iplt,i)=0.0
          if (i.le.5) then
            pltpkop(feascomb,curinth,iplt,i)=0.0
          endif
105     continue
c
c         end of initialisation
c
c         so long as there are time steps remaining AND the combination
c         did not become infeasible maintain energy consumption record
c         keeping
c
        do while (istp.lt.stpnm.and.mtchfnd)
          mtchfnd=.false.
          istp=istp+1
c
c         scan through the combinations available at this time step to
c         see if the one we are looking for (icmb1) is there
c
          cmbctr2=0
          do while (cmbctr2.lt.plteqno(iplt,istp).and.
     -              .not.mtchfnd)
          cmbctr2=cmbctr2+1
            icmb=pltcmbrec(cmbctr2,iplt,1,istp)
            if (icmb.eq.icmb1) then
c
c                     match found begin or continue record keeping on this
combination
c               set match found flag equal to true
c
               mtchfnd=.true.
               numofch=pltcmbrec(cmbctr2,iplt,2,istp)
               numofbl=pltcmbrec(cmbctr2,iplt,3,istp)
               do 103 i=1,5+numofch+numofbl
                 if (i.le.5) then
                 pltavop(feascomb,curinth,iplt,i)=
     -           pltavop(feascomb,curinth,iplt,i)+
     -           pltoprec(cmbctr2,iplt,i,istp)*pltstp(iplt,istp)
                 pltpkop(feascomb,curinth,iplt,i)=
     -           amax1(pltpkop(feascomb,curinth,iplt,i),
     -                 pltoprec(cmbctr2,iplt,i,istp))
                 else
                 pltavop(feascomb,curinth,iplt,i)=
     -           pltavop(feascomb,curinth,iplt,i)+
     -           pltopdat(cmbctr2,iplt,i-5,stpnm,1)*
     -           pltstp(iplt,istp)
                 endif
103            continue
            endif
          enddo
        enddo
        if (.not.mtchfnd) then
c
c            no match found - this combination became infeasible during
c            the time step so reset feascomb to its value before the
c            current combination was started.  Go on to the next
c            combination
c
```

```
            feascomb=feascomb-1
          endif
102     continue
c
c          how many feasible combinations were there this time step?
c
           numfeas(curinth,iplt)=feascomb
101     continue
        return
        end
C
*DECK optavrgh
      SUBROUTINE optavrgh
CD    TITLE:=  optavrgh - compute hourly averages of
CD             plant energy consumption
CD             for determining optimal equipment schedule
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  MARCH 1995
CD    PURPOSE:=
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c      cmbctr - keeps track of how many of the total number of equipment
c               combinations have been considered
c      cmbctr2 - keeps track of how many of the total number of equipment
c               combinations have been considered
c      feascomh - counter to determine the number of feasible combinations
c                 this hour
c      i
c      icmb - the equipment combination reference number.  Enables the
c             actual equipment being used to be determined from the
c             eqpcomb array set up in EQCOMB.
c      icmb1 - the equipment combination reference number.  Enables the
c             actual equipment being used to be determined from the
c             eqpcomb array set up in EQCOMB.
c      iplt
c      time - keeps track of how many sub hourly time steps have elapsed
c      mtchfnd - used to determine if a combination became infeasible
c                during a TINC interval.  In SETEQUIP an infeasible (i.e.
c                insufficient capacity) system would not be simulated so
c                it is conceivable that a combination which appears at the
c                beginning of the time step would not appear at the end.
c
c   INPUTS
c
c      stdtim
c      pltopcmb(array)
c      pltavop(array)
c      pltpkop(array)
c      numfeas(array)
c
c
c   OUTPUTS
c
c      numfeash(HOUR,PLANT#)
c      pltavoph(EQUIP,HOUR,PLANT#,ENERGY TYPE)
c      pltcmbh(EQUIP,HOUR,PLANT#)
```

```
c      pltpkoph(EQUIP,HOUR,PLANT#,ENERGY TYPE)
c
c
c
CD
CD     METHOD:=
C
C      INCLUDES:-
C
       implicit none
       integer iplt,i,numofch,numofbl
       integer cmbctr,cmbctr2,feascomh,icmb,icmb1,time,feascomb
       logical mtchfnd
c
       INCLUDE 'param.inc'
       INCLUDE 'pltpar.inc'
c
       INCLUDE 'absenv.inc'
       INCLUDE 'boilrdcl.inc'
       INCLUDE 'chlrdcl.inc'
       INCLUDE 'cndrdcl.inc'
       INCLUDE 'gnrtrdcl.inc'
       INCLUDE 'hortot.inc'
       INCLUDE 'optiblst.inc'
       INCLUDE 'pltrckp.inc'
       INCLUDE 'runctl.inc'
       INCLUDE 'sysldc.inc'
       INCLUDE 'tepdcr.inc'
       INCLUDE 'tepdcl.inc'
       INCLUDE 'vartic.inc'
       INCLUDE 'zoncon.inc'
c
c*******Main Program*******
c
       do 101 iplt=1,ninpl
             feascomh=0
c
c            go through all the combinations available at the beginning of
c            the hour
c
         do 102 feascomb=1,numfeas(1,iplt)
            feascomh=feascomh+1
            icmb1=pltopcmb(feascomb,1,1,iplt)
            pltcmbh(feascomh,stdtim,1,iplt)=icmb1
            pltcmbh(feascomh,stdtim,2,iplt)=
     -                pltopcmb(feascomb,1,2,iplt)
            pltcmbh(feascomh,stdtim,3,iplt)=
     -                pltopcmb(feascomb,1,3,iplt)
           pltinelch(feascomh,stdtim)=pltinelec(feascomb,stdtim)
           do 110 i=1,pltopcmb(feascomb,1,2,iplt)+
     -             pltopcmb(feascomb,1,3,iplt)
             pltinoph(feascomh,stdtim,i,1)=pltintop(feascomb,stdtim,i,1)
             pltinoph(feascomh,stdtim,i,2)=pltintop(feascomb,stdtim,i,2)
             pltinoph(feascomh,stdtim,i,3)=pltintop(feascomb,stdtim,i,3)
110        continue
c
c            for all the "icont" time steps look for the selected combination
c            and compute its total and peak energy consumption
```

```
c
c          initialisations
c
         mtchfnd=.true.
         do 105 i=1,25
           pltavoph(feascomh,stdtim,iplt,i)=0.0
           if (i.le.5) then
             pltpkoph(feascomh,stdtim,iplt,i)=0.0
           endif
105      continue
c
c          end of initialisation
c
c          so long as there are time steps remaining AND the combination
c          did not become infeasible maintain energy consumption record
c          keeping
c
         time=0
         do while (time.lt.inth.and.mtchfnd)
           mtchfnd=.false.
           time=time+1
c
c          scan through the combinations available at this time step to
c          see if the one we are looking for (icmbl) is there
c
           cmbctr2=0
           do while (cmbctr2.lt.numfeas(time,iplt).and.
     -               .not.mtchfnd)
           cmbctr2=cmbctr2+1
             icmb=pltopcmb(cmbctr2,time,1,iplt)
             if (icmb.eq.icmbl) then
c
c                    match found begin or continue record keeping on this
combination
c               set match found flag equal to true
c
               mtchfnd=.true.
               numofch=pltopcmb(cmbctr2,time,2,iplt)
               numofbl=pltopcmb(cmbctr2,time,3,iplt)
               do 103 i=1,5+numofch+numofbl
                 pltavoph(feascomh,stdtim,iplt,i)=
     -           pltavoph(feascomh,stdtim,iplt,i)+
     -           pltavop(cmbctr2,time,iplt,i)
                 if (i.le.5) then
                   pltpkoph(feascomh,stdtim,iplt,i)=
     -             amax1(pltpkoph(feascomb,stdtim,iplt,i),
     -                 pltpkop(cmbctr2,time,iplt,i))
                 endif
103            continue
             endif
         enddo
         enddo
         if (.not.mtchfnd) then
c
c             no match found - this combination became infeasible during
c             the time step so reset feascomh to its value before the
c             current combination was started.  Go on to the next
c             combination
```

```
c
            feascomh=feascomh-1
          endif
102     continue
c
c           how many feasible combinations were there this time step?
c
            numfeash(stdtim,iplt)=feascomh
101     continue
        return
        end
*DECK dumpopt
      SUBROUTINE dumpopt
CD    TITLE:=  dumpopt - dump report for hourly averages of
CD             plant energy consumption
CD             for determining optimal equipment schedule
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  MARCH 1995
CD    PURPOSE:=
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c      i
c      iplt
c      time - keeps track of how many sub hourly time steps have elapsed
c
c   INPUTS
c
c      numfeash(HOUR,PLANT#)
c      pltavoph(EQUIP,HOUR,PLANT#,ENERGY TYPE)
c      pltcmbh(EQUIP,HOUR,PLANT#)
c      pltpkoph(EQUIP,HOUR,PLANT#,ENERGY TYPE)
c
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
      integer iplt,i,j
      integer time,numblch
      logical mtchfnd
      real totengy,pkengy
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
c

      INCLUDE 'absenv.inc'
      INCLUDE 'blsfls.inc'
      INCLUDE 'boilrdcl.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'gnrtrdcl.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
```

```
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
      INCLUDE 'tepdcl.inc'
      INCLUDE 'vartic.inc'
      INCLUDE 'zoncon.inc'
c
c*******Main Program*******
c
      do 101 iplt=1,ninpl
c
c       report energy consumption values
c
c       header info
c
         write(output,203)
         write(output,204)
c
c       day loop
c
        do 102 time=1,24
         write(output,*)
         write(output,201) time
c
c       combination info for this time step
c
         do 103 i=1,numfeash(time,iplt)
           totengy=0.0
           pkengy=0.0
           do 104 j=1,4
             totengy=totengy+pltavoph(i,time,iplt,j)
             pkengy=pkengy+pltpkoph(i,time,iplt,j)
104        continue
           write(output,*)
           write(output,202) pltcmbh(i,time,1,iplt),(pltavoph
     -           (i,time,iplt,j),pltpkoph(i,time,iplt,j),j=1,5),
     -           totengy,pkengy
           write(output,*)
103      continue
102     continue
c
c       report plant equipment utilisation
c
c       day loop
c
        do 107 time=1,24
          write(output,*)
          write(output,201) time
c
c       combination info for this time step
c
          do 105 i=1,numfeash(time,iplt)
            write(output,*)
            write(output,205) (pltcmbh(i,time,j,iplt),j=1,3)
            numblch=pltcmbh(i,time,2,iplt)+pltcmbh(i,time,3,iplt)
            if (numblch.gt.0) then
              write(output,206) (pltinoph(i,time,j,1),j=1,numblch)
              write(output,207) (pltavoph(i,time,iplt,j+5),j=1,numblch)
```

```
            endif
105       continue
107     continue
101   continue
c
      return
201   format(t5,'hour of simulation: ',i2)
202   format(t3,i4,t10,6(1x,1p1e9.2,'/',1p1e9.2))
203   format(t3,'plant',t18,'gas',t35,'diesel fuel',t53,
     -'total electric',t74,'boiler fuel',t93,'util. electric',t114,
     -'total energy')
204   format(t2,'comb. #',t15,6('Total/Peak',10x))
205   format(t3,'comb # ',i4,2x,'# of chillers ',i2,2x,
     -'# of boilers ',i2)
206   format(t2,'initial fraction',t20,15(f4.2,1x))
207   format(t2,'average fraction',t20,15(f4.2,1x))
c
c  format for dump report
c
c  plant              gas                   diesel  fuel          total  electric
boiler fuel        util. electric       total energy
c comb. #          Total/Peak              Total/Peak                Total/Peak
Total/Peak           Total/Peak          Total/Peak
c   1234      +1.12E+12/+1.12E+12  +1.12E+12/+1.12E+12  +1.12E+12/+1.12E+12
+1.12E+12/+1.12E+12 +1.12E+12/+1.12E+12 +1.12E+12/+1.12E+12
c
      end
C
*DECK zeropt
      SUBROUTINE zeropt
CD    TITLE:=  zeropt - zero certain plant record keeping variables
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  APRIL 1995
CD    PURPOSE:=
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c      i
c      iplt
c      time - keeps track of how many sub hourly time steps have elapsed
c
c
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
      integer iplt,j
      integer time,cmbctr
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
c

      INCLUDE 'absenv.inc'
      INCLUDE 'blsfls.inc'
```

```
      INCLUDE 'boilrdcl.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'gnrtrdcl.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
      INCLUDE 'tepdcl.inc'
      INCLUDE 'vartic.inc'
      INCLUDE 'zoncon.inc'
c
c*******Main Program*******
c
      do 101 iplt=1,ninpl
        do 102 stpnm=1,nts
          PLTSTP(IPLT,STPNM)=0.0
          PLTEQNO(IPLT,STPNM)=0
          do 103 cmbctr=1,mxncmb
            PLTCMBREC(CMBCTR,IPLT,1,STPNM)=0
            PLTCMBREC(CMBCTR,IPLT,2,STPNM)=0
            PLTCMBREC(CMBCTR,IPLT,3,STPNM)=0
            do 104 j=1,5
              PLTOPREC(CMBCTR,IPLT,j,STPNM)=0.0
104         continue
103       continue
102     continue
101   continue
      return
      end
*DECK OPTIMISE
      SUBROUTINE OPTIMISE(STINC,IPLCTR,PHOVRLD,PCOVRLD,CLPROV,HTPROV)
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  APRIL 1995
CD    PURPOSE:=
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c      i
c      iplt
c      time - keeps track of how many sub hourly time steps have elapsed
c
c
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
c
      INCLUDE 'absenv.inc'
      INCLUDE 'blsfls.inc'
```

```
      INCLUDE 'boilrdcl.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'cool.inc'
      INCLUDE 'gnrtrdcl.inc'
      INCLUDE 'heat.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
      INCLUDE 'tepdcl.inc'
      INCLUDE 'vartic.inc'
      INCLUDE 'zoncon.inc'
      INCLUDE 'nigel.inc'
      INCLUDE 'iceopt.inc'
c
      LOGICAL PHOVRLD,PCOVRLD,PLHEAT,PLCOOL,FCLFLG
      integer iplt,i,j
      integer time,cmbctr,iplctr
      integer iavail,ichlr,maxsz,isize,iblr
      real stinc,clprov,htprov,clngfrac
c
      INTEGER NNVARS,MAXBAS,MAXHES,NNOBJ
      LOGICAL INPRNT,OTPRNT
      CHARACTER*19 TTITLE
      Real*8 VARLB(NVARBS),VARUB(NVARBS),BLCON(NFUN),BUCON(NFUN)
      REAL*8 XX(NVARBS),FCNS(NFUN),RMULTS(NFUN),DEFAUL(19)
      REAL*8 RAMCON(NFUN),RAMVAR(NVARBS),INBIND(NFUN),Z(NCORE)
      REAL*8 NONBAS(NVARBS),REDGR(NVARBS)
      real optchmin(nvarbs),optchmax(nvarbs),optchact(nvarbs)
      real optblmin(nvarbs),optblmax(nvarbs),optblact(nvarbs)
      REAL*8 FPNEWT,FPINIT,FPSTOP,FPSPIV,PPH1EP
      INTEGER NNSTOP,IITLIM,LLMSER,IIPR,IIPN4,IIPN5,
     -        IIPN6,IIPER,IIDUMP,IIQUAD,LDERIV,MMODCG
      INTEGER NBIND,NNONB,INFORM
      integer pltno,nchvarbs,nblvarbs
      logical covrflg
      logical hovrflg
      real sststp,htng,clng,plhtcnst
      COMMON /XFER/ sststp,pltno,covrflg,hovrflg,htng,clng
      COMMON /BOUNDS/ optchmin,optchmax,optblmin,optblmax
c
c*******Main Program*******
c
c        compute cooling load as a fraction of the max available chiller
c        capacity
c
c        Initialisations
c
      sststp=stinc
      pltno=iplctr
      nchvarbs=0
      nblvarbs=0
c
c
      DO 10 ICHLR=1,MXCHTYP
```

```
          MAXSZ=NUMCHSIZ(ICHLR,IPLCTR)
          DO 11 ISIZE = 1,MAXSZ
            DO 12 IAVAIL=1,NCHAVAIL(ISIZE,ICHLR,IPLCTR)
              nchvarbs=nchvarbs+1
              if (cumchlcap.gt.0.0) then
                OPTCHFRAC(IAVAIL,ISIZE,ICHLR)=
     -          CHNOMCAP(ISIZE,ICHLR,IPLCTR)*maxchfr(ichlr,iplctr)
     -          /CUMCHLCAP
              else
                OPTCHFRAC(IAVAIL,ISIZE,ICHLR)=0.0
              endif
12          continue
11        continue
10      continue
        DO 20 IBLR=1,MXBLRTYP
          MAXSZ=NUMBLRSIZ(IBLR,IPLCTR)
          DO 21 ISIZE = 1,MAXSZ
            DO 22 IAVAIL=1,NBLRAVAIL(ISIZE,IBLR,IPLCTR)
              nblvarbs=nblvarbs+1
              IF (CUMBLRCAP.GT.0.0) THEN
                OPTBLFRAC(IAVAIL,ISIZE,IBLR)=
     -          BLRNOMCAP(ISIZE,IBLR,IPLCTR)*maxblrfr(iblr,iplctr)
     -          /CUMBLRCAP
              ELSE
                OPTBLFRAC(IAVAIL,ISIZE,IBLR)=0.0
              ENDIF
22          continue
21        continue
20      continue
c
c          simulate ice storage equipment to meet cooling loads if
c          available
c
        if (callice.and..not.noice) then
          CALL ICESIM(iplctr,stinc)
        endif
c
c          now call the plant simulation routine and simulate
c          chillers, boilers, etc
c
        first=.true.
        CALL SIMPLT(STINC,IPLCTR,PHOVRLD,PCOVRLD,CLPROV,HTPROV)
        pltclng=ecool
        plthtng=eboler
        plhtcnst=consth
c
c          set upper and lower bounds on variables and calculate
c          current values of variables
c
        if (.not.warmup.and.(ecool.gt.0.0.or.eboler.gt.0.0)) then
        first=.false.
        nchvarbs=0
        nblvarbs=0
        do 301 i=1,nvarbs
          varub(i)=0.0
          varlb(i)=0.0
          xx(i)=0.0
301     continue
```

```
c
      DO 30 ICHLR=1,MXCHTYP
        MAXSZ=NUMCHSIZ(ICHLR,IPLCTR)
        DO 31 ISIZE = 1,MAXSZ
          DO 32 IAVAIL=1,NCHAVAIL(ISIZE,ICHLR,IPLCTR)
            nchvarbs=nchvarbs+1
            if (cumchlcap.gt.0.0) then
              optchmin(nchvarbs)=minchfr(ichlr,iplctr)
              optchmax(nchvarbs)=maxchfr(ichlr,iplctr)
              optchact(nchvarbs)=ECOOL*OPTCHFRAC(IAVAIL,ISIZE,ICHLR)
     -                          /CHNOMCAP(ISIZE,ICHLR,IPLCTR)
            else
              OPTCHFRAC(IAVAIL,ISIZE,ICHLR)=0.0
              optchmin(nchvarbs)=0.0
              optchmax(nchvarbs)=0.0
              optchact(nchvarbs)=0.0
            endif
32        continue
31      continue
30    continue
      DO 40 IBLR=1,MXBLRTYP
        MAXSZ=NUMBLRSIZ(IBLR,IPLCTR)
        DO 41 ISIZE = 1,MAXSZ
          DO 42 IAVAIL=1,NBLRAVAIL(ISIZE,IBLR,IPLCTR)
            nblvarbs=nblvarbs+1
            IF (CUMBLRCAP.GT.0.0) THEN
              optblmin(nblvarbs)=minblrfr(iblr,iplctr)
              optblmax(nblvarbs)=maxblrfr(iblr,iplctr)
              optblact(nblvarbs)=EBOLER*OPTBLFRAC(IAVAIL,ISIZE,IBLR)
     -                          /BLRNOMCAP(ISIZE,IBLR,IPLCTR)
            ELSE
              OPTBLFRAC(IAVAIL,ISIZE,IBLR)=0.0
              optblmax(nblvarbs)=0.0
              optblmin(nblvarbs)=0.0
              optblact(nblvarbs)=0.0
            ENDIF
42        continue
41      continue
40    continue
c
c        set up call to grg subroutine using results of this
c        first call to the central plant
c
c        1) define number of variables
c
      nnvars=nchvarbs+nblvarbs
      if (nnvars.gt.0) then
c
c        2) define number of functions - 2 functional constraints +
c          1 objective
c          function #1 - cooling provided to the system
c          function #2 - heating provided to the system
c          function #3 - diff between plant steam required and supplied
c          objective function - total energy consumption
c
c
c        3a) set function upper bounds
c
```

```
      if (pltclng.eq.0.0) then
        blcon(1)=-1.0e31
      else
        blcon(1)=0.0
      endif
      if (plthtng.eq.0.0) then
        blcon(2)=-1.0e31
      else
        blcon(2)=0.0
      endif
      blcon(3)=-1.0e31
c      blcon(3)=0.0
c      blcon(4)=-1.0e31
c
c        3b) set function lower bounds
c
      if (pltclng.eq.0.0) then
        bucon(1)=1.0e31
      else
        bucon(1)=0.0
      endif
      if (plthtng.eq.0.0) then
        bucon(2)=1.0e31
      else
        bucon(2)=0.0
      endif
      bucon(3)=1.0e31
c      bucon(3)=0.0
c      bucon(4)=1.0e31

c
c        4) set variable upper and lower bounds and initialise
c           variable values
c
      do 101 i=1,nchvarbs
c      if (pltclng.eq.0.0) then
c        varub(i)=1.0e31
c        varlb(i)=-1.0e-31
c        xx(i)=0.0
c      else
        varub(i)=optchmax(i)
        varlb(i)=optchmin(i)
        xx(i)=optchact(i)
c      endif
101   continue
      do 102 j=1,nblvarbs
        i=j+nchvarbs
c      if (plthtng.eq.0.0) then
c        varub(i)=1.0e31
c        varlb(i)=-1.0e31
c        xx(i)=0.0
c      else
        varub(i)=optblmax(j)
        varlb(i)=optblmin(j)
        xx(i)=optblact(j)
c      endif
102   continue
```

```
c
c        5) Initialise all other variables
c
      INPRNT=.false.
      OTPRNT=.false.
      MAXBAS=nfun
      MAXHES=nnvars
      NNOBJ=nfun
      TTITLE='NONE'
c---      FPNEWT
      DEFAUL(1)=1.0
c---      FPINIT
      DEFAUL(2)=1.0
c---      FPSTOP
      DEFAUL(3)=1.0
c---      FPSPIV
      DEFAUL(4)=1.0
c---      PPH1EP
      DEFAUL(5)=1.0
c---      NNSTOP
      DEFAUL(6)=1.0
c---      IITLIM
      DEFAUL(7)=1.0
c---      LLMSER
      DEFAUL(8)=1.0
c---      IIPR
      DEFAUL(9)=1.0
c---      IIPN4
      DEFAUL(10)=1.0
c---      IIPN5
      DEFAUL(11)=1.0
c---      IIPN6
      DEFAUL(12)=1.0
c---      IIPER
      DEFAUL(13)=1.0
c---      IIDUMP
      DEFAUL(14)=1.0
      IIQUAD=1
      DEFAUL(15)=0.0
      LDERIV=1
      DEFAUL(16)=0.0
c---      MMODCG
      DEFAUL(17)=1.0
c---      RAMCON and RAMVAR
      DEFAUL(18)=1.0
c---
      DEFAUL(19)=1.0
c
c        6) call grg to begin optimising
c
      CALL GRGSUB(INPRNT,OTPRNT,NCORE,NNVARS,NFUN,MAXBAS,
     1 MAXHES,NNOBJ,TTITLE,VARLB,VARUB,BLCON,BUCON,DEFAUL,FPNEWT,FPINIT,
     2 FPSTOP,FPSPIV,PPH1EP,NNSTOP,IITLIM,LLMSER,IIPR,IIPN4,IIPN5,
     3 IIPN6,IIPER,IIDUMP,IIQUAD,LDERIV,MMODCG,
     4 RAMCON,RAMVAR,XX,FCNS,INBIND,RMULTS,NONBAS,REDGR,
     5 NBIND,NNONB,INFORM,Z)
c
c       7) continue on as normal
```

```
c
      htprov=htng
      clprov=clng
      pcovrld=covrflg
      phovrld=hovrflg
      numofch=nchvarbs
      numofbl=nblvarbs
      do 501 i=1,nchvarbs
        optchfin(i)=xx(i)
501   continue
      do 502 j=1,nblvarbs
        i=j+nchvarbs
        optblfin(j)=xx(i)
502   continue
      endif
c
      endif
c
      return
      end
*DECK GCOMP
      SUBROUTINE GCOMP(G,X)
C
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  APRIL 1995
CD    PURPOSE:= Evaluate objective function for grg
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c      i
c      iplt
c      time - keeps track of how many sub hourly time steps have elapsed
c
c
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
      INCLUDE 'absenv.inc'
      INCLUDE 'blsfls.inc'
      INCLUDE 'boilrdcl.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'cool.inc'
      INCLUDE 'gnrtrdcl.inc'
      INCLUDE 'heat.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
```

```
      INCLUDE 'tepdcl.inc'
      INCLUDE 'vartic.inc'
      INCLUDE 'zoncon.inc'
      INCLUDE 'nigel.inc'
c
c     Local variables
c
      INTEGER M,N,MP1,NPMP1,NBMAX,NNBMAX,NPNBMX,MPNBMX,NRTOT
      COMMON /DIMEN/ M,N,MP1,NPMP1,NBMAX,NNBMAX,NPNBMX,MPNBMX,NRTOT
      COMMON /XFER/ sststp,pltno,covrflg,hovrflg,htng,clng
      COMMON /BOUNDS/ optchmin,optchmax,optblmin,optblmax
      real*8 g(MP1),x(NPMP1)
      integer pltno,nvars,isize,maxsz,iblr,iavail,ichlr,nch
      logical covrflg
      logical hovrflg
      real sststp,htng,clng,sumxch,sumxbl,pkYorN
      real optchmin(nvarbs),optchmax(nvarbs)
      real optblmin(nvarbs),optblmax(nvarbs)
c
c     initialisations
c
      covrflg=.false.
      hovrflg=.false.
      nvars=0
      sumxch=0.0
      sumxbl=0.0
c
c        first transfer x's to correct simulation variables
c
      nch=0
      DO 10 ICHLR=1,MXCHTYP
        MAXSZ=NUMCHSIZ(ICHLR,PLTNO)
        DO 11 ISIZE = 1,MAXSZ
          DO 12 IAVAIL=1,NCHAVAIL(ISIZE,ICHLR,PLTNO)
            nvars=nvars+1
            x(nvars)=dble(amin1(amax1(optchmin(nvars),sngl(x(nvars))),
     -                optchmax(nvars)))
            OPTCHFRAC(IAVAIL,ISIZE,ICHLR)=x(nvars)
            sumxch=sumxch+x(nvars)
12        continue
11      continue
10    continue
      nch=nvars
      DO 20 IBLR=1,MXBLRTYP
        MAXSZ=NUMBLRSIZ(IBLR,PLTNO)
        DO 21 ISIZE = 1,MAXSZ
          DO 22 IAVAIL=1,NBLRAVAIL(ISIZE,IBLR,PLTNO)
            nvars=nvars+1
            x(nvars)=dble(amin1(amax1(optblmin(nvars-nch),
     -               sngl(x(nvars))),optchmax(nvars-nch)))
            OPTBLFRAC(IAVAIL,ISIZE,IBLR)=x(nvars)
            sumxbl=sumxbl+x(nvars)
22        continue
21      continue
20    continue
c
c       next call simulation and simulate
c        chillers, boilers, etc
```

```
c
      CALL SIMPLT(sststp,pltno,hovrflg,covrflg,clng,htng)
c
c      evaluate objective function and functional constraints
c
      if (stdtim.ge.bgnonpk.and.stdtim.le.endonpk) then
         pkYorN=onpkmult
      else
         pkYorN=1.0
      endif
      g(1)=clldrem
      g(2)=htldrem
      g(3)=cstfac(1)*EFUELG(PLTNO)+cstfac(2)*EFUELD(PLTNO)+
     -     pkYorN*cstfac(3)*EELECT(PLTNO)+cstfac(4)*EFUELB(PLTNO)
c      g(3)=stmdiff
c      g(4)=EFUELG(PLTNO)+EFUELD(PLTNO)+EELECT(PLTNO)+EFUELB(PLTNO)
c
      return
      end
*DECK PLTOPSCHD
      SUBROUTINE PLTOPSCHD
CD    TITLE:=  PLTOPSCHD - GENERATE PLANT OPTIMAL EQUIPMENT SCHEDULE
CD             BASED ON RESULTS OF HOURLY PLANT OPTIMISATIONS
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  MAY 1995
CD    PURPOSE:=
CD
CD   Routine to determine optimal schedule from feasible
CD   operating combinations of plant equipment
CD
CD    OUTLINE:
CD
CD   1. First - determine schedule which minimises total energy consumption
CD            or cost and determine the peak energy consumption associated
CD            with this path for each energy type (some energy types will
CD            not incur a penalty for high peak usage
CD   2. Next -  Eliminate nodes which have a higher peak consumption than
CD            the least consumption path
CD   3. Next -  For the remaining feasible nodal points calculate the
CD            peak energy consumption associated transitions from the
CD            nodes at one time step to the nodes at the next step
CD   4. Next -  Eliminate nodal transitions with a higher peak consumption
CD            than the lowest cost path
CD   5. Next -  Compute the total consumption and associated peak for all
CD            paths through the remaining nodal points (hopefully not
CD            many and compute the energy cost
CD   6. Finally - Compare the cost of each path to the least consumption
CD              path and the others to determine the path with the
CD              global minimum cost.
CD
CD    VARIABLE DICTIONARY:=
c      inputs
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
```

```
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pathfind.inc'
      INCLUDE 'blsfls.inc'
      INCLUDE 'runctl.inc'
c
C     VARIABLES
c
c***** Variable Dictionary **********
c
c       i,j,j2,iplt,time,tmtim - loop counters
c
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 1 - find path with lowest energy consumption or lowest cost
c              when peak consumption is neglected
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      INTEGER iplt,time,i,j,sboptcb(nhrsindy,mxplts,3),sboppkdt,
     -        curcmb,newcmb,eqpdiff(mxneq),timofdypk(mxplts),j1,
     -        neweq(mxneq),cureq(mxneq),sbpktrtim,sseqdiff(mxneq)
      REAL sboptot(nhrsindy,mxplts),sboptpk(nhrsindy,mxplts),totcost,
     -     pkengy,sboppkdy,neweqop(mxneq),pkcondy,cureqop(mxneq),
     -     sboptcst,subpeakt,sboppktr(nhrsindy),sbopten(nhrsindy,mxplts)
     -    ,totengy,sbopchen(nhrsindy),sbopblen(nhrsindy),pkYorN
c
c     cureq - array that indicates which of the available
c             equipment is turned on at the current time
c     neweq - array that indicates which of the available equipment
c             is turned on one time step later than cureq
c    curcmb - pointer to the current combination used to evaluate cureq
c    newcmb - pointer to the combination used to evaluate neweq
c   neweqop - array of energy consumption for each piece of equipment,
c             ordered so that it can be directly related to a specific
c             equipment combination
c   sboptot - sub optimal total cost of operating plant equipment at this
c             time step
c   sboptpk - peak electric consumption on suboptimal path
c   sboptcb - pointer to suboptimal combination at this time step
c sbpktrtim - time at which peak electric consumption due to change in
c             equipment usage occurs
c  sboppkdy - peak steady state electric consumption
c  sboppkdt - time at which peak ss electric consumption occurs
c  sboppktr - peak electric consumption due to equipment change at given
c             time step
c  subpeakt - actual peak electric consumption due to equipment combination
c             change for the whole day
c   eqpdiff - calculated by taking neweq-cureq (vectorially)
c  sseqdiff -
c   totcost - interim variable of total hourly energy cost
c    pkengy - interim variable of peak energy consumption during an hour
c   sboptin -
c   pkcondy -
c timofdypk -
c  sboptcst -
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     STEP 2 - Eliminate all nodal points with higher peak consumption during
c              on-peak hours than the lowest consumption path.
```

```
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      INTEGER noinfeascb(nhrsindy),combinfeas(nhrsindy,nfsph)
      REAL chcap,blcap
c
c  noinfeascb -
c  combinfeas -
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     STEP 3 - Calculate peak utility electric consumption associated with
c              all possible remaining transitions
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      LOGICAL feasflg1,feasflg2
      INTEGER i2,i3,j2,noftrans(nhrsindy),itrans(nhrsindy,400,2),
     -        nchdiff,nbldiff
      REAL pkelec,rtrans(nhrsindy,400),newchcp,newblcp,curchcp,
     -     curblcp
c
c  feasflg1   -
c  feasflg2   -
c  noftrans   -
c  pkelec     -
c  itrans     -
c  rtrans     -
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 4 - Eliminate transitions with higher peak consumption than lowest
c               cost path
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      INTEGER noinfstrn(nhrsindy),transinfeas(nhrsindy,nfstph)
      REAL chenchng,blenchng,delchcap,delblcap
c
c  noinfstrn   -
c  transinfeas -
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 5 - Compute total cost associated with all remaining possible paths
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      INTEGER nofpaths,pthcmb(nhrsindy,mxpth,mxplts),icomb,newnofpaths,
     -        tmpcmb(nhrsindy,mxpth),tmtim,ccomb,fincmb
      REAL pktrans(nhrsindy,mxpth,mxplts),tmppktr(nhrsindy,mxpth),
c    -     sscost(mxpth),sspeak(mxpth),trpeak(mxpth),pthpeak(mxpth),
     -    sscost,sspeak,trpeak,pthpeak(mxpth),
     -    pthcst(mxpth),totfincst,finpktr,fincost,finpeak,delcost
c
c  nofpaths   -
c  pthcmb     -
c  pktrans    -
c  icomb      -
c  newnofpaths -
c  tmpcmb     -
c  tmppktr    -
c  sscost     -
c  sspeak     -
c  trpeak     -
c  ccomb      -
c  pthpeak    -
c  pthcst     -
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 6 - Search for global minimum lowest cost path
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      INTEGER optpth,pth
```

```
      REAL optcost,optpeak,costtot
c
c  optcost    -
c  optpeak    -
c  optpth     -
c
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 7 - Report result of optimisation
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL trnspk
c
c  trnspk    -
c
c*******Main Program*******
c
      do 101 iplt=1,ninpl
c
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 1 - find path with lowest energy consumption or lowest cost
c              when peak consumption is neglected
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c       day loop
c
        do 102 time=1,24
          if (time.ge.bgnonpk.and.time.le.endonpk) then
            pkYorN=onpkmult
          else
            pkYorN=1.0
          endif
c
c       combination info for this time step
c
         sboptot(time,iplt)=1.0e20
         sboptpk(time,iplt)=0.0
         do 103 i=1,numfeash(time,iplt)
           totcost=0.0
           totengy=0.0
           pkengy=0.0
           do 104 j=1,4
            if (j.eq.3) then
             totcost=totcost+pkYorN*cstfac(j)*pltavoph(i,time,iplt,j)
            else
             totcost=totcost+cstfac(j)*pltavoph(i,time,iplt,j)
            endif
             totengy=totengy+pltavoph(i,time,iplt,j)
             if (j.eq.3) then
c
c            we are only interested in peak ELECTRIC consumption between
c            specified on and off peak hours
c
               if (time.ge.bgnonpk.and.time.le.endonpk) then
                 pkengy=pltpkoph(i,time,iplt,j)
               endif
             endif
104        continue
           if (totcost.lt.sboptot(time,iplt)) then
            sboptot(time,iplt)=totcost
```

```
            sbopten(time,iplt)=totengy
            sboptpk(time,iplt)=pkengy
            do 105 j=1,3
             sboptcb(time,iplt,j)=pltcmbh(i,time,j,iplt)
105         continue
            do 106 j=1,sboptcb(time,iplt,2)+sboptcb(time,iplt,3)
             sboptin(time,j,1)=pltinoph(i,time,j,1)
             sboptin(time,j,2)=pltinoph(i,time,j,2)
106         continue
            sbopchen(time)=0.0
            do 107 j=1,sboptcb(time,iplt,2)
             sbopchen(time)=sbopchen(time)+pltinoph(i,time,j,3)
107         continue
            sbopblen(time)=0.0
            do 108 j=1,sboptcb(time,iplt,3)
             j1=j+sboptcb(time,iplt,2)
             sbopblen(time)=sbopblen(time)+pltinoph(i,time,j1,3)
108         continue
           endif
103      continue
102     continue
c
c       determine peak steady state electric consumption for the day
c
        sboppkdy=0.0
        do 113 time=bgnonpk,endonpk
          sboppkdy=amax1(sboppkdy,sboptpk(time,iplt))
          if (sboptpk(time,iplt).eq.sboppkdy) then
            sboppkdt=time
          endif
113     continue
c
c       compute peak consumptions based on equipment transitions from
c       time step to time step
c
        curcmb=sboptcb(1,iplt,1)
        call eqptrans(curcmb,time,cureq,cureqop,iplt)
        subpeakt=0.0
        sbpktrtim=25
c
        do 111 time=2,24
          sboppktr(time)=0.0
          newcmb=sboptcb(time,iplt,1)
c
c         if the equipment combination changed then we need to see if
c         anything was turned on
c
            call eqptrans(newcmb,time,neweq,neweqop,iplt)
c
c          if (curcmb.eq.newcmb) then
c            do 121 i=1,nalleqp(iplt)
c              sboppktr(time)=sboppktr(time)+neweqop(i)
c121         continue
c            sboppktr(time)=pltinelch(newcmb,time)
c          else
          do 121 i=1,numfeash(time,iplt)
            if(pltcmbh(i,time,1,iplt).eq.newcmb) then
              sboppktr(time)=pltinelch(i,time)
```

```
          endif
121     continue
        if (curcmb.ne.newcmb) then
c
c        take the difference between the current combination and the previous
c        one i.e. literally neweq-cureq
c
          do 112 i=1,nalleqp(iplt)
            eqpdiff(i)=amax0(0,neweq(i)-cureq(i))
c             sseqdiff(i)=amax0(0,neweq(i)-eqpdiff(i))
c
c        compute total peak electric consumption due to startup of equipment
c        1.2 is an empirical factor to account for the startup transient
c        based on observed data
c
c             sboppktr(time)=sboppktr(time)+
c    -          float(eqpdiff(i))*neweqop(i)*1.2
c    -                           +sseqdiff(i)*neweqop(i)
            sboppktr(time)=sboppktr(time)+
     -          float(eqpdiff(i))*neweqop(i)*0.2
c
c        transfer neweq to cureq for next time step
c
            cureq(i)=neweq(i)
c
112       continue
        endif
        if (time.ge.bgnonpk.and.time.le.endonpk) then
          subpeakt=amax1(sboppktr(time),subpeakt)
          if (subpeakt.eq.sboppktr(time)) then
             sbpktrtim=time
          endif
        endif
        curcmb=newcmb
111   continue
c
c      calculate overall peak for the day
c
      pkcondy=amax1(subpeakt,sboppkdy)
      if (pkcondy.eq.subpeakt) then
         timofdypk(iplt)=sbpktrtim
      else
         timofdypk(iplt)=sboppkdt
      endif
c
c      report sub-optimal path
c
    sboptcst=0.0
    write (output,130)
    write (output,*)
    write (output,131)
    write (output,135)
    write (output,*)
    do 120 time=1,24
      write(output,132) time,sboptcb(time,iplt,1),sbopten(time,iplt),
     -           sboptot(time,iplt),sboptpk(time,iplt),sboppktr(time)
      sboptcst=sboptcst+sboptot(time,iplt)
120 continue
```

```
      sboptcst=sboptcst+dmndfac*pkcondy
      write (output,*)
      write(output,133) sboptcst
c
c       if utility imposes demand charges look for improvement on
c       suboptimal path
c
      if (bgnonpk.lt.endonpk.and.bgnonpk.lt.24.and.endonpk.lt.24) then
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     STEP 2 - Eliminate all nodal points with higher peak consumption during
c              on-peak hours than the lowest consumption path.
c              Also eliminate nodes which have capacity but no consumption
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        do 201 time=bgnonpk,endonpk
          noinfeascb(time)=0
          do 203 i=1,numfeash(time,iplt)
            if (pltpkoph(i,time,iplt,3).gt.pkcondy) then
c
c         this is an infeasible combination
c
              noinfeascb(time)=noinfeascb(time)+1
              combinfeas(time,noinfeascb(time))=i
            else
              curcmb=pltcmbh(i,time,1,iplt)
              curchcp=eqpcomb(curcmb,neqtot(iplt)+1,iplt)
              curblcp=eqpcomb(curcmb,neqtot(iplt)+2,iplt)
              do 204 i2=1,numfeash(time,iplt)

c
c             if curcmb has boiler or chiller capacity but
c             a feasible combination exists without
c             boiler or chiller capacity then disallow it
c
                if (i2.ne.i) then
                  newcmb=pltcmbh(i2,time,1,iplt)
                  newchcp=eqpcomb(newcmb,neqtot(iplt)+1,iplt)
                  newblcp=eqpcomb(newcmb,neqtot(iplt)+2,iplt)
                  if (curchcp.eq.newchcp.and.curblcp.gt.0.0.and.newblcp
     -            .eq.0.0) then
                    nchdiff=0
                    do 205 i3=1,nchts(iplt)
                      nchdiff=nchdiff+abs(eqpcomb(newcmb,i3,iplt)-
     -                        eqpcomb(curcmb,i3,iplt))
205                 continue
                    if (nchdiff.eq.0) then
                       noinfeascb(time)=noinfeascb(time)+1
                       combinfeas(time,noinfeascb(time))=i
                    endif
                  elseif (curblcp.eq.newblcp.and.curchcp.gt.0.0.and.
     -              newchcp.eq.0.0) then
                    nchdiff=0
                    do 206 i3=nchts(iplt)+1,nchts(iplt)+nblts(iplt)
                      nbldiff=nbldiff+abs(eqpcomb(newcmb,i3,iplt)-
     -                        eqpcomb(curcmb,i3,iplt))
206                 continue
                    if (nbldiff.eq.0) then
                       noinfeascb(time)=noinfeascb(time)+1
                       combinfeas(time,noinfeascb(time))=i
```

```
               endif
             endif
           endif
204        continue
         endif
203      continue
       write(*,*) time,numfeash(time,iplt),noinfeascb(time)
201    continue
      write (*,*)
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     STEP 3 - Calculate peak utility electric consumption associated with
c              all possible remaining transitions
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      write(output,138)
        noftrans(bgnonpk)=0.0
        curcmb=sboptcb(bgnonpk-1,iplt,1)
        call eqptrans(curcmb,time-1,cureq,cureqop,iplt)
        do 293 j=1,numfeash(bgnonpk,iplt)
          feasflg2=.true.
          if (noinfeascb(bgnonpk).ge.1) then
            do 296 j2=1,noinfeascb(bgnonpk)
              if(j.eq.combinfeas(bgnonpk,j2))
     -          feasflg2=.false.
296         continue
          endif
c
c           if feasible --> continue
c
          if (feasflg2) then
            newcmb=pltcmbh(j,bgnonpk,1,iplt)
            noftrans(bgnonpk)=noftrans(bgnonpk)+1
c
c         if the equipment combination changed then we need to see if
c         anything was turned on
c
            call eqptrans(newcmb,bgnonpk,neweq,neweqop,iplt)
c
            pkelec=0.0
c            if (curcmb.eq.newcmb) then
c              do 299 i2=1,nalleqp(iplt)
c                pkelec=pkelec+neweqop(i2)
c299           continue
c              pkelec=pltinelch(newcmb,time)
c            else
c              pkelec=pltinelch(newcmb,time)
            do 299 i2=1,numfeash(time,iplt)
              if(pltcmbh(i2,time,1,iplt).eq.newcmb) then
                pkelec=pltinelch(i2,time)
              endif
299         continue
            if (curcmb.ne.newcmb) then
c
c         take the difference between the current combination and the previous
c         one i.e. literally neweq-cureq
c
              do 297 i2=1,nalleqp(iplt)
                eqpdiff(i2)=amax0(0,neweq(i2)-cureq(i2))
c                 sseqdiff(i2)=amax0(0,neweq(i2)-eqpdiff(i2))
```

```
c
c        compute total peak electric consumption due to startup of equipment
c        1.2 is an empirical factor to account for the startup transient
c        based on observed data
c
c                 pkelec=pkelec+float(eqpdiff(i2))*neweqop(i2)*1.2
c     -                       +sseqdiff(i2)*neweqop(i2)
                 pkelec=pkelec+float(eqpdiff(i2))*neweqop(i2)*0.2
297            continue
             endif
c
c         record peak consumption and transition states
c
             itrans(bgnonpk,noftrans(bgnonpk),1)=curcmb
             itrans(bgnonpk,noftrans(bgnonpk),2)=newcmb
             rtrans(bgnonpk,noftrans(bgnonpk))=pkelec
           endif
293      continue
      write(*,*) bgnonpk,noftrans(bgnonpk)
c
      do 301 time=bgnonpk+1,endonpk
        noftrans(time)=0.0
        do 302 i=1,numfeash(time-1,iplt)
c
c          look to see if this is a feasible node
c
          feasflg1=.true.
          if (noinfeascb(time-1).ge.1) then
            do 305 j=1,noinfeascb(time-1)
              if(i.eq.combinfeas(time-1,j))
     -           feasflg1=.false.
305         continue
          endif
c
c          if feasible --> continue
c
          if (feasflg1) then
            curcmb=pltcmbh(i,time-1,1,iplt)
            call eqptrans(curcmb,time-1,cureq,cureqop,iplt)
            do 303 j=1,numfeash(time,iplt)
              feasflg2=.true.
              if (noinfeascb(time).ge.1) then
                do 306 j2=1,noinfeascb(time)
                  if(j.eq.combinfeas(time,j2))
     -               feasflg2=.false.
306             continue
              endif
c
c          if feasible --> continue
c
              if (feasflg2) then
                newcmb=pltcmbh(j,time,1,iplt)
                noftrans(time)=noftrans(time)+1
c
c        if the equipment combination changed then we need to see if
c        anything was turned on
c
                call eqptrans(newcmb,time,neweq,neweqop,iplt)
```

```
c
                pkelec=0.0
c                if (curcmb.eq.newcmb) then
c                  do 309 i2=1,nalleqp(iplt)
c                    pkelec=pkelec+neweqop(i2)
c309              continue
c                pkelec=pltinelch(newcmb,time)
c                else
c                pkelec=pltinelch(newcmb,time)
                do 309 i2=1,numfeash(time,iplt)
                  if(pltcmbh(i2,time,1,iplt).eq.newcmb) then
                    pkelec=pltinelch(i2,time)
                  endif
309             continue
                if (curcmb.ne.newcmb) then
c
c         take the difference between the current combination and the previous
c         one i.e. literally neweq-cureq
c
                  do 307 i2=1,nalleqp(iplt)
                    eqpdiff(i2)=amax0(0,neweq(i2)-cureq(i2))
c                     sseqdiff(i2)=amax0(0,neweq(i2)-eqpdiff(i2))
c
c         compute total peak electric consumption due to startup of equipment
c         1.2 is an empirical factor to account for the startup transient
c         based on observed data
c
c                     pkelec=pkelec+float(eqpdiff(i2))*neweqop(i2)*1.2
c     -                        +sseqdiff(i2)*neweqop(i2)
                    pkelec=pkelec+float(eqpdiff(i2))*neweqop(i2)*0.2
307               continue
                endif
c
c          record peak consumption and transition states
c
                itrans(time,noftrans(time),1)=curcmb
                itrans(time,noftrans(time),2)=newcmb
                rtrans(time,noftrans(time))=pkelec
              endif
303         continue
          endif
302     continue
      write(output,136) time,noftrans(time)
      do 310 i=1,noftrans(time)
        write(output,137) itrans(time,i,1),itrans(time,i,2),
     -                     rtrans(time,i)
310   continue
301   continue
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 4 - Eliminate transitions with higher peak consumption than lowest
c              cost path
c     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      do 401 time=bgnonpk,endonpk
c
c       what is trend in energy consumption?
c
        chenchng=sbopchen(time)-sbopchen(time-1)
        blenchng=sbopblen(time)-sbopblen(time-1)
```

```
c
c       if engychng is + disallow transitions to equipment with less
c       capacity
c       if engychng is - disallow transitions to combinations with more
c       capacity
c
        noinfstrn(time)=0
        do 402 i=1,noftrans(time)
          curcmb=itrans(time,i,1)
          newcmb=itrans(time,i,2)
          delchcap=eqpcomb(newcmb,neqtot(iplt)+1,iplt)-
     -              eqpcomb(curcmb,neqtot(iplt)+1,iplt)
          delblcap=eqpcomb(newcmb,neqtot(iplt)+2,iplt)-
     -              eqpcomb(curcmb,neqtot(iplt)+2,iplt)
          if (rtrans(time,i).gt.pkcondy) then
            noinfstrn(time)=noinfstrn(time)+1
            transinfeas(time,noinfstrn(time))=i
          elseif (chenchng.ge.0.0.and.delchcap.lt.0.0) then
            noinfstrn(time)=noinfstrn(time)+1
            transinfeas(time,noinfstrn(time))=i
          elseif (chenchng.le.0.0.and.delchcap.gt.0.0) then
            noinfstrn(time)=noinfstrn(time)+1
            transinfeas(time,noinfstrn(time))=i
          elseif (blenchng.ge.0.0.and.delblcap.lt.0.0) then
            noinfstrn(time)=noinfstrn(time)+1
            transinfeas(time,noinfstrn(time))=i
          elseif (blenchng.le.0.0.and.delblcap.gt.0.0) then
            noinfstrn(time)=noinfstrn(time)+1
            transinfeas(time,noinfstrn(time))=i
          endif
402     continue
      write(*,*) time,noftrans(time),noinfstrn(time)
401   continue
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 5 - Compute total cost associated with all remaining possible paths
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     If the time is off-peak then use the suboptimal lowest consumption path
c     because peak demand is irrelevant.  If time is on-peak then need to
c     compute consumption and peak associated with all possible remaining
c     paths.
      pth=0
      do 503 i=1,noftrans(bgnonpk)
        feasflg1=.true.
        j=1
        do while (j.le.noinfstrn(bgnonpk).and.feasflg1)
c        do 504 j=1,noinfstrn(bgnonpk)
c          feasflg1=.true.
          if (i.eq.transinfeas(bgnonpk,j)) then
            feasflg1=.false.
          endif
          j=j+1
        enddo
c504     continue
        if (feasflg1) then
          pth=pth+1
c
c           compute energy consumption/cost, peak consumption, node
c           transitions
```

```
c
          pthcmb(bgnonpk-1,pth,iplt)=itrans(bgnonpk,i,1)
          pthcmb(bgnonpk,pth,iplt)=itrans(bgnonpk,i,2)
          pktrans(bgnonpk,pth,iplt)=rtrans(bgnonpk,i)
        endif
503   continue
      nofpaths=pth
      do 501 time=bgnonpk+1,endonpk-1
        newnofpaths=0
        do 505 pth=1,nofpaths
          icomb=pthcmb(time-1,pth,iplt)
          do 506 i=1,noftrans(time)
            feasflg1=.true.
            j=1
            do while (j.le.noinfstrn(time).and.feasflg1)
c            do 507 j=1,noinfstrn(time)
c              feasflg1=.true.
             if (i.eq.transinfeas(time,j)) then
               feasflg1=.false.
             endif
             j=j+1
            enddo
c507          continue
            if (feasflg1) then
c
c           look for a transition with a starting point equal to the previous
c           endpoint
c
              if (itrans(time,i,1).eq.icomb) then
                newnofpaths=newnofpaths+1
                tmpcmb(bgnonpk-1,newnofpaths)=pthcmb(bgnonpk-1,pth,iplt)
                do 508 tmtim=bgnonpk,time-1
                  tmpcmb(tmtim,newnofpaths)=pthcmb(tmtim,pth,iplt)
                  tmppktr(tmtim,newnofpaths)=pktrans(tmtim,pth,iplt)
508             continue
                tmpcmb(time,newnofpaths)=itrans(time,i,2)
                tmppktr(time,newnofpaths)=rtrans(time,i)
              endif
            endif
506       continue
505     continue
c
c           transfer path information from temporary arrays
c
        nofpaths=newnofpaths
        do 509 pth=1,nofpaths
          pthcmb(bgnonpk-1,pth,iplt)=tmpcmb(bgnonpk-1,pth)
          do 510 tmtim=bgnonpk,time
            pthcmb(tmtim,pth,iplt)=tmpcmb(tmtim,pth)
            pktrans(tmtim,pth,iplt)=tmppktr(tmtim,pth)
510       continue
509     continue
        write(*,*) time,nofpaths
501   continue
c
c     compute total energy consumption associated with each path
c     and evaluate peak consumption
c
```

```
      do 511 pth=1,nofpaths
        sscost=0.0
        sspeak=0.0
        trpeak=0.0
        pthcst(pth)=0.0
        do 512 time=bgnonpk-1,endonpk-1
          if (time.ge.bgnonpk) then
            pkYorN=onpkmult
          else
            pkYorN=1.0
          endif
          icomb=pthcmb(time,pth,iplt)
          if (time.ge.bgnonpk) then
            trpeak=amax1(trpeak,pktrans(time,pth,iplt))
          endif
          do 513 i=1,numfeash(time,iplt)
            ccomb=pltcmbh(i,time,1,iplt)
            if (ccomb.eq.icomb) then
              do 514 j=1,4
               if (j.eq.3) then
                sscost=sscost+pkYorN*cstfac(j)*
     -                      pltavoph(i,time,iplt,j)
                sspeak=amax1(sspeak,pltpkoph(i,time,iplt,j))
               else
                sscost=sscost+cstfac(j)*
     -                      pltavoph(i,time,iplt,j)
               endif
514           continue
            endif
513       continue
512     continue
c
c         include the costs of the last on-peak hour
c
          pthcst(pth)=1.0e+30
          icomb=pthcmb(endonpk-1,pth,iplt)
          do 516 i=1,noftrans(endonpk)
            feasflg1=.true.
            j=1
            do while (j.le.noinfstrn(endonpk).and.feasflg1)
              if (i.eq.transinfeas(endonpk,j)) then
                feasflg1=.false.
              endif
              j=j+1
            enddo
            if (feasflg1) then
c
c           look for a transition with a starting point equal to the previous
c           endpoint
c
              if (itrans(endonpk,i,1).eq.icomb) then
                fincmb=itrans(endonpk,i,2)
                finpktr=rtrans(endonpk,i)
                do 523 i2=1,numfeash(endonpk,iplt)
                  ccomb=pltcmbh(i2,endonpk,1,iplt)
                  if (ccomb.eq.fincmb) then
                    delcost=0.0
                    do 524 j2=1,4
```

```
                    if (j2.eq.3) then
                      delcost=delcost+onpkmult*cstfac(j2)*
     -                        pltavoph(i2,endonpk,iplt,j2)
                      finpeak=amax1(sspeak,
     -                              pltpkoph(i2,endonpk,iplt,j2))
                    else
                      delcost=delcost+cstfac(j2)*
     -                        pltavoph(i2,endonpk,iplt,j2)
                    endif
524               continue
                  fincost=sscost+delcost
                  totfincst=fincost+dmndfac*amax1
     -              (amax1(finpeak,sspeak),amax1(trpeak,finpktr))
                  if(totfincst.lt.pthcst(pth)) then
                    pthpeak(pth)=amax1(amax1(finpeak,sspeak),
     -                           amax1(trpeak,finpktr))
                    pthcst(pth)=totfincst
                    pthcmb(endonpk,pth,iplt)=fincmb
                  endif
                endif
523           continue
            endif
          endif
516     continue
511   continue
c      do 511 pth=1,nofpaths
c        sscost(pth)=0.0
c        sspeak(pth)=0.0
c        trpeak(pth)=0.0
c        do 512 time=bgnonpk-1,endonpk
c          if (time.ge.bgnonpk.and.time.le.endonpk) then
c            pkYorN=onpkmult
c          else
c            pkYorN=1.0
c          endif
c          icomb=pthcmb(time,pth,iplt)
c          if (time.ge.bgnonpk) then
c            trpeak(pth)=amax1(trpeak(pth),pktrans(time,pth,iplt))
c          endif
c          do 513 i=1,numfeash(time,iplt)
c            ccomb=pltcmbh(i,time,1,iplt)
c            if (ccomb.eq.icomb) then
c              do 514 j=1,4
c                if (j.eq.3) then
c                 sscost(pth)=sscost(pth)+pkYorN*cstfac(j)*
c    -                        pltavoph(i,time,iplt,j)
c                 sspeak(pth)=amax1(sspeak(pth),pltpkoph(i,time,iplt,j))
c                else
c                 sscost(pth)=sscost(pth)+cstfac(j)*
c    -                        pltavoph(i,time,iplt,j)
c                endif
c514          continue
c            endif
c513        continue
c512      continue
c        pthpeak(pth)=amax1(sspeak(pth),trpeak(pth))
c        pthcst(pth)=sscost(pth)+dmndfac*pthpeak(pth)
c511   continue
```

```
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 6 - Search for global minimum lowest cost path
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      optcost=pthcst(1)
      optpeak=pthpeak(1)
      optpth=1
      do 601 pth=2,nofpaths
        if (pthcst(pth).lt.optcost) then
c
c       a lower cost path exists
c
          optcost=pthcst(pth)
          optpeak=pthpeak(pth)
          optpth=pth
        endif
601   continue
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Step 7 - Report results of search process
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     From hour 1 to the beginning of on-peak the optimal path is the
c     same as the sub-optimal path based on the lowest cost neglecting
c     peak consumption and the effect of demand charges
c
      write (output,140)
      write (output,*)
      write (output,131)
      write (output,135)
      write (output,*)
      costtot=0.0
      do 701 time=1,bgnonpk-1
        write(output,134) time,sboptcb(time,iplt,1),sbopten(time,iplt),
     -                   sboptot(time,iplt),sboptpk(time,iplt)
      costtot=costtot+sboptot(time,iplt)
701   continue
c
c     During the on-peak hours use the optimal path developed in the previous
c     steps
      pth=optpth
      do 702 time=bgnonpk,endonpk
        pkYorN=onpkmult
        do 705 i=1,numfeash(time,iplt)
          if(pthcmb(time,pth,iplt).eq.pltcmbh(i,time,1,iplt)) then
           if(time.gt.bgnonpk) then
             do 710 i2=1,noftrans(time)
               if(pthcmb(time-1,pth,iplt).eq.itrans(time,i2,1).and.
     -            pthcmb(time,pth,iplt).eq.itrans(time,i2,2)) then
                trnspk=rtrans(time,i2)
               endif
710          continue
            endif
            totcost=0.0
            totengy=0.0
            pkengy=0.0
            do 706 j=1,4
             totengy=totengy+pltavoph(i,time,iplt,j)
             if (j.eq.3) then
              totcost=totcost+pkYorN*cstfac(j)*pltavoph(i,time,iplt,j)
             else
```

```
             totcost=totcost+cstfac(j)*pltavoph(i,time,iplt,j)
            endif
c
c           we are only interested in peak ELECTRIC consumption between
c           specified on and off peak hours
c
706        continue
           pkengy=pltpkoph(i,time,iplt,3)
         endif
705     continue
        write(output,132) time,pthcmb(time,pth,iplt),totengy,totcost,
     -                    pkengy,trnspk
       costtot=costtot+totcost
702   continue
c
c     after the end of the on-peak hours revert to the suboptimal path
c
      do 703 time=endonpk+1,24
        write(output,134) time,sboptcb(time,iplt,1),sbopten(time,iplt),
     -                    sboptot(time,iplt),sboptpk(time,iplt)
      costtot=costtot+sboptot(time,iplt)
703   continue
      costtot=costtot+dmndfac*optpeak
      write(output,*)
      write(output,133) costtot
c
c     report total energy consumption by energy type, total cost, peak
c     utility electric
c
c     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      endif
101   continue
      RETURN
130   format(t5,'SUBOPTIMAL PATH - LEAST COST IGNORING PEAK DEMAND')
131   format(t2,'TIME',t9,'COMB.',t16,'TOT. HOURLY',t30,'TOT. HOURLY',
     -t43,'PK SS ENERGY',t57,'PK INST. ENERGY')
135   format(t8,'NUMBER',t16,'CONSUMPTION',t34,'COST',t44,'CONSUMPTION',
     -t59,'CONSUMPTION')
132   format(t3,i3,t10,i4,t17,1p1e10.3,t31,1p1e10.3,t45,1p1e10.3,
     -t59,1p1e10.3)
134   format(t3,i3,t10,i4,t17,1p1e10.3,t31,1p1e10.3,t45,1p1e10.3)
133   format(t5,'TOTAL DAILY ENERGY COST: ',1p1e10.3)
C TIME   COMB.  TOT. HOURLY   TOT. HOURLY  PK SS ENERGY  PK INST. ENERGY
C       NUMBER  CONSUMPTION       COST      CONSUMPTION    CONSUMPTION
C  24     999    +1.000E+00    +1.000E+00    +1.000E+00    +1.000E+00
136   format(t5,'TIME: ',i2,t16,'# OF TRANSITIONS: ',i3)
138   format(t2,'COMB 1',t10,'COMB 2',t19,'PK ENERGY')
137   format(t3,i3,t11,i3,t18,1pe10.3)
140   format(t5,'OPTIMAL PATH - LEAST COST INCLUDING PEAK DEMAND')
      END
c
*DECK EQPTRANS
      SUBROUTINE EQPTRANS(curcmb,time,cureq,cureqop,iplt)
CD    TITLE:=  EQPTRANS - transfer equipment electric consumption data
CD             to the same format as the equipment combination matrix
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  MAY 1995
CD    PURPOSE:=
```

```
CD
CD   Routine to determine optimal schedule from feasible
CD   operating combinations of plant equipment
CD
C     INCLUDES:-
C
      implicit none
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pathfind.inc'
      INCLUDE 'runctl.inc'
c
C     VARIABLES
      INTEGER curcmb,time
      INTEGER i,i2,j,cureq(50),iplt,eqctr
      INTEGER ieqno,ncbmax,ncbcur,ncbdiff
      REAL cureqop(50)
c
c
c*******Main Program*******
c
        ieqno=0
        do 112 i=1,neqtot(iplt)
          ncbmax=eqpcomb(maxcom(iplt),i,iplt)
          ncbcur=eqpcomb(curcmb,i,iplt)
          ncbdiff=ncbmax-ncbcur
          do 113 j=1,ncbmax
            ieqno=ieqno+1
            if (ncbdiff.eq.0) then
              cureq(ieqno)=1
            else
              if (j.le.ncbdiff) then
                cureq(ieqno)=0
              else
                cureq(ieqno)=1
              endif
            endif
113       continue
112     continue
c
        eqctr=0
        do 114 i=1,nalleqp(iplt)
          cureqop(i)=0.0
          if (cureq(i).eq.1) then
            eqctr=eqctr+1
            do 115 i2=1,numfeash(time,iplt)
              if (pltcmbh(i2,time,1,iplt).eq.curcmb) then
                cureqop(i)=pltinoph(i2,time,eqctr,2)
              endif
115         continue
          endif
114     continue
        return
        end
c
```

Appendix B: Subroutines Required for Ice Storage Optimization

```
*DECK ICESIM
      SUBROUTINE ICESIM(iplctr,stinc)
C
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  JULY 1995
CD    PURPOSE:= Simulate Ice storage Equipment
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
      INCLUDE 'absenv.inc'
      INCLUDE 'blsfls.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'cool.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
      INCLUDE 'tepdcl.inc'
      INCLUDE 'vartic.inc'
      INCLUDE 'zoncon.inc'
      INCLUDE 'nigel.inc'
      INCLUDE 'iceopt.inc'
      INCLUDE 'istore.inc'
      INCLUDE 'store.inc'
                                    I N C L U D E      ' t e s d c l . i n c '
modrtts
      INCLUDE 'iceloads.inc'
      INCLUDE 'tesaux.inc'
c
c     Local variables
c
      INTEGER i,i2,statctr,iprev,iplctr
      REAL STINC
      LOGICAL pcovrld
c
c     Code
c
      IF (IICECAP(IPLCTR).GT.0.0) THEN
c
        nicests(stdtim,iplctr)=0
        ecool=plcld/1000.
```

```
        CALL AMIOP(IPLCTR)
c
c         call ice storage simulations
c
        IF (OFFPK(IPLCTR).or.warmup.or.icefinal) THEN
          nicests(stdtim,iplctr)=1
          if (icefinal) then
           ichload=finpthfr(stdtim,iplctr)*compcm(iplctr)
           if (stdtim.eq.1.and.stpnm.eq.1.and.curinth.eq.1) then
             cicapi(iplctr)=tnkice(1,24,iplctr,2)
           endif
          elseif (offpk(iplctr)) then
           ichload=compcm(iplctr)
           cicapi(iplctr)=tnkicap(1,iplctr)
          else
           ichload=0.5*compcm(iplctr)
           cicapi(iplctr)=tnkicap(1,iplctr)
          endif
          CALL IISSS(IPLCTR,PCOVRLD,STINC)
          tnkfcap(1,iplctr)=cicapin(iplctr)
          offpkld(1,iplctr)=isoplan(iplctr)
          elecpwr(1,iplctr)=iiceelec(iplctr)+codelc(iplctr)+
     -                     parice1(iplctr)+parice2(iplctr)
        ELSE
          if (stdtim.eq.1) then
            statctr=nicests(24,iplctr)
          else
            statctr=nicests(stdtim-1,iplctr)
          endif
          do 101 iprev=1,statctr
            do 102 i2=1,2
              nicests(stdtim,iplctr)=nicests(stdtim,iplctr)+1
              ichload=iceldfr(i2,stdtim,iplctr)*compcm(iplctr)
              if (stpnm.eq.1.and.curinth.eq.1) then
                cicapi(iplctr)=tnkicap(iprev,iplctr)
              else
                cicapi(iplctr)=tnkicap(nicests(stdtim,iplctr),iplctr)
              endif
c
c                                       Model Indirect Ice Storage
c
              CALL IISSS(IPLCTR,PCOVRLD,STINC)
c
c                                       store results of simulation
c
              tnkfcap(nicests(stdtim,iplctr),iplctr)=cicapin(iplctr)
              chopfr(nicests(stdtim,iplctr),iplctr)=
     -                                   iceldfr(i2,stdtim,iplctr)
              offpkld(nicests(stdtim,iplctr),iplctr)=isoplan(iplctr)
              elecpwr(nicests(stdtim,iplctr),iplctr)=iiceelec(iplctr)+
     -           codelc(iplctr)
     -                     +parice1(iplctr)+parice2(iplctr)
102         continue
101       continue
        ENDIF
      ENDIF
      RETURN
      END
```

```
*DECK ICEAVRGH
      SUBROUTINE ICEAVRGH
C
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  JULY 1995
CD    PURPOSE:= Hourly averaging of ice storage performance
CD              data
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
      INCLUDE 'absenv.inc'
      INCLUDE 'blsfls.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'cool.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
      INCLUDE 'tepdcl.inc'
      INCLUDE 'vartic.inc'
      INCLUDE 'zoncon.inc'
      INCLUDE 'nigel.inc'
      INCLUDE 'iceopt.inc'
      INCLUDE 'istore.inc'
      INCLUDE 'store.inc'
                                      I N C L U D E      ' t e s d c l . i n c '
modrtts
      INCLUDE 'iceloads.inc'
c
c     Local variables
c
      INTEGER i,i2,iplt,icnt
c
c     Code
c
c        Set initial values to final values of last time step of
c        previous hour
c
      do 100 iplt=1,ninpl
      IF (doice(iplt)) THEN
       if(.not.warmup) then
        if (.not.dayoff(iplt)) then
          if (stdtim.eq.1) then
            tnkice(1,stdtim,iplt,1)=tnkice(1,24,iplt,2)
```

```
          tnkice(1,stdtim,iplt,2)=tnkfcap(1,iplt)
          tnkengh(1,stdtim,iplt)=icelecav(1,iplt)
          choffpk(1,stdtim,iplt)=offpkav(1,iplt)
          ISOPLA(stdtim,iplt)=offpkav(1,iplt)
          chopfrh(1,stdtim,iplt)=1.0
        elseif (stdtim.ge.bgnice(iplt).and.stdtim.le.
     -          endice(iplt)) then
          icnt=0
          do 101 i=1,nicests(stdtim-1,iplt)
            do 102 i2=1,nicepths(iplt)/nicests(stdtim-1,iplt)
              icnt=icnt+1
              if (stdtim.eq.bgnice(iplt)) then
               tnkice(icnt,stdtim,iplt,1)=tnkice(i,stdtim-1,iplt,2)
              else
               tnkice(icnt,stdtim,iplt,1)=tnkice(icnt,stdtim-1,iplt,2)
              endif
102         continue
101       continue
          icnt=0
          do 103 i=1,nicests(stdtim,iplt)
            do 104 i2=1,nicepths(iplt)/nicests(stdtim,iplt)
              icnt=icnt+1
              tnkice(icnt,stdtim,iplt,2)=tnkfcap(i,iplt)
              tnkengh(icnt,stdtim,iplt)=icelecav(i,iplt)
              choffpk(icnt,stdtim,iplt)=offpkav(i,iplt)
              ISOPLA(stdtim,iplt)=0.0
              chopfrh(icnt,stdtim,iplt)=chopfr(i,iplt)
104         continue
103       continue
        elseif (stdtim.eq.endice(iplt)+1) then
          tnkice(1,stdtim,iplt,1)=optnk(stdtim-1,iplt,2)
          tnkice(1,stdtim,iplt,2)=tnkfcap(1,iplt)
          choffpk(1,stdtim,iplt)=offpkav(1,iplt)
          ISOPLA(stdtim,iplt)=offpkav(1,iplt)
          chopfrh(1,stdtim,iplt)=1.0
          tnkengh(1,stdtim,iplt)=icelecav(1,iplt)
        else
          tnkice(1,stdtim,iplt,1)=tnkice(1,stdtim-1,iplt,2)
          tnkice(1,stdtim,iplt,2)=tnkfcap(1,iplt)
          choffpk(1,stdtim,iplt)=offpkav(1,iplt)
          ISOPLA(stdtim,iplt)=offpkav(1,iplt)
          chopfrh(1,stdtim,iplt)=1.0
          tnkengh(1,stdtim,iplt)=icelecav(1,iplt)
        endif
      else
        if (stdtim.eq.1) then
          tnkice(1,stdtim,iplt,1)=tnkice(1,24,iplt,2)
          tnkice(1,stdtim,iplt,2)=tnkfcap(1,iplt)
          choffpk(1,stdtim,iplt)=offpkav(1,iplt)
          ISOPLA(stdtim,iplt)=offpkav(1,iplt)
          chopfrh(1,stdtim,iplt)=1.0
          tnkengh(1,stdtim,iplt)=icelecav(1,iplt)
        else
          tnkice(1,stdtim,iplt,1)=tnkice(1,stdtim-1,iplt,2)
          tnkice(1,stdtim,iplt,2)=tnkfcap(1,iplt)
          choffpk(1,stdtim,iplt)=offpkav(1,iplt)
          ISOPLA(stdtim,iplt)=offpkav(1,iplt)
          chopfrh(1,stdtim,iplt)=1.0
```

```
            tnkengh(1,stdtim,iplt)=icelecav(1,iplt)
          endif
        endif
       else
        if (stdtim.eq.1) then
          tnkice(1,stdtim,iplt,1)=tnkice(1,24,iplt,2)
          tnkice(1,stdtim,iplt,2)=tnkfcap(1,iplt)
          tnkengh(1,stdtim,iplt)=icelecav(1,iplt)
          choffpk(1,stdtim,iplt)=offpkav(1,iplt)
          ISOPLA(stdtim,iplt)=offpkav(1,iplt)
          chopfrh(1,stdtim,iplt)=1.0
        else
          tnkice(1,stdtim,iplt,1)=tnkice(1,stdtim-1,iplt,2)
          tnkice(1,stdtim,iplt,2)=tnkfcap(1,iplt)
          choffpk(1,stdtim,iplt)=offpkav(1,iplt)
          ISOPLA(stdtim,iplt)=offpkav(1,iplt)
          chopfrh(1,stdtim,iplt)=1.0
          tnkengh(1,stdtim,iplt)=icelecav(1,iplt)
        endif
       endif
      endif
100   continue
      RETURN
      END
*DECK INICEOPT1
      SUBROUTINE INICEOPT1
C
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  JULY 1995
CD    PURPOSE:= Perform necessary initialisations of ice storage
CD              optimisation daily storage arrays and variables
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
      INCLUDE 'absenv.inc'
      INCLUDE 'blsfls.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'cool.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
      INCLUDE 'tepdcl.inc'
      INCLUDE 'vartic.inc'
```

```
      INCLUDE 'zoncon.inc'
      INCLUDE 'nigel.inc'
      INCLUDE 'iceopt.inc'
      INCLUDE 'istore.inc'
      INCLUDE 'store.inc'
                                           I N C L U D E       ' t e s d c l . i n c '
modrtts
      INCLUDE 'iceloads.inc'
c
c     Local variables
c
      INTEGER i,i2,j,iplt,icnt
c
c     Code
c
      noice=.true.
      icefinal=.false.
      iceit=0
      do 100 iplt=1,ninpl
       IF (IICECAP(IPLT).gt.0.0) then
        noice=.false.
        doice(iplt)=.true.
        do 102 i2=1,64
          tnkicap(i2,iplt)=0.0
          do 101 i=1,24
            optnk(i,iplt,1)=0.9
            optnk(i,iplt,2)=0.9
            do 103 j=1,2
              tnkice(i2,i,iplt,j)=0.9
103         continue
            choffpk(i2,i,iplt)=0.0
            tnkengh(i2,i,iplt)=0.0
101        continue
102     continue
        tnkicap(1,iplt)=0.9
        do 105 i=1,24
          iceldfr(1,i,iplt)=1.0
          iceldfr(2,i,iplt)=1.0
105     continue
        nicepths(iplt)=1
        do 104 i=bgnice(iplt),endice(iplt)
          nicepths(iplt)=nicepths(iplt)*2
          iceldfr(1,i,iplt)=0.33
          iceldfr(2,i,iplt)=0.67
104     continue
       endif
100   continue
      RETURN
      END
*DECK INICEOPT2
      SUBROUTINE INICEOPT2
C
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  JULY 1995
CD    PURPOSE:= Perform necessary initialisations of ice storage
CD              optimisation hourly arrays and variables
CD
CD    VARIABLE DICTIONARY:=
```

```
c   LOCAL
c
c
CD
CD     METHOD:=
C
C      INCLUDES:-
C
       implicit none
c
       INCLUDE 'param.inc'
       INCLUDE 'pltpar.inc'
       INCLUDE 'absenv.inc'
       INCLUDE 'blsfls.inc'
       INCLUDE 'chlrdcl.inc'
       INCLUDE 'cndrdcl.inc'
       INCLUDE 'cool.inc'
       INCLUDE 'hortot.inc'
       INCLUDE 'optiblst.inc'
       INCLUDE 'pltrckp.inc'
       INCLUDE 'runctl.inc'
       INCLUDE 'sysldc.inc'
       INCLUDE 'tepdcr.inc'
       INCLUDE 'tepdcl.inc'
       INCLUDE 'vartic.inc'
       INCLUDE 'zoncon.inc'
       INCLUDE 'nigel.inc'
       INCLUDE 'iceopt.inc'
       INCLUDE 'istore.inc'
       INCLUDE 'store.inc'
                                            I N C L U D E       ' t e s d c l . i n c '
modrtts
       INCLUDE 'iceloads.inc'
c
c      Local variables
c
       INTEGER i,i2,j,iplt,icnt
c
c      Code
c
       do 100 iplt=1,ninpl
         do 102 i2=1,64
           icelecav(i2,iplt)=0.0
           offpkav(i2,iplt)=0.0
102      continue
100    continue
       RETURN
       END
*DECK INICEOPT3
       SUBROUTINE INICEOPT3
C
CD     AUTHOR:=  RUSS TAYLOR
CD     DATE WRITTEN:=  JULY 1995
CD     PURPOSE:= Perform necessary initialisations of ice storage
CD               optimisation arrays and variables to allow
CD               next iteration day to be simulated
CD
CD     VARIABLE DICTIONARY:=
```

```
c   LOCAL
c
c
CD
CD     METHOD:=
C
C      INCLUDES:-
C
       implicit none
c
       INCLUDE 'param.inc'
       INCLUDE 'pltpar.inc'
       INCLUDE 'absenv.inc'
       INCLUDE 'blsfls.inc'
       INCLUDE 'chlrdcl.inc'
       INCLUDE 'cndrdcl.inc'
       INCLUDE 'cool.inc'
       INCLUDE 'hortot.inc'
       INCLUDE 'optiblst.inc'
       INCLUDE 'pltrckp.inc'
       INCLUDE 'runctl.inc'
       INCLUDE 'sysldc.inc'
       INCLUDE 'tepdcr.inc'
       INCLUDE 'tepdcl.inc'
       INCLUDE 'vartic.inc'
       INCLUDE 'zoncon.inc'
       INCLUDE 'nigel.inc'
       INCLUDE 'iceopt.inc'
       INCLUDE 'istore.inc'
       INCLUDE 'store.inc'
                                          INCLUDE      'tesdcl.inc'
modrtts
       INCLUDE 'iceloads.inc'
c
c      Local variables
c
       INTEGER i,i2,j,iplt,maxiceit,optit
       REAL intrvl,oldldfr(2,24,mxplts),oldopfr(24,mxplts),oldcst(mxplts)
       parameter (maxiceit=10)
c
c      Code
c
c         increment iteration counter
c
       iceit=iceit+1
c
c          select new chiller fractions for each on peak hour
c
       if (iceit.le.maxiceit) then
         do 100 iplt=1,ninpl
           IF (doice(iplt)) THEN
             IF (iceit.gt.1) then
c
c               see if iteration is going in the right direction
c
               IF (optcst(iplt).gt.oldcst(iplt)) then
c
c               objective growing: set step smaller
```

```
c
              optit=optit+1
              do 111 i=bgnice(iplt),endice(iplt)
                intrvl=abs(oldldfr(1,i,iplt)-oldldfr(2,i,iplt))
     -               /(2.0**float(optit)+1)
                iceldfr(1,i,iplt)=amax1(0.0,oldopfr(i,iplt)-intrvl)
                iceldfr(2,i,iplt)=amin1(1.0,oldopfr(i,iplt)+intrvl)
111           continue
             ELSE
c
c            objective decreasing: keep going in same direction
c
               optit=0
               oldcst(iplt)=optcst(iplt)
               do 112 i=bgnice(iplt),endice(iplt)
                 intrvl=abs(iceldfr(1,i,iplt)-iceldfr(2,i,iplt))/3.0
                 oldldfr(1,i,iplt)=iceldfr(1,i,iplt)
                 oldldfr(2,i,iplt)=iceldfr(2,i,iplt)
                 oldopfr(i,iplt)=opthfr(i,iplt)
                 iceldfr(1,i,iplt)=amax1(0.0,opthfr(i,iplt)-intrvl)
                 iceldfr(2,i,iplt)=amin1(1.0,opthfr(i,iplt)+intrvl)
112            continue
             ENDIF
          ELSE
            optit=0
            oldcst(iplt)=optcst(iplt)
            do 101 i=bgnice(iplt),endice(iplt)
              intrvl=abs(iceldfr(1,i,iplt)-iceldfr(2,i,iplt))/3.0
              oldldfr(1,i,iplt)=iceldfr(1,i,iplt)
              oldldfr(2,i,iplt)=iceldfr(2,i,iplt)
              oldopfr(i,iplt)=opthfr(i,iplt)
              iceldfr(1,i,iplt)=amax1(0.0,opthfr(i,iplt)-intrvl)
              iceldfr(2,i,iplt)=amin1(1.0,opthfr(i,iplt)+intrvl)
101         continue
          ENDIF
        endif
100     continue
      else
c
c         set chiller fractions for final pass through simulation
c
        icefinal=.true.
        do 102 iplt=1,ninpl
         IF (doice(iplt)) THEN
          do 104 i=1,24
            finpthfr(i,iplt)=1.0
104       continue
          do 103 i=bgnice(iplt),endice(iplt)
            finpthfr(i,iplt)=opthfr(i,iplt)
103       continue
            write(output,205)
            write(output,207)
            do 105 i=1,24
              write(output,206) i,finpthfr(i,iplt)
105         continue
        endif
102     continue
       endif
```

```
      RETURN
205   format(t5,'Results of ice storage optimisation')
207   format(t5,'TIME',t15,'CHILLER FRACTION')
206   FORMAT(t6,i2,t18,1p1e10.3)
      END
*DECK ICESCHDOPT
      SUBROUTINE ICESCHDOPT
C
CD    AUTHOR:=  RUSS TAYLOR
CD    DATE WRITTEN:=  JULY 1995
CD    PURPOSE:= Optimise on-peak ice storage usage profile
CD
CD    VARIABLE DICTIONARY:=
c   LOCAL
c
c
CD
CD    METHOD:=
C
C     INCLUDES:-
C
      implicit none
c
      INCLUDE 'param.inc'
      INCLUDE 'pltpar.inc'
      INCLUDE 'absenv.inc'
      INCLUDE 'blsfls.inc'
      INCLUDE 'chlrdcl.inc'
      INCLUDE 'cndrdcl.inc'
      INCLUDE 'cool.inc'
      INCLUDE 'hortot.inc'
      INCLUDE 'optiblst.inc'
      INCLUDE 'pltrckp.inc'
      INCLUDE 'runctl.inc'
      INCLUDE 'sysldc.inc'
      INCLUDE 'tepdcr.inc'
      INCLUDE 'tepdcl.inc'
      INCLUDE 'vartic.inc'
      INCLUDE 'zoncon.inc'
      INCLUDE 'nigel.inc'
      INCLUDE 'iceopt.inc'
      INCLUDE 'istore.inc'
      INCLUDE 'store.inc'
                                            INCLUDE      'tesdcl.inc'
modrtts
      INCLUDE 'iceloads.inc'
c
c     Local variables
c
      INTEGER i,i2,iplt,icnt,time,pth,optpth
c
      REAL chtnkice,ofpkeng,totnkeng,offpkcst,puicecst,pkpthen
      REAL pthfr(24),pthcst,tnk(24,2),optpken,deltnkice
c
c     Code
c
c
c     1st step: Compute cost of stored ice per unit storage
```

```
c
      do 100 iplt=1,ninpl
      if (doice(iplt)) then
       if(stdtim.eq.endice(iplt)) then
        chtnkice=0.0
        ofpkeng=0.0
        totnkeng=0.0
        do 101 time=endice(iplt)+1,24
          deltnkice=amax1(0.0,tnkice(1,time,iplt,2)-
     -              tnkice(1,time,iplt,1))
          if (deltnkice.gt.0.0) then
            chtnkice=chtnkice+deltnkice
            ofpkeng=ofpkeng+choffpk(1,time,iplt)
            totnkeng=totnkeng+tnkengh(1,time,iplt)
          endif
101     continue
        do 102 time=1,bgnice(iplt)-1
          deltnkice=amax1(0.0,tnkice(1,time,iplt,2)-
     -              tnkice(1,time,iplt,1))
          if (deltnkice.gt.0.0) then
            chtnkice=chtnkice+deltnkice
            ofpkeng=ofpkeng+choffpk(1,time,iplt)
            totnkeng=totnkeng+tnkengh(1,time,iplt)
          endif
102     continue
c
c         compute energy cost
c
          offpkcst=cstfac(3)*totnkeng
c
c         totnkeng is the total amount of energy consumed by the chiller
c         during the off-peak hours.  But all this energy may not go to
c         the creation of ice so cost must be based on the total amount
c         of cooling provided.
c
          puicecst=offpkcst/(chtnkice*tnkcap(iplt)+ofpkeng)
c
c     2nd step: Compute cost of all possible paths through on-peak
c               hours based on utility rate structure, including demand
c               charges, and evaluate least cost path.
c         a/  Compute cost of first path as reference
c
          optcst(iplt)=0.0
          pkpthen=0.0
          chtnkice=0.0
          optpth=1
          do 105 time=bgnice(iplt),endice(iplt)
            optnk(time,iplt,1)=tnkice(1,time,iplt,1)
            optnk(time,iplt,2)=tnkice(1,time,iplt,2)
            chtnkice=tnkice(1,time,iplt,1)-tnkice(1,time,iplt,2)
            optcst(iplt)=optcst(iplt)+puicecst*chtnkice*tnkcap(iplt)+
     -               onpkmult*cstfac(3)*tnkengh(1,time,iplt)
            pkpthen=amax1(pkpthen,tnkengh(1,time,iplt))
            opthfr(time,iplt)=chopfrh(1,time,iplt)
105       continue
          optcst(iplt)=optcst(iplt)+dmndfac*pkpthen
          optpken=pkpthen
c
```

```fortran
c         b/compute cost associated with remaining paths and keep
c           running evaluation of optimal path
c
        do 103 pth=2,nicepths(iplt)
          pthcst=0.0
          pkpthen=0.0
          chtnkice=0.0
          do 104 time=bgnice(iplt),endice(iplt)
            tnk(time,1)=tnkice(pth,time,iplt,1)
            tnk(time,2)=tnkice(pth,time,iplt,2)
            chtnkice=tnkice(pth,time,iplt,1)-tnkice(pth,time,iplt,2)
            pthcst=pthcst+puicecst*chtnkice*tnkcap(iplt)+onpkmult*
     -              cstfac(3)*tnkengh(pth,time,iplt)
            pkpthen=amax1(pkpthen,tnkengh(pth,time,iplt))
            pthfr(time)=chopfrh(pth,time,iplt)
104       continue
          pthcst=pthcst+dmndfac*pkpthen
          if (pthcst.lt.optcst(iplt)) then
            optpth=pth
            optcst(iplt)=pthcst
            optpken=pkpthen
            do 106 time=bgnice(iplt),endice(iplt)
              opthfr(time,iplt)=pthfr(time)
              optnk(time,iplt,1)=tnk(time,1)
              optnk(time,iplt,2)=tnk(time,2)
106         continue
          endif
103     continue
c
c       3rd step/ report results of optimisation
c
        write(output,901) optcst(iplt)
        write(output,902) optpken
        write(output,903)
        write(output,905)
        do 107 time=bgnice(iplt),endice(iplt)
          write(output,904) time,opthfr(time,iplt),optnk(time,iplt,1),
     -     optnk(time,iplt,2),tnkengh(optpth,time,iplt)
107     continue
       endif
      endif
100   continue
      RETURN
901   format(t5,'Optimal path cost this iteration: ',1p1e10.3)
902   format(t5,'Max. on-peak electric consumption: ',1p1e10.3)
903   format('   Time     Chiller     Initial     Final     Chiller Ener
     -gy')
905   format('             Fraction    Capacity    Capacity     Consumptio
     -n')
904   format(t5,i2,t11,1p1e10.3,t23,1p1e10.3,t35,1p1e10.3,t50,1p1e10.3)
c   Time     Chiller     Initial     Final     Chiller Energy
c           Fraction    Capacity    Capacity     Consumption
c    12    +1.234e+12  +1.234e+12  +1.234e+12     +1.234e+12
      END
```

Appendix C: Sample IBLAST Input File

```
BEGIN INPUT;
   RUN CONTROL:
     NEW ZONES,
     NEW AIR SYSTEMS,
     PLANT,
     REPORTS(ZONE LOADS,ZONES,SYSTEM LOADS,SYSTEM),
     UNITS(IN=ENGLISH, OUT=METRIC);
     TEMPORARY WALLS:
**
**   A2 - 4IN Dense Face Brick
**   B3 - 2IN Insulation
**   BB17 - 3/4IN Acoustic Tile
**   C8 - 8IN HW Concrete
**   CO17 - 4IN Concrete, Dried Sand and Gravel
**   DIRT 12 IN -
**   FF5 - 1/16IN Tile Flooring
**   IN3 - 6IN Mineral Fibre, Fibrous
**   IN71 - Preformed Roof Insulation
**   PL4 - Plaster - Gypsum LWA 5/8IN
**   RF4 - 3/8IN Built-up Roofing
**
       CONST1
         = (A2 ,
            C8 ,
            IN3 ,
            PL4);
       CONST3
         = (PL4 ,
            B3 ,
            PL4);
     END;
     TEMPORARY ROOFS:
       CONST5
         = (RF4 ,
            IN71 ,
            BB17);
     END;
     TEMPORARY FLOORS:
       CONST6
         = (DIRT 12 IN ,
            CO17 ,
            FF5);
     END;
     PROJECT="Modified Ft Monmouth Building for
              Testing Central Plant Optimisation";
     LOCATION=NYC ;
     DESIGN DAYS=NYC SUMMER;
**                 NYC WINTER;
   BEGIN BUILDING DESCRIPTION;
     BUILDING="2 ZONE FT MONMOUTH EDUCATION CENTER";
       NORTH AXIS=165.00;
       SOLAR DISTRIBUTION=-1;
       ZONE 1 "ZONE #1 ":
         ORIGIN:(0.00, 0.00, 0.00);
         NORTH AXIS=0.00;
```

```
EXTERIOR WALLS :
  STARTING AT(0.00, 0.00, 0.00)
  FACING(180.00)
  TILTED(90.00)
  CONST1 (50.00 BY 10.00)
    WITH WINDOWS OF TYPE
    DPW (8.70 BY 7.00)
      REVEAL(0.00)
      AT (0.01, 0.01),
  STARTING AT(50.00, 0.00, 0.00)
  FACING(90.00)
  TILTED(90.00)
  CONST1 (16.00 BY 10.00)
    WITH WINDOWS OF TYPE
    DPW (6.97 BY 6.00)
      REVEAL(0.00)
      AT (0.01, 0.01),
  STARTING AT(25.00, 124.66, 0.00)
  FACING(0.00)
  TILTED(90.00)
  CONST1 (25.00 BY 10.00)
    WITH WINDOWS OF TYPE
    DPW (6.97 BY 6.00)
      REVEAL(0.00)
      AT (0.01, 0.01)
    WITH DOORS OF TYPE
    ALD (3.50 BY 3.00)
      AT (0.01, 0.01),
  STARTING AT(0.00, 124.66, 0.00)
  FACING(270.00)
  TILTED(90.00)
  CONST1 (124.66 BY 10.00)
    WITH WINDOWS OF TYPE
    DPW (44.53 BY 7.50)
      REVEAL(0.00)
      AT (0.01, 0.01)
    WITH DOORS OF TYPE
    ALD (4.07 BY 4.00)
      AT (0.01, 0.01);
PARTITIONS :
  STARTING AT(50.00, 16.00, 0.00)
  FACING(90.00)
  TILTED(90.00)
  CONST3 (13.00 BY 10.00),
  STARTING AT(50.00, 29.00, 0.00)
  FACING(90.00)
  TILTED(90.00)
  CONST3 (3.66 BY 10.00),
  STARTING AT(50.00, 32.66, 0.00)
  FACING(0.00)
  TILTED(90.00)
  CONST3 (25.00 BY 10.00),
  STARTING AT(25.00, 32.66, 0.00)
  FACING(90.00)
  TILTED(90.00)
  CONST3 (92.00 BY 10.00);
SLAB ON GRADE FLOORS :
  STARTING AT(0.00, 0.00, 0.00)
```

```
    FACING(0.00)
    TILTED(180.00)
    CONST6 (157.33 BY 25.00);
  ROOFS :
    STARTING AT(0.00, 0.00, 10.00)
    FACING(0.00)
    TILTED(0.00)
    CONST5 (157.33 BY 25.00);
  INTERNAL MASS: CONST3
    ( 238.00 BY 10.00);
  PEOPLE=100,OFFICE OCCUPANCY,
    AT ACTIVITY LEVEL 0.45, 70.00 PERCENT RADIANT,
    FROM 01JAN THRU 31DEC;
  LIGHTS=30.00,OFFICE LIGHTING ,
      0.00 PERCENT RETURN AIR, 20.00 PERCENT RADIANT,
     20.00 PERCENT VISIBLE, 0.00 PERCENT REPLACEABLE,
    FROM 01JAN THRU 31DEC;
  INFILTRATION=470.00,CONSTANT ,
    WITH COEFFICIENTS (0.606000, 0.020200, 0.000598, 0.000000),
    FROM 01JAN THRU 31DEC;
  CONTROLS=NWS2,
    75.6 HEATING, 86.4 COOLING,
    FROM 01JAN THRU 31DEC;
END ZONE;
ZONE 2 "ZONE #2 ":
  ORIGIN:(25.00, 32.66, 0.00);
  NORTH AXIS=0.00;
  EXTERIOR WALLS :
    STARTING AT(25.00, 16.66, 0.00)
    FACING(90.00)
    TILTED(90.00)
    CONST1 (75.34 BY 10.00)
      WITH WINDOWS OF TYPE
      DPW (27.84 BY 7.50)
        REVEAL(0.00)
        AT (0.01, 0.01),
    STARTING AT(25.00, 92.00, 0.00)
    FACING(0.00)
    TILTED(90.00)
    CONST1 (25.00 BY 10.00)
      WITH WINDOWS OF TYPE
      DPW (6.97 BY 6.00)
        REVEAL(0.00)
        AT (0.01, 0.01)
      WITH DOORS OF TYPE
      ALD (3.50 BY 3.00)
        AT (0.01, 0.01);
  PARTITIONS :
    STARTING AT(0.00, 0.00, 0.00)
    FACING(180.00)
    TILTED(90.00)
    CONST3 (25.00 BY 10.00),
    STARTING AT(25.00, 0.00, 0.00)
    FACING(90.00)
    TILTED(90.00)
    CONST3 (3.66 BY 10.00),
    STARTING AT(25.00, 3.66, 0.00)
    FACING(90.00)
```

```
          TILTED(90.00)
          CONST3 (13.00 BY 10.00),
          STARTING AT(0.00, 92.00, 0.00)
          FACING(270.00)
          TILTED(90.00)
          CONST3 (92.00 BY 10.00);
        SLAB ON GRADE FLOORS :
          STARTING AT(0.00, 0.00, 0.00)
          FACING(0.00)
          TILTED(180.00)
          CONST6 (25.00 BY 92.00);
        ROOFS :
          STARTING AT(0.00, 0.00, 10.00)
          FACING(0.00)
          TILTED(0.00)
          CONST5 (25.00 BY 92.00);
        INTERNAL MASS: CONST3
          ( 88.00 BY 10.00);
        PEOPLE=55,OFFICE OCCUPANCY,
          AT ACTIVITY LEVEL 0.45, 70.00 PERCENT RADIANT,
          FROM 01JAN THRU 31DEC;
        LIGHTS=15.00,OFFICE LIGHTING ,
            0.00 PERCENT RETURN AIR, 20.00 PERCENT RADIANT,
           20.00 PERCENT VISIBLE, 0.00 PERCENT REPLACEABLE,
          FROM 01JAN THRU 31DEC;
        INFILTRATION=262.00,CONSTANT ,
          WITH COEFFICIENTS (0.606000, 0.020200, 0.000598, 0.000000),
          FROM 01JAN THRU 31DEC;
        CONTROLS=NWS2,
          49.14 HEATING, 56.2 COOLING,
          FROM 01JAN THRU 31DEC;
      END ZONE;
  END BUILDING DESCRIPTION;
  BEGIN FAN SYSTEM DESCRIPTION;
    SINGLE ZONE DRAW THRU SYSTEM 1
    "DUAL PURPOSE VAV " SERVING ZONES
       1;
      FOR ZONE 1:
        SUPPLY AIR VOLUME=4000.;
        EXHAUST AIR VOLUME=0;
*        REHEAT CAPACITY=100.0;
*        REHEAT ENERGY SUPPLY=HOT WATER;
        BASEBOARD HEAT CAPACITY=0.0;
        BASEBOARD HEAT ENERGY SUPPLY=HOT WATER;
*        MINIMUM AIR FRACTION=0.30;
**      REHEAT COIL DESIGN PARAMETERS:
*            COIL TYPE=FLAT FIN HOT WATER;
*            RH PRIMARY SURFACE AREA=10.95658;
*            RH SECONDARY SURFACE AREA=383.61500;
*            RH INSIDE SURFACE AREA=10.64981297;
*            RH MINIMUM AIR FLOW AREA=0.9012443;
*            RH COIL DEPTH=0.8125;
*            RH COIL HEIGHT=1.0625;
*            RH FIN THICKNESS=0.000416666667;
*            RH INNER DIAMETER OF TUBES=0.047083333;
*            RH OUTER DIAMETER OF TUBES=0.052083333;
*            RH NUMBER OF CIRCUITS PER ROW=8.0;
*            RH THERMAL CONDUCTIVITY OF TUBES=223.175929;
```

```
*              RH THERMAL CONDUCTIVITY OF FINS=117.9479;
*              RH NUMBER OF ROWS=6.0;
*              RH FIN DISTANCE=0.005952392857;
*              RH TUBE DEPTH SPACING=0.080708661;
*          COIL TYPE=SIMPLE H/RH COIL;
*          RH UA VALUE=10000.0;
*              RH WATER VOLUME FLOW RATE=10.0;
*              RH ENTERING WATER TEMPERATURE=145.4;
*              REHEAT COIL CONTROL=WILD COIL;
**        END REHEAT COIL DESIGN PARAMETERS;
          ZONE MULTIPLIER=1;
        END ZONE;
        OTHER SYSTEM PARAMETERS:
          SUPPLY FAN PRESSURE=2.48914;
          SUPPLY FAN EFFICIENCY=0.7;
          RETURN FAN PRESSURE=0.0;
          RETURN FAN EFFICIENCY=0.7;
          EXHAUST FAN PRESSURE=1.00396;
          EXHAUST FAN EFFICIENCY=0.7;
          COLD DECK CONTROL=FIXED SET POINT;
          COLD DECK TEMPERATURE=55.04;
          COLD DECK THROTTLING RANGE=7.2;
          COLD DECK CONTROL SCHEDULE=(55 AT 90, 65 AT 70);
          MIXED AIR CONTROL=TEMPERATURE ECONOMY CYCLE;
          DESIRED MIXED AIR TEMPERATURE=COLD DECK TEMPERATURE;
          OUTSIDE AIR VOLUME=0.0;
          PREHEAT COIL LOCATION=NONE;
          PREHEAT TEMPERATURE=46.4;
          PREHEAT ENERGY SUPPLY=HOT WATER;
          PREHEAT COIL CAPACITY=0;
          GAS BURNER EFFICIENCY=0.8;
          VAV VOLUME CONTROL TYPE=INLET VANES;
          ** FAN POWER COEFFICIENTS=(0,0,0,0,0);
          HUMIDIFIER TYPE=NONE;
          HUMIDISTAT LOCATION=1;
          HUMIDISTAT SET POINT=50;
          SYSTEM ELECTRICAL DEMAND=0.0;
          REHEAT TEMPERATURE CONTROL=FIXED SET POINT;
          REHEAT TEMPERATURE LIMIT=140;
          REHEAT CONTROL SCHEDULE=(140 AT 0,70 AT 70);
        END OTHER SYSTEM PARAMETERS;
** IF ANY ONE OF THE FOLLOWING BLOCK IS CHANGED, CHANGE THE REST ACCORDINGLY
**
        COOLING COIL DESIGN PARAMETERS:
**        HEATING COIL DESIGN PARAMETERS:
*         COIL TYPE=SIMPLE H/RH COIL;
           HC UA VALUE=10000.0;
**        END HEATING COIL DESIGN PARAMETERS;
            COIL TYPE=FLAT FIN CHILLED WATER;
            CC PRIMARY SURFACE AREA=10.95658;
            CC SECONDARY SURFACE AREA=383.61500;
            CC INSIDE SURFACE AREA=10.64981297;
            CC MINIMUM AIR FLOW AREA=0.9012443;
            CC COIL DEPTH=0.8125;
            CC COIL HEIGHT=1.0625;
            CC FIN THICKNESS=0.000416666667;
            CC INNER DIAMETER OF TUBES=0.047083333;
            CC OUTER DIAMETER OF TUBES=0.052083333;
```

```
          CC NUMBER OF CIRCUITS PER ROW=8.0;
          CC THERMAL CONDUCTIVITY OF TUBES=223.175929;
          CC THERMAL CONDUCTIVITY OF FINS=117.9479;
          CC NUMBER OF ROWS=6.0;
          CC FIN DISTANCE=0.005952392857;
          CC TUBE DEPTH SPACING=0.080708661;
** END ADDITIONS
          AIR VOLUME FLOW RATE=1500;
          BAROMETRIC PRESSURE=405.489;
          AIR FACE VELOCITY=492.126;
          ENTERING AIR DRY BULB TEMPERATURE=84.92;
          ENTERING AIR WET BULB TEMPERATURE=64.04;
          LEAVING AIR DRY BULB TEMPERATURE=55.04;
          LEAVING AIR WET BULB TEMPERATURE=52.7;
          ENTERING WATER TEMPERATURE=44.96;
          LEAVING WATER TEMPERATURE=55.04;
          WATER VOLUME FLOW RATE=10.0;
          WATER VELOCITY=275.59;
      END COOLING COIL DESIGN PARAMETERS;
      HEAT RECOVERY PARAMETERS:
        HTREC1(0.85,0.0,0.0);
        HTREC2(0.0,0.0,0.0);
        HTREC3(0.0,0.0,0.0);
        HTREC4(0.0,0.0,0.0);
        HTREC5(0.0,0.0,0.0);
        HTREC6(0.0,0.0,0.0);
        HTPWR(0.0,0.0,0.0);
        HEAT RECOVERY CAPACITY=3412000;
      END HEAT RECOVERY PARAMETERS;
      EQUIPMENT SCHEDULES:
        SYSTEM OPERATION=INT,FROM 01JAN THRU 31DEC;
        EXHAUST FAN OPERATION=ON,FROM 01JAN THRU 31DEC;
        PREHEAT COIL OPERATION=ON,FROM 01JAN THRU 31DEC;
        HUMIDIFIER OPERATION=OFF,FROM 01JAN THRU 31DEC;
        COOLING COIL OPERATION=INT,FROM 01MAY THRU 30SEP;
        COOLING COIL OPERATION=OFF,FROM 01OCT THRU 30APR;
        REHEAT COIL OPERATION=ON,FROM 01JAN THRU 31DEC;
        TSTAT BASEBOARD HEAT OPERATION=OFF,FROM 01JAN THRU 31DEC;
        HEAT RECOVERY OPERATION=OFF,FROM 01JAN THRU 31DEC;
        MINIMUM VENTILATION SCHEDULE=MINOA,FROM 01JAN THRU 31DEC;
        MAXIMUM VENTILATION SCHEDULE=MAXOA,FROM 01JAN THRU 31DEC;
        SYSTEM ELECTRICAL DEMAND SCHEDULE=OFF,FROM 01JAN THRU 31DEC;
      END EQUIPMENT SCHEDULES;
    END SYSTEM;
    SINGLE ZONE DRAW THRU SYSTEM 2
    "DUAL PURPOSE VAV " SERVING ZONES
       2;
      FOR ZONE 2:
        SUPPLY AIR VOLUME=2600.;
        EXHAUST AIR VOLUME=0;
*        REHEAT CAPACITY=60.0;
*        REHEAT ENERGY SUPPLY=HOT WATER;
        BASEBOARD HEAT CAPACITY=0.0;
        BASEBOARD HEAT ENERGY SUPPLY=HOT WATER;
        MINIMUM AIR FRACTION=0.25;
**      REHEAT COIL DESIGN PARAMETERS:
*            COIL TYPE=FLAT FIN HOT WATER;
*            RH PRIMARY SURFACE AREA=10.95658;
```

```
*              RH SECONDARY SURFACE AREA=383.61500;
*              RH INSIDE SURFACE AREA=10.64981297;
*              RH MINIMUM AIR FLOW AREA=0.9012443;
*              RH COIL DEPTH=0.8125;
*              RH COIL HEIGHT=1.0625;
*              RH FIN THICKNESS=0.000416666667;
*              RH INNER DIAMETER OF TUBES=0.047083333;
*              RH OUTER DIAMETER OF TUBES=0.052083333;
*              RH NUMBER OF CIRCUITS PER ROW=8.0;
*              RH THERMAL CONDUCTIVITY OF TUBES=223.175929;
*              RH THERMAL CONDUCTIVITY OF FINS=117.9479;
*              RH NUMBER OF ROWS=6.0;
*              RH FIN DISTANCE=0.005952392857;
*              RH TUBE DEPTH SPACING=0.080708661;
*          COIL TYPE=SIMPLE H/RH COIL;
*          RH UA VALUE=10000.0;
*              RH WATER VOLUME FLOW RATE=6.4;
*              RH ENTERING WATER TEMPERATURE=145.4;
*              REHEAT COIL CONTROL=WILD COIL;
**        END REHEAT COIL DESIGN PARAMETERS;
          ZONE MULTIPLIER=1;
        END ZONE;
        OTHER SYSTEM PARAMETERS:
          SUPPLY FAN PRESSURE=2.48914;
          SUPPLY FAN EFFICIENCY=0.7;
          RETURN FAN PRESSURE=0.0;
          RETURN FAN EFFICIENCY=0.7;
          EXHAUST FAN PRESSURE=1.00396;
          EXHAUST FAN EFFICIENCY=0.7;
          COLD DECK CONTROL=FIXED SET POINT;
          COLD DECK TEMPERATURE=55.04;
          COLD DECK THROTTLING RANGE=7.2;
          COLD DECK CONTROL SCHEDULE=(55 AT 90, 65 AT 70);
          MIXED AIR CONTROL=TEMPERATURE ECONOMY CYCLE;
          DESIRED MIXED AIR TEMPERATURE=COLD DECK TEMPERATURE;
          OUTSIDE AIR VOLUME=0.0;
          PREHEAT COIL LOCATION=NONE;
          PREHEAT TEMPERATURE=46.4;
          PREHEAT ENERGY SUPPLY=HOT WATER;
          PREHEAT COIL CAPACITY=0;
          GAS BURNER EFFICIENCY=0.8;
          VAV VOLUME CONTROL TYPE=INLET VANES;
          ** FAN POWER COEFFICIENTS=(0,0,0,0,0);
          HUMIDIFIER TYPE=NONE;
          HUMIDISTAT LOCATION=2;
          HUMIDISTAT SET POINT=50;
          SYSTEM ELECTRICAL DEMAND=0.0;
          REHEAT TEMPERATURE CONTROL=FIXED SET POINT;
          REHEAT TEMPERATURE LIMIT=140;
          REHEAT CONTROL SCHEDULE=(140 AT 0,70 AT 70);
        END OTHER SYSTEM PARAMETERS;
** IF ANY ONE OF THE FOLLOWING BLOCK IS CHANGED, CHANGE THE REST ACCORDINGLY
**
        COOLING COIL DESIGN PARAMETERS:
**      HEATING COIL DESIGN PARAMETERS:
*        COIL TYPE=SIMPLE H/RH COIL;
        HC UA VALUE=10000.0;
**      END HEATING COIL DESIGN PARAMETERS;
```

```
          COIL TYPE=FLAT FIN CHILLED WATER;
          CC PRIMARY SURFACE AREA=10.95658;
          CC SECONDARY SURFACE AREA=383.61500;
          CC INSIDE SURFACE AREA=10.64981297;
          CC MINIMUM AIR FLOW AREA=0.9012443;
          CC COIL DEPTH=0.8125;
          CC COIL HEIGHT=1.0625;
          CC FIN THICKNESS=0.000416666667;
          CC INNER DIAMETER OF TUBES=0.047083333;
          CC OUTER DIAMETER OF TUBES=0.052083333;
          CC NUMBER OF CIRCUITS PER ROW=8.0;
          CC THERMAL CONDUCTIVITY OF TUBES=223.175929;
          CC THERMAL CONDUCTIVITY OF FINS=117.9479;
          CC NUMBER OF ROWS=6.0;
          CC FIN DISTANCE=0.005952392857;
          CC TUBE DEPTH SPACING=0.080708661;
** END ADDITIONS
          AIR VOLUME FLOW RATE=1500;
          BAROMETRIC PRESSURE=405.489;
          AIR FACE VELOCITY=492.126;
          ENTERING AIR DRY BULB TEMPERATURE=84.92;
          ENTERING AIR WET BULB TEMPERATURE=64.04;
          LEAVING AIR DRY BULB TEMPERATURE=55.04;
          LEAVING AIR WET BULB TEMPERATURE=52.7;
          ENTERING WATER TEMPERATURE=44.96;
          LEAVING WATER TEMPERATURE=55.04;
          WATER VOLUME FLOW RATE=10.0;
          WATER VELOCITY=275.59;
      END COOLING COIL DESIGN PARAMETERS;
      HEAT RECOVERY PARAMETERS:
        HTREC1(0.85,0.0,0.0);
        HTREC2(0.0,0.0,0.0);
        HTREC3(0.0,0.0,0.0);
        HTREC4(0.0,0.0,0.0);
        HTREC5(0.0,0.0,0.0);
        HTREC6(0.0,0.0,0.0);
        HTPWR(0.0,0.0,0.0);
        HEAT RECOVERY CAPACITY=3412000;
      END HEAT RECOVERY PARAMETERS;
      EQUIPMENT SCHEDULES:
        SYSTEM OPERATION=INT,FROM 01JAN THRU 31DEC;
        EXHAUST FAN OPERATION=ON,FROM 01JAN THRU 31DEC;
        PREHEAT COIL OPERATION=ON,FROM 01JAN THRU 31DEC;
        HUMIDIFIER OPERATION=OFF,FROM 01JAN THRU 31DEC;
        COOLING COIL OPERATION=INT,FROM 01MAY THRU 30SEP;
        COOLING COIL OPERATION=OFF,FROM 01OCT THRU 30APR;
        REHEAT COIL OPERATION=ON,FROM 01JAN THRU 31DEC;
        TSTAT BASEBOARD HEAT OPERATION=OFF,FROM 01JAN THRU 31DEC;
        HEAT RECOVERY OPERATION=OFF,FROM 01JAN THRU 31DEC;
        MINIMUM VENTILATION SCHEDULE=MINOA,FROM 01JAN THRU 31DEC;
        MAXIMUM VENTILATION SCHEDULE=MAXOA,FROM 01JAN THRU 31DEC;
        SYSTEM ELECTRICAL DEMAND SCHEDULE=OFF,FROM 01JAN THRU 31DEC;
      END EQUIPMENT SCHEDULES;
    END SYSTEM;
  END FAN SYSTEM DESCRIPTION;
  BEGIN CENTRAL PLANT DESCRIPTION;
    PLANT 1 "myplant " SERVING ALL SYSTEMS;
      EQUIPMENT SELECTION:
```

```
        BOILER :
           1 OF SIZE 175;
        CHILLER :
           4 OF SIZE 100.0;
      END EQUIPMENT SELECTION;
      PART LOAD RATIOS:
        BOILER(MIN=.0100,MAX=1.0000,BEST=.8700,ELECTRICAL=.0500);
        CHILLER(MIN=.0700,MAX=1.05,BEST=.6500,ELECTRICAL=.2275);
      END PART LOAD RATIOS;
      SCHEDULE:
        HOT WATER=10.0,OO,FROM 1JAN THRU 31DEC,
                     AT 140.0 SUPPLIED BY BOILER;
      END SCHEDULE;
      FOR SYSTEM 1:
        SYSTEM MULTIPLIER=1;
      END SYSTEM;
    END PLANT;
  END CENTRAL PLANT DESCRIPTION;
END INPUT;
```

Vita

Russell Derek Taylor was born 12 April, 1964 in Preston, Lancashire, England to F. Alan and Nancy Taylor. He grew up in England, and attended Ormskirk Grammar School, until his parents emigrated to the United States of America in 1978. He continued his secondary education at Schuylkill Valley High School in Leesport, Pennsylvania, graduating in 1981. The same year he enrolled at the Berks Campus of The Pennsylvania State University where he majored in Physics. In his second year at Berks, he served as President of the Student Government Association and was named the Eric A. and Josephine Walker Award recipient for the campus in 1983. He enlisted in the United States Marine Corps Reserve in September 1982 and attended boot camp at United States Marine Corps Recruit Depot, Parris Island during the summer of 1983. He served in the Marine Corps Reserve as an Aviation Ordnance Technician until honorably discharged in 1990 with the rank of Sergeant. Russ completed his undergraduate studies at The Pennsylvania State University, University Park, PA. and received a Bachelor of Science degree in Aerospace Engineering in 1985. While an undergraduate at University Park, he recieved the George F. Wislecenius Undergraduate Research Assistantship and worked on methods for visualizing flow separation and transition in a compressor cascade.

Russ remained at Penn State, after completion of his degree, to pursue graduate studies and perform research in solid rocket motor instabilities. In addition, he served as Graduate Student Association President for the 1986-87 school year. He received a Master of Science Degree in Aerospace Engineering from Penn State in 1988. In 1987, he was accepted into the University of Illinois at Urbana-Champaign as a graduate student in Aeronautical and Astronautical Engineering. In 1990, he began working for the BLAST Support Office where he worked to develop an integrated building energy analysis program, IBLAST. He has published several conference papers related to this work and given building energy analysis seminars and training courses in the U.S. and Europe.

Russ holds memberships in several organizations: American Society of Heating, Refrigerating, and Air Conditioning Engineers (ASHRAE), student member; Tau Beta Pi, National Engineering Honor Society; Sigma Gamma Tau, National Aerospace Engineering Honor Society, the Academy of Model Aeronautics; and the Experimental Aircraft Association. In addition, he is working on his Private Pilot Certificate.