Fall 2009

Application of computational intelligence in modeling and optimization of HVAC systems

Mingyang Li
University of Iowa

Follow this and additional works at: https://ir.uiowa.edu/etd
Part of the Industrial Engineering Commons

APPLICATION OF COMPUTATIONAL INTELLIGENCE IN MODELING AND OPTIMIZATION OF HVAC SYSTEMS

by

Mingyang Li

A thesis submitted in partial fulfillment
of the requirements for the Master of Science
degree in Industrial Engineering
in the Graduate College of
The University of Iowa

December 2009

Thesis Supervisor: Professor Andrew Kusiak

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

____

MASTER'S THESIS

____

This is to certify that the Master's thesis of

Mingyang Li

has been approved by the Examining Committee
for the thesis requirement for the Master of Science
degree in Industrial Engineering at the December 2009 graduation.

Thesis Committee: ___________________________________
Andrew Kusiak, Thesis Supervisor

____
Yong Chen

____
Nick Street

To My Parents and Friends

Everything we call real is made of things that cannot be regarded as real.

Niels Bohr

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Professor Andrew Kusiak for his devotion to this research. He has been the most instrumental person for my academic, research, and teaching accomplishments. He provided the motivation, encouragement, guidance and advice which have prepared me for the challenging life that lies ahead. I was exposed to real-world applications while working in the Intelligent Systems Laboratory. This invaluable experience has allowed me to maintain a balance between theory and practice leading to realistic solutions.

I would like to thank my other members of the Master Thesis Committee: Professor Yong Chen and Professor Nick Street for serving on my Thesis Committee and providing valuable suggestions and feedback on the Thesis.

I am also grateful for the financial support from Iowa Energy Center. Discussions with energy experts: Curt Klaassen (Iowa Energy Center), Xiaohui Zhou (Iowa Energy Center), William Haman (Iowa Energy Center), Jiong Zhou (Facilities Management. The University of Iowa) and George Paterson (Facilities Management, The University of Iowa) have provided me invaluable information for this research.

I thank the members of the Intelligent Systems Laboratory who have worked with me and provided advice, critic, reviews and suggestions. Special thanks to my former colleagues: Zhe Song and Haiyang Zheng, who provided me with research insights during my first year in Iowa; Zijun Zhang who have worked with me on solving real-world problems.

I am also thankful to my friends, Guoyuan Liang and Xu Liu for helping me in my research and life.

And finally, and most importantly, I would like to express my sincere gratitude to my parents, who solidly supported me in pursuing my academic goals.

ABSTRACT

HVAC (Heating Ventilating and Air-Conditioning) system is multivariate, nonlinear, and shares time-varying characteristics. It poses challenges for both system modeling and performance optimization. Traditional modeling approaches based on mathematical equations limit the nature of the optimization models and solution approaches.

Computational intelligence is an emerging area of study which provides powerful tools for modeling and analyzing complex systems. Computational intelligence is concerned with discovery of structures in data and recognition of patterns. It encompasses techniques such as neural networks, fuzzy logic, and so on. These techniques derive rules, patterns, and develop complex mappings from the data. The recent advances in information technology have enabled collection of large volumes of data. Computational intelligence embraces biology-inspired paradigms like evolutionary computation and particle swarm intelligence in solving complex optimization problems.

Successful applications of computational intelligence have been found in business, marketing, medical and manufacturing domains. The focus of this thesis is to apply computational intelligence approach in modeling and optimization of HVAC systems. In this research, four HVAC sub-systems are investigated: the AHU (Air Handling Unit), VAV (Variable Air Volume), ventilation system, and thermal zone. Various computational intelligence approaches are used to identify parameters or problem solving. Energy savings are accomplished by minimizing the cooling output, reheating output or fan running time as well as on-line monitoring. One contribution of the research reported in the thesis is the use of computational intelligence algorithms to establish nonlinear mappings among different parameters. Another major contribution is in using heuristics algorithms to solve multi-objective optimization problems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER1.

INTRODUCTION

HVAC (Heating Ventilating and Air-Conditioning) systems provide thermal comfort and air quality in buildings. Due to rapidly growing energy use, energy savings are stressed in almost every aspect of our lives. HVAC systems are major energy consumers in buildings,. According to the published statistics, HVAC systems account for almost 31% of the electricity consumed by U.S. households [1]. Therefore, energy conservation in HVAC system is an issue.

Energy conservation problem in HVAC systems is multi-dimensional. Minimizing the system cost and energy requirements should not compromise an acceptable level of occupancy comfort and indoor air quality.

In addition, as the size of the buildings increase, the complexity of HVAC system expands to meet the various building functions.

1.1 Analytical approaches for modeling and optimization of HVAC systems

An analytical approach to modeling HVAC systems relies on detailed physics-based models and simulation software. Analytical models are often derived from fundamental laws of energy, mass, heat transfer, and so on in the form of mathematical equations. Cumali et al [2] modeled the HVAC system as an optimization problem based on the first principles and applied a generalized reduced gradient method to provide consistent results in handling the equality constraints. Yu et al [3] developed dynamic models for both dry and wet cooling coils using the mass balance and energy equations. Zhang et al [4] proposed a physics-based supervisory control strategy to minimize the net external energy consumption under a series of constraints. If basic assumptions are satisfied, these physics-based models are reliable. However, detailed physics-based models often involve high computational cost and memory demand due to their complexity, which makes them difficult to use in real online applications [5]. To overcome this obstacle, many simplified models are incorporated in simulation programs and on-line applications. Wang et al [6] presented a simple yet accurate model for cooling coil unit and yield better real time control and optimization. Several commonly used simulation programs are TRNSYS [7], HVACSIM+ [8] and SIMBAD [9]. Henze et al [10] modeled a building in TRNSYS and proposed a model predict control strategy to control active and passive building thermal storage inventory in real time. One restriction in these simulation programs is that many components models are steady-state or quasi-steady-state, which makes them not suitable for handling high frequency disturbances [11].

To solve optimization problems formulated by analytical models, many nonlinear local optimization techniques can be used. Rink et al [12] solved the optimization problem of multi-zone HVAC system with substantial energy storage by using state-increment dynamic programming. Kota et al [13] applied the DDP (differential dynamic programming) technique of optimal control to HVAC systems and compared its performances with sequential quadratic programming method. Other nonlinear local techniques like direct search [14], conjugate gradient method [15], univariate search [16], etc were also used in HVAC field.

1.2 Data-driven approaches

Unlike the analytical approaches, a data-driven approach relies on empirical behavior of the system or process. Data mining algorithms like classification or regression map relationships between the input and output variables without requiring detailed prior knowledge. Xi et al [17] built a 2-by-2 nonlinear dynamic model of a HVAC system by SVM and designed a nonlinear model predictive controller based on that model. Chow et al [18] identified the thermal system by applying the reduced-order functional link neural network. Namburu et al [19] developed a generic FDD (fault detection and diagnosis) scheme for centrifugal chillers by employing three well-known statistical inference techniques, namely SVM (Support Vector Machine), PCA (principal component analysis) and PLS (partial least squares) to isolate faults. A nominal data-driven model of the chiller was also developed by PLS to predict the system response under new loading conditions. One shortcoming for data-driven approach is that insufficient training data will result in instability of models since prediction error becomes obvious in ranges out of which training data has covered.

Since some models are derived from advanced techniques like NN (Neural Network), no explicit knowledge is available. Nonlinearity and opaqueness characteristics of these optimization models pose challenge for traditional mathematical programming methods. Some heuristic search algorithms like SA (Simulated annealing) [20], EA (Evolutionary Algorithms) [21, 22, 23], and so on, are suited for finding near-optimal solutions for complex problems in building energy.

1.3 Computational Intelligence in modeling and optimization of HVAC systems

CI (Computational Intelligence) was first proposed by Bezdek [24] and gained widespread attention. Originated from emulating intelligent phenomenon in nature, computational intelligence centers on the study of adaptive mechanism to enable or facilitate intelligent behavior in complex and changing environment. It includes paradigms like neural network [25, 26], evolutionary computation [27, 28], swarm intelligence [29], fuzzy system [30] and so forth [31]. Computational intelligence methods provide great functions in nonlinear mapping and optimization. Thus, in recent years, many researches have been done in HVAC modeling, control and optimization areas [32, 33, 34].

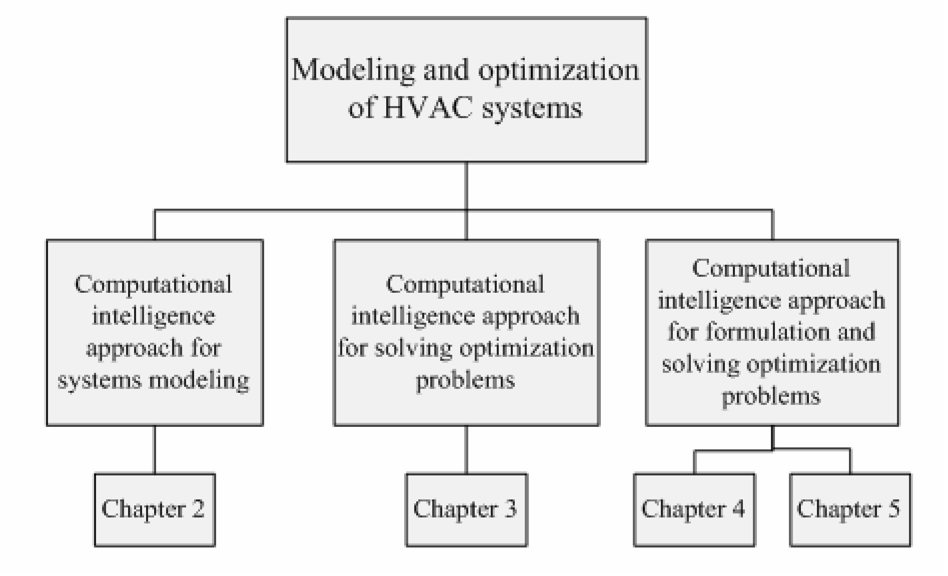
Modeling and optimization of HVAC systems
Computational intelligence approach for systems modeling
Computational intelligence approach for solving optimization problems
Computational intelligence approach for formulation and solving optimization problems
Chapter 2
Chapter 3
Chapter 4
Chapter 5

Figure 1.1 Structure of the thesis

Figure 1.1 illustrates the structure of the thesis. Different applications of computational intelligence in HVAC systems are introduced from Chapter 2 through Chapter 5. Chapter 2 focuses on modeling HVAC systems in a computational intelligence approach. A neural network is employed to establish nonlinear mappings among different parameters. Chapter 3 focuses on applying multi-objective evolutionary algorithm for solving the optimization model formulated by physics-based equations. Chapter 4 and 5 discuss the use of computational intelligence methods in model formulation and solving. In each chapter, an appropriate HVAC sub-system is selected. Figure 1.2 describes typical HVAC system. Chapter 2 focuses on establishing mappings among different parameters in a single HVAC zone. Chapter 3 presents an optimal ventilation control scheme in a single zone. Chapter 4 is concerned with optimization of the AHU (Air Handling Unit). Chapter 5 presents optimization of the VAV (Variable Air Volume) Box.

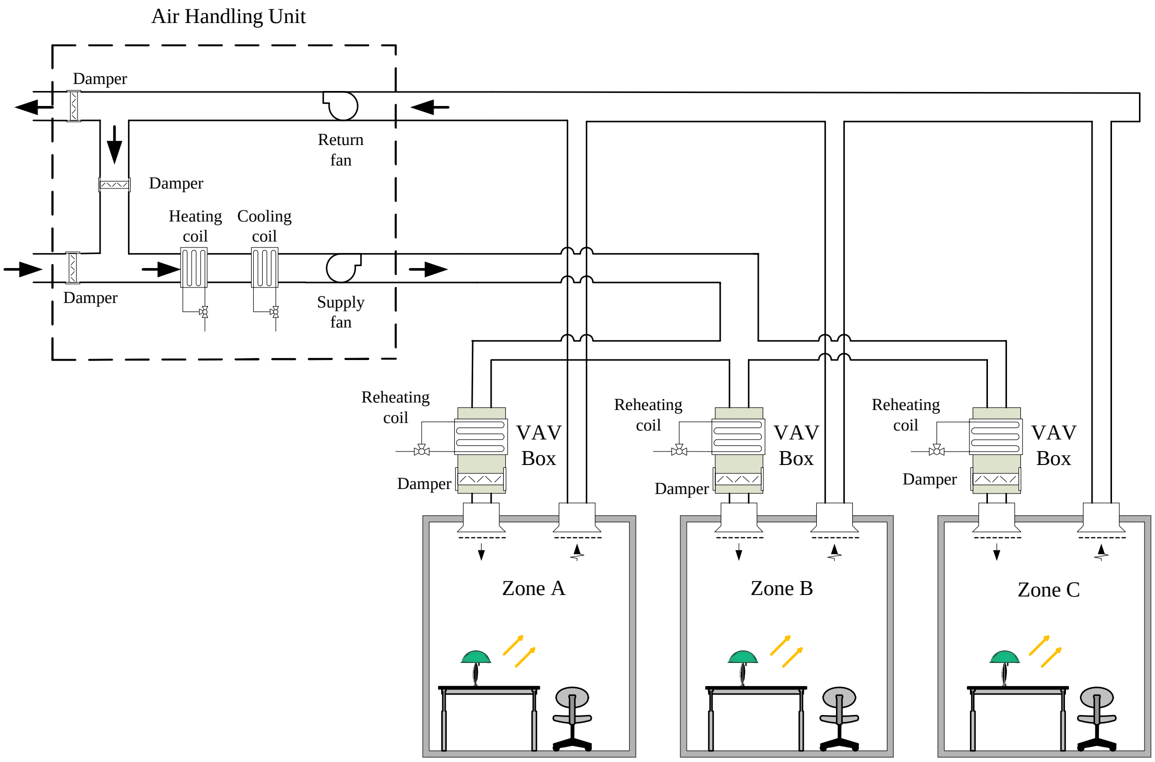

Figure 2.2 Diagram of a typical HVAC system

Computational intelligence allows modeling systems of different complexity. In Chapter 2, NN is used for modeling IAQ (indoor-air-quality) sensors used in HVAC systems. The IAQ sensors considered in the chapter measure three basic parameters, namely temperature, CO_2, and relative humidity. The computational results produced by models built with different data mining algorithms are discussed. In comparison with other data mining algorithms, NN produced the best results for all three IAQ sensors among all algorithms tested on the HVAC data set collected at an office-type facility. The models built with NN can serve as virtual IAQ sensors in buildings and be used for on-line monitoring and calibration of the IAQ sensors.

Computational intelligence methods also offer powerful searching techniques in solving complex optimization problems. In chapter 3, based on analytical model of indoor CO_2 concentration, a two-mode ventilation control of a single facility is formulated as a scheduling model over multiple time horizons. Using the CO_2 concentration as the major indoor air quality index and expected room occupancy schedule, optimal solutions leading to reduced CO_2 concentration and energy costs are obtained by solving the multi-objective optimization model formulated in the chapter. A modified evolutionary strategy algorithm is used to solve the model at different time horizons. The optimized ventilation schedules result in energy savings and maintain an acceptable level of indoor CO_2 concentration.

In Chapter 4, computational intelligence methods are first applied in both optimization problems formulating and solving. A model extracted by a neural network is selected for identifying the functional mapping between specific outputs and controllable and non-controllable inputs of the AHU. To minimize the cooling output while maintaining the corresponding thermal properties of the supply air within a certain range, a bi-objective optimization model is proposed. The evolutionary strategy algorithm is applied to solve the optimization problem with the optimal control settings obtained at each time stamp. The minimized AHU’s cooling output reduces the chiller’s load, which leads to energy savings.

In Chapter 5, computational intelligence methods are applied for optimizing the reheating process in VAV box. Compared to Chapter 4, time delay among inputs and outputs are investigated more deeply in this chapter. NN is selected to derive the temporal predictive models from real data and the reheating process is transformed into a bi-objective optimization model. Unlike previous two chapters of using evolutionary algorithms in solving optimization problems, a modified PSO (Particle Swarm Optimization) based on two levels of non-dominated solutions is introduced to find near-optimal solutions with decent computation cost in consideration of on-line implementation. Based on the model predictive control strategy, recommended control output is obtained to minimize the reheating output while maintaining the thermal comfort at certain acceptable level in the future.

Chapter 6 summarizes the concepts presented in the thesis and provides future research directions.

CHAPTER 2.

INDOOR-AIR-QAULITY VIRTUAL SENSOR MODELING AND ON-LINE MONITORING

2.1 Introduction

The growing complexity of building HVAC systems has become a major challenge for applying strategies for reducing costs and enhancing air quality [35]. Degraded equipment, failed sensors, improper installation, poor maintenance and aged control systems have contributed to poor performance of HVAC systems of various commercial buildings.

Indoor air quality is usually measured by the level of temperature [°C], CO_2 [PPM] and relative humidity [%]. The performance of IAQ sensors greatly impacts the air quality and energy savings of HVAC systems. The physical sensors installed in any HVAC system degrade over time, and this leads to inferior performance, poor air quality, and energy waste due to incorrect feedback from the degraded IAQ sensors. Various researchers have focused on modeling sensors, detection of sensor faults and developing cost-efficient control strategies for HVAC systems. Namburu et al. [19] proposed a real-time fault diagnosis scheme for chillers based on a data-driven approach. Cho et al. [36] developed a model based on pattern classification and residual ratios to diagnose, identify, and detect multiple-faults occurring in HVAC systems. Salsbury et al. [37] used simulation to predict performance targets and compare monitored system outputs for performance validation and energy analysis. Hou et al. [38] combined a rough set approach with a neural network algorithm to build a model based on past HVAC performance data. The model was intended to detect and diagnose sensor faults in HVAC systems. Schein et al. [39] developed a rule-based method for fault detection in air handling systems. The rules were derived by experts from mass and energy balances.

In this chapter, ANN has been applied to build models of IAQ sensors. The constructed models are used as on-line profiles and virtual sensors for indoor air quality. They can be also used to monitor the performance of IAQ sensors installed in HVAC systems. The IAQ sensor models are built based on historical data from an HVAC system at an existing facility. Identifying sensor faults and on-line monitoring of the IAQ sensors is useful for optimizing the performance of HVAC systems.

2.2 Methodology for IAQ Sensor Modeling and On-line Monitoring

Instead of having a real physical sensor, virtual sensor refers to its word meaning of using other sensed data collected from related sensors in replacement of measuring the process condition or product properties directly. Virtual sensor technology is often applied when the direct observation is impossible, expensive or unreliable. In this chapter, neural network is applied in building the virtual sensor to achieve the function of on-line monitoring. Neural networks have been widely used to model and control dynamic processes because of their capability of approximating nonlinear model functions.

Control charts are used in process monitoring to determine and eliminate the sources of process variation so that the process returns to its normal state. They have been widely researched in the statistical quality and process control literature [40]. Recent advances in profile monitoring have led to applications in manufacturing, calibration, logistics, service, marketing, finance and accounting [41, 42, 43, 44, 45].

Figure 2.1 illustrates the basic concept of IAQ sensor modeling and on-line monitoring presented in this paper. A data mining algorithm is used to identify IAQ sensor models based on the historical HVAC process data. The model can be updated to reflect the process change over time. The update frequency could be, for example, two weeks. The operational update frequency depends on the HVAC system operational conditions and the accuracy requirements. Alternatively, a separate routine could monitor the model performance and refresh the model once its performance degraded. A control chart generated from the HVAC data can be used for on-line monitoring of an IAQ sensor. The IAQ models and control chart monitor the IAQ sensor performance at a certain time interval, e.g., every five minutes.

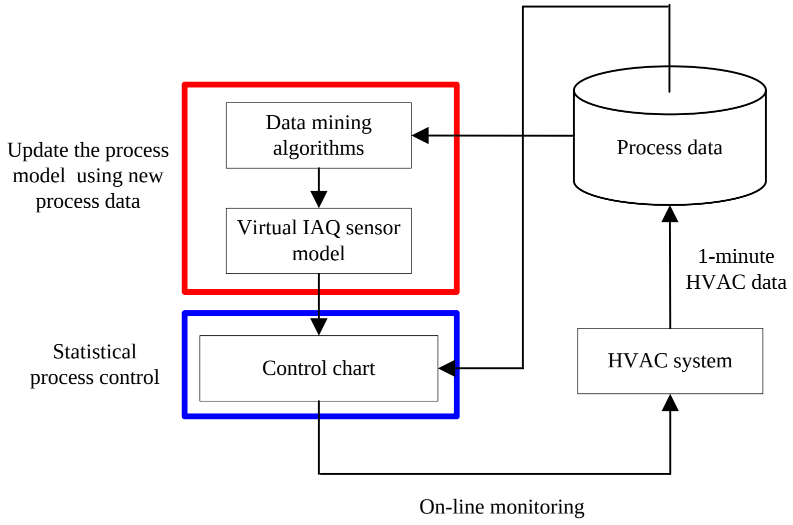

Figure 2.1. Modeling and on-line monitoring of IAQ sensors

The residual control chart approach (statistical quality control) [42, 43] is used to analyze residuals between the model predicted IAQ value and the observed (measured by sensor) IAQ value. The residual ε is expressed in Equation (2.1) [43]:

$$\varepsilon = \hat{y} - y \qquad (2.1)$$

where y is the observed IAQ value, and $\hat{y}$ is the reference value predicted by an IAQ sensor model.

The control chart approach [41, 42, 43] allows the residuals and their variations to be monitored, and thus detect abnormal conditions and an IAQ sensor fault. A training data set of N_{train} observations with outliers removed was selected to build a control chart. The training data set is represented as $y\_TrainSet = \left[y(i), \hat{y}(i) \right]$, $i = 1, \ldots, N_{train}$.

Using the training data set, the residual ε for each point is computed, as well as the mean and the standard deviation of ε. The mean residual μ_{Train} and the standard deviation σ_{Train} are shown in Equations (2.2, 2.3) [46]:

$$\mu_{Train} = \frac{1}{N_{train}} \sum_{i=1}^{N} (\hat{y}(i) - y(i)) \qquad (2.2)$$

$$\sigma_{Train} = \sqrt{\frac{1}{N_{train}-1} \sum_{i=1}^{N} ((\hat{y}(i) - y(i)) - \mu_{Train})^2} \qquad (2.3)$$

The test data set $y\_TestSet = \left[y(i), \hat{y}(i) \right]$ includes N_{test} consecutive data points drawn in time sequence from the test data set.

Similarly, the mean residual μ_{Test} and the standard deviation σ_{Test} of the test data set are expressed as Equations (2.4, 2.5) [46]:

$$\mu_{Test} = \frac{1}{N_{test}} \sum_{i=1}^{n} (\hat{y}(i) - y(i)) \qquad (2.4)$$

$$\sigma_{Test} = \sqrt{\frac{1}{N_{test}-1} \sum_{i=1}^{n} ((\hat{y}(i) - y(i)) - \mu_{Test})^2} \qquad (2.5)$$

Once μ_{Train} and σ_{Train} are known, the upper and lower control limits of the control chart are computed and used to detect anomalies. Based on Equation (2.2, 2.3), the control limits of the control chart are derived from Equation (2.6) [41, 42]:

$$\begin{aligned} UCL_1 &= \mu_{Train} + \eta \frac{\sigma_{Train}}{\sqrt{N_{test}}} \\ CenterLine_1 &= \mu_{Train} \\ LCL_1 &= \mu_{Train} - \eta \frac{\sigma_{Train}}{\sqrt{N_{test}}} \end{aligned} \qquad (2.6)$$

N_{test} is the number of points in $y\_TestSet$, η is the integer multiple for the control limits, and N_{test} (usually fixed as 3) can be adjusted to make the control chart less sensitive to the data variability and thus reduce the risk of false alarms. In this paper, N_{test} was set at 5 to make the control chart less sensitive to the data variability. If μ_{Test} is above UCL_1 or below LCL_1, the IAQ parameter value at the sampling time $y\_TestSet$ is considered to be deficient, and this type of fault detected by the control chart is defined as Fault Type I.

Similarly, the control limits for $\sigma^2{}_{Test}$ are calculated from Equation (2.7) [41, 42]:

$$\begin{aligned} &UCL_2 = \frac{\sigma^2_{Train}}{N_{test}-1} \times \chi^2_{\alpha/2,N_{test}-1} \\ &CenterLine_2 = \sigma^2_{Train} \\ &LCL_2 = 0 \end{aligned} \qquad (2.7)$$

where $\chi^2_{\alpha/2,N_{test}-1}$ denotes the right $\alpha/2$ percentage points of the chi-square distribution, $N_{test}-1$ is the degree of freedom of the chi-square distribution. The parameter α needs to be adjusted to make the control chart less sensitive to the variability of the data. LCL_2 is set to 0 to indicate that the variation of residuals in the test data is 0, so that the measured IAQ value matches the reference IAQ value in the normal status. If σ^2_{Test} is above UCL_2, the IAQ parameter value at the sampling time $y\_TestSet$ is considered as deficient, and this type of fault detected by control chart is defined as Fault Type II.

2.3 Case Study and Computational Results

2.3.1 Data Description

The data used in this research was collected at the Iowa Association of Municipal Utilities (IAMU) Office Building and Training Complex, which included 12,500 square feet of office space, maintenance, and shop facilities. The HVAC control system collected data for more than 60 parameters with a sampling interval of one minute. Due to the type of data collecting system, the values stored for all HVAC parameters are the (last-measured) point data rather than the one-minute average data. The HVAC data from the sensors installed at the IAMU was supplemented with the outside weather data collected by the Iowa Energy Center over a two-month period.

For each IAMU office/room of interest to this research, the data was collected for the same set of IAQ parameters: temperature, CO_2, and relative humidity, as well as other parameters. In this paper, the IAMU auditorium was selected for in-depth analysis;

however, the approach presented can be applied to any office in the building. Table 2.1 lists the HVAC parameters of the IAMU auditorium used in this research. The first three parameters in Table 2.1 are the indoor air quality parameters collected from the IAQ sensors; the parameter “Aud_IAQ_Temp” is the temperature measured at the thermostat installed in the auditorium; the last eight parameters indicate the outside weather conditions which were collected by the Iowa Energy Center.

Table 2.1 Parameter description of the HVAC data set

Parameter	Description	Unit
Aud_IAQ_Temp	Auditorium temperature from IAQ sensor	°C
Aud_IAQ_ CO_2	Auditorium CO_2 from IAQ sensor	PPM
Aud_IAQ_RH	Auditorium Relative Humidity from IAQ sensor	% RH
Aud_Temp	Controllable auditorium temperature	°C
Aud_Lite	Auditorium light level	FC
BAR-PRES	Barometric pressure	Bar
OA-HUMD	Relative humidity	% RH
OA-TEMP	Dry-bulb temperature	°C
SOL-BEAM	Direct normal solar irradiation	Btu/hr-ft^2
SOL-HORZ	Total horizontal irradiation	Btu/hr-ft^2
WIND-DIR	Wind direction	°
WIND-VEL	Wind speed	Mile/hour

The data set used for IAQ sensor modeling was divided into two independent data subsets, a training data set and a test data set. The training data set was used to develop models of IAQ sensors, while the test data set was used to validate the performance of the models learned from the training data set. Figures 2.2 to 2.4 show typical plots of temperature, CO_2 and relative humidity, respectively over a one-day period. It can be observed that the IAQ parameter values change during the day. X axis refers to the data point sequence number.

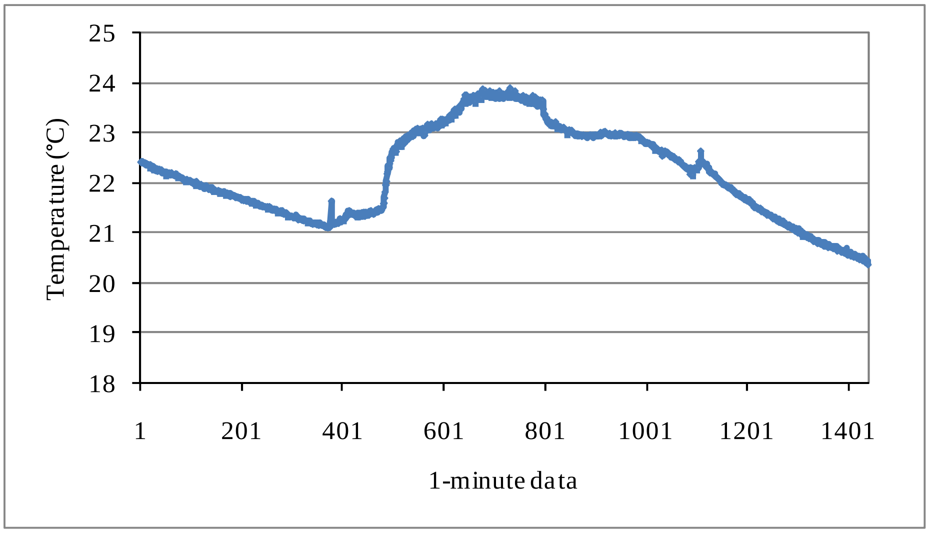

Figure 2.2. Illustrative plot of IAQ temperature parameter

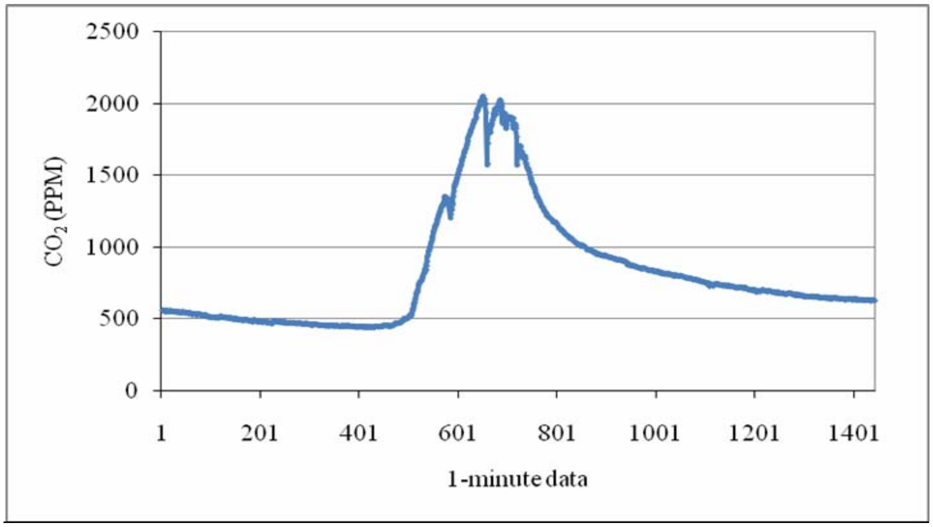

Figure 2.3. Illustrative plot of IAQ CO2 parameter

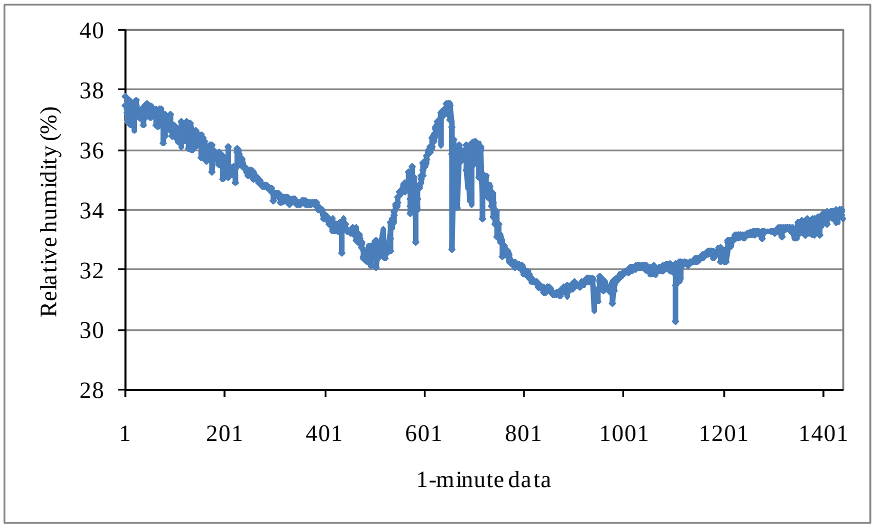

Figure 2.4. Illustrative plot of IAQ humidity parameter

The values of the temperature, CO_2, and relative humidity are continuously changing. However, the IAQ sensor plots shown in Figures 2.2 to 2.4 are not all continuously-changing curves, rather they contain steep changes. Various reasons could explain the abnormal spikes, including the degraded performance of sensors, faulty HVAC control, and unexpected working conditions around sensors. If accurate and robust virtual models of IAQ sensors were available, the physical IAQ sensors could be monitored and calibrated on-line or even replaced by the virtual IAQ sensors.

2.3.2 Algorithm Selection for IAQ Sensor Modeling

Data mining algorithms were used to build models for the IAQ sensors in the HVAC system. Virtual IAQ sensor modeling used other HVAC parameters as predictors to predict IAQ parameters as dependent, and the IAQ parameters included temperature, CO_2 and relative humidity. The relationship between IAQ parameters and various other HVAC parameters are complicated, and thus it is hard to identify the model and accurately predict IAQ parameters with high-dimension HVAC parameters as input using mathematical modeling. However, data mining is a powerful tool in extracting knowledge from voluminous data.

A virtual IAQ sensor model represents the underlying function between the IAQ parameter and the other HVAC parameters. Equations (2.8) to (2.10) show the sensor models for predicting temperature, CO_2 and relative humidity, respectively.

$$\begin{gathered} y_{Aud\_IAQ\_Temp} = f(x_{Aud\_IAQ\_CO2},\ x_{Aud\_IAQ\_RH},\ v_{Aud\_Temp}, v_{Aud\_Lite} v_{BAR\text{-}PRES}, v_{OA\text{-}HUMD}, \\ v_{OA\text{-}TEMP},\ v_{SOL\text{-}BEAM}, v_{SOL\text{-}HORZ}, v_{WIND\text{-}DIR}, v_{WIND\text{-}VEL}) \end{gathered} \tag{2.8}$$

$$\begin{gathered} y_{Aud\_IAQ\_CO2} = f(\ x_{Aud\_IAQ\_RH},\ x_{Aud\_IAQ\_Temp}, v_{Aud\_Temp}, v_{Aud\_Lite} v_{BAR\text{-}PRES}, v_{OA\text{-}HUMD}, \\ v_{OA\text{-}TEMP},\ v_{SOL\text{-}BEAM}, v_{SOL\text{-}HORZ}, v_{WIND\text{-}DIR}, v_{WIND\text{-}VEL}) \end{gathered} \tag{2.9}$$

$$\begin{gathered} y_{Aud\_IAQ\_RH} = f(\ x_{Aud\_IAQ\_CO2},\ x_{Aud\_IAQ\_Temp}, v_{Aud\_Temp}, v_{Aud\_Lite} v_{BAR\text{-}PRES}, v_{OA\text{-}HUMD}, \\ v_{OA\text{-}TEMP},\ v_{SOL\text{-}BEAM}, v_{SOL\text{-}HORZ}, v_{WIND\text{-}DIR}, v_{WIND\text{-}VEL}) \end{gathered} \tag{2.10}$$

In Equations (2.8) to (2.10), y is the dependent IAQ parameter, x is the IAQ parameter used in this model as a predictor, v is the parameter indicating the outside weather conditions. The v and x parameters are listed in Table 2.1. The model $f(\bullet)$ is learned by a data mining algorithm. An obvious advantage of the data-driven approach is that $f(\bullet)$ can be easily and timely updated by the most current HVAC process data. Deriving an accurate virtual IAQ model that maps complicated relationships among the parameters of the HVAC system is a challenge.

The selection of an appropriate data mining algorithm is important for building an accurate, stable, and robust IAQ model. Different data mining algorithms were applied for IAQ sensor modeling, and the performance of the various data mining algorithms was analyzed. Two basic metrics, the MAE (mean absolute error) and Std (standard deviation of absolute error) were used to compare prediction accuracy. They were computed to select the best data mining algorithm to extract the accurate IAQ sensor model (Equations. (2.8) to (2.10)). The small value of the MAE and Std implies the superior prediction performance of the IAQ model. The AE (absolute error), MAE (mean absolute error), and the Std (standard deviation) are expressed in Equations (2.11) to (2.13) [46].

$$AE = \left|\hat{y} - y\right| \qquad (2.11)$$

$$MAE = \frac{\sum_{i=1}^{N} AE(i)}{N} \qquad (2.12)$$

$$Std = \sqrt{\frac{\sum_{i=1}^{N} (AE(i) - MAE)}{N-1}} \qquad (2.13)$$

where $\hat{y}$ is the predicted IAQ parameter value, y is the observed (measured by sensor) IAQ parameter value, and N is the number of test data points used to validate the performance of the IAQ sensor model learned by a data mining algorithm.

To select the best data mining algorithm for constructing accurate and robust IAQ sensor models, a two-day-long data set was used. Table 2.2 shows data set 1 with a beginning time stamp of “3/30/2005 0:00” and an ending time stamp of “3/31/2005 23:50”. Data set 1 was divided into two data subsets using a random sampling method, data set 2 and data set 3. Data set 2 contains 2304 randomly selected data points and was used to develop a model $f(\bullet)$ estimating the IAQ parameters. Data set 3 includes 576 randomly selected data points and was used to test the performance of the model $f(\bullet)$ learned from data set 2.

Table 2.2 The two-day-long data set characterization

Data set	Sample size (%)	Description
1	100	Total data set; 2880 observations
2	80	Training data set; 2304 observations
3	20	Test data set; 576 observations

Four different algorithms were used to learn (identify) the IAQ sensor models. They included the MLP (Multi-layer perceptron) NN (neural network) [47, 48], RBF (Radio-basis-function) NN [47, 49], support vector machine regression (SVM) [50, 51], and Pacereg (Pace regression) [52, 53]. The MLP NN and RBF NN algorithms are usually used in non-linear regression and classification modeling due to their ability to capture complex relationships between parameters. The SVM regression is a supervised machine learning algorithm. It avoids difficulties of using linear functions in the high dimensional feature space and optimization problem is transformed into dual convex quadratic programming problem. In regression case the loss function is used to penalize errors. The Pace regression algorithm consists of a group of estimators that are either optimal overall or optimal under certain conditions. It is a relatively new approach for developing linear models in high dimensional spaces.

In this research 35 MLP NN and 35 RBF NN models with different kernels and structures were built, and the most accurate and robust model was selected based on the objective function of sum of square errors between observed and predicted values. Five different activation functions were used for the hidden and output neurons, namely, the logistic, identity, tanh, exponential, and sine function. The number of hidden units was set between 4 to 14, and the weight decay for both hidden and output layer varied from 0.0001 to 0.001.

Table 2.3 summarizes the prediction performance of IAQ sensor models learned by different data mining algorithms. In the figure, mean and std are computed from Equations (2.12) and (2.13). Max and min are the maximum and minimum absolute error from data set. The performance of data mining algorithms is ranked using the MAE (Equations (2.12) and (2.13)) metrics for the test data set from Table 2.2. The small value of the MAE and Std indicates accurate, stable, and robust prediction performance. Figures 2.5 to 2.7 illustrate the mean absolute error for each of the three IAQ sensor models (temperature, CO_2 and relative humidity). Y axis refers to the mean absolute error of the model obtained by different algorithm. The MLP algorithm performed best among the four data mining algorithms due to its smallest MAE and Std in all three IAQ sensor models, while the RBF performed the worst, as indicated by the largest value of MAE and Std. The Pacereg and SVM algorithms performed quite well for prediction of temperature and relative humidity. The MLP algorithm provided high quality predictions for the test data set and captured the HVAC system dynamics with high fidelity. The performance of models built by the MLP algorithm is validated in Section 2.3.3, based on a larger data set.

Table 2.3 Prediction accuracy of different algorithms

Algorithm	Parameter	Mean	Std	Max	Min
MLP	Temp (°)	0.0311	0.0276	0.3052	0.0000
	CO_2 (PPM)	6.4238	8.5631	86.1721	0.0133
	RH (%)	0.1111	0.1096	1.2297	0.0004
RBF	Temp (°)	0.9478	0.6593	2.9145	0.0096
	CO_2 (PPM)	184.1855	208.7004	799.9020	0.1212
	RH (%)	1.7544	0.9513	4.7719	0.0069
Pacereg	Temp (°)	0.1668	0.1474	0.7421	0.0007
	CO_2 (PPM)	71.3895	78.2013	534.2520	0.3677
	RH (%)	0.5458	0.4051	2.7928	0.0019
SVM	Temp (°)	0.1473	0.1951	0.9691	0.0000
	CO_2 (PPM)	61.1816	106.5812	575.9267	0.0069
	RH (%)	0.5343	0.4859	3.3849	0.0015

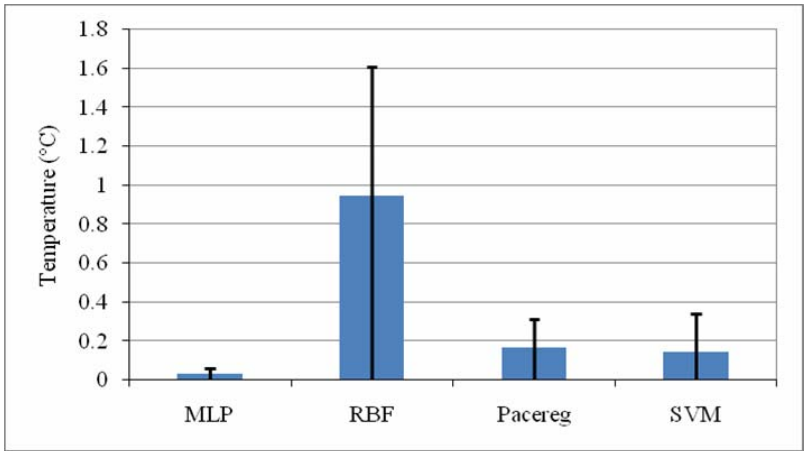

Figure 2.5 The bar-chart of prediction performance of temperature for test data set 3 of Table 2.2.

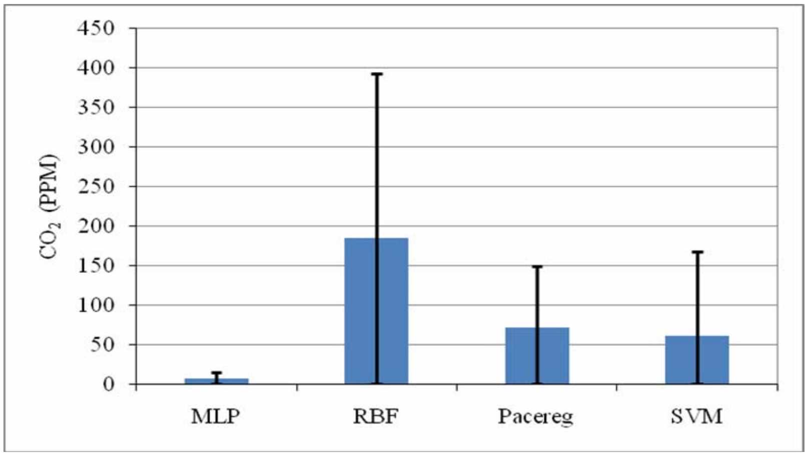

Figure 2.6 The bar-chart of prediction performance of CO_2 for test data set 3 of Table 2.2.

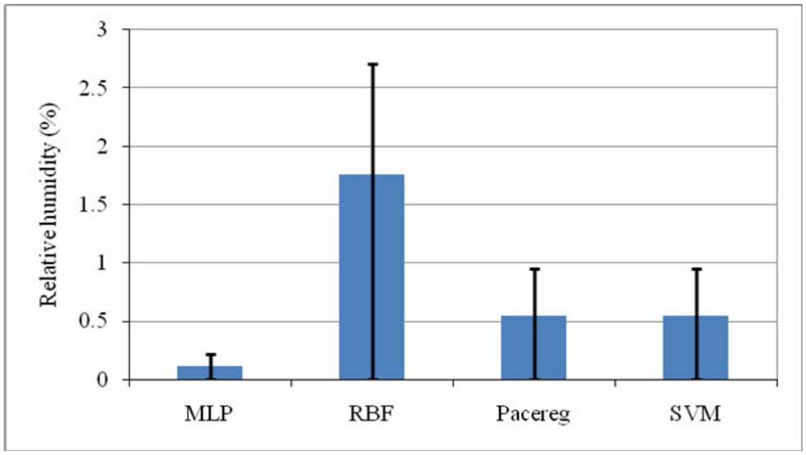

Figure 2.7 The bar-chart of prediction performance of humidity for test data set 3 of Table 2.2.

Figures 2.8 to 2.10 show respectively the first 100 predicted (by MLP) and observed temperature, CO_2 and relative humidity computed for the test data set of Table 2.2. It is easy to see that the observed and predicted values of IAQ parameters are almost identical and even overlap, and the predicted value follows exactly the trend of the observed value. The three IAQ sensor models built by MLP algorithm predict accurately CO_2, temperature and relative humidity, respectively, on the data set 3 of Table 2.2. X axis refers to the data point sequence number.

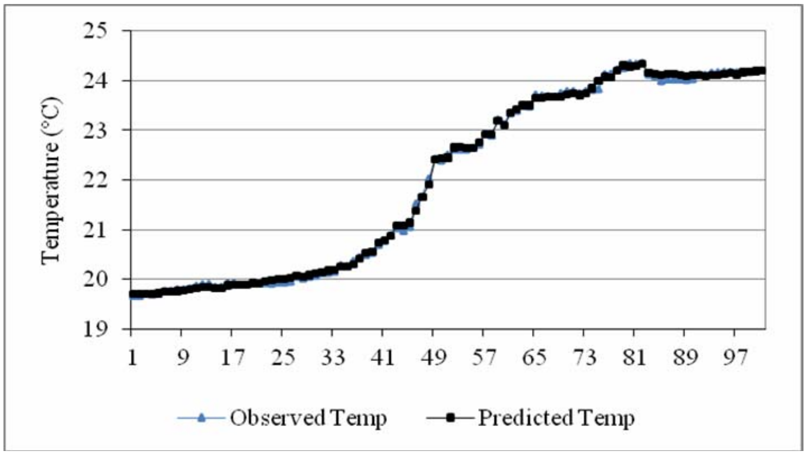

Figure 2.8 Predicted and observed value of IAQ temperature of the test data set from Table 2.2.

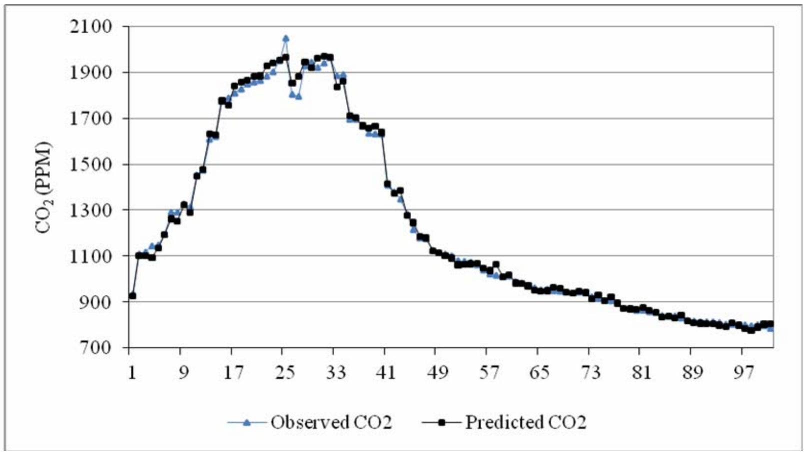

Figure 2.9 Predicted and observed value of IAQ CO$_2$ of the test data set from Table 2.2.

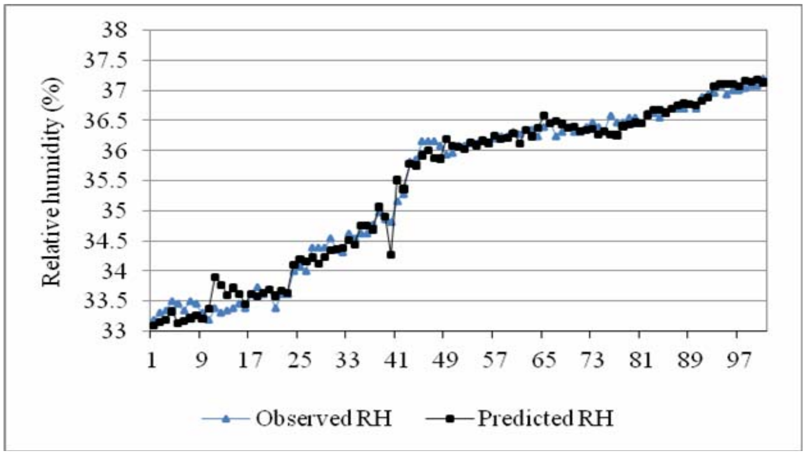

Figure 2.10 Predicted and observed value of IAQ humidity of the test data set from Table 2.2.

2.3.3 Validation of IAQ Models Based on a Two-Week Data Set

The MLP NN algorithm performed best among the four different data mining algorithms based on the two-day data set, and thus it was selected for further investigation on a two-week data set. Table 2.4 shows the two-week data set with a beginning time stamp of “5/1/2007 0:00” and an ending time stamp of “5/11/2007 23:50”. Data set 1 was divided into three data subsets, data set 2, data set 3 and data set 4. All the data sets contain consecutive data points drawn in a time sequence. Considering that the HVAC system of the IAMU building operates at different conditions during weekdays and weekends, the weekend data of 5/5/2007 and 5/6/2007 was removed from data set 1, and thus two training data sets were created due to a weekend (two-day) data set between the two weekday data sets. Data set 2, containing 5760 time-consecutive data points, and data set 3, containing 4320 time-consecutive data points, were used to develop a model $f(\bullet)$ estimating the IAQ parameters. Data set 4, containing 2880 time-consecutive data points, was used to test the performance of the models $f(\bullet)$ learned from data set 2 and data set 3.

Table 2.4 The description of two-week data set

Data Set	Start Time	End Time	Description
1	5/1/2007 0:00	5/11/2007 23:50	Total data set; 15840 observations
2	5/1/2007 0:00	5/4/2007 23:50	Training data set;5760 observations
3	5/7/2007 0:00	5/9/2007 23:50	Training data set; 4320 observations
4	5/10/2007 00:00	5/11/2007 23:50	Test data set; 2880 observations

Figures 2.11 to 2.13 show the first 200 predicted (by MLP) and observed temperature, CO_2 and relative humidity for the test data set of Table 2.4. The observed and predicted values of IAQ parameters match well, and the predicted values follow exactly the trend of the observed value or even overlap. On the two-day data set shown in the Table 2.4, the three IAQ models constructed by MLP algorithm perform satisfactorily. X axis refers to the data point sequence number.

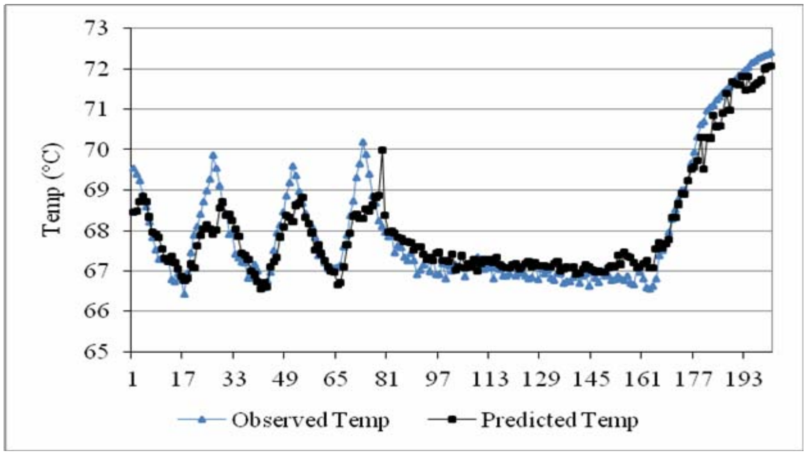

Figure 2.11 Predicted and observed values of IAQ temperature for the data set 4 from Table 2.4

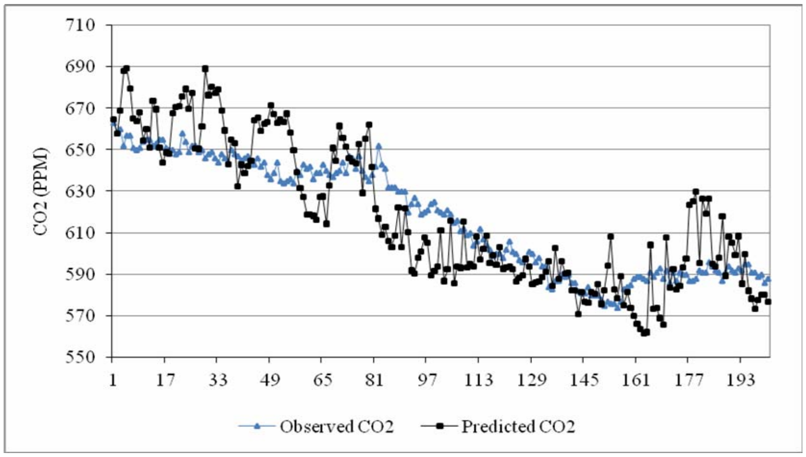

Figure 2.12 Predicted and observed values of IAQ CO_2 for the data set 4 from Table 2.4

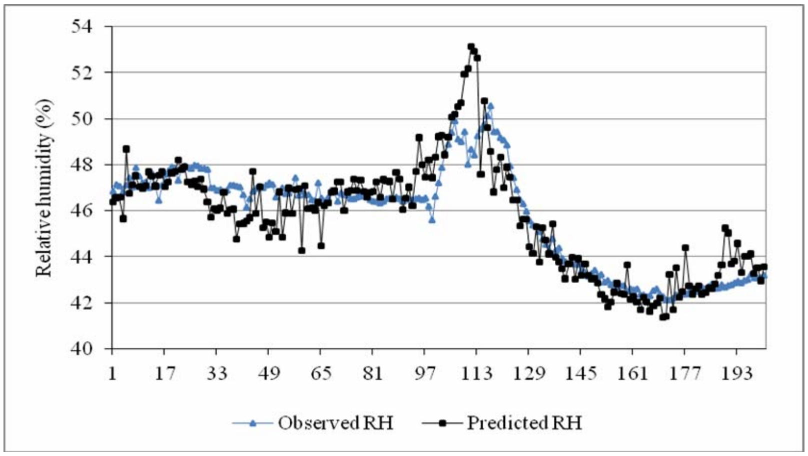

Figure 2.13 Predicted and observed values of IAQ humidity for the data set 4 from Table 2.4

Table 2.5 summarizes the prediction accuracy of three IAQ models built by the MLP algorithm. The MLP algorithm performs well on both the two-day data set (data set 2 and 3 in the Table 2.4) and the two-week data set (data set 4 in the Table 2.4). The performance is maintained accurate and robust. However, in real practice, updating the learned model with the new data is necessary for processes that are temporal. The temporal process modeling task is accomplished by using data mining algorithms. A large prediction error (MAE and Std) indicates that the model built on historical data needs to be updated. The update time is impacted by various factors, e.g., if the seasonal and rapid change of weather conditions.

Table 2.5 Prediction accuracy of different algorithms for the test data set of Table 2.4

Parameter	Mean	Std	Max	Min
Temp (°)	0.2749	0.2815	2.1389	0.0001
CO_2 (PPM)	25.4649	28.9067	277.6084	0.0010
RH (%)	0.7443	0.6962	4.8757	0

The sensitivity of a neural network's output to its input perturbation is an important issue in theory and practice. The sensitivity analysis [47, 49] indicates input variables that are most important for a particular neural network. It often identifies variables that can be safely ignored in subsequent analyses and key variables that must always be retained. Sensitivity analysis ranks variables according to the deterioration of the model performance if that variable is no longer available for the model. In so doing, it assigns a single rank value to each variable. Table 2.6 shows the sensitivity analysis of the three IAQ sensor models built by the MLP based on the two-week data set. Different parameters have different sensitivity ranks in the three IAQ sensor models. The sensitivity analysis offers insight into the complicated and non-linear relationship between various HVAC parameters and the outside weather condition parameters.

Table 2.6 Sensitivity analysis of MLP NN for test data set of Table 2.4

Temperature		CO_2		Relative Humidity	
Parameter	Sensitivity	Parameter	Sensitivity	Parameter	Sensitivity
Aud_Temp	797.5280	Aud_IAQ_Temp	207.6935	Aud_IAQ_Temp	305.3133
Aud_IAQ_CO_2	235.3234	Aud_Temp	183.0385	Aud_IAQ_ CO_2	236.4275
BAR-PRES	94.3404	BAR-PRES	181.5638	BAR-PRES	188.0380
Aud_IAQ_RH	66.1309	Aud_IAQ_RH	101.6903	Aud_Temp	73.9207
OA-TEMP	34.5833	SOL-HORZ	66.4172	OA-TEMP	22.9534
OA-HUMD	10.6605	SOL-BEAM	54.0379	SOL-BEAM	15.5511
SOL-BEAM	6.3443	OA-TEMP	8.7827	SOL-HORZ	9.5619
SOL-HORZ	5.2809	Aud_Lite	3.2909	Aud_Lite	7.0401
Aud_Lite	2.8174	OA-HUMD	1.4673	OA-HUMD	5.7857
WIND-DIR	1.3425	WIND-DIR	1.1229	WIND-DIR	1.2634
WIND-VEL	1.0893	WIND-VEL	1.0449	WIND-VEL	1.0526

2.3.4 On-line IAQ Sensor Monitoring With Virtual Sensor Model

The approach and equations of Section 2.3 have been implemented to form control charts for the three IAQ sensor models. The three IAQ sensor models (Equations (2.8) - (2.10)) are constructed by the MLP NN algorithm based on data set 2 and data set 3 of Table 2.4, and the parameters of three control charts for CO_2, temperature and relative humidity are computed based on the same training data set. On-line monitoring simulation of the three virtual IAQ sensor models is performed and validated on the testing data set of Table 2.4.

Table 2.7 shows the control chart parameters of the three IAQ sensor models formed based on the training data set of Table 2.4. The parameter α in Equation (2.7) is fixed as 2, and η of Equation (2.6) is set to 3 to enhance the confidence of detecting anomalies in the IAQ sensors. Two important parameters α and η can be adjusted according to the real situation in practice to reduce the risk of a false alarm in mistaking the normal state as an abnormal one. A control chart monitors the residual mean and variation of the sampling data at the same time. In this research, the UCL_1 and LCL_1 calculated in Equation (2.6) monitor the residual mean of the test data set, and the sensor fault detected due to the abnormal residual mean is defined as Fault Type I; $UCL2$ calculated from Equation (2.7) monitors the residual variation of the test data set, and the sensor fault detected due to the abnormal residual variation is defined as Fault Type II.

Table 2.7 Control limit values for IAQ parameters

Parameter	UCL1	LCL1	UCL2	LCL2
Temp (°C)	0.2794	-0.2780	0.2330	0.0000
CO_2 (PPM)	51.6453	-49.6820	4278.0080	0.0000
RH (%)	1.1661	-1.1550	2.2448	0.0000

Figures 2.14 to 2.16 show the on-line monitoring simulation results of the control limits of the three IAQ sensor models trained by the MLP NN algorithm based on the training data set of Table 2.4; the observed IAQ parameter curves and the curve of the upper and lower limits were constructed from the test data set of Table 2.4; the first three hundred data points among the 2800 test data points of Table 2.4 are shown in three figures. The upper limits (boundary) of three normal IAQ parameters curves are calculated based on the UCL_1 of Equation (2.6). The lower limits (boundary) of three normal IAQ parameter charts are calculated based on the LCL_1 of Equation (2.6). X axis refers to the data point sequence number.

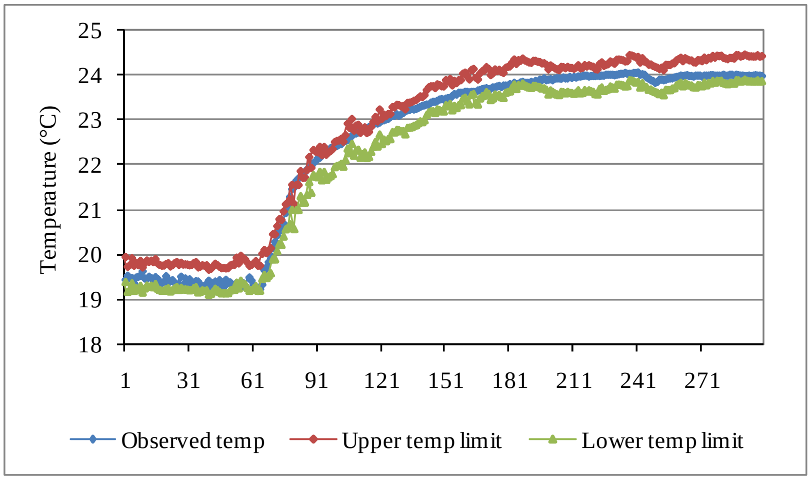

Figure 2.14 Temperature control chart

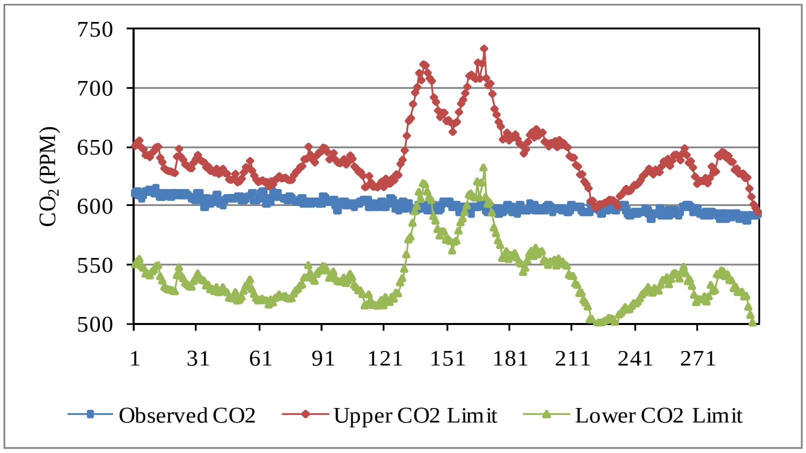

Figure 2.15 CO_2 control chart

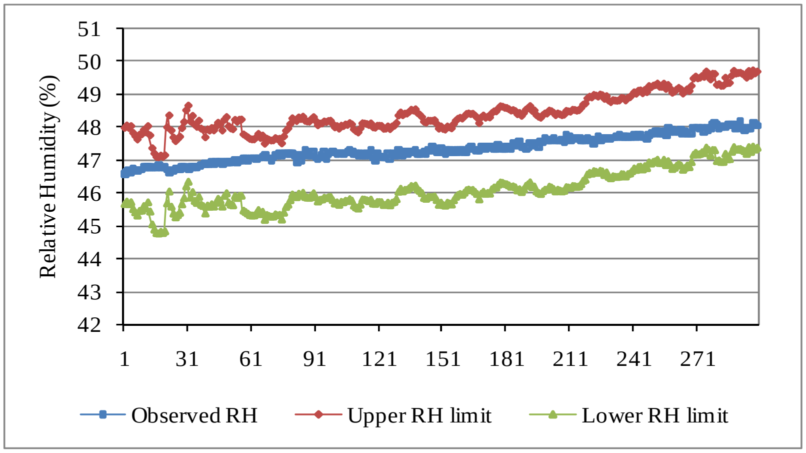

Figure 2.16 Humidity control chart

Table 2.8 shows the statistics of the on-line monitoring simulation results of three IAQ sensor control charts based on data set 4 of Table 2.4. Fault Type I is the abnormal data point detected by the residual mean (Equation (2.6)), and Fault Type II is the sensor fault detected by residual variation. Table 2.8 shows the number of Type I faults and Type II faults and their percentage among the total test data set (2800 data points in total) of Table 2.4. The RH control chart detected more sensor faults than CO_2 and temperature. There several possible explanation for this: the control chart could be sensitive, the relative humidity could be hard to measure by sensors, the sensor could have drifted or operated in abnormal working conditions, or the prediction of IAQ models may not have been accurate and robust enough. As shown in the Table 2.8, fault type II has less number of faults detected. Therefore, the parameters in Equations (2.6, 2.7) could make the three control charts more robust and less sensitive. These parameters could be adjusted dynamically based on the practical operations of an individual HVAC system.

Table 2.8 The statistics of IAQ sensor fault detection of control charts

IAQ parameter	Fault Type I (%)	Fault Type II (%)	Fault Type I	Fault Type II
Temperature(°)	9.4097	2.7431	271	79
CO_2 (PPM)	10.7292	1.8750	296	54
RH (%)	20.3125	0.5903	585	17

2.4 Summary

A comprehensive comparative analysis of the IAQ sensor models built with different data mining algorithms and a sensitivity analysis of the three IAQ sensor models built by the MLP was reported in this chapter. The performance of the selected MLP algorithm was validated on a two-week data set. MLP neural network identified IAQ sensor models from the actual HVAC process data, and capture the complicated relationship between HVAC parameters and outside weather condition parameters.

The IAQ sensor models were used as the reference IAQ sensors (on-line profile) for monitoring the performance of physical IAQ sensors and the indoor air quality of the HVAC system. The control chart approach can be used to monitor the residuals between the observed and the reference IAQ parameter values, and variation of the residuals. The IAQ sensor faults in non-stationary HVAC processes with certain structures can be detected in an on-line fashion. On-line IAQ sensor monitoring provides a data-driven approach for IAQ sensor fault detection for the HVAC industry. Three IAQ sensor models could be used as virtual sensors for monitoring, calibration, and even replacement of the physical sensor installed in HVAC systems. Virtual sensor provides the estimated value of a parameter in place of the actual physical sensor. The approach developed in this paper provides a basis for HVAC control optimization sustaining indoor air quality at lowered energy costs.

CHAPTER 3.

OPTIMAL DECISION MAKING IN VENTILATION CONTROL

3.1 Introduction

Maintaining air quality and providing thermal comfort is important for facilities supported by heating, ventilating and air-conditioning (HVAC) systems. According to published statistics, HVAC systems account for large proportion of energy consumption in both commercial and residential buildings [1]. Therefore, appropriate consumption of energy while maintaining the desired air quality has an impact on energy cost and indoor comfort.

The traditional approach to ventilation is to provide a fixed minimum ventilation rate per person based on the maximum occupancy of a facility. To provide air quality guidelines, ASHRAE Standard 90.1 [54] specifies the minimum ventilation rate of 2.5 l/s per person, while ASHRAE Standard 62-2004 [55] has been revised to the minimum ventilation rate of 10 l/s per person [56]. The number of occupants in any facility varies over time, and it is rare that the facility is fully occupied. This provides a good opportunity to save energy by ventilating facilities on demand [57]. Thus, the demand-control ventilation (DCV) is a commonly used strategy in HVAC systems based on signals from the indoor sensors, e.g., a CO_2 sensor. Both simulations and field tests of the CO_2-based DCV have demonstrated the potential to save energy [58], especially in facilities with a high occupancy density. A major difficulty with this approach is that CO_2 can only be used as a surrogate of human generated pollutants, whereas a CO_2 sensor cannot respond to pollutants such as emissions from furniture or painted materials. The location and stability of CO_2 sensors are also problematic. Therefore, different control strategies [59, 60] have been developed to deal with these issues. Other sensors like the VOC (volatile organic compound) sensor, occupancy sensor, humidity sensor, particle sensor, and so on, are used to modulate the ventilation rate over time under various conditions. In addition, devices such as air-side economizers are also used in ventilation systems to reduce energy consumption [61]. The quantity of fresh air supply is determined on the basis of the outside air dry-bulb temperature, enthalpy or other thermal properties. These approaches are usually cost-effective in areas where the heating or cooling cost is high.

Various optimization models [62, 63, 64] and algorithms [65, 66] have been discussed in the HVAC literature. In this chapter, the on-off ventilation control is formulated as an optimization model. The model involves three objectives, namely the fan-on time period, the average CO_2 above threshold, and the time period corresponding to the CO_2 above a threshold. The model is solved by an evolutionary algorithm. By optimizing fan on-and-off schedules on the basis of the trade-off among the three objectives, energy savings can be achieved, while proper air quality can be ensured by maintaining the CO_2 concentration in an acceptable range without installing any analog indoor sensors.

3.2 Methodology for Optimal Decision Making in Ventilation Control

3.2.1 Multiobjective Optimization by Evolutionary Algorithms

Many optimization problems are not only non-linear, which asks for heuristic approaches in solving these problems, but also involve multiple non-commeasurable objectives. Evolutionary algorithms are good candidates in solving multi-objective optimization problems [67], and Fonsecan et al [68] classified them into three categorical approaches: aggregating approaches, non-Pareto-based approaches and Pareto-based approaches. Most popular aggregating approaches are the weighted-sum, goal programming, and ε-constrained methods [69]. Most popular non-Pareto-based approaches are vector evaluated genetic algorithm (VEGA) [70], multisexual genetic algorithm (MGA) [71] and weighted min-max approach [72] , while some Pareto-based approaches are niched Pareto genetic algorithm(NPGA) [73], nondominated sorting genetic algorithm (NSGA) [74] and strength Pareto evolutionary algorithm (SPEA) [75].

The generalized form of multi-objective optimization problem can be defined as [76]:

$$\begin{aligned}&\text{Minimize } f_i(x)\ \ i=1,\ldots,N_{obj}\\&\text{Subject to: }\begin{cases}g_j(x)=0 & j=1,\ldots,M\\ h_k(x)\le 0 & k=1,\ldots,K\end{cases}\end{aligned} \tag{3.1}$$

Where f_i refers to the i^{th} objective function, x is the decision variable and N_{obj} is the number of objectives. Several objectives are optimized simultaneously while decision variables should satisfy both the equality and inequality constraints.

A more formal definition of Pareto optimality can be defined as follow [68] :assume $F(x)=(f_1(x),\ldots f_{N_{obj}}(x))$. A decision variable x_u is said to be Pareto-optimal if and only if there is no x_v for which $v=F(x_v)=(v_1,\ldots,v_{N_{obj}})$ dominates $u=F(x_u)=(u_1,\ldots,u_{N_{obj}})$. In other words, there is no x_v such that

$$\forall i\in\{1,\ldots,N_{obj}\}, v_i\le u_i \cap \exists i\in\{1,\ldots,N_{obj}\}, v_i\le u_i \tag{3.2}$$

The set of all Pareto-optimal decision vectors is called the Pareto-optimal set, while the corresponding set of objective vectors is called the non-dominated set.

To solve the optimization problem formulated in this chapter, one of the Pareto-based approaches, the Strength Pareto Evolutionary Algorithm (SPEA) [77, 78] is used to search the space of non-dominated solutions and update them to the elite set at each generation. To solve the model at hand, the modified evolutionary strategy algorithm presented next has been used.

Step 1. Initialize a population P_0 as the current population $P_{current}$ and create an empty external population $P_{external}$ to store elite solutions;

Step 2. Find non-dominated solutions in $P_{current}$ and copy them into $P_{external}$;

Step 3. Find non-dominated solutions in $P_{external}$ and update the elite population $P_{external}$;

Step 4. Cluster solutions, if the size of $P_{external}$ exceeds the limit $\overline{N}$.

Step 5. Assign fitness values to each individual in $P_{current}$ and $P_{external}$;

Step 6. Select N_{parent} individuals into P_{parent} from $P_{current} + P_{external}$ by using the binary tournament selection scheme with replacement;

Step 7. Randomly select two individuals and retain the fitter individual for inclusion in $P_{offspring}$. $P_{offspring}$ has the same population size as $P_{current}$;

Step 8. Apply recombination and mutation operators to $P_{offspring}$; replace $P_{current}$ with $P_{offspring}$ and go back to Step 2 until the stopping criterion (here the maximum number of generations) is met.

To reduce computational effort, clustering takes place in Step 4. Euclidean distance is used as a distance metric between the data points. If the size of the external population is larger than the limit $\overline{N}$, clustering will be performed by setting a distance limit to filter similar points in each cluster while remaining a representative point in each group. For solutions with $Obj2 = 0$ and $Obj3 = 0$, all possible solutions are kept until the final generation by choosing the lowest $Obj1$.

The fitness functions assigned in Step 5 to the individuals in $P_{current}$ and $P_{external}$ differ. For individuals in the elite population $P_{external}$, the fitness function is:

$$F_i = \frac{M_i}{N_{current} + 1} \qquad (3.3)$$

where F_i is the fitness of i^{th} elite individual in the external population $P_{external}$, M_i is the number of individuals in $P_{current}$ that solution i dominates, $N_{current}$ is the population size of $P_{current}$.

For the individuals in the current population $P_{current}$, the fitness function is:

$$F_j = 1 + \sum_{i \in A} F_i \qquad (3.4)$$

where F_j is the fitness of j^{th} individual in the current population $P_{current}$, F_i is the fitness of i^{th} elite which dominates the solution j , A is an elite set including all elite solutions that dominate the solution j .

Individuals with smaller fitness values have a higher probability to reproduce (Step 6). The mutation operation of Step 8 is realized by adding noise $\vartriangle r_i$ to r_i , where r_i is the i^{th} solution variable and $\vartriangle r_i$ is a Gaussian distribution with a mean of zero and standard deviation σ [79]. Solution r_i is updated to r_i ' by $r_i^{'} = [r_i + N(0,\sigma)]$, where $[\bullet]$ is the nearest integer. The value of σ is selected for each variable, and it remains fixed for all generations. The mutated solutions are checked for possible constraint violations. When a constraint is violated, the value of the violating solution is replaced with a corresponding constraint-bound value to make sure all solutions remain in the specified search space.

3.2.2 CO2 Predictive Model

Carbon dioxide concentration in indoor air is commonly used as an indicator of the outside air ventilation rate [80]. CO_2 is a practical and widely used metric for measuring air quality. Though it does not reflect all air containments, a high level of CO_2 concentration points to insufficient ventilation of indoor space. In facilities, such as classrooms with relatively stable occupancy rates during certain time periods, a high concentration of CO_2 can degrade the productivity of students [81, 82]. In this paper, CO_2 is used as the index to optimize ventilation control.

The equilibrium CO_2 concentration in a single facility can be derived based on the number of occupants, the CO_2 generation rate of the occupants, and the supply quantity of the outside air. A diagram of a single facility ventilation system is shown in Figure 3.1.

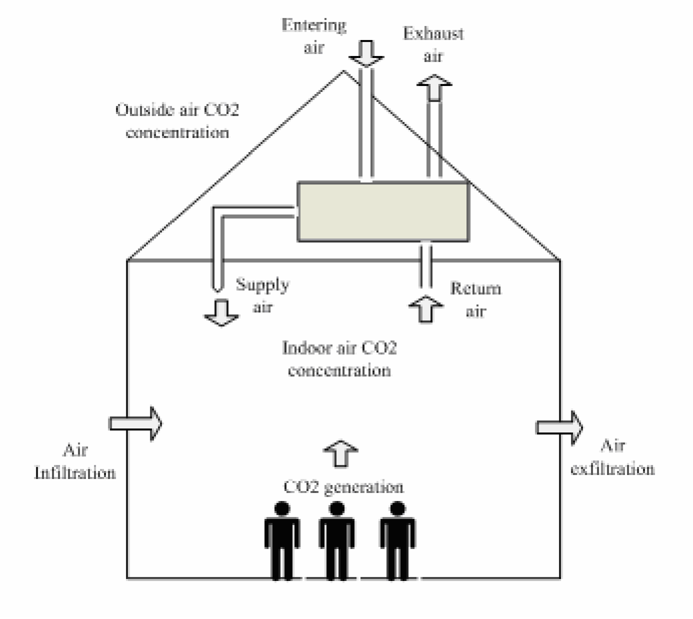

Figure 3.1. Single facility ventilation system

Define the air exchange rate $\lambda_v = Q/V$, where Q is the overall outside air ventilation rate and V is the volume of the facility. The steady-state of indoor CO_2 concentration is obtained from the mass balance in Equation (3.5) [81].

$$\frac{dC}{dt} = \frac{G}{V} + \lambda_v C_{out} - \lambda_v C - v_d \frac{A}{V} C - \frac{Q_{ac}}{V} C \varepsilon_{ac} \qquad (3.5)$$

where: C: the indoor CO_2 concentration,

G: the CO_2 generation rate of occupants,

λ_v: the air exchange rate (defined above),

C_{out}: the outside air CO_2 concentration,

A: the surface area on which indoor pollutants are deposited,

v_d: the deposition velocity of the pollutant,

Q_{ac}: the air flow rate in an air cleaner,

ε_{ac}: the efficiency of an air cleaner.

Equation (3.5) is based on the assumption that the indoor air CO_2 is completely mixed and the air flow rate in and out of the facility, including mechanical ventilation, infiltration, exfiltration, and so on, is balanced.

Note that the unit of C and C_{out} is a fractional concentration (v/v). A conversion factor between the fractional concentration and ppm is 10^6. All concentration-related variables used in this paper have been expressed in ppm. Assuming no pollutant deposition and no air cleaning is taking place, Equation (3.6) is derived.

$$V\frac{dC}{dt} = G + Q(C_{out} - C) \qquad (3.6)$$

Replacing G with the product of the number of people S and the average CO_2 generation rate R , Equation (3.6) can be transformed into an iterative form which is convenient for simulation with a fixed time step [83].

$$C_{next} = (RS/Q)(1 - e^{-Q\Delta T/V}) + C_{last}e^{-Q\Delta T/V} \qquad (3.7)$$

where C_{next} is the difference between the CO_2 concentration in the indoor and the outside air at t_{next} ; C_{last} is the difference between the CO_2 concentration in the indoor and the outside air at t_{last} .

Using the CO_2 predictive model, the optimal ventilation rate can be determined as a trade-off between the energy savings and the air quality.

3.2.3 Optimization Model of Ventilation in a Single Facility

A facility ventilation control can be represented as a fan on-off scheduling model. Unlike demand-control ventilation relying on indoor sensor measurements, the indoor CO_2 concentration can be estimated by applying the CO_2 steady-state Equation (3.7) and the statistically described occupancy pattern drawn from the occupancy information, e.g., meeting or course schedules. In this case, ventilation can be determined without using any sensor feedback signals. Based on simplifying assumptions a scheduling model is formulated.

Denote: t_0, t_N the time stamps when a facility opens and closes for a work day, e.g., 8:00 AM to 6:00 PM. In other words, occupants may only appear in the interval between t_0 and t_N (see Figure 3.2).

$x_i, \vartriangle x_i$ in the time period (t_{i-1}, t_i) represents the fan-on time period.

C_t is the relative CO_2 concentration at time stamp t.

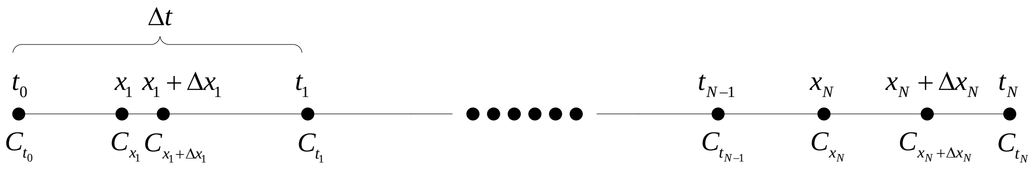

Figure 3.2. On-off schedule of the ventilation fan

Assumption 1: Assume the time horizon is divided into N equal time intervals, $t_0, t_1, ..., t_{N-1}, t_N$. The time interval length is $\Delta t = \frac{t_N - t_0}{N}$. During each time period (t_{i-1}, t_i), define a random variable S_i, $i = 1, ..., N$. $S_i = 0$ represents the number of occupants in this period with a probability density function of $f_i(\cdot)$. For example, if the probability of $S_i = 0$ is 1, it refers to the unoccupied facility during the time period (t_{i-1}, t_i). If the probability of $S_i = \text{constant}$ is high, it implies meetings or courses where the number of occupants is relatively stable over a certain time period. With a more complex probability density function, S_i may describe a more dynamic occupant pattern during a certain time period.

Assumption 2: Assume during time period (t_{i-1}, t_i) the ventilation fan is to be turned on at most once. This assumption is practical if the time interval Δt is relatively short. Note that a user may define Δt as short as needed. Another reason is that frequently turning the ventilation fan on and off may adversely impact the lifetime of the mechanical motor.

Assumption 3: Assume a fan operates in two modes, on and off. Furthermore, assume that the ventilation speed of the fan is constant. The proposed methodology applies to multi-mode ventilation as well as continuous ventilation fans by discretizing continuous values into discrete ones. Thus, the overall ventilation rate Q discussed in [83] is shown in Equation (3.8).

$$Q = IQ \cdot Q_{mech} + Q_{nat} \tag{3.8}$$

Where $IQ = \begin{cases} 0 & t \in [t_{i-1}, x_i) \cup [x_i + \Delta x_i, t_i) \\ 1 & t \in [x_i, x_i + \Delta x_i) \end{cases} \quad i = 1, 2, ..., N$

IQ : denotes the status of the ventilation fan,

Q_{mech} : is the mechanical ventilation rate of the fan, and

Q_{nat} : is the natural ventilation.

In a single facility ventilation system, optimal fan control involves determining the start time of the ventilation fan and the duration of its run. Minimizing the run time of the fan decreases the power consumption of the fan and reduces the AHU heating/cooling load due to the reduced amount of air used while maintaining air quality at the desired level. Quantification of energy savings could be provided once the proposed methodology is fully implemented. Assume a maximum allowable level of an indoor CO_2 concentration of 1600 ppm and 100 occupants using an indoor facility for an hour. Vary the fan start time in five-minute intervals from the first hour to the second hour with the occupants present. The fan run time is fixed at 30 minutes. The simulation result is shown in Figure 3.3.

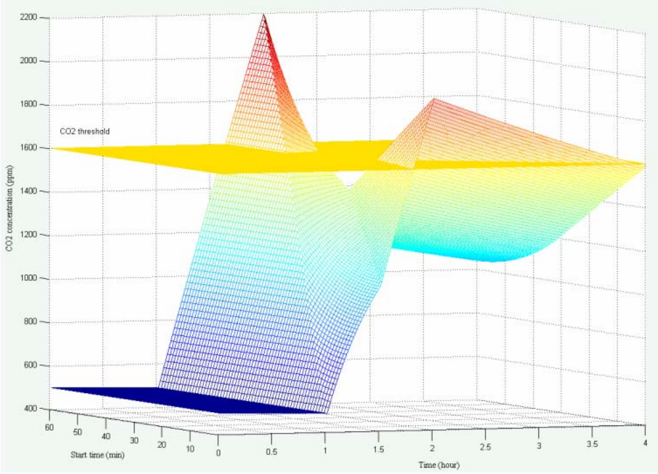

Figure 3.3. The simulated CO_2 curve

As shown in Figure 3.3, different start times of the fan result in various CO_2 curves, some of which reach but others do not, the preset threshold of 1600 ppm. Therefore, an appropriate start time for the fan not only can maintain indoor CO_2 concentration under the specified threshold, but offers a potential for operating the fan for much less time than the original 30-minute-long run time. Thus, formulating an optimization model to determine the optimal schedule is worthy of further research.

For a specific facility and time period (t_{i-1}, t_i) , V is constant. Assume S_i is the number of occupants and the average generation rate is determined in (t_{i-1}, t_i) . Equation (3.7) can be generalized as Equation (3.9).

$$C_{next} = g(Q, t_{next} - t_{last}, C_{last}) \qquad (3.9)$$

where C_{next} is the difference between the CO_2 concentration in the indoor and the outside air at t_{next} , and C_{last} is the difference between the CO_2 concentration in the indoor and the outside air at t_{last} .

Therefore, the CO_2 concentration at any time during the interval (t_0, t_N) can be represented as follows:

$$\begin{cases} C_t = C_{t_0} & t = t_0 \\ C_t = g(Q_{nat}, t - t_{i-1}, C_{t_{i-1}}) & t \in (t_{i-1}, x_i] \\ C_t = g(Q_{nat} + Q_{mech}, t - x_i, g(Q_{nat}, x_i - t_{i-1}, C_{t_{i-1}})) & t \in (x_i, x_i + \vartriangle x_i] \\ C_t = g(Q_{nat}, t - x_i - \vartriangle x_i, g(Q_{nat} + Q_{mech}, \vartriangle x_i, g(Q_{nat}, x_i - t_{i-1}, C_{t_{i-1}}))) & t \in (x_i + \vartriangle x_i, t_i],\ i = 1, 2, ..., N \end{cases} \quad (3.10)$$

As the total run time of all intervals is included in the objective function, the single facility ventilation control can be formulated as model (3.11).

$$\begin{aligned} & \min_{x_i, \vartriangle x_i} \sum_{i=1}^{N} \Delta x_i \\ & s.t. \\ & h(C_t) \le C_{\lim} \\ & t_{i-1} \le x_i \le t_i \\ & t_{i-1} \le x_i + \vartriangle x_i \le t_i \\ & S_i \sim f_i(\cdot) \end{aligned} \qquad (3.11)$$

In this model (3.11), constraint $h(C_t) \le C_{\lim}$ limits the CO_2 concentration. For example, when $h(C_t) = C_t$, the constraint indicates that the CO_2 concentration at any time interval will reach the set limit $C_{\lim}$. In a typical application scenario, the upper CO_2 threshold $C_{\lim}$ may be replaced with a tolerance or a preference function. The more it exceeds a certain value, the higher dissatisfaction value produced. In Section 3, a tri-objective optimization model is presented.

3.3 Computational Study and Results

A single room ventilation model is formulated by evenly dividing working time intervals into N intervals with a specific probability density function of the number of occupants (see Section 3.2.3). Two scenarios, a single-time interval and two-time intervals, are considered. The model is generalized to a ten-interval model discussed towards the end of this section.

3.3.1 Optimization Model for a Single-time Interval

Assume the expected number of occupants to appear at this time interval is known and fixed. This is a reasonable assumption in practice, e.g., for classrooms and conference rooms, because the number of students or conference participants registered can be used as the expected value in the probability density function. As the number of occupants to appear in the interval is defined as a random variable S, consider the constraint function $h(C_t) = E(C_t)$ and Equation (3.7) shown in (3.12).

$$E(C_{next}) = (RE(S)/Q)(1-e^{-Q\Delta T/V}) + E(C_{last})e^{-Q\Delta T/V} \tag{3.12}$$

Let $N_p = E(S)$, where N_p is the number of occupants.

To include different additional situations when indoor CO_2 exceeds the threshold, the CO_2 constraint in optimization model (3.11) can be transformed into two objective functions.

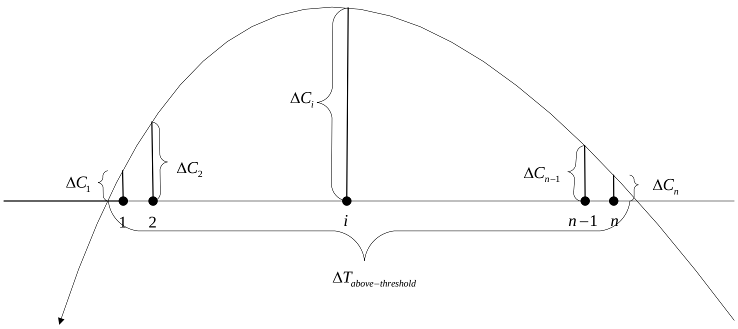

Figure 3.4. CO_2 exceeding a threshold.

Based on Figure 3.4, define the average CO_2 concentration above the threshold. Assume n sampling points of the CO_2 concentration above the threshold; then the second optimization objective is expressed in (3.13). Note the first objective represents the total amount of the fan run time.

$$Obj2 = \text{Average CO}_2\text{ concentration above the threshold} = \frac{1}{n}\sum_{i=1}^{n}\Delta C_i \tag{3.13}$$

where ΔC_i denotes the difference between the i^{th} point CO_2 concentration and CO_2 threshold.

The third objective is expressed in (3.14)

$$Obj3 = \text{Elapsed time} = \Delta T_{above-threshold} \qquad (3.14)$$

where $\Delta T_{above-threshold}$, shown in Figure 3.4, is the total time the indoor CO_2 concentration is above the threshold.

The three objectives, $Obj1$, $Obj2$, and $Obj3$, are integrated into the optimization model for time interval (t_0, t_1) shown in (3.15):

$$\begin{aligned} &\min_{x_1, \triangle x_1} \{Obj1, Obj2, Obj3\} \\ &s.t. \\ &t_0 \le x_1 \le x_1 + \triangle x_1 \le t_1 \end{aligned} \qquad (3.15)$$

where x_1, $\triangle x_1$ are the start time and running time of the fan (both integers):

$Obj1 = \triangle x_1$,

$Obj2 = \frac{1}{n}\sum_{i=1}^{n} \Delta C_i$

$Obj3 = \Delta T_{above-threshold}$

For $Obj2 = 0$ and $Obj3 = 0$, model (3.15) ensures that the indoor CO_2 concentration is below the threshold value at any time. Transforming the constrained model into a non-constrained one, a bi-objective optimization problem can be built by constructing objective functions $Obj1 = \triangle x_1$ and $Obj2 = \max(0, \frac{1}{n}\sum_{i=1}^{n} \Delta C_i) + \max(0, \Delta T_{above-threshold})$. In this paper, a more general tri-objective optimization model is built to include all these situations. In solving this model, a Pareto-based evolutionary strategy is proposed for finding an elite set which includes non-dominated solutions. To find solutions for CO_2 strictly under the CO_2 threshold, simply choose solutions in this elite set with both $Obj2 = 0$ and $Obj3 = 0$.

2.3.2 Model Solving by the Evolutionary Strategy Algorithm

To solve model (3.15), some parameters need to be initialized. The parameters used in this research are listed in Table 3.1. The unit associated with the solution variables, the start time and the run time of the fan (shown in Tables 3.2, 3.3 and 3.5), is ten seconds.

Table 3.1 Parameters' descriptions

Variable	Value	Description	Unit
V	2000	Facility volume	m^3
N_p	100	Number of occupants	person
Q_{nat}	240	Natural ventilation rate	m^3/hr
Q_{mech}	3000	Mechanical ventilation rate	m^3/hr
C_{out}	400	Outside CO2 concentration	ppm
C_{t_0}	500	Initial indoor CO2 concentration	ppm
$C_{threshold}$	1500	CO2 threshold	ppm
R	0.01	Average CO2 generation rate	$\mathrm{l/s}$

To solve the model at hand, the Strength Pareto Evolutionary Algorithm (SPEA) has been used. In the one-time interval scenario, solution variables r_1, r_2 are $x_1, \triangle x_1$ respectively. Set $\sigma = 2$. The maximum number of generations (the stopping criterion) is set as 30, as there is no significant difference in the solution quality when the number of generations exceeds 30. The values of the parameters used in the evolutionary strategy algorithm, such as the ratio of the parent and offspring size and the initial population size, are set as follows. The evaluation criterion is based on solutions with $Obj2 = 0$ and $Obj3 = 0$ in the final elite set.

For each of the four runs of the SPEA algorithm, the initial population size varies from 30 to 390. Figure 3.5 demonstrates $Obj1$ of the solution in the final elite set, with $Obj2 = 0$ and $Obj3 = 0$, for different values of the initial population size. Figure 3.6 shows the average values of $Obj1$ based on four runs of the algorithm.

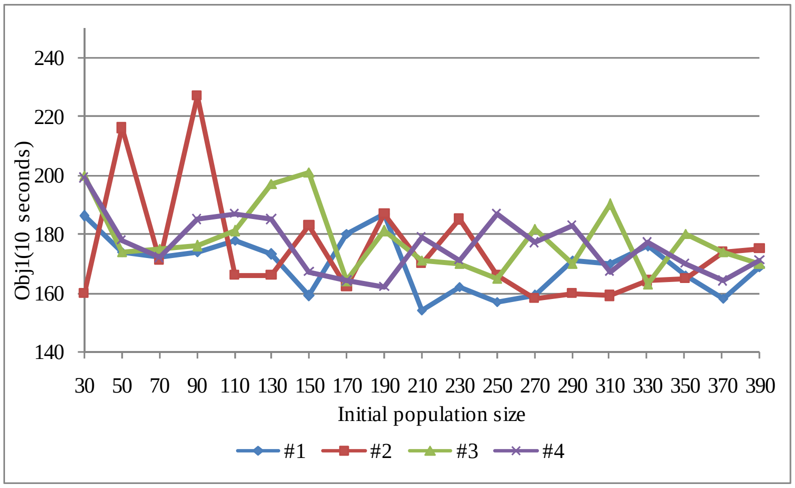

Figure 3.5 Values of objective function $Obj1$ for different initial population size based on four runs of the algorithm.

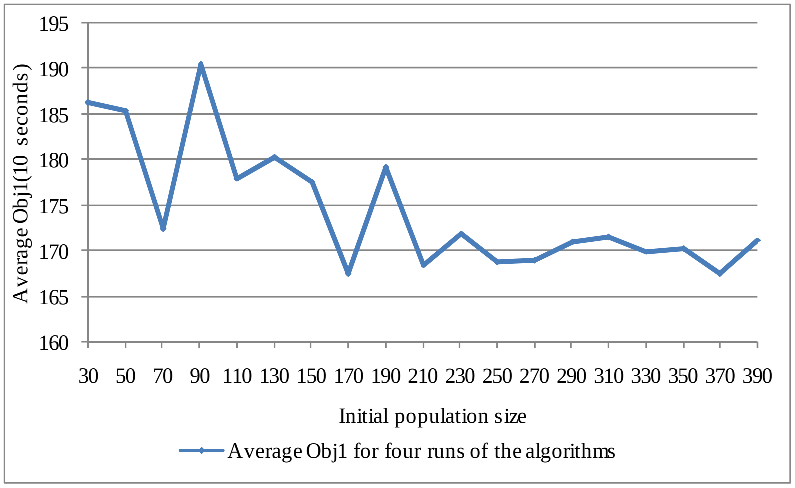

Figure 3.6 Average $Obj1$ for four runs of the algorithm for different initial population sizes.

As illustrated in Figure 3.6, the initial population size of 250 has a relatively low average value and therefore is selected as the initial size. Figure 3.7 demonstrates the change of $Obj1$ for different parent and offspring ratios.

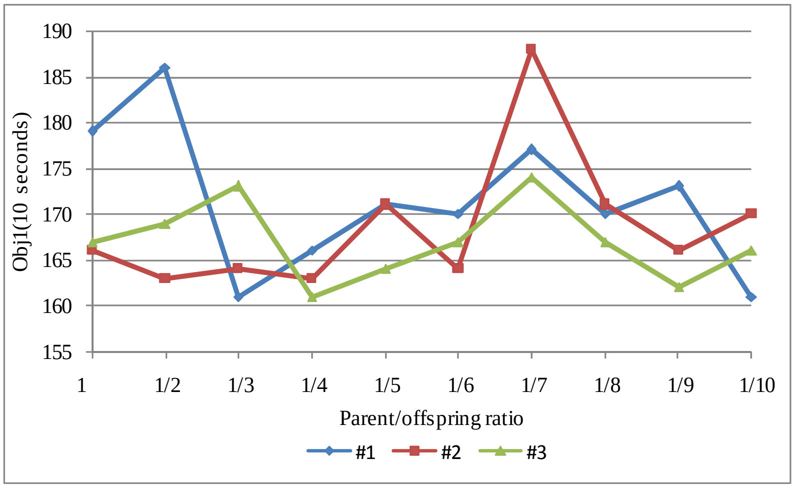

Figure 3.7 Values of the objective function $Obj1$ for different parent offspring ratios for four runs of the algorithm.

The parent offspring ratio of 1/4 is selected, as it has produced the best quality results (the lowest value *Obj*1, as shown in Figure 3.7). The algorithm is run using the tuned parameters above. Distributions of the offspring and the elite objective values at different generations are shown in Figures 3.8 and 3.9.

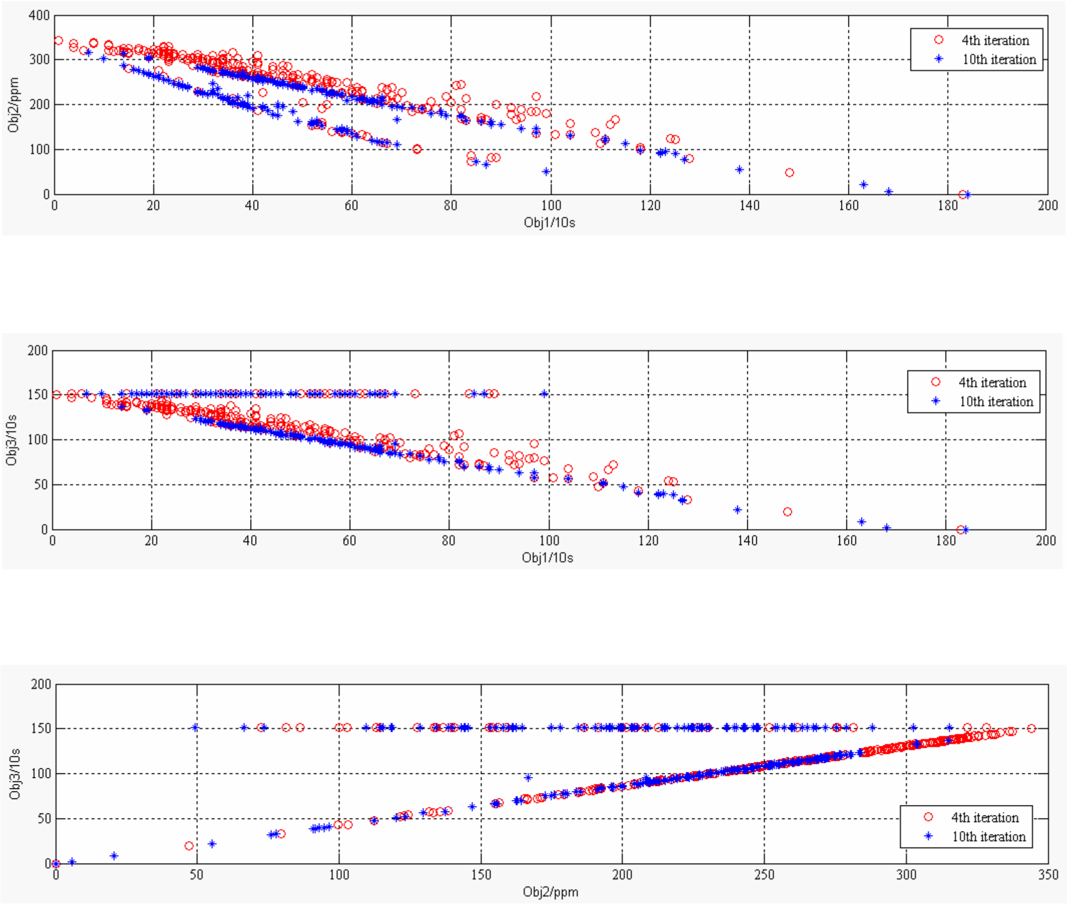

Figure 3.8 Distributions of the offspring in two dimensional space of objective values at different iterations.

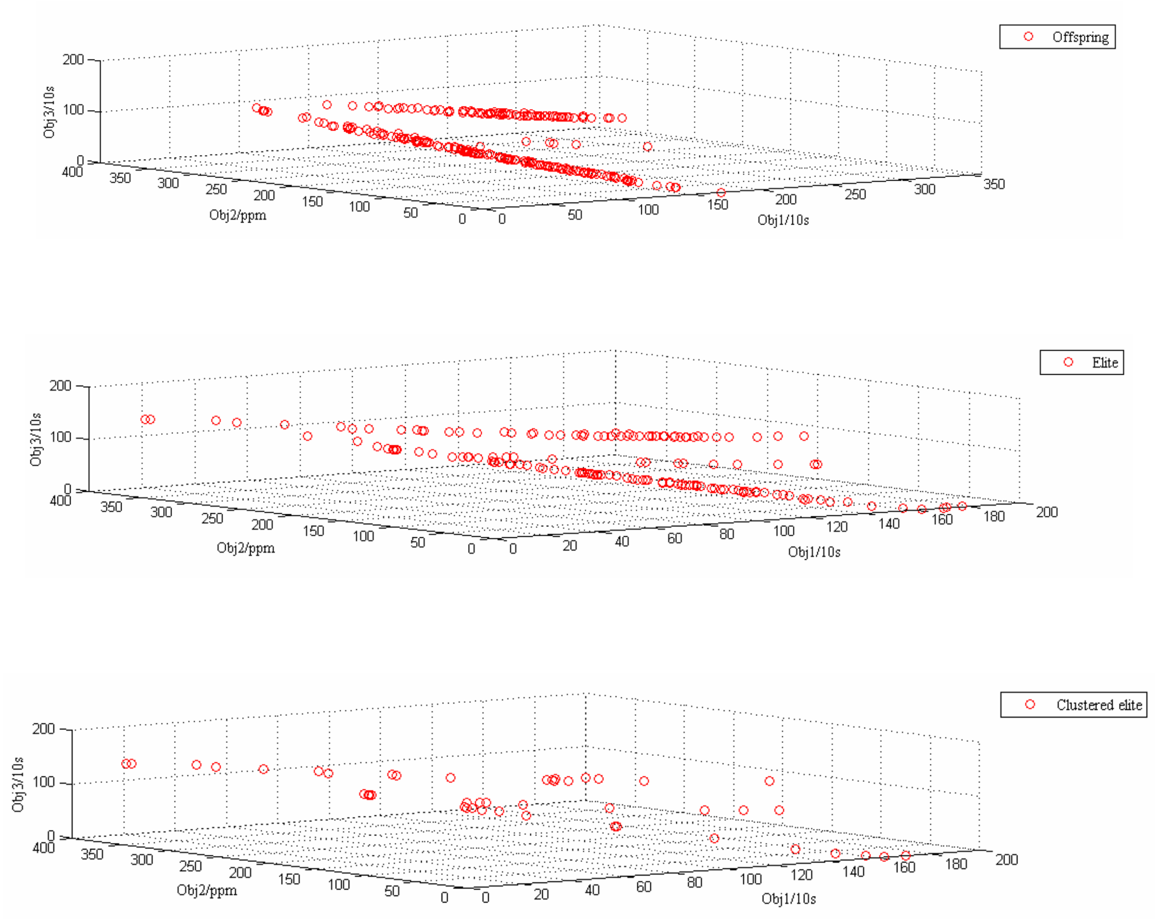

Figure 3.9 Distributions of the offspring and the elite objective values before and after clustering at 30th generation.

As shown in Figure 3.8, with the increase of the number of iterations, the points in the offspring set move towards the origin point (0, 0) in the two dimensional objective space. It indicates that the distribution front of the offspring moves towards the origin point (0, 0, 0) in the three dimensional space as the number of iteration increases. Figure 3.9 outlines the shape of the Pareto front in the space of objective values It also demonstrates the difference in the elite set before and after the clustering. Each point in the elite set cannot dominate another. Points with a small Euclidean distance are removed, while one remains to represent that cluster for computational cost consideration. Here the elite size limit $\overline{N}$ is set as 10. The Euclidean distance limit is set to 50, which means that if a point's distance is less than 50, it will be considered for clustering. When considering the diversity of solutions with both $Obj2 = 0, Obj3 = 0$, points on the $Obj1$ axis are not clustered. At the last iteration, the smallest $Obj1$ value is kept, while the larger ones are filtered out.

2.3.3 Optimal Solution Selection

The selection of the optimal solution from the elite set depends on the importance of each objective. Assigning weights to each objective and transforming them into a single objective is a commonly used approach. The final solution corresponds to the minimum value of Obj in (3.16).

$$Obj = w_1 \frac{Obj1 - Obj1_{\min}}{Obj1_{\max} - Obj1_{\min}} Obj1 + w_2 \frac{Obj2 - Obj2_{\min}}{Obj2_{\max} - Obj2_{\min}} Obj2 + w_3 \frac{Obj3 - Obj3_{\min}}{Obj3_{\max} - Obj3_{\min}} Obj3 \tag{3.16}$$

where w_1, w_2, w_3 are the user-defined weights indicating the importance of each objective and $Obj1_{\max}$ and $Obj1_{\min}$ are the maximum and the minimum values of $Obj1$ in the final elite set. Similar notation is used for $Obj2_{\max}, Obj2_{\min}, Obj3_{\max}$, and $Obj3_{\min}$.

Note that $\sum_{m=1}^{3} w_m = 1$, with w_1, w_2, w_3 being either constants or functions of other objectives. For example, if the indoor CO_2 concentration is required to be below a certain CO_2 threshold, w_2 and w_3 are assigned relatively large values compared to w_1. If the value of $Obj2$ is the range $[0, a]$, w_2 is constructed as follows

$$w_2 = \begin{cases} 0 & Obj2 \in [0,\ a] \\ \text{large compared to } w_1 & Obj2 \in (a, +\infty) \end{cases}$$

Figure 3.10 shows five cases of the change in the indoor CO_2 concentration. The ventilation schedule varies among the five cases. In all cases, the occupants arrive at 8:00 AM and leave the facility at 9:00 AM.

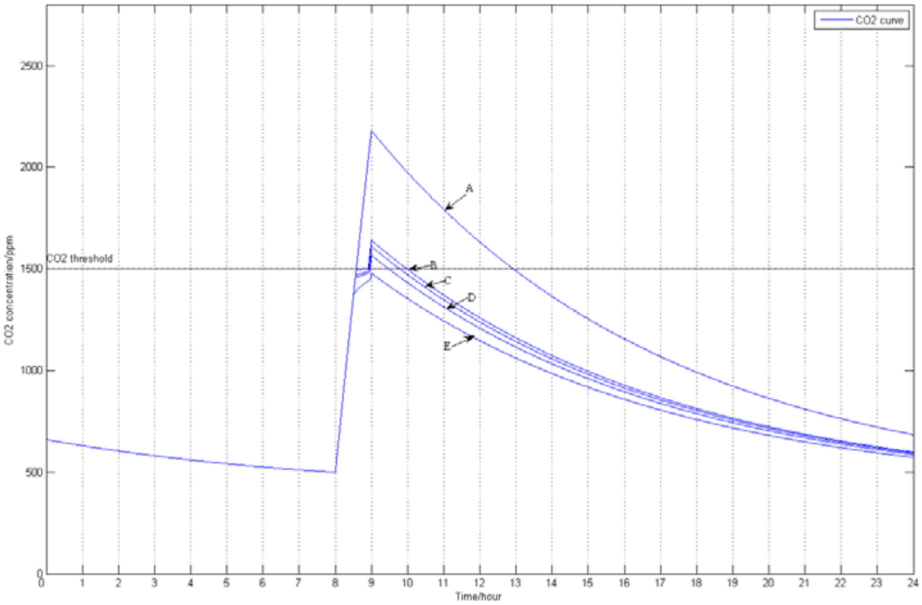

Figure 3.10 Five cases of the indoor CO_2 concentration.

The details of each of the five cases are illustrated in Figure 3.10. The weights, optimal solutions, and the experiment description are provided in Table 3.2.

Table 3.2 Description of the weight assignment and the results

Curve	w_1	w_2	w_3	x_1	$\vartriangle x_1$	Description
A				0	0	Fan is not on
B	1, $w_2 = w_3 = 0$ 0.002, $w_2 = w_3 = 0.499$	0, $Obj2 \in [0,100]$ 0.499, $Obj2 \in (100,+\infty)$	0, $Obj3 \in [0,20]$ 0.499, $Obj3 \in (20,+\infty)$	202	124	$Obj2$ admitted in bound [0,100] $Obj3$ admitted in bound [0,20]
C	0.5	0.4	0.1	208	128	Different weights
D	1, $w_2 = w_3 = 0$ 0.002, $w_2 = w_3 = 0.499$	0, $Obj2 \in [0,50]$ 0.499, $Obj2 \in (50,+\infty)$	0, $Obj3 \in [0,10]$ 0.499, $Obj3 \in (10,+\infty)$	199	142	$Obj2$ admitted in bound [0,50] $Obj3$ admitted in bound [0,10]
E	0.002	0.499	0.499	182	171	$Obj2 = 0, Obj3 = 0$

As presented in Table 3.2, by assigning different weights to the three objectives, the fan-on time period varies. Maintaining the CO_2 concentration strictly below the CO_2 threshold, e.g., 1500 ppm, implies higher fan energy consumption, while allowing the CO_2 concentration to remain at a certain threshold interval can reduce the run time of the fan, thus resulting in higher energy savings.

2.3.4 Statistical Analysis

In the one-time-interval optimization model, the expectation of the number of occupants is used. Other assumptions based on different probability functions and probability of the number of occupants can also be made. For example, assume the occupancy distribution function is the Poisson distribution [46].

$$f(S_1) = \frac{\lambda^{S_1} e^{-\lambda}}{S_1 !} \qquad S_1 = 0, 1, 2, \ldots \qquad (3.17)$$

If $P(S_1 \le 100) = 0.95$, λ is 85. Consider the optimal solution chosen in section 3.3 to keep the CO_2 strictly under threshold. Because Equation (3.7) is a non-decreasing function, fewer people than 100 can always satisfy the constraints. The maximum number of people is 101. Therefore, statistical confidence of this optimal decision making with $Obj2 = 0, Obj3 = 0$ can be computed as $P(S_1 \le 101) = 0.96$.

The probability of the number of people less than 102 is 0.96. If the number of people less than that particular number is true, the indoor air CO_2 concentration will be strictly under 1500 ppm, using the proposed optimal solution of $x_1 = 1820s, \Delta x_1 = 1710s$. Define $f_{CO2\_Max}(z)$ as the maximum CO_2 concentration for $S_1 = z$. Then $P(f_{CO2\_Max}(z) <= 1500) = 0.96$. If the CO_2 threshold is lowered, the probability will decrease. Assume the acceptance probability is 0.95 and introduce the variation of the CO_2 threshold.

$$P(f_{CO2\_Max}(z) <= 1500 - \Delta CO2_{threshold}) >= 0.95 \qquad (3.18)$$

From that calculation, $\Delta CO2_{threshold} <= 19\,ppm$. In other words, statistical confidence of this optimal decision making can also hold if the variation of the CO_2 threshold is less than a limited value. The distribution of the occupants can be any other probability density function, e.g., exponential family or probability density functions learned from the occupancy data.

3.3.5 Optimal Model for Two-time Intervals

Like the one-time interval, the optimization model of two consecutive time intervals can be formulated as follows:

$$\begin{gathered} \min_{x_1, \Delta x_1, x_2, \Delta x_2} \{Obj1, Obj2, Obj3\} \\ s.t. \\ t_0 \le x_1 \le x_1 + \Delta x_1 \le t_1 \\ t_1 \le x_2 \le x_2 + \Delta x_2 \le t_2 \end{gathered} \tag{3.19}$$

where: $x_1, \Delta x_1$ are the start time and run time of the fan during the first interval $[t_0, t_1]$;

$x_2, \Delta x_2$ are the start time and run time of the fan during the second interval $[t_1, t_2]$;

$Obj1 = \Delta x_1 + \Delta x_2$, $Obj2, Obj3$ are defined the same as in the one-time interval.

Searching four solution variables by applying the evolutionary algorithm mentioned above, distribution of points in the offspring and the elite and clustered elite set at the last iteration is shown in Figure 3.11. The maximum iteration here is 40.

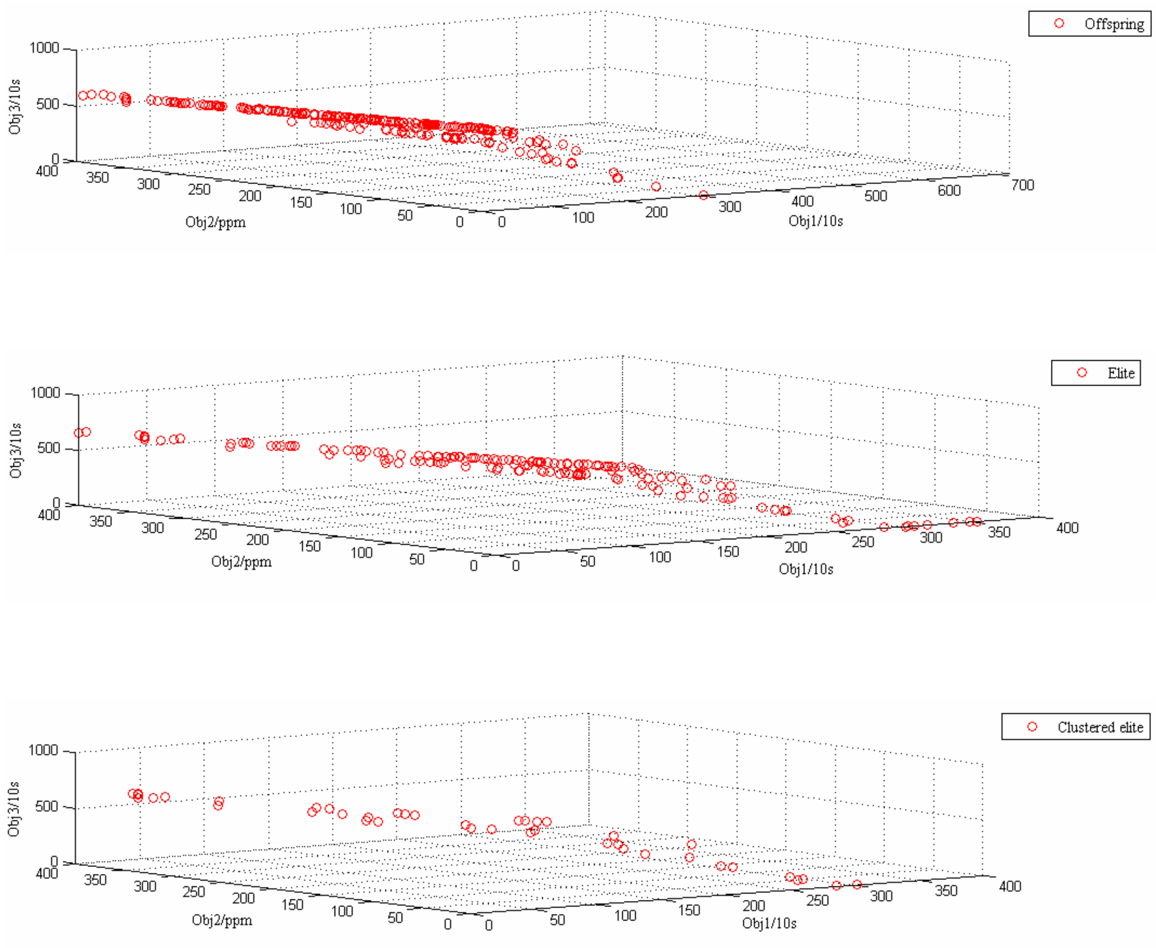

Figure 3.11 Distributions of the offspring and elite objective values before and after clustering.

The optimal solutions selecting criteria is the same by assigning weights to each objective. Figure 3.12 demonstrates the change of indoor CO_2 concentrations in five cases. People appear from 8:00 AM to 10:00 AM. The number of people is 100 and 30, respectively, in each one-hour interval.

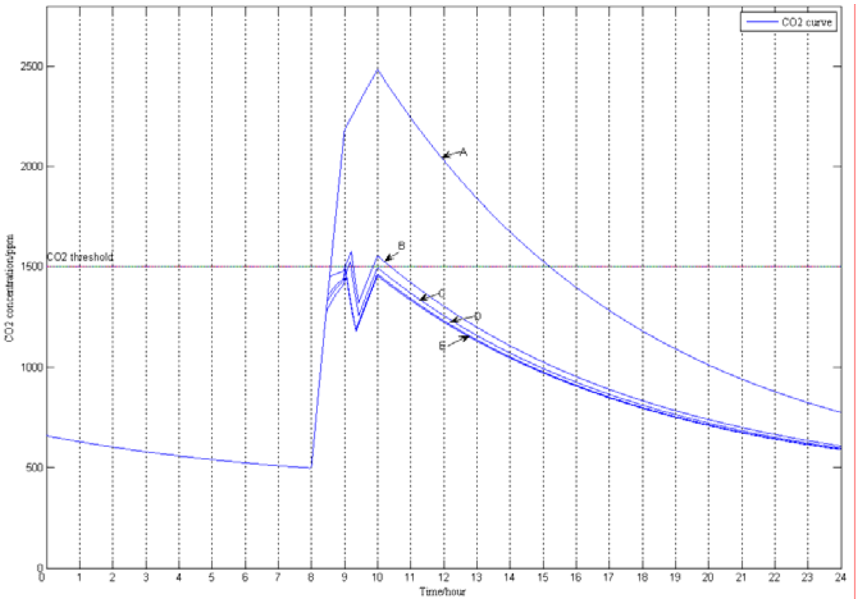

Figure 3.12 Change of indoor CO_2 concentration in five cases

The weight assignment and solutions to model (3.19) are shown in Table 3.3.

Table 3.3 The weight assignment and the results

Curve	A	B	C	D	E
w_1		1, $w_2 = w_3 = 0$ 0.002, $w_2 = w_3 = 0.499$	0.5	1, $w_2 = w_3 = 0$ 0.002, $w_2 = w_3 = 0.499$	0.002
w_2		0, $Obj2 \in [0,100]$ 0.499, $Obj2 \in (100,+\infty)$	0.4	0, $Obj2 \in [0,50]$ 0.499, $Obj2 \in (50,+\infty)$	0.499
w_3		0, $Obj3 \in [0,100]$ 0.499, $Obj3 \in (100,+\infty)$	0.1	0, $Obj3 \in [0,50]$ 0.499, $Obj3 \in (50,+\infty)$	0.499
x_1	0	198	178	172	160
$\triangle x_1$	0	159	175	183	197
x_2	0	75	14	65	22
$\triangle x_2$	0	80	112	91	105
Total run time	0	239	287	274	302
Description	Fan is not on	$Obj2$ admitted in range [0,100] $Obj3$ admitted in range [0,100]	Different weights	$Obj2$ admitted in range [0,50] $Obj3$ admitted in range [0,50]	$Obj2 = 0, Obj3 = 0$

3.3.6 The Scenario With Ten-time Intervals

For N consecutive intervals, the number of solution variables is $2n$ and $Obj1 = \sum_{i=1}^{N} \triangle x_i$. For example, assume the working hours are from 8:00 AM to 6:00 PM. The occupancy schedule is established on an hourly basis (see Table 3.4). Therefore, there are ten intervals and $N = 10$. Some intervals may not include occupants, e.g., the lunch time period, 12:00 AM to 1:00 PM.

Table 3.4 Occupancy schedules for ten-time periods.

	8:00 AM - 9:00 AM	9:00 AM - 10:00 AM	10:00 AM - 11:00 AM	11:00 AM - 12:00 AM	12:00 AM - 1:00 PM	1:00 PM - 2:00 PM	2:00 PM - 3:00 PM	3:00 PM - 4:00 PM	4:00 PM - 5:00 PM	5:00 PM - 6:00 PM
1	50	10	100	0	0	40	0	30	0	10
2	30	20	35	15	10	43	0	30	0	27
3	10	50	45	9	0	17	40	30	14	5
4	13	25	14	9	5	17	16	21	14	35

Optimal solutions for the scenario with $Obj2 = 0, Obj3 = 0$ are to be determined. To reduce the solution search space, the fan is turned off during intervals in which the number of occupants is 0. This is feasible because the CO$_2$ concentration can never exceed the threshold in these intervals provided that the CO$_2$ concentration at the start point of such intervals is below the threshold. For a population size to 2000, the number of generations at 30, and an elite population size of 100, the optimal solutions for each of the four occupancy schedules of Table 3.4 are shown in Table 3.5. The units of values below are listed in 10s.

Table 3.5 Computed ventilation schedules.

	$x_1, \triangle x_1$	$x_2, \triangle x_2$	$x_3, \triangle x_3$	$x_4, \triangle x_4$	$x_5, \triangle x_5$	$x_6, \triangle x_6$	$x_7, \triangle x_7$	$x_8, \triangle x_8$	$x_9, \triangle x_9$	$x_{10}, \triangle x_{10}$
1	131,41	111, 119	64, 291	0,0	0,0	67,103	0,0	93,114	0,0	214,83
2	218,19	27,53	48,96	161,36	228,74	73,117	0,0	78,121	0,0	196,42
3	258,18	194,119	118,129	40,77	0,0	194,101	207,85	80,83	128,16	146,54
4	32,9	281,10	149,33	17,49	226,47	245,66	5,4	56,6	11,195	261,98

Note that search is performed in parallel so that the entire schedule is considered at the same time. Solutions can be also determined based on the time interval by a time interval basis. The optimal solution for the first interval is determined first, and then a solution for the second interval is determined based on the previous one. In this paper, the parallel approach is followed. The CO_2 charts for the four schedules are shown in Figure 3.13.

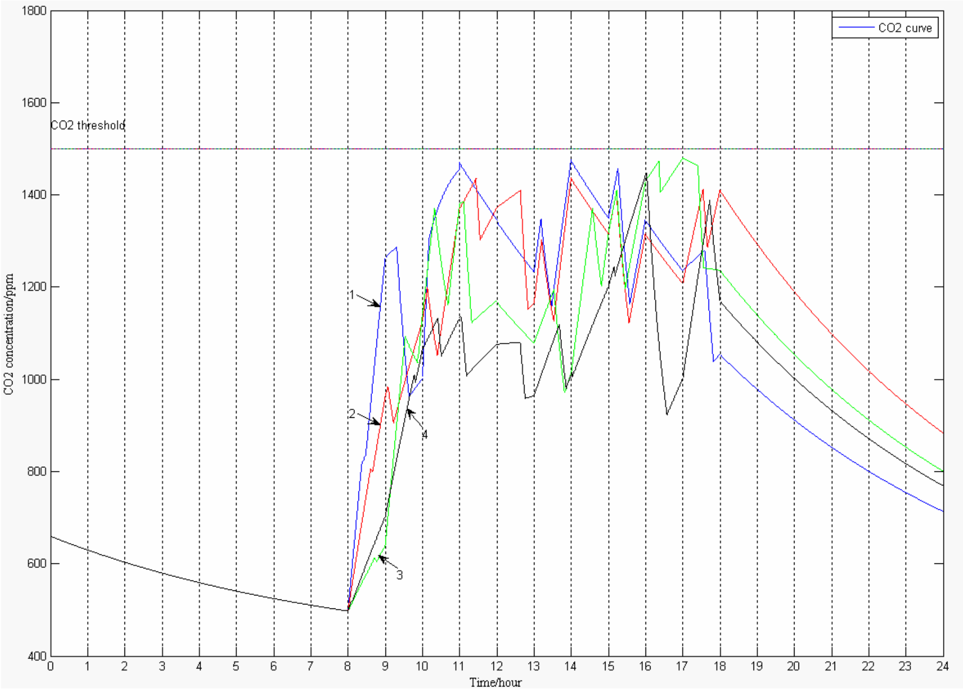

Figure 3.13 CO_2 concentration for different optimized schedules.

3.4 Summary

In this chapter, a scheduling model was developed for a ventilation control problem. Optimal ventilation was determined for different occupancy schedules. The fan start and run times were determined to maintain the CO_2 level below a pre-selected threshold. The optimized run time of the mechanical ventilation system reduced power consumption of the fan and the heating and cooling costs. These savings were accomplished while the indoor air quality was guaranteed without installing any analog indoor sensors. Future research will focus on modeling cases where the fan operates in more than two modes. The CO_2 model needs to be improved by considering the quasi-steady state and CO_2 spatial distribution.

CHAPTER 4.

COOLING OUTPUT OPTIMIZATION OF AN AIR HANDLING UNIT

4.1 Introduction

In modern buildings, heating, ventilating and air conditioning systems consumes a large part of total building energy consumption to maintain suitable thermal comfort and acceptable air quality for occupants. Therefore, HVAC systems play a significant role in the energy savings equation. Since it is a complex, nonlinear, discrete system involving numerous constrains, it poses big challenge for system modeling and optimization.

Numerous studies on optimization of HVAC systems and energy efficiency have been published. Fong *et al.* [23] discussed reducing energy consumption by optimizing set points of the chilled water supply temperature and supply air temperature in response to the dynamic cooling load and changing weather. Mathews *et al.* [35] applied simulation to evaluate the impact of control strategies on energy savings while ensuring sufficient indoor comfort. Huang *et al.* [1] introduced new functions for an energy management control system based on the dynamic model of a VAV-HVAC system and demonstrated energy reduction by using simulation. Nassif *et al.* [84] developed a simple control strategy for minimum energy use which is suitable for online implementation. Using mathematical models of the cooling load and energy consuming devices, Lu *et al.* [85] proposed a systematic approach to the optimization of the overall system energy consumption. Ari *et al.* [86] presented an optimization approach to minimize energy consumption of a typical building with a constraint on the individual dissatisfaction generated. Based on the thermodynamic modeling of the air-conditioning system, Kintner-Meyer *et al.* [87] investigated an optimal protocol for indoor temperature and humidity control, as well as operating point settings for the chiller to achieve energy savings.

Unlike many studies centered on mathematical models [88] and simulation approaches [89], this chapter presents a computational intelligence approach as a viable alternative for building predictive models. Such models are solved with evolutionary computation algorithms. In this chapter, ANN is applied to extract dynamic models based on the data collected from HVAC systems. The identified models are validated and tested with independent HVAC data sets.

4.2 Methodology for Cooling Output Optimization of an Air Handling Unit

The energy used by a chiller accounts for a large portion of the total energy consumption of any HVAC system. Designed to meet the peak load of a building, the chiller operates at its full capacity only a limited time. Most of time, it operates at a part load, and its power input changes in response to the cooling load.

Define the chiller part load ratio PLR [90] as shown in Equation (4.1).

$$PLR = Q_{chl} / Q_{chr} \qquad (4.1)$$

where Q_{chl} is the part cooling load and Q_{chr} is the full cooling capacity, a design parameter specified for a certain type of chiller. The relationship between the PLR and the power input of the chiller [91] is expressed in Equation (4.2).

$$kW = a + bPLR + cPLR^2 + dPLR^3 \qquad (4.2)$$

where a, b, c, d are coefficients determined by the chiller design. Typically, as the chiller's load ratio increases from, for example, 60% to 100%, the power input increases [92]. Therefore, by optimizing the PLR, or more specifically Q_{chl}, as shown in (4.3), energy is saved due to the reduced power input.

$$Q_{chl} = C_{pw} m_w \left(T_{chwr} - T_{chws} \right) \qquad (4.3)$$

where: Q_{chl} is the cooling load,

C_{pw} is the specific heat capacity of chilled water,

m_w is the mass flow rate of the chilled water,

T_{chwr} and T_{chws} is the chiller water supply and return temperature, respectively [93].

In the AHU side, the cooling output of the cooling coil accounts for the major load of a chiller. As the chilled water flows through the pipes to the cooling coil in the air handling unit (AHU) and returns to the chiller for cooling after absorbing heat, both the measured values of the cooling coil entering water temperature $T_{chwc-ewt}$ and the cooling coil exiting water temperature $T_{chwc-lwt}$ are selected as approximations of the T_{chws}, T_{chwr}. Therefore, the load Q_{chl} is represented by cooling output Q_{chl}' using the approximations of T_{chwr} and T_{chws} shown in (4.4).

$$Q_{chl}' = C_{pw} m_w \left(T_{chwc-lwt} - T_{chwc-ewt}\right) \qquad (4.4)$$

To minimize Q_{chl}', the function $y_1(t) = f_1(\bullet)$ should be established between the output Q_{chl}' and the input variables of the AHU. The input variables fall into three categories: the previous status of Q_{chl}', controllable variables, and uncontrollable variables. The function $y_1(t)$ [94] is shown in (4.5).

$$y_1(t) = f_1([y_1(t-d)]_{d \in D_y}, [\boldsymbol{x}(t-d)]_{d \in D_x}, [\boldsymbol{v}(t-d)]_{d \in D_v}) \qquad (4.5)$$

where: $y_1(t-d) \in \boldsymbol{R}$ includes previous states of $y_1(t)$, e.g., $y_1(t-1)$ and $y_1(t-2)$;

$\boldsymbol{x} \in \boldsymbol{R}^k$ is a vector of k controllable variables with time variables, e.g., x_1 is the cooling coil valve position variable and $x_1(t-1)$ refers to its previous state at $t-1$ time stamp;

$\boldsymbol{v} \in \boldsymbol{R}^m$ is a vector of m uncontrollable variables with time variables, e.g., v_1 is the outside air temperature variable and $v_1(t-2)$ refers to its previous state at $t-2$ time stamp;

D_y, D_x, D_v are sets containing time variables of the corresponding variables; e.g., $D_{x_1} = \{0,1\}$, which means there are two values for variables x_1, and they are $x_1(t)$, $x_1(t-1)$, respectively.

Let $y_1 = f_1(\bullet)$ be an objective function to be optimized by global search algorithms determining optimal controllable variables in Section 3.4 and 3.5. Models of

$y_2 = f_2(\bullet)$ and $y_3 = f_3(\bullet)$, where y_2 is the supply air temperature and y_3 is the supply air humidity, should be built using the same controllable and uncontrollable inputs, since different values of both controllable and uncontrollable variables not only influence the cooling output, but also affect the supply air quality. The optimization model considered in this research is presented in (4.6).

$$\begin{array}{l}
\mathit{min}\ y_1 \\
\mathit{subject\ to}: \\
y_1(t) = f_1([y_1(t-d)]_{d \in D_y}, [\mathbf{x}(t-d)]_{d \in D_x}, [\mathbf{v}(t-d)]_{d \in D_v}) \\
y_2(t) = f_2([y_2(t-d)]_{d \in D_y}, [\mathbf{x}(t-d)]_{d \in D_x}, [\mathbf{v}(t-d)]_{d \in D_v}) \\
y_3(t) = f_3([y_3(t-d)]_{d \in D_y}, [\mathbf{x}(t-d)]_{d \in D_x}, [\mathbf{v}(t-d)]_{d \in D_v}) \\
x_i \in S_{x_i} \quad i\ \mathit{is\ the\ number\ of\ controllable\ vectors} \\
\mathrm{y}_j \in S_{y_j} \quad j = 2{,}3
\end{array} \tag{4.6}$$

A modified evolutionary strategy algorithm is used to search for optimal control settings. The AHU supply air temperature and humidity are treated as constraints, with their values changing in a certain range so that the air thermal properties are not compromised as the cooling output is minimized to reduce the energy consumption.

4.3 Case Study and Computational Results

The approach proposed in this paper has been tested on the data collected at the Energy Resource Station (ERS) in Ankeny, Iowa. The ERS is designed for testing and demonstrating commercial HVAC systems. The data includes more than 300 variables captured at the AHU, heating and chilling plants, and different testing zones. The sampling time interval is one minute, and data points collected are the last values recorded at one-minute time intervals, rather than the average data used in other applications.

4.3.1 Variable Selection and Data Dimensionality Reduction

In this paper, a data set collected at the AHU over a two-day period was used for model development, testing, and analysis. The data set includes over 90 variables, including the data collected at coils, dampers, fans, ducts, and pipes. A schematic diagram of the AHU is shown in Figure 4.1.

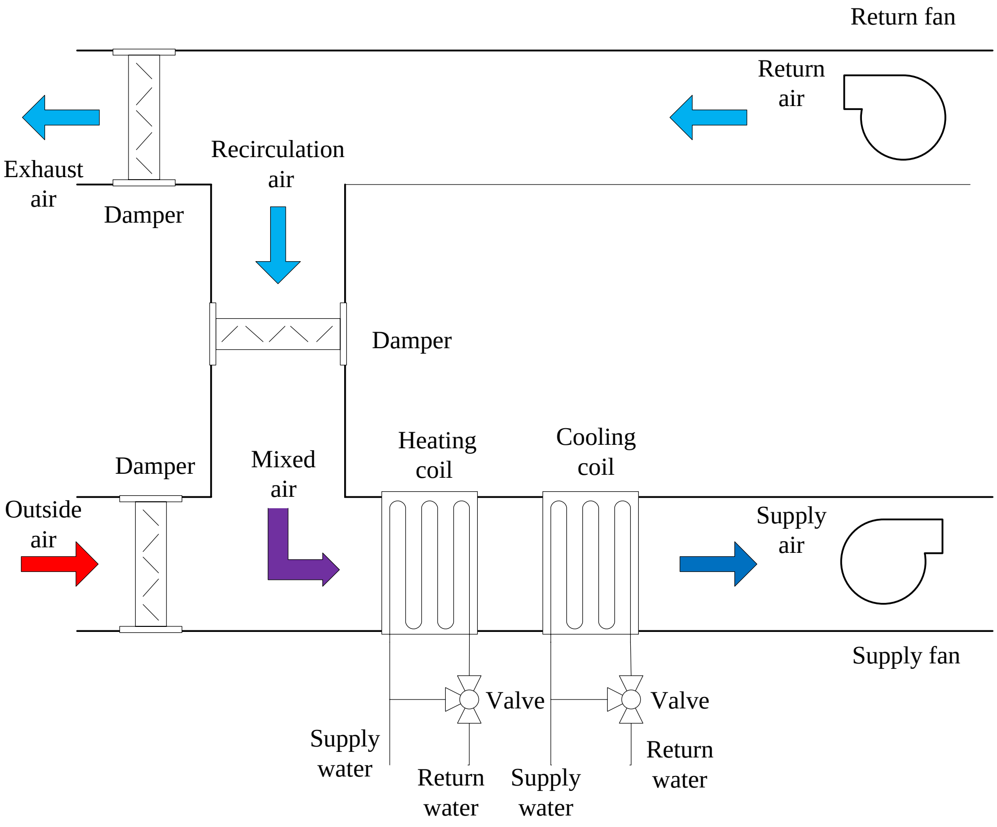

Figure 4.1 Schematic diagram of the AHU

Variable selection is performed to eliminate variables of relatively less importance in this two-day data set, since it can improve the comprehensibility, scalability, and possibly, accuracy of the resulting models [95]. For example, relative open damper values are important in the AHU, and they directly determine the amount of air volume in or out. But in this scenario, these variables are removed in the variable selection process because their values remain almost the same in the two-day data set. This is also true for the heating coil because, in this case, the heating water coil valve is 100% closed in the two-day data set and no heating process is performed. Many methods, like wrappers [96], can be used to reduce the dimensionality of the variable space. However, the wrapper-type approaches can be computationally expensive, as a specific search algorithm searches the space of all possible variables and evaluates each subset of variables after building a model based on this subset. In this paper, a boosting tree algorithm [97, 98] was applied to reduce the number of variables at a relatively low computational effort. Boosting tree methods have been applied in many areas [99]. They generate standard measures of variable importance that can be used to develop optimal feature sets. The boosting tree algorithm is robust and effective in eliminating variables (input variables) that are not relevant.

Four significant variables have been selected as inputs to the model of cooling output, supply air temperature, and supply air humidity. Two of them are controllable variables, i.e., the cooling coil chilled water valve position and the relative speed of the supply air fan. The other two are uncontrollable variables, the outside air temperature and the chilled water supply temperature. The outside air temperature measured at the inlet duct is an independent variable. The chilled water supply temperature determined by the control settings of the chillers is relatively independent of the controllable variables in AHU.

The time delay should also be considered in modeling the AHU. For instance, past values of some variables may have more impact on the model accuracy of the HVAC system than their current values. The boosting tree algorithm was used to select the most significant time delays (periods) at which the values of the variables should be considered. Table 4.1 lists the variables selected to model the cooling output, supply air temperature, and supply air humidity at time stamp t. Two of them, namely $x_1(t)$ and $x_2(t)$, are current time controllable variables, which will be optimized in the next section.

Table 4.1 Variables selected for building model (4.6) at t time stamp

Variable	Point Name	Description	Unit
$x_1(t)$	CHWC-VLV	Chilled water coil valve position at time t	%Open
$x_1(t-4)$	CHWC-VLV	Chilled water coil valve position at time $t-4$	%Open
$x_2(t)$	SF-SPD	Supply fan VFD speed at time t	% Spd
$v_1(t-4)$	CHWC-EWT	Chilled water coil entering water temperature at time $t-4$	°C
$v_2(t-3)$	OAD-TEMP	OA duct temperature at time $t-3$	°C

4.3.2 Cooling Output Modeling and Validation

As shown in Equation (4.4), Q_{chl}' can be computed from the measured values of the water temperature entering the cooling coil, the water temperature leaving the cooling coil and the flow rate of the chilled water. The value computed from Equation (4.4) approximates the part load of a chiller. Values of Q_{chl}' obtained at the previous time periods also affect the current value of Q_{chl}'. Therefore, $y_1(t-1)$, $y_1(t-2)$, $y_1(t-3)$, $y_1(t-4)$ have been selected as the previous values (states) of Q_{chl}' together with the variables selected above to build the cooling output model $f_1(\bullet)$ at t time stamp.

$$y_1(t) = f(y_1(t-1), y_1(t-2), y_1(t-3), y_1(t-4), x_1(t), x_2(t), x_1(t-4), v_1(t-4), v_2(t-3)) \qquad (4.7)$$

Four data mining algorithms, including the MLP (Multiple-layer Perceptron) neural network, boosting tree [100], random forest [101], and support vector machine [102] have been applied to build predictive models of the cooling output. The multi-layer perceptron is a commonly used feed forward neural network consisting of a number of units organized into multiple layers. Through adaptive adjusting weights among units under supervised learning, the MLP is capable of identifying and learning patterns based on input data sets and the corresponding target values. The boosting tree [97, 98] is a data mining algorithm involving the application of boosting methods to regression trees used in classification as well as in regression. The random forest is a data mining method for classification and regression introduced by Breiman and Cutler [101]. Unlike the standard classification trees that use the best split among all variables at each node for splitting, the random forest algorithm uses the best split among a subset of randomly selected predictors at that node. The support vector machine is a supervised learning method based on kernel functions and is used for classification and function approximation. SVM regression is applied here. It avoids difficulties of using linear functions in the high dimensional feature space and optimization problem is transformed into dual convex quadratic programming problem. In regression case the loss function is used to penalize errors.

By applying four data mining algorithms to build the cooling output model, a comparative analysis of the generated results has been performed. The data set of 1440 observations (1 day data, July 19, 2005) was divided into a training data set (70% were randomly sampled data points) and a testing data set (the remaining 30% of the data). The quality of the predictive models has been compared using the following metrics: MAE (Mean Absolute Error), Std (Standard Deviation), and MRE (Mean Relative Error) defined in Equations (2.11) to (2.13).

Table 4.2 summarizes the prediction performance of the cooling output models built by four data mining algorithms.

Table 4.2 Prediction accuracy of cooling output models
built by four different data mining algorithms

Cooling Output					
	MAE	Std	MRE	Max	Min
MLP NN	0.1841	0.1666	0.20%	1.5686	0.0008
Random Forest	0.2018	0.1770	0.22%	1.3473	0.0001
Boosting Tree	0.2475	0.2280	0.27%	1.5004	0.0004
SVM	0.1938	0.1532	0.22%	0.7682	0.0002

The results in Table 4.2 demonstrate that the model built by the MLP neural network produces the most accurate and stable results. Therefore, the MLP neural network algorithm is selected for building a predictive model of a cooling output from a larger data set. The first-day data from the two-day data set is used as the training set to build the model, while the second day is used as the testing data set to validate the model. Table 4.3 shows the description of the two data sets.

Table 4.3 The description of the two-day data set

Data Set	Start Time	End Time	Description
1	6/19/2005 0:00	6/19/2005 23:59	Training data set; 1440 observations
2	6/20/2005 0:00	6/20/2005 23:59	Testing data set; 1440 observations

For the hidden and output neurons of the MLP neural network algorithm, five different activation functions were selected as candidates, namely, the logistic, identity, tanh, exponential, and sine functions. The number of hidden units was selected between 5 and 18, and the weight decay for both the hidden and output layer varied from 0.0001 to 0.001. In the final neural network model, nine hidden neurons are chosen. The activation functions in the hidden and the output layers are exponential functions. Figure 4.2 illustrates 200 one-minute data points drawn from the test data set to validate the accuracy of the model.

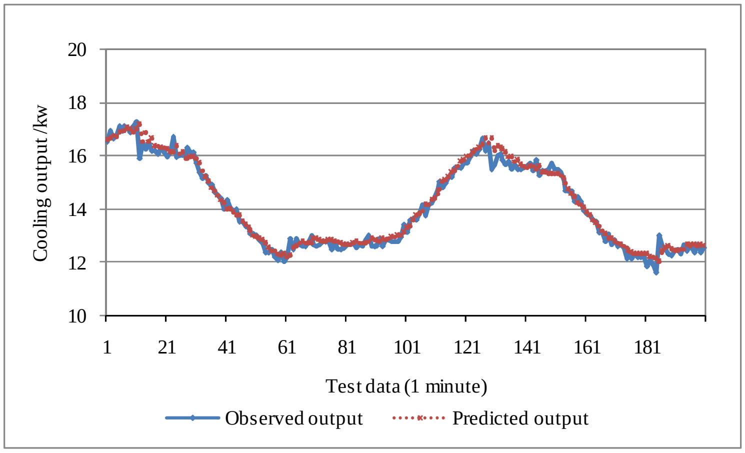

Figure 4.2 Validation of the cooling output model with 200 test points

The predicted load function in Figure 4.2 follows fairly closely the observed load curve. Table 4.4 summarizes the numerical values

Table 4.4 Predictive cooling output model accuracy by using the MLP neural network

Cooling Output					
Data Set	MAE	Std	MRE	Max	Min
Training Data	0.1732	0.1492	0.19%	1.5983	1.3701E-05
Test Data	0.2013	0.1669	0.22%	1.2797	0.0004

4.3.3 Supply Air Temperature and Humidity Modeling and Validation

In a typical HVAC system, the temperature of the supply air leaving the AHU is maintained around 55 °F (12.8°C) to accommodate air condensation. As controllable variables such as the position of the chilled water cooling coil valve and the supply air fan speed change, the overall thermal conditions of the supply air are also affected. Models of supply air temperature and humidity should be built so that cooling output optimization can conform to the acceptable fluctuation range of the supply air temperature and humidity. Using the previously introduced methodology (Section 4.2) of building a cooling output model, the temperature and humidity predictive models are represented as $f_2(\bullet)$ and $f_3(\bullet)$, as seen below (see Equations (4.8) and (4.9)).

$$y_2(t) = f(y_2(t-1), y_2(t-2), y_2(t-3), y_2(t-4), x_1(t), x_2(t), x_1(t-4), v_1(t-4), v_2(t-3)) \tag{4.8}$$

$$y_3(t) = f(y_3(t-1), y_3(t-2), y_3(t-3), y_3(t-4), x_1(t), x_2(t), x_1(t-4), v_1(t-4), v_2(t-3)) \tag{4.9}$$

where $y_2(t-1)$, $y_2(t-2)$, $y_2(t-3)$, $y_2(t-4)$ are supply air temperature at $t-1$, $t-2$, $t-3$ and $t-4$ time stamp; $y_3(t-1)$, $y_3(t-2)$, $y_3(t-3)$, $y_3(t-4)$ are supply air humidity at $t-1$, $t-2$, $t-3$ and $t-4$ time stamp; other input variables are the same as those in the cooling output model.

Using the MLP neural network algorithm and the same data split shown in Table 3 for training and testing, the model validation results with 200 test points are shown in Figure 3 and Figure 4. The supply air temperature model includes 12 hidden units and a hyperbolic tangent function. Exponential functions are used as the activation functions in the hidden and output layers. The supply air humidity model has five hidden neurons and a hyperbolic tangent function is used as the activation function in the hidden and output layers.

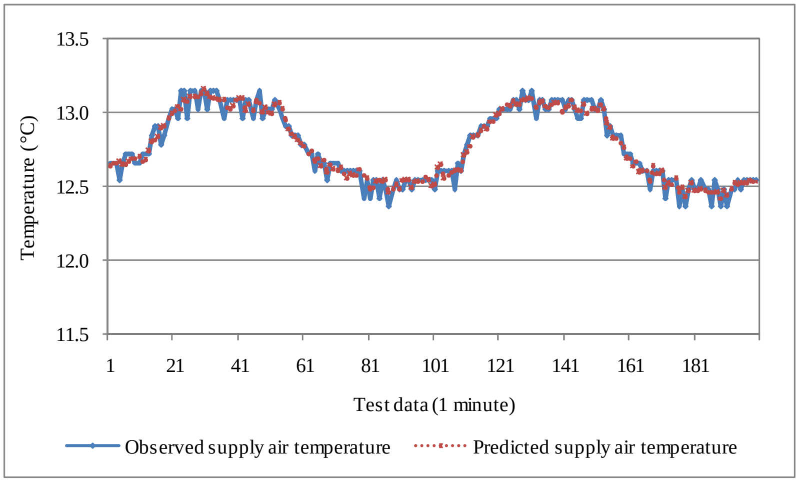

Figure 4.3 Validation of the supply air temperature model with 200 test points

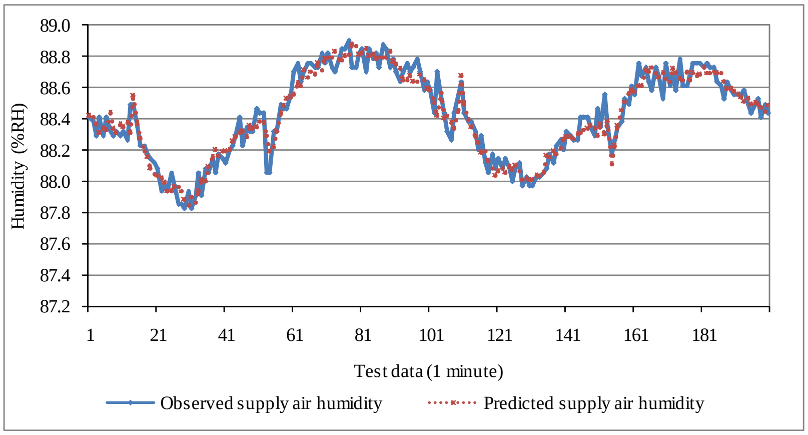

Figure 4.4 Validation of the supply air humidity model with 200 test points

The results in Table 4.5 present the accuracy of the temperature and humidity model.

Table 4.5 Prediction accuracy of the MLP neural network model

Supply Air Temperature					
Data Set	MAE	Std	MRE	Max	Min
Training Data	0.0283	0.0274	0.05%	0.1628	2.0444E-05
Test Data	0.0298	0.0290	0.06%	0.1618	2.8944E-05
Supply Air Humidity					
Data Set	MAE	Std	MRE	Max	Min
Training Data	0.0583	0.0519	0.07%	0.5034	0.0002
Test Data	0.0618	0.057	0.07%	0.5365	1.9700E-05

4.3.4 Optimization Model

Models learned by the MLP neural network have been used for optimization. To minimize the cooling output, the objective function is represented as $y_1 = f_1(\bullet)$. Major constraints are supply air temperature and humidity, so $y_2 = f_2(\bullet)$ and $y_3 = f_3(\bullet)$ need to be maintained in a certain range. Here, the supply air temperature is constrained between 54°F (12.2°C) to 56°F (13.3°C), and the humidity range is set between 87% RH to 90% RH. As uncontrollable input variables are essentially independent of the controllable ones, the values of uncontrollable variables, such as the outside air temperature or chilled water supply temperature, can be fixed in seeking the optimal control settings at each minute time stamp. Therefore, only the values of controllable variables are modified at each time. The search space of the two controllable variables, the chilled water cooling coil valve position and the relative speed of the supply air fan, are set based on their learning range. The optimization model is stated in (4.10).

$$\begin{gathered}
\min_{x_1(t),x_2(t)}(y_1(t)) \\
\textit{subject to :} \\
y_1(t) = f(y_1(t-1), y_1(t-2), y_1(t-3), y_1(t-4), x_1(t), x_2(t), x_1(t-4), v_1(t-4), v_2(t-3)) \\
y_2(t) = f(y_2(t-1), y_2(t-2), y_2(t-3), y_2(t-4), x_1(t), x_2(t), x_1(t-4), v_1(t-4), v_2(t-3)) \\
y_3(t) = f(y_3(t-1), y_3(t-2), y_3(t-3), y_3(t-4), x_1(t), x_2(t), x_1(t-4), v_1(t-4), v_2(t-3)) \\
12.2 \leqslant y_2(t) \leqslant 13.3 \\
87 \leqslant y_3(t) \leqslant 90 \\
50 \leqslant x_1(t) \leqslant 90 \\
70 \leqslant x_2(t) \leqslant 75
\end{gathered} \tag{4.10}$$

The constraints $y_2(t)$ and $y_3(t)$ in (4.10) are neural network derived models. Solving such a nonlinear constrained optimization problem is a challenge. To solve model (4.10), its constraints are placed into the objective function shown in (4.11).

$$\max\{0, 12.2 - y_2(t)\} + \max\{0, y_2(t) - 13.3\} + \max\{0, 70 - y_3(t)\} + \max\{0, y_3(t) - 90\} \tag{4.11}$$

Each of the four terms in (4.11) remains 0, and the sum equals 0 when $y_2(t)$ and $y_3(t)$ are within their bounds. When either $y_2(t)$ or $y_3(t)$ violates a constraint, the sum becomes a positive number. The larger this constraint is, the larger the sum. Therefore, minimizing the objective function (4.11) can push $y_2(t)$ and $y_3(t)$ towards their bounds. Then, the optimization model can be modified to (4.12).

$$\begin{gathered}
\min_{x_1(t),x_2(t)}(obj_1, obj_2) \\
\textit{subject to :} \\
y_1(t) = f(y_1(t-1), y_1(t-2), y_1(t-3), y_1(t-4), x_1(t), x_2(t), x_1(t-4), v_1(t-4), v_2(t-3)) \\
y_2(t) = f(y_2(t-1), y_2(t-2), y_2(t-3), y_2(t-4), x_1(t), x_2(t), x_1(t-4), v_1(t-4), v_2(t-3)) \\
y_3(t) = f(y_3(t-1), y_3(t-2), y_3(t-3), y_3(t-4), x_1(t), x_2(t), x_1(t-4), v_1(t-4), v_2(t-3)) \\
50 \leqslant x_1(t) \leqslant 90 \\
70 \leqslant x_2(t) \leqslant 75
\end{gathered} \tag{4.12}$$

where $obj_1 = y_1$,

$obj_2 = \max\{0, 12.2 - y_2(t)\} + \max\{0, y_2(t) - 13.3\} + \max\{0, 70 - y_3(t)\} + \max\{0, y_3(t) - 90\}$.

To solve this bi-objective optimization problem, an evolutionary strategy algorithm is used. Note that this optimization model is solved at different time stamps in Section 4.3.5. Observed values rather than the predicted values are used at the next point optimization. For example, to obtain $y_1(t)$ at t stamp, the observed values $y_1(t-1)$, $y_1(t-2)$, $y_1(t-3)$, and $y_1(t-4)$ are used rather than the predicted ones.

4.3.5 Problem Solving by the Evolutionary Strategy

Unlike single objective optimization, the solution to a bi-objective optimization problem converges to the Pareto-optimal front. Different solutions trade off differently between the two objectives. Optimal solutions change with the weights assigned to the objectives. It is possible that one solution is better than the other for one objective (e.g., cooling output minimization) but is worse for the other objective (e.g., supply air temperature). Neither can dominate the other, and they are called non-dominated solutions. The Strength Pareto Evolutionary Algorithm (SPEA) [77, 78] is used to search the space of non-dominated solutions and update them to the elite set at each generation. Detailed algorithm is presented in the section 3.2.1 of Chapter 3.

The mutation operation in Step 7 is realized by adding noise $\triangle x_i$ to the x_i, where x_i is the i^{th} controllable variable and $\triangle x_i$ is a Gaussian distribution with zero mean and standard deviation σ [79]. Solution x_i is updated to x_i' by $x_i' = x_i + N(0, \sigma)$. The value of σ is selected for each variable, and it remains fixed for all generations. The mutated solutions are checked for possible constraint violations. When a constraint is violated, the value of the violating solution is replaced with a corresponding constraint-bound value to make sure all solutions remain in the specified search space. For $x_1(t)$, set $\sigma_1 = 0.4$ and for $x_2(t)$, set $\sigma_2 = 0.1$. In case no feasible solution satisfying the supply air temperature and humidity constraints exists, the evolutionary strategy algorithm is repeated. The parameters of this algorithm are tuned to meet the solution quality and the computational cost requirements. The values of the parameters used in the evolutionary strategy algorithm, such as the ratio of the parent and offspring size, the initial population size, and the maximum number of generations (the stopping criterion), are set as follows.

For three time stamps, namely 12:09 AM, 8:57 AM and 8:08 PM, Fig. 5 demonstrates the fitness function values for different ratios of parent and offspring sizes. Note that the objective function at the vertical axis refers to the obj_1 in model (4.12), not the fitness values computed in Equations (3.3) and (3.4) at each generation.

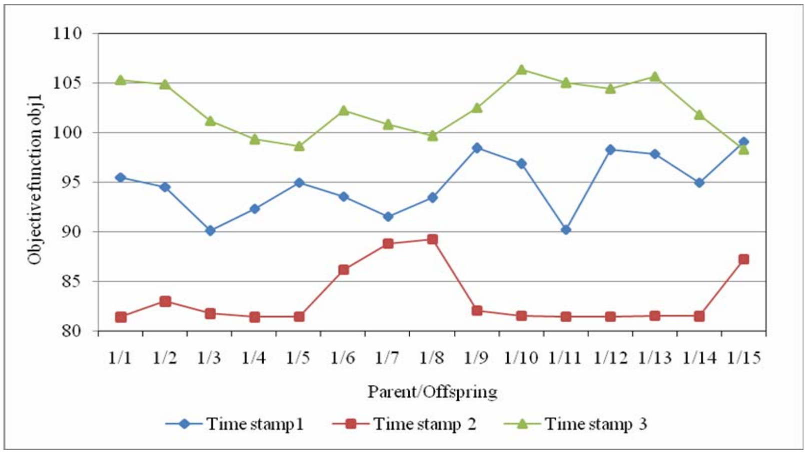

Figure 4.5 Values of the objective function obj_1 for different ratios of the parent and offspring sizes

As shown in the Figure 4.5, the parent offspring ratio of 1/3 is selected as producing the best quality results. Using this ratio, initial population sizes from 40 to 200 individuals have been tested, and the results are shown in Figure 4.6.

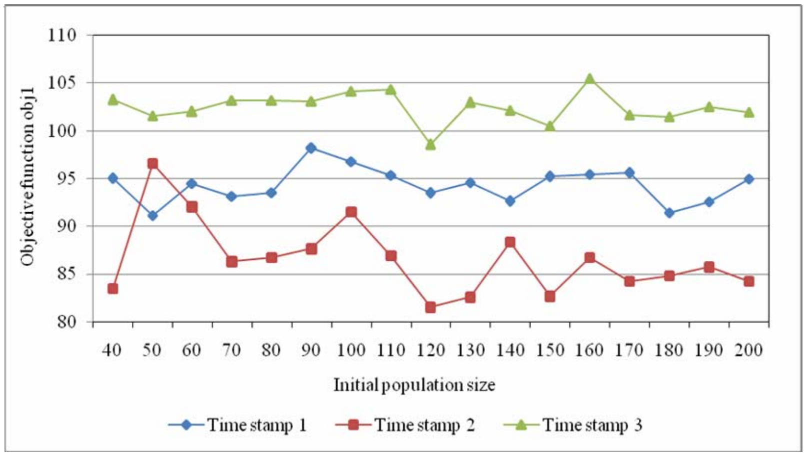

Figure 4.6 Values of the objective function obj_1 for different initial population sizes

Figure 4.6 shows that the initial population size of 120 minimizes the objective value and therefore is selected as the optimal initial population size. Figure 4.7 shows the average value of the objective function for the elite solutions at each generation. The elite population is updated at each generation, and the average value converges as the number of iterations increases.

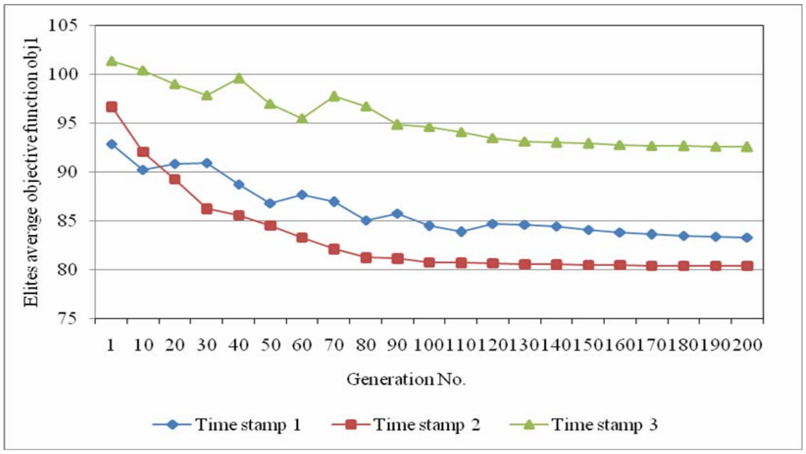

Figure 4.7 Average value of the objective function obj_1 of elite solutions at each generation

As shown in the Figure 4.7, when the number of iterations exceeds 110, the average value of the objective function obj_1 of elite solutions converges to a small value that slightly fluctuates. Therefore, the maximum generation number is set to 110 as the stopping criterion for the evolutionary computation algorithm.

Figure 4.8 to Figrue 4.11 present distributions of the offspring and elite population sets at different generations, respectively. The horizontal axis refers to obj_1 and the vertical axis to obj_2 . As the number of generations increases, the distributions of offspring and elite individuals move from the right to the left, thus following a minimizing trend. The algorithm terminates when the generation number reaches the maximum number, and the minimum obj_1 with obj_2 equals 0 ($y_2(t)$ and $y_3(t)$ obey the constraints) and will be chosen as the final optimal value for optimized cooling output. The optimal controllable settings which are obtained correspond to the minimum cooling output.

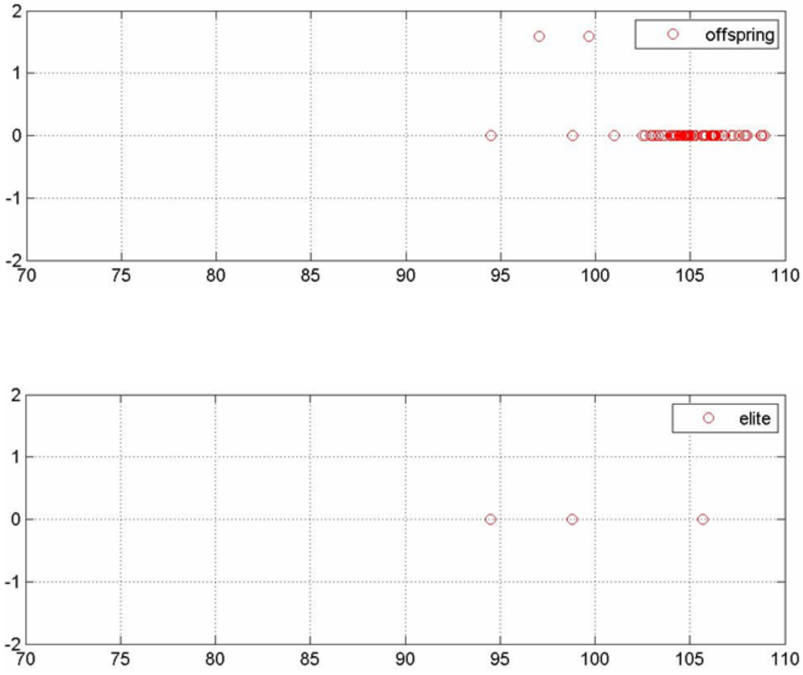

Figure 4.8 Distribution of the offspring and elite objective values (2nd generation)

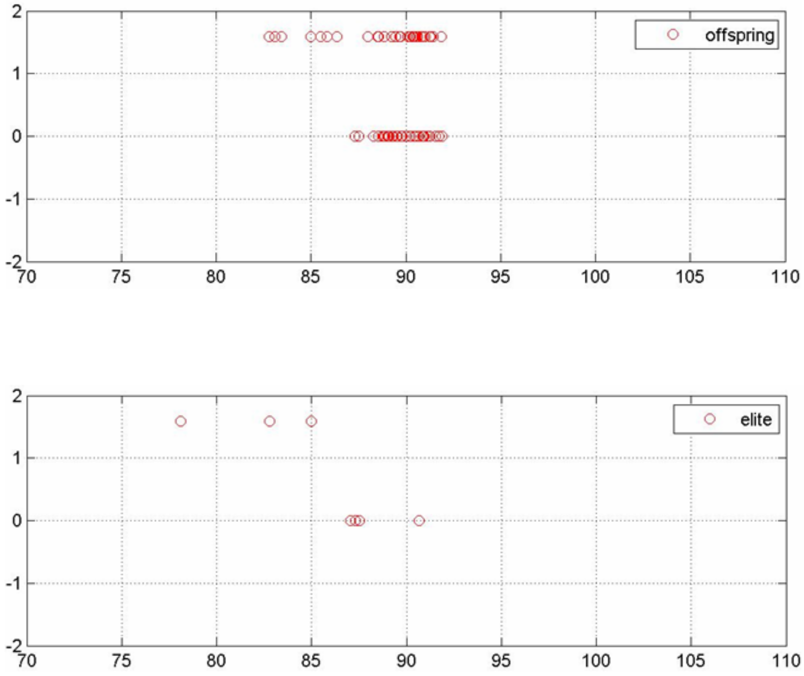

Figure 4.9 Distribution of the offspring and elite objective values (10th generation)

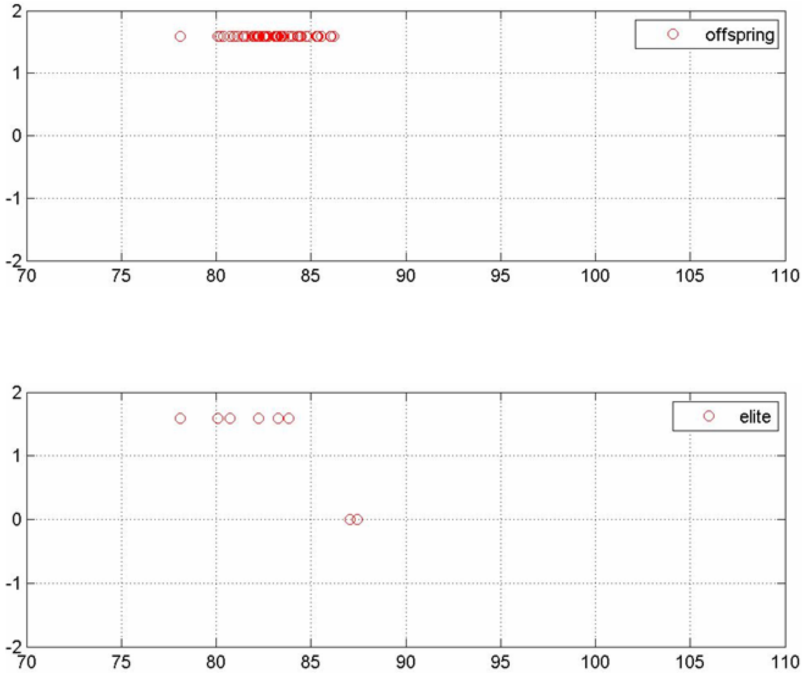

Figure 4.10 Distribution of the offspring and elite objective values (20th generation)

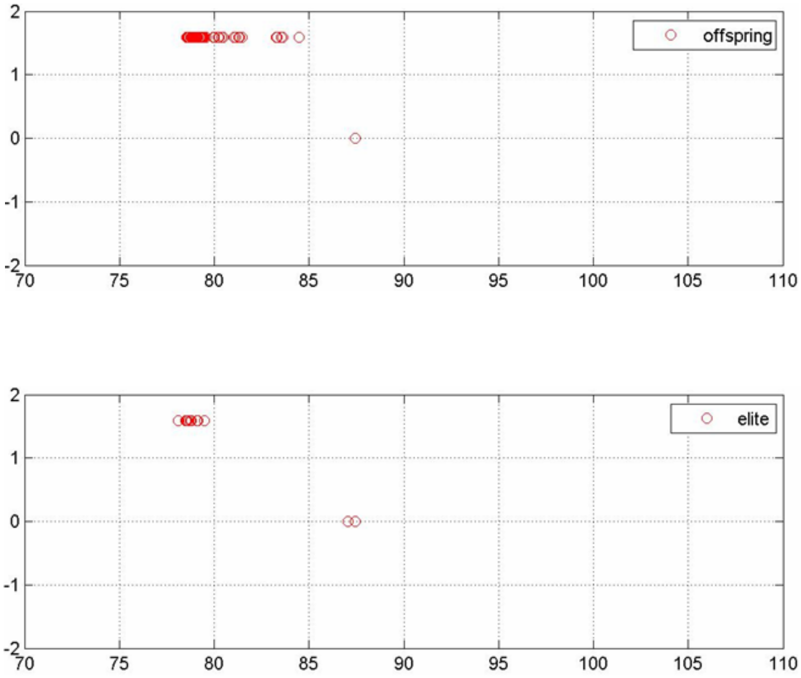

Figure 4.11 Distribution of the offspring and elite objective values (30th generation)

4.3.6 Optimization Results and Discussion

At each time stamp, the evolutionary strategy algorithm is applied to solve the optimization model (4.12). Optimal solutions of controllable settings, namely the chilled water cooling coil valve position and the supply air relative fan speed, satisfy the two objective functions. The simulation results for the first 100 points of training data set in the Table 4.3 are shown in Figure 4.12 to Figure 4.16. It should be noted that observed values are used as the previous status values in solving the dynamic equations of $y_1 = f_1(\bullet)$, $y_2 = f_2(\bullet)$ and $y_3 = f_3(\bullet)$.

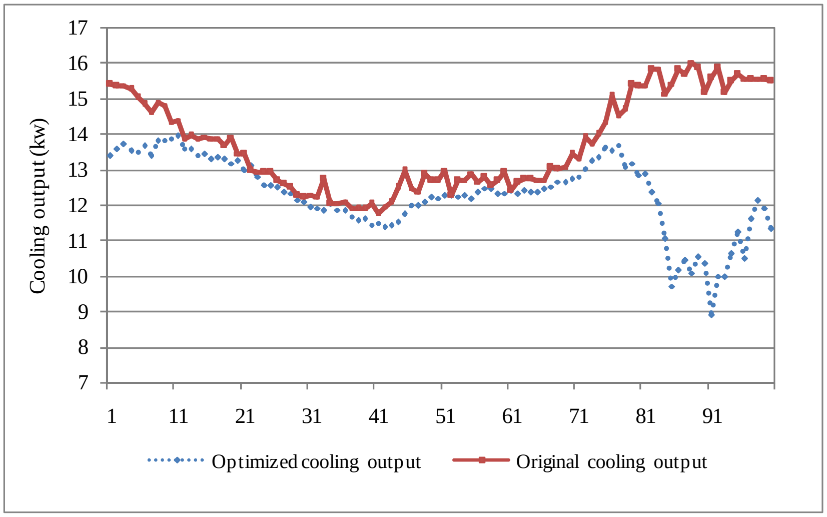

Figure 4.12 The first 100 points of the optimized cooling output

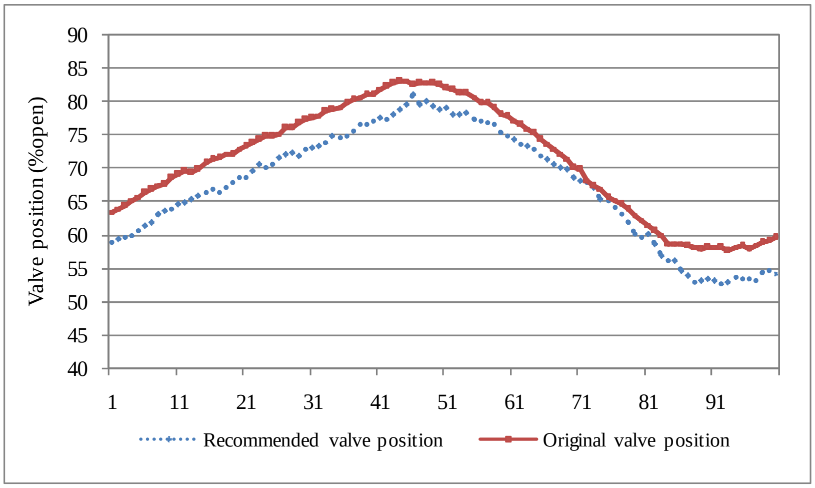

Figure 4.13 The first 100 points of the recommended valve position

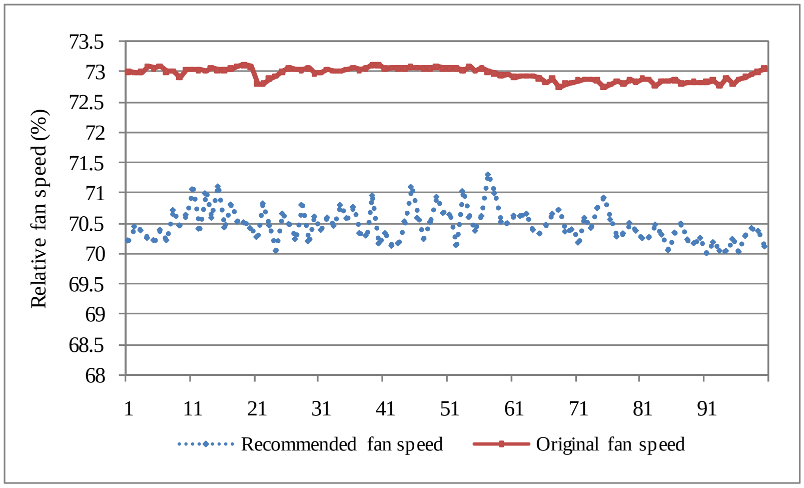

Figure 4.14 The first 100 points of the recommended supply air relative fan speed

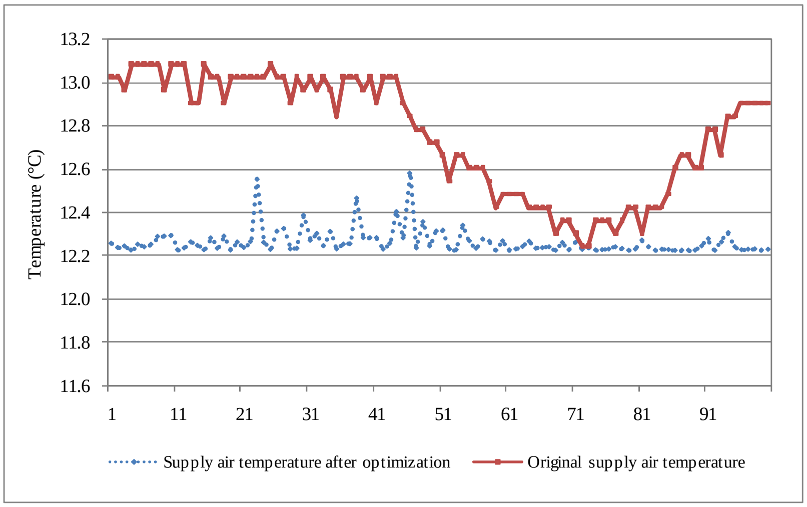

Figure 4.15 The first 100 points of the measured and optimized supply air temperature

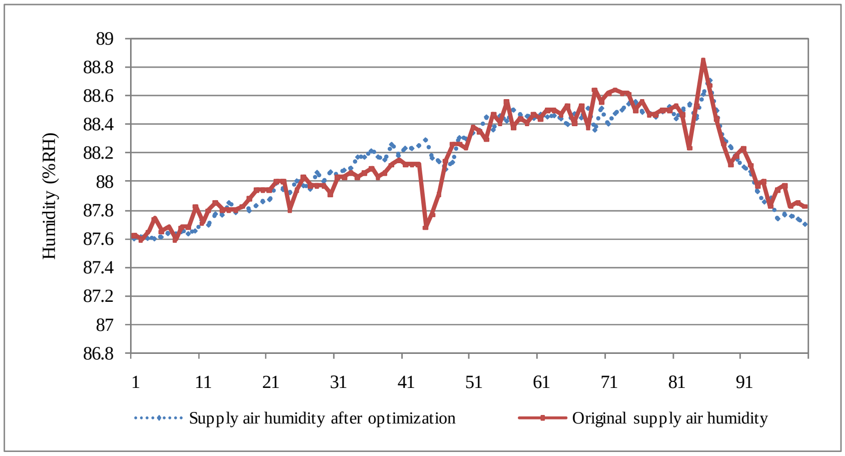

Figure 4.16 The first 100 points of the measured and optimized supply air humidity

The graph in Figure 4.12 shows that the cooling output at each time stamp is reduced by adjusting the controllable variables. Both the value position and the relative fan speed have smaller values compared to the original control settings, as illustrated in Figure 4.13 and Figure 4.14. Meanwhile, Figure 4.15 and Figure 4.16 show that the supply air temperature and the humidity remain in an acceptable range. By lowering the speed of a supply fan, the amount of cooling air discharged out of the AHU is reduced. In this scenario, heat transferred from the chilled water is decreased, because there is relatively more cooling air at the cooling coil end part. For this reason, adjusting the chilled water entering the cooling coil valve position at the same time to decrease the heat transfer from the water side to the air side can decrease the difference in temperature of the chilled water entering and leaving. A lower temperature difference refers to a smaller cooling output for the cooling coil and the chiller. Therefore, by adjusting two controllable variables simultaneously at each time stamp to minimize the cooling output, an energy saving potential can be achieved from the chiller side.

4.4 Summary

In this chapter, the cooling output of an Air Handling Unit (AHU) was minimized by using a computational intelligence approach. MLP neural networks were applied to build dynamic models for controllable and uncontrollable input variables and output variables, such as cooling output, supply air temperature, and humidity. A modified evolutionary strategy algorithm was applied to solve the bi-objective optimization model while providing values of optimal controllable variables. The simulation results demonstrated that the cooling output was reduced while the supply air temperature and humidity in the AHU remained in an acceptable range. A reduced cooling output decreased the part load for the chiller, and therefore energy consumption was reduced. Further research involving wider ranges of data is needed. In this chapter, one step optimization was discussed. In the future research, iterative replacement of the observed values with the predicted ones will be considered.

CHAPTER 5

OPTIMIZATION OF REHEAT PROCESS IN A VARIABLE-AIR-VOLUME BOX

5.1 Introduction

With upheavals of energy cost in recent years, operations efficiency of HVAC (Heating, Ventilating and Air-conditioning) system becomes more and more significant. Therefore, many research has been done in modeling and optimizing the local devices and process or even the overall HVAC system from an optimization perspective. Traditional approaches in optimizing HVAC systems are based on analytical models and mathematical programming methods. Lu et al [103] propose a systematic approach in optimizing the general HVAC systems based on mathematical models of the energy consuming components. Based on steady state models, Zheng et al [54] formulate the thermal process in a variable air volume as a constrained problem and provide optimal daily operating trajectories in achieving both energy saving and optimal outdoor air-flow rates. Rink et al [12] consider the minimum-cost control of multi-zone cooling system as a three-dimensional periodic optimization problem and solve it by using state-increment dynamic programming. Substantial savings of energy coast have been demonstrated by providing the optimal use of cooling storage. Due to the nonlinear, highly constrained and multidimensional nature of HVAC system, recent efforts incorporate computational intelligence aspects into the systems modeling and optimization. Yang et al [104] present an application of ANN (Artificial Neural Network) in establishing non-linear mapping in a building control system to determine the optimal start time for a heating system. Xi et at [105] develop the 2-by-2 nonlinear dynamic model of a HVAC system by SVR (Support Vector Regression) and generate online control signals to improve control performance by using SMO (Sequential Minimal Optimization) algorithm to solve the optimization problem. To tackle two HVAC energy management problems, Fong et al [106] present the robust evolutionary algorithm to improve the optimization effectiveness.

The research reported in this chapter focuses on using computational approach to transform the reheating process in VAV (Variable Air Volume) box into a bi-objective optimization problem. Compared to other data mining algorithms, ANN is applied to derive the dynamic models from real data collected in HVAC system. To solve the problem, a modified PSO based on two levels of non-dominated solutions is introduced with fast convergence features. Based on the model predictive control (MPC) strategy, recommended current reheating coil valve positions are obtained in minimizing reheating output while maintaining VAV discharge air temperature and room humidity under certain acceptable level in the future.

5.2 Methodology for Optimizing reheating process in a VAV box

5.2.1 Model Predictive Control Based Reheating Process Optimization

Reheat system in the VAV terminal provides a quantity of conditioned air to the specific thermal zone in the building envelop. For the reheat system with hydronic reheat coil, hot water goes through coils adjusted by the valve position and heat exchange takes place between water side and air side. Amount of air entering the zone is accomplished by dampers in the VAV box. Based on the idea of model predictive control strategy, which is commonly used in the process control area, reheating process is optimized under some criterion by giving the recommended control output. Some common elements in MPC algorithms are employed in this research, namely predictive model, objective function and control law [107].

Predictive models refer to fully capturing the dynamic process while providing reasonable predictions of outputs in the future with known values up to current time.

Predictive models could be analytical models on the basis of mathematical equations, empirical models developed from the real data or mixed models combining both advantages of the previous two types. In this paper, since reheating process in the VAV box is a dynamic process with nonlinear and time-varying features, temporal process models are derived from the real data based on certain data mining algorithms.

Considering the process as a multi-input-single-output (MISO) system, it can be represented as $y = f(\boldsymbol{x}, \boldsymbol{v})$, where $f(\cdot)$ is the function capturing the process, $\boldsymbol{x} \in R^m$ is a vector of m controllable variables, $\boldsymbol{v} \in R^n$ is a vector of n uncontrollable variables and y is the response variable. For the reheating process, y could be target variable to be optimized, e.g. reheating output, or performance variables to be satisfied under some constraints, e.g. acceptable discharge air temperature or humidity. x may refer to controllable variables like reheating valve position or damper position while v may refer to uncontrollable variables like VAV entering air temperature. In a discrete-time form, nonlinear process model can be viewed as mappings between those variables that are available for predicting system behavior up to the current time and those to be predicted at or after that instant.[108] It can be expressed by the general equation below:[94]

$$y(t) = f([y(t-d)]_{d \in D_y}, [x_1(t-d)]_{d \in D_{x_1}}, \ldots, [x_m(t-d)]_{d \in D_{x_m}}, [v_1(t-d)]_{d \in D_{v_1}}, \ldots [v_n(t-d)]_{d \in D_{v_n}}) \quad (5.1)$$

Where D_y is the set containing all the possible time delays related to response variable $y(t)$ at the time stamp t; D_{x_i} is the set containing all the possible time delays related to i^{th} controllable variable $x_i(t)$ at the time stamp t; D_{v_j} is the set containing all the possible time delays related to j^{th} uncontrollable variable $v_j(t)$ at the time stamp t.

It is noted that considering the real implementation of providing recommended control output at current time stamp, other input variables should only take values up to the current time. For example if the predictive model is expressed as $y(t) = f(y(t-1), x(t-2), v(t-3))$. The function could be rearranged as $y(t+2) = f(y(t+1), x(t), v(t-1))$. In determining the controllable variable $x(t)$ at time stamp t, information after this time stamp is unknown and therefore could not be obtained. Therefore, $y(t+1)$ cannot be used in this function.

After developing the temporal process model, optimization model could be formulated with certain objective functions and constraints. Optimization model considered in this chapter is to minimize the reheating output while maintaining the corresponding VAV discharge air temperature and room humidity under certain threshold. Minimization of reheating output will realize some energy saving potential from the hot water supplied side. Reheating output can be calculated as [90]:

$$Q_{heat} = C_{pm} m_w (T_{hwst} - T_{hwlt}) \tag{5.2}$$

Where C_{pm} is the specific heat capacity of the hot water, m_w is the mass flow rate of the hot water, T_{hwst} and T_{hwlt} are entering and leaving water temperature of the hot water respectively.

The Optimization model can be presented in the following Equation (5.3)

$$\begin{aligned}
&\underset{\boldsymbol{x}^*(t)}{\arg\min} \quad y_1^*(t+d_{y_1}) \\
&y_1^*(t+d_{y_1}) = f(y_1(t), y_1(t-1), \ldots, \boldsymbol{x}^*(t), \boldsymbol{x}(t-1)\ldots, \boldsymbol{v}(t), \boldsymbol{v}(t-1), \ldots) \\
&y_2^*(t+d_{y_2}) = f(y_2(t), y_2(t-1), \ldots, \boldsymbol{x}^*(t), \boldsymbol{x}(t-1)\ldots, \boldsymbol{v}(t), \boldsymbol{v}(t-1), \ldots) \\
&y_3^*(t+d_{y_3}) = f(y_3(t), y_3(t-1), \ldots, \boldsymbol{x}^*(t), \boldsymbol{x}(t-1)\ldots, \boldsymbol{v}(t), \boldsymbol{v}(t-1), \ldots) \\
&s.t. \\
&\boldsymbol{x}^*(t) \in S_{\boldsymbol{x}}, y_2^*(t+d_{y_2}) \in S_{y_2}, y_3^*(t+d_{y_3}) \in S_{y_3}
\end{aligned} \tag{5.3}$$

Where $\boldsymbol{x}^*(t)$ refer to the vector of recommended adjusting controllable variables at current time stamp t; $y_1^*(t+d_{y_1})$, $y_2^*(t+d_{y_2})$ and $y_3^*(t+d_{y_3})$ are predicted values with response time delay d_{y_1}, d_{y_2} and d_{y_3} to the adjusted controllable variables; other variables without asterisks are real measured values.

Due to the nonlinearity, complexity and opaqueness of the temporal models built in this paper, it poses challenge for solving by traditional mathematical programming methods. Therefore, multiple objective Particle swarm optimization (MOPSO) algorithm is employed in solving the optimization problem by searching out near-optimal solutions.

Fast converging advantage of particle swarm algorithms make it very promising in real on-line optimization.

5.2.2 Multiobjective Optimization by Particle Swarm Optimization Algorithms

Unlike evolutionary algorithms, Particle Swarm Optimization (PSO) [29] is a stochastic optimization technique inspired by the bird flocks. As one of the swarm intelligence algorithms, PSO has a well-balance mechanism to enhance and adapt the global and local exploration abilities. PSO first obtains great success in single objective optimization [109, 110] with its fast convergence advantage. Recently, multi-objective PSO has gained attention gradually in the research communities. In this chapter, a modified PSO based on two level of non-dominated solutions is introduced which is first proposed by M. A. Abido [76].

The basic algorithm of the canonical PSO can be described as follow [29]:

Step 1: Initialize n particles $x_i \in R^{N_{\mathrm{dim}}}$ and velocities $v_i \in R^{N_{\mathrm{dim}}}$;

Step 2: Compute fitness function $f(\bullet)$ for each particle;

Step 3: Find current best position $\hat{l}_i$ for each particle and let $\hat{g}$ be the global best;

Step 4: For each particle, update the particle velocities and positions:

$$\begin{aligned} &v_i \leftarrow \omega v_i + c_1 rand()(\hat{l}_i - x_i) + c_2 rand()(\hat{g} - x_i) \\ &x_i \leftarrow x_i + v_i \end{aligned}$$

Step 5: If the stop criterion is satisfied, global best $\hat{g}$ is the final optimal solution with fitness $f(\hat{g})$; otherwise, return to Step 2.

Where $\mathrm{x_i}, v_i$ refer to the position and velocity of i^{th} particle. N_{dim} is the dimension of searching space. $\hat{l}_{\mathrm{i}}$ is the local best particle for i^{th} particle while $\hat{\mathrm{g}}$ is the global best particle for all particles. For the velocity updating part, ω is the inertia factor, which is used to balance the global and local search. Two random generated coefficients are drawn from uniform distribution $\mathrm{U}(0,1)$. $\mathrm{c}_1, \mathrm{c}_2$ are learning factors which control the influence of the social and cognitive components. In this paper, set $w = 0.95$ and set $c_1 = c_2 = 2$.

The canonical PSO is often applied to solve the single objective optimization problem. To make it adapt into solving the multi-objective optimization problem, some modifications have been done as follow [76]:

Modification 1: Generate a storing set S_i for each particle x_i to store the non-dominated solutions through iterations for that particle;

Modification 2: Generate a storing set G to store the non-dominated solutions from all S_i at each iteration;

Modification 3: Generate an external set E to store the non-dominated solutions from G through iterations;

Modification 4: Local non-dominated set S_i updating process: at each iteration, compare current particle solution and stored solutions in the previous rounds in the objective values space, dominated solutions are removed from the set while non-dominated solutions are kept.

Modification 5: Global non-dominated set G updating process: at each iteration, copy all local non-dominated sets at this iteration into global non-dominated set. Solutions are compared among one to another and only non-dominated solutions are kept.

Modification 6: External non-dominated set E updating process: at each iteration, copy the global non-dominated set G to E. Previous solutions at E and newly entered solutions from G at current iterations are compared while only non-dominated solutions are kept.

Modification 7: Local and global best solution generating process: for each particle at each iteration, Euclidean distance among solutions from corresponding local non-dominated set and global non-dominated set are measured, the pair with minimum distance in the objective value space is selected as the local and global best for this particle in undertaking the later velocity and position updating process.

In contrast to the canonical PSO which mainly solve the single objective problem, idea of Pareto-optimality is incorporated into the algorithm to expand its capability in solving problems with several conflicting objectives simultaneously. In the modified algorithm, several sets are generated to store the non-dominated solutions under different scales.

5.3 Industrial Case Study and Computational Results

5.3.1 Data Description and Feature Selection

Data used in this research is collected at Energy Resource Station (ERS), home to the Iowa Energy Center.

More than 50 data points for each test room are recorded and data sampling time is 1 minute. Data from May 8th to May 26th in 2009 of one test room is used to investigate the reheat process in this paper. During this period, set back heating and cooling set points are set as 66°F and 78°F from 18:00 to 6:00 on the next day while occupant heating and cooling set points are set as 70°F and 74°F on the rest of the day. Reheating process data for each day has been selected from the original data set. Table 5.1 describes the data used for feature selection, model training and testing.

Table 5.1 Data description

Data set	Time period	Data description
1	5/8/2009~5/10/2009	Feature selection, algorithms selection, 404 observations
2	5/8/2009~5/25/2009	Model training,1526 observations
3	5/26/2009	Model testing, 114 observations

Due to the larger number of parameters in the original data set, many parameters are relatively irrelevant or redundant for the reheating process and therefore feature selection needs to be preformed. For example, damper position is important in the reheating process by determining amounts of air entering the room. But in this data set, it is kept as 30% open and could be discarded as irrelevant information. In this paper, a boosting tree algorithm is used to perform the feature selection, as it shares advantages of the decision tree induction and tends to be robust in removal of irrelevant parameters [97, 98]. In the boosting method, a sequence of binary trees is built. Each tree focuses on learning instances misclassified by the previous trees based on the prediction error. All trees are integrated with different weights in a single model. In the boosting tree algorithm, a split at every node of any regression tree is based on certain criteria, e.g., minimization of the total regression error used in this chapter. In the process of generating successive trees, the statistical importance of each variable at each split of every tree is accumulated and normalized. Predictors with a higher importance rank indicate a larger contribution to the predicted output parameter. Based on both the domain knowledge and boosting tree results, four variables are selected for building the models of VAV reheating output, VAV supply air temperature and room humidity. Parameters are described in the Table 5.2.

Table 5.2 Parameters description for developing models

Variable	Point name	Description	Unit
y_1	HEAT	Output variable, reheating output	Watt
y_2	VAV-DAT	Output variable, VAV box discharging air temperature	°F
y_3	RM-HUMD	Output variable, room humidity	RH
v_1	INT-WIN	Input uncontrollable variable, light level on window	Ftc
v_2	PLN-HUMD	Input uncontrollable variable, plenum air humidity	RH
v_3	VAV-EAT	Input uncontrollable variable, VAV box entering air temperature	°F
x_1	VAVHCVLV	Input controllable variable, VAV box heating coil valve position	%Open

As shown in the table, v_1, v_2 and v_3 are measured variables which cannot be manipulated. x_1 is the controllable variable and the recommended adjusting value will be searched in minimizing the reheating output while maintaining the VAV box discharging air temperature and room humidity under certain tolerance level. Since the reheating process is a dynamic process, time delay of current valve position changes influencing on reheating output and thermal comfort may vary. Both correlation coefficients and boosting tree algorithms are applied to investigate strengths of linearity and nonlinearity among different previous status values of inputs and outputs in building the dynamic models and three models to be built could be represented as the functional mapping described below.

$$y_1(t+1) = f_1[y_1(t), y_1(t-1), x_1(t), x_1(t-1), v_1(t), v_2(t), v_3(t)] \qquad (5.4)$$

$$y_2(t+4) = f_2[y_2(t), y_2(t-1), x_1(t), x_1(t-1), v_1(t), v_2(t), v_3(t)] \qquad (5.5)$$

$$y_3(t+8) = f_3[y_3(t), y_3(t-1), x_1(t), x_1(t-1), v_1(t), v_2(t), v_3(t)] \qquad (5.6)$$

It is noted that the minimum time interval is 1 minute. The adjustment of valve position value at current time stamp t will influence most on reheating output at $t+1$ time stamp, VAV discharge air temperature at $t+4$ time stamp and room humidity at $t+8$ time stamp.

5.3.2 Temporal Predictive Model Building and Validating

Several data mining algorithms, namely MLP neural network, SVM, Boosting tree, Pace regression and Random forest, have been applied to extract the models from data based on these three mapping equations. MLP (Multi-layer Perceptron) neural network [111, 112] is a commonly used feed-forward neural network involving numerous units organized into multiple layers. Through adaptive adjusting weights among units under supervised learning, the MLP is capable of identifying and learning patterns based on input data sets and the corresponding target values. In this paper, four different activation functions (the logistic, identity, tanh, and exponential functions) are selected for the hidden and output neurons of the MLP neural network algorithm. The number of hidden units is set between 5 and 25, and the weight decay for both the hidden and output layer varies from 0.0001 to 0.001. SVM [98] (Support Vector Machine) is a supervised learning method based on kernel functions, and it is used for classification and function approximation. The support vector machine is a supervised learning method based on kernel functions and is used for classification and function approximation. SVM regression is applied here. It avoids difficulties of using linear functions in the high dimensional feature space and optimization problem is transformed into dual convex quadratic programming problem. In regression case the loss function is used to penalize errors. In this paper, the radial basis function (RBF) is used as the kernel function in the SVM algorithm. Apart from feature selection, boosting tree [97, 98] can also applied in applied in regression and classification. In this paper, the maximum number of additive trees equals 200 and to avoid overfitting, each consecutive tree is built using a subset of data, and the subset proportion is set to 0.5. The Pace regression algorithm [52, 53] is a relatively new approach for developing linear models in high dimensional spaces. It consists of a group of estimators that are either optimal overall or optimal under certain conditions. In this paper, empirical Bayes is used as the estimator. The random forest is a data mining method for classification and regression introduced by Breiman and Cutler [101]. Unlike the standard classification trees that use the best split among all variables at each node for splitting, the random forest algorithm uses the best split among a subset of randomly selected predictors at that node. In this paper, the maximum number of trees from the forest is set to 200.

As shown in the table 5.1, data set 1 of three days have been used to select the algorithms for building the models. The following metrics are used to measure the prediction accuracy of the model: the MAE (mean absolute error), Std_AE (standard deviation of absolute error) [46]. They are defined in Equations (2.11) to (2.13)

Tables 5.3 to 5.5 summarize the prediction performance of the reheating output model, VAV discharge air temperature model and room humidity model built by five data mining algorithms respectively. Max and Min refer to the maximum and minimum absolute error between observed and predicted values.

Table 5.3 Prediction accuracy comparison of reheating output model

Algorithms	MAE	Std_ AE	Max	Min
MLP	58.1708	66.8506	486.3586	0.0141
Random Forest	66.4441	93.5182	663.5640	0.7633
SVM	64.1313	81.2788	560.7002	0.1322
Boosting	66.3925	88.6143	522.4716	0.7342
Pace Regression	68.7535	88.5757	562.6107	0.2320

Table 5.4 Prediction accuracy comparison of VAV discharge air temperature model

Algorithms	MAE	Std_AE	Max	Min
MLP	0.3991	0.4235	3.1832	0.0016
Random Forest	0.9591	0.8009	3.9914	0.0002
SVM	0.8264	0.8630	5.1879	0.0017
Boosting	0.9718	0.8370	4.4633	0.0121
Pace Regression	1.1049	0.9405	5.2009	0.0033

Table 5.5 Prediction accuracy comparison of room humidity model

Algorithms	MAE	Std_ AE	Max	Min
MLP	0.2240	0.2212	2.5902	0.0008
Random Forest	0.3896	0.7951	6.9207	0.0009
SVM	0.8088	0.4894	3.2795	0.0005
Boosting	0.7230	1.0365	9.9106	0.0056
Pace Regression	0.5523	0.6604	5.5368	0.0039

As shown in the tables, MLP neural network overperforms other data mining algorithms with the smallest mean absolute error and standard deviation of absolute error. Therefore, it is selected as the best algorithm for further development of three models. Training data set of data set 2 and testing data set of data set 3 shown in Table 1 are selected for building and validating the models respectively. Training and testing results are present in the Table 5.6.

Table 5.6 Prediction accuracy of models training and testing

Reheating output				
Data set	MAE	Std of AE	Max	Min
Training	57.9976	70.1547	633.0374	0.0960
Testing	64.5382	54.4437	242.8076	0.1180
VAV discharge air temperature				
Data set	MAE	Std of AE	Max	Min
Training	0.5073	0.5081	3.4058	0.0016
Testing	0.6425	0.4952	2.6043	0.0096
Room humidity				
Data set	MAE	Std of AE	Max	Min
Training	0.2457	0.2160	1.5635	0.0006
Testing	0.3717	0.3728	2.0252	0.0007

In developing the neural networks, sum of squared error is used as the cost function and weights are modified in minimizing that cost function accordingly. Table 5.7 describes the detailed information of three neural networks

Table 5.7 Detailed information of three models

Model output	Hidden units	Activation function on hidden units	Activation function on output units
Reheating output	15	Logistic function	Exponential function
VAV discharge air temperature	24	Hyperbolic tangent function	Logistic function
room temperature	19	Hyperbolic tangent function	Hyperbolic tangent function

Model testing results of 114 data points are shown in the Figures 5.1 to 5.3.

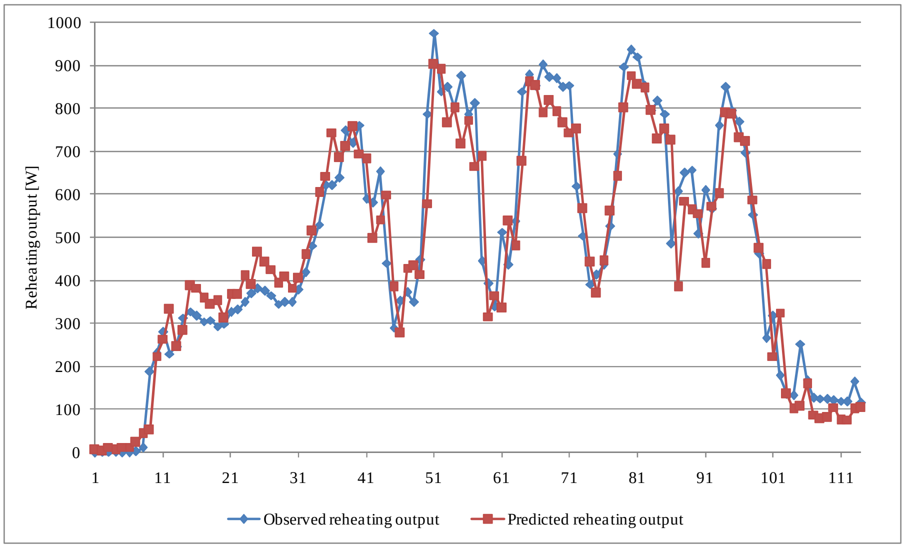

Figure 5.1 Testing results of reheating output model

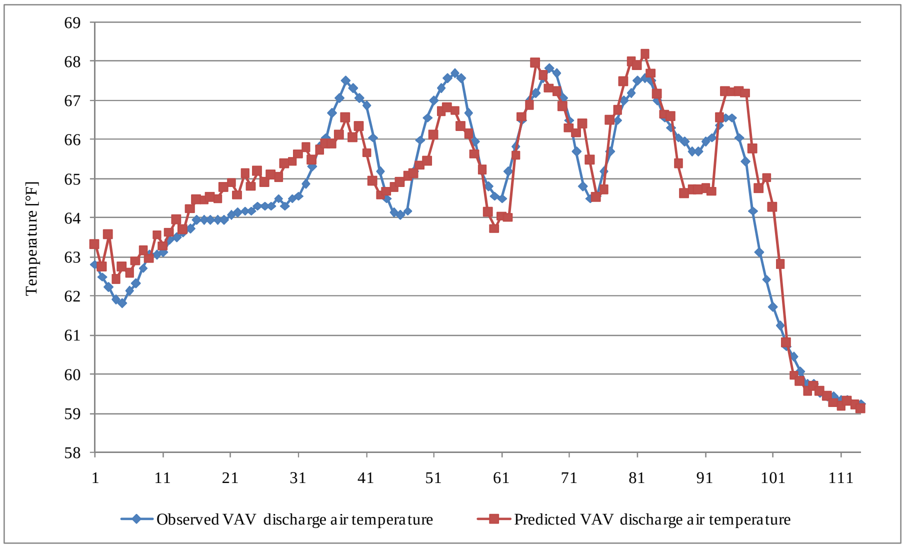

Figure 5.2 Testing results of VAV discharge air temperature model

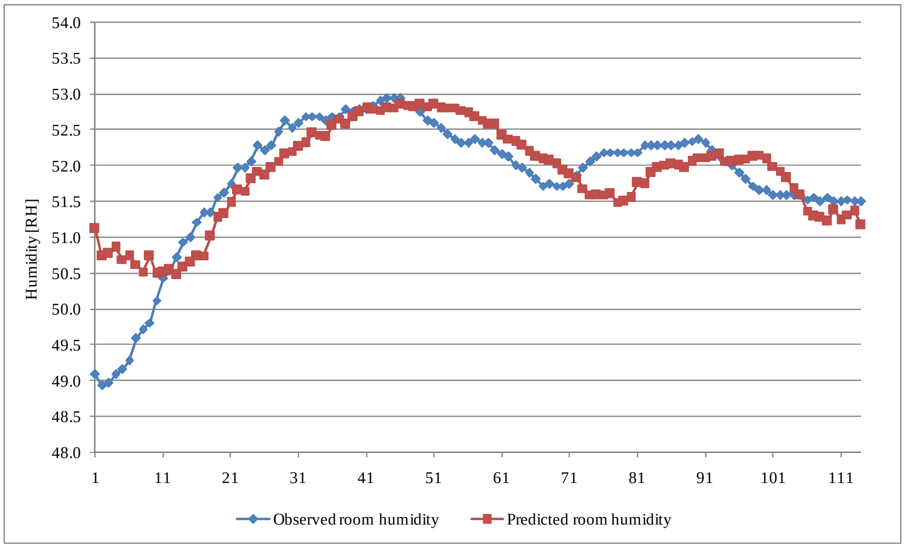

Figure 5.3 Testing results of VAV room humidity model

As shown in the figures, predicted values track relatively close to the observed ones. Therefore, three models will be applied in formulating the optimization model.

5.3.3 Optimization Model Formulation

Three models of reheating output, VAV discharge air temperature and room humidity have been established by the MLP neural networks. To minimize the reheating output so as to achieve some energy saving potentials, the reheating process needs to be transformed into an optimization model. $y_1(t+1) = f_1(\bullet)$ is selected as the objective function to be minimized by searching the optimal solution of reheating valve position at the current time t. In the mean time, VAV discharge air temperature and room humidity will be affected in the future and therefore they should be included in constraints. In this paper, future temperature is constrained not to go beyond 1 °F while humidity is limited not to go beyond 0.5 RH. These tolerance values could be replaced by any other values based on people’s preference and the methodology will be remained the same. The optimization model could be formulated as:

$$\begin{gathered}\min_{x_1(t)}(y_1(t+1))\\ \text{subject to:}\\ y_1(t+1)=f(y_1(t),y_1(t-1),x_1(t),x_1(t-1),v_1(t),v_2(t),v_3(t))\\ y_2(t+4)=f(y_2(t),y_2(t-1),x_1(t),x_1(t-1),v_1(t),v_2(t),v_3(t))\\ y_3(t+8)=f(y_3(t),y_3(t-1),x_1(t),x_1(t-1),v_1(t),v_2(t),v_3(t))\\ 0\le x_1(t)\le 80\\ y_2^*(t+4)-1\le y_2(t+4)\le y_2^*(t+4)+1\\ y_3^*(t+8)-0.5\le y_3(t+8)\le y_3^*(t+8)+0.5\end{gathered}\qquad(5.7)$$

where y_2^*, y_3^* are predicted values of temperature and humidity by applying the original reheating coil valve position value. In minimizing the reheating output at $t+1$ time stamp by manipulating valve to the optimal position, VAV discharge air temperature and room humidity are maintained in certain acceptable level relative to original values without optimization. Transforming the constrained problem into a non-constrained problem, a bi-objective optimization problem can be built by constructing objective functions as $Obj1 = y_1(t+1)$,

$$Obj2=\max\{0,y_2^*(t+4)-1-y_2(t+4)\}+\max\{0,y_2(t+4)-y_2^*(t+4)-1\}+\max\{0,y_3^*(t+8)-0.5-y_3(t+8)\}$$
$$+\max\{0,y_3(t+8)-y_3^*(t+8)-0.5\}$$

Then the bi-objective model is formulated as:

$$\begin{gathered}\min_{x_1(t)}(Obj1,Obj2)\\ \text{subject to:}\\ y_1(t+1)=f(y_1(t),y_1(t-1),x_1(t),x_1(t-1),v_1(t),v_2(t),v_3(t))\\ y_2(t+4)=f(y_2(t),y_2(t-1),x_1(t),x_1(t-1),v_1(t),v_2(t),v_3(t))\\ y_3(t+8)=f(y_3(t),y_3(t-1),x_1(t),x_1(t-1),v_1(t),v_2(t),v_3(t))\\ 0\le x_1(t)\le 80\end{gathered}\qquad(5.8)$$

It is noted that when $Obj2=0$, all the constraints are satisfied. In solving this optimization problem, a multi-objective PSO (Particle Swarm Optimization) algorithm is employed.

5.3.4 Problem solving by MOPSO

In solving the optimization problem shown in Equation (5.8), one instance is randomly selected from the training data set described in the table to tune certain parameters of the algorithm. Two parameters are tuned here, namely the initial population size and the number of iterations. Table 5.8 shows the detailed description of the instance selected.

Table 5.8 Instance selected for tuning parameters

DATE	TIME	INT-WIN(T)	PLN-HUMD(T)
2009-5-8	7:14:00	22.813	45.063
VAV-EAT(T)	VAVHCVLV(T-1)	VAVHCVLV(T)	HEAT(T-1)
57.875	20.547	20.641	2.25504
HEAT(T)	HEAT(T+1)	VAV-DAT(T-1)	VAV-DAT(T)
2.318	2.5365	65.438	64.688
VAV-DAT(T+4)	RM-HUMD(T-1)	RM-HUMD(T)	RM-HUMD(T+8)
64.688	52.906	53	53.5

Vary the initial population size from 1 to 30. The maximum number of iterations is set as 100. Since the final results are not a single solution, but a set of solutions, optimal solution selecting criteria are needed for making the final decision. Here the selecting criterion is to satisfy constraints as much as possible. Figure 5.4 shows the optimal results with different initial population size.

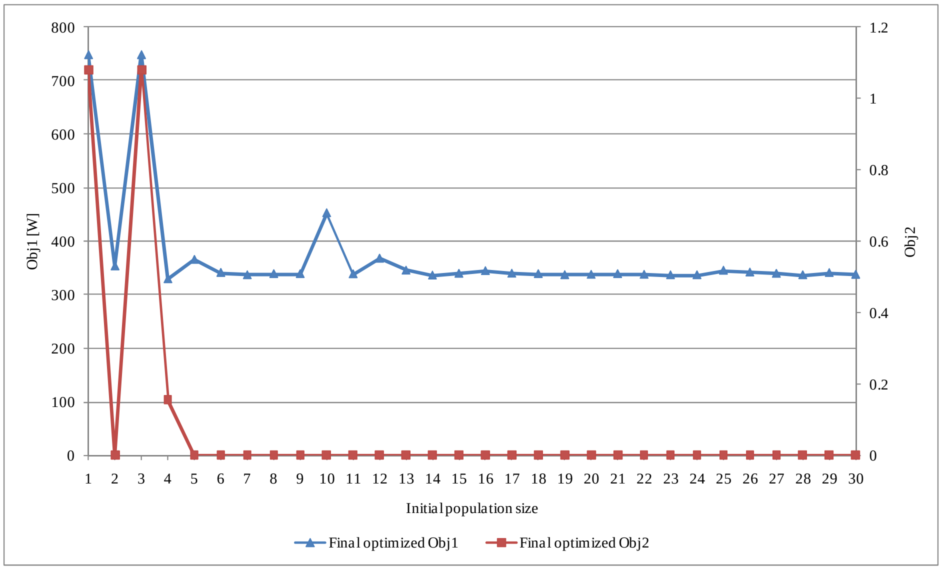

Figure 5.4 Optimal results with different initial population size

As shown in the figure, when the number of particles initially generated is small, there is no guarantee that constraints could be satisfied since searching range is limited. As the initial size becomes larger, final optimal results are relatively the same after 100 iterations. Figure 5.5 demonstrates the computation cost for each running with different initial population size.

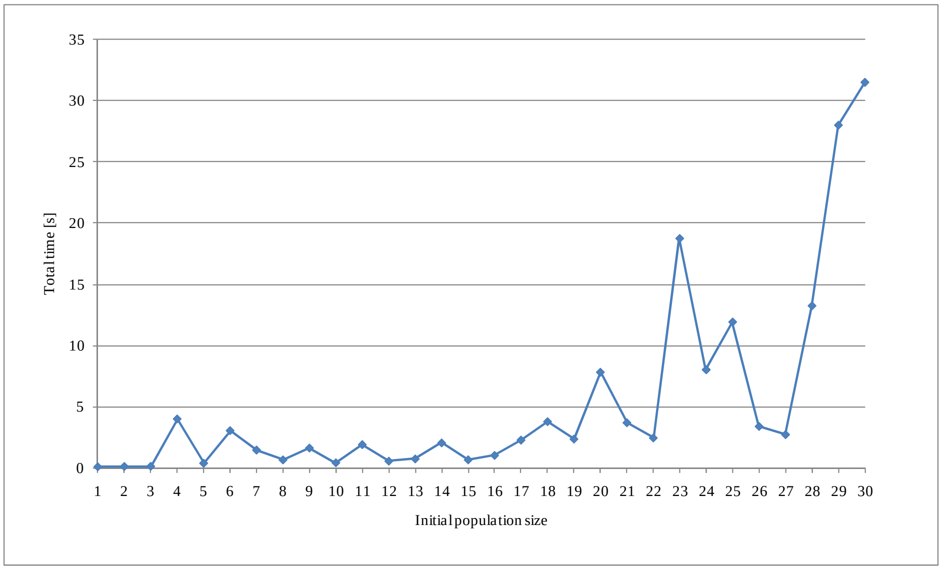

Figure 5.5 Computation cost with different initial population size

As shown in the figure, although the larger size of initial population size can increase the possibility of finding out near-optimal solutions, computation cost will become more obvious. Considering the on-line implementation, time for searching out good solutions should be short. Therefore, initial population size is set as 15 since it can both give good results and reasonable computation cost.

Another factor influencing the computation cost is the maximum number of iterations. Set initial population size as 15 and record the best solution at different iterations. Figure 5.6 demonstrates the optimal results at different iterations.

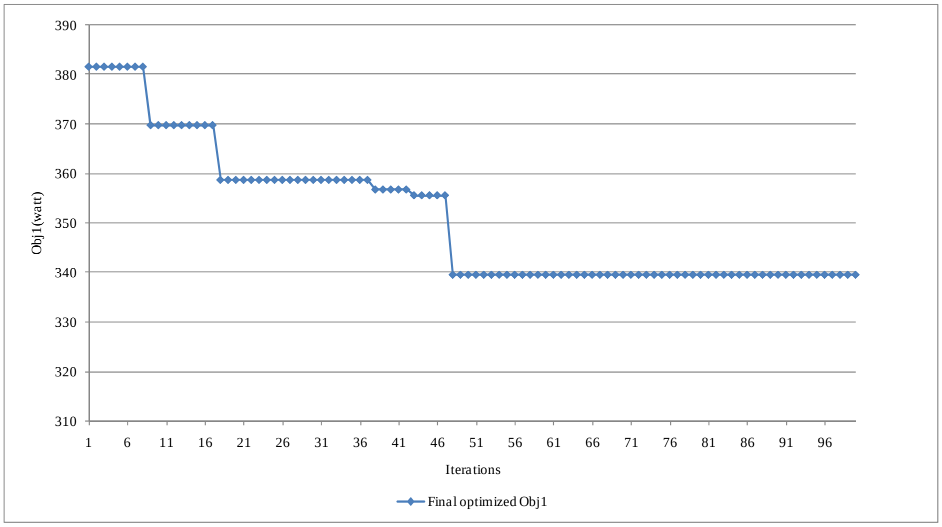

Figure 5.6 Optimal results at each iteration

As shown in the figure, as the number of iterations goes beyond certain threshold, the Obj1 converges. The larger the number of iterations, the more running time that will be taken (shown in the Figure 5.7). Therefore, the maximum number of iterations is set as 50. Major time cost comes from the local, global and external non-dominated set updating process. To further reduce the computation cost, clustering technique is applied to these three procedures. When the updating process finishes, if the number of non-dominated solution exceed certain threshold, clustering will be preformed to randomly select one solution as the representative from a group of solutions with similar close distances in objective value space. Here set size to trigger clustering in local, global and external non-dominated set are 5, 15 and 30 respectively. The tolerance distance along Obj1 and Obj2 axis are 0.2 and 0.05. Solutions with distance smaller than these two tolerance distance in objective value space will be clustered as a group and one will be randomly selected as the representative. Noted that solutions with Obj2=0 will not be involved in the clustering. Figure 5.7 shows the comparison of computation cost with and without employing clustering techniques.

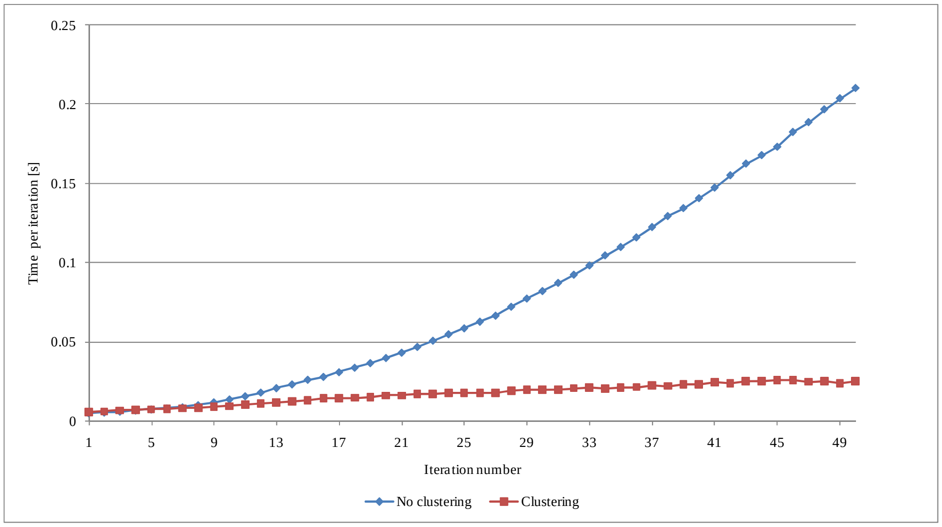

Figure 5.7 Comparison of computational cost with and without clustering

As seen in the figure, as the number of iterations goes up, clustering technique apparently restricts the further increase of the computation cost. Figure 5.8 describe the distribution of solutions from external set at 30th iteration in objective value space with and without clustering performed.

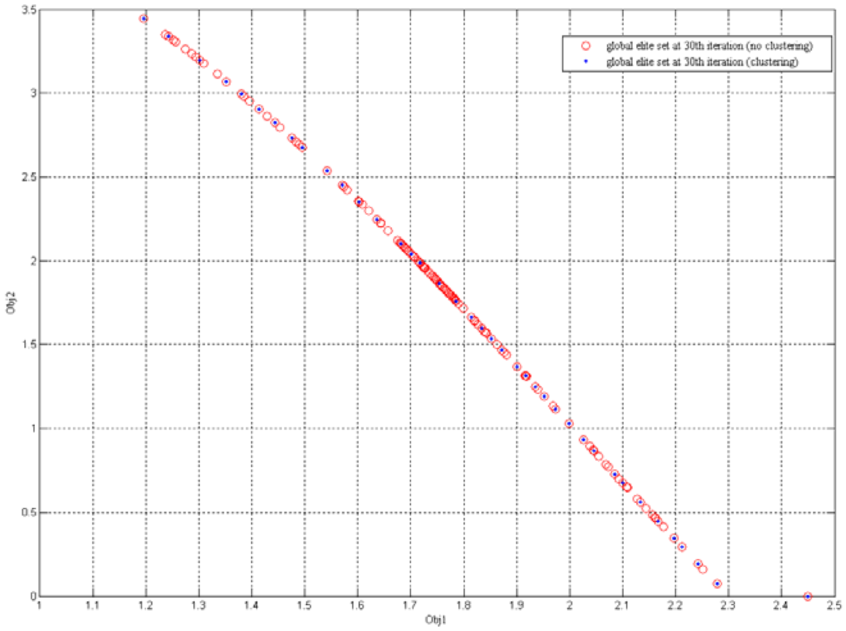

Figure 5.8 Comparison of solutions distribution in objective value space with and without clustering

As shown in the figure, the number of solutions has been dramatically reduced after clustering. But the shape of the distribution front is not destroyed since points in the small neighborhood are well represented.

5.3.5 Optimization Results and Discussion

Proposed multiple objective PSO algorithm is applied to solve the model on the one time stamp at a time fashion. 70 data points from 6:36 AM to 7:45 AM on May 8th, 2009 are selected. It is noted that observed values, rather than predicted ones are used in the next time stamp optimization. Figures 5.9 to 5.12 show the optimization results.

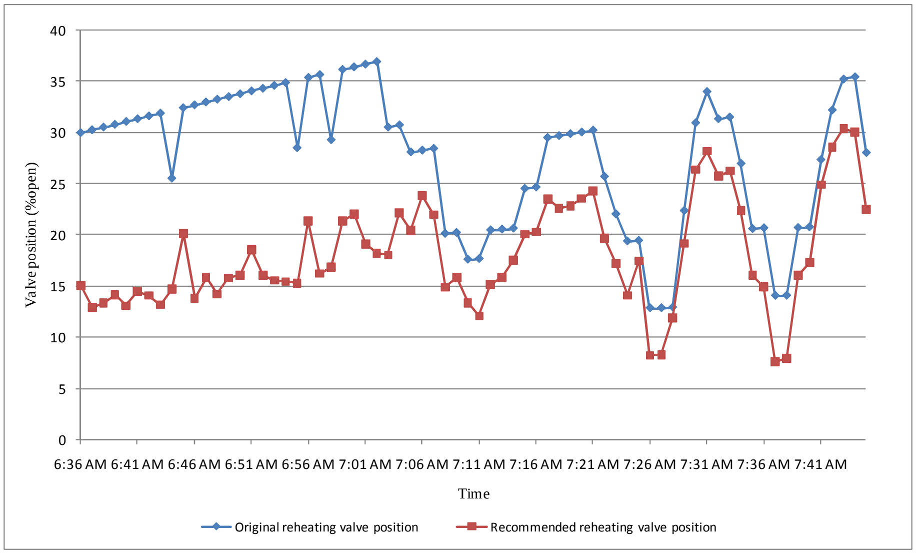

Figure 5.9 Comparison of original and recommend reheating valve position

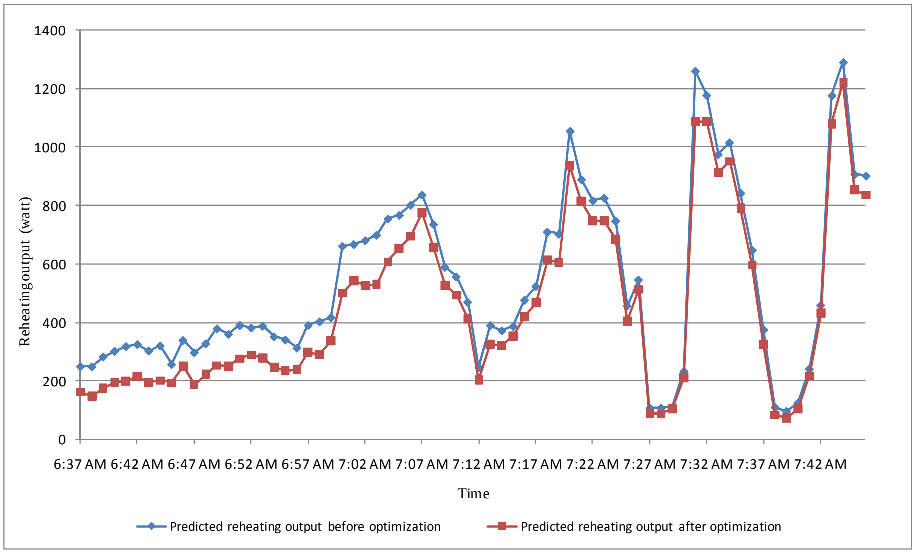

Figure 5.10 Comparison of reheating output before and after optimization

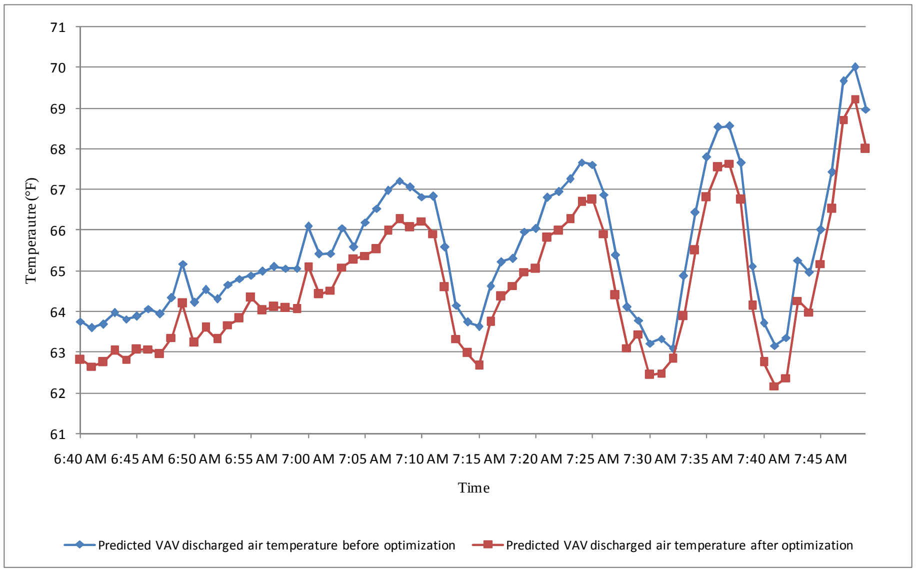

Figure 5.11 Comparison of VAV discharge air temperature before and after optimization

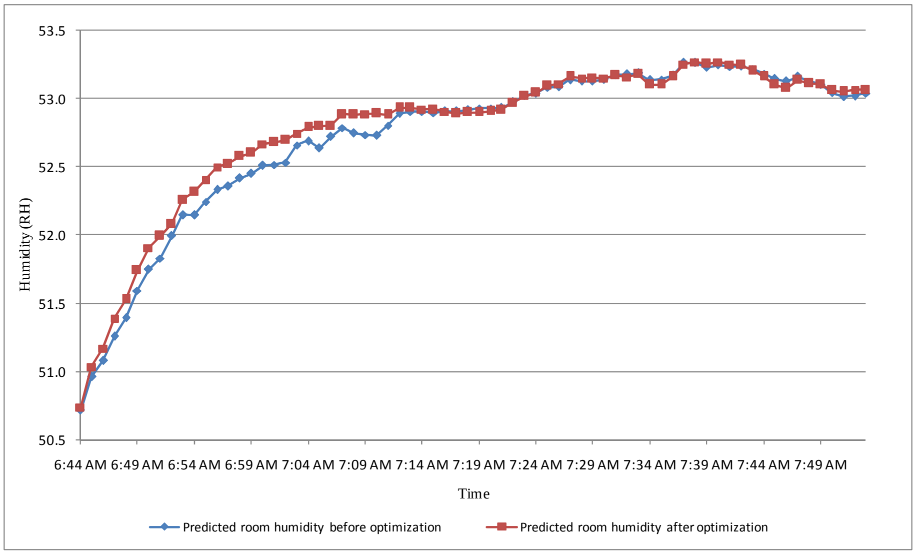

Figure 5.12 Comparison of room humidity before and after optimization

As shown in the figures, recommended values of reheating valve position could be given by minimizing the reheating output in the future time stamp and maintaining the VAV discharge air temperature and room humidity under certain acceptable range. It is straightforward that less reheating output will hinder the speed of temperature increase, and there exist a trade-off. If a small tolerance of temperature and humidity in the future is admitted, a quantitative adjusting value of reheating valve position could be given at the current time in achieving some energy saving potentials by minimizing the reheating output at next time stamp.

5.4 Summary

In this chapter, a computational intelligence approach for optimization the reheating process in the VAV box was presented. The MLP neural network outperformed other data mining algorithms and therefore was selected to build the model of reheat output, VAV discharge air temperature, and room humidity. The reheat process was solved by the modified PSO algorithm Based on the model predictive control strategy, the optimal control output value of the reheat valve position was generated. Computational results demonstrate that energy savings could be achieved by the recommended reheating valve position as the trade-off between energy consumption and thermal comfort. Future research will focus on on-line implementation of the proposed methodology.

CHAPTER 6.

CONCLUSION

This thesis is focused on applying computational intelligence in HVAC systems modeling and optimization. The framework presented in the thesis includes three major parts. The first part (Chapters 1 and 2) introduces a computational intelligence approach in HVAC modeling 2. Neural network is applied to establish nonlinear mappings among different air quality parameters applicable for construction of control charts for on-line monitoring of sensors. It may provide information to operators and thus decrease the risk due to the sensor failures.

Second part (Chapter 3) is centered on applying computational intelligence in optimization. A bi-objective optimization model is formulated based on the steady state equations of indoor CO_2 concentration. Due to the model complexity, a multi-objective evolutionary algorithm is employed. The run time of the fan is minimized while level of CO_2 concentration is maintained below a preference threshold.

In the third part (Chapters 4 and 5), computational intelligence is applied to process optimization. Neural networks are used to develop temporal models of air handling unit and VAV box. Different optimization models are established with conflicting objectives of energy consumption and thermal comfort. Due to the nonlinearity and complexity of the models, two heuristics algorithms, an evolutionary algorithm and a particle swarm algorithm are considered in searching for near optimal solutions. Energy savings are realized (minimization of cooling output of AHU or reheating output of VAV box) while supply air temperature and humidity are maintained at acceptable level.

The future research will involve integration of the analytical models and data-driven model. Issues such as models overfitting, bias-variance trade off, and computational complexity will be also addressed.

REFERENCES

[1] W. Huang, M. Zaheeruddin and S. H. Cho, "Dynamic simulation of energy management control functions for HVAC systems in buildings," Energy Conversion and Management, vol. 47, no. 7-8, pp. 926-943, 2006.

[2] Z. Cumali, “Global optimization of HVAC system operations in real time,” ASHRAE Transaction, Vol. 94, No. 1, pp. 1729-1744, 1988.

[3] X. Yu, J. Wen and T. F. Smith, “A model for the dynamic response of a cooling coil,” Energy and Buildings, Vol. 37, No. 12, pp. 1278-1289, 2005.

[4] Y. Zhang and V. I. Hanby, “Model-based control of renewable energy systems in buildings,” HVAC&R Research, Vol. 12, No. 3, pp. 739-760, 2006.

[5] S. Wang and Z. Ma, “Supervisory and Optimal Control of Building HVAC Systems: A Review,” Vol. 14, No. 1 HVAC&R Research, pp.3-32 , 2008.

[6] Y. Wang, W. Cai, Y. Soh, S. Li, L. Lu and L. Xie, “A simplified modeling of cooling coils for control and optimization of HVAC systems,” Energy Conversion and Management, Vol. 45, No. 18-19, pp. 2915-2930, 2004.

[7] TRNSYS: A transient system simulation program, Volume 1 (reference manual). Solar Energy Laboratory, University of Wisconsin-Madison, WI, 1996.

[8] D. R. Clark, HVACSIM+ Building System and Equipment Simulation Program Reference Manual: NBSIR 84-2996. Gaithersburg, MA: National Institute of Standard Technology, 1985.

[9] SIMBAD: Building and HVAC Toolbox. Center Scientifique et Technique du Batiment, 1999.

[10] G. P. Henze, D. E. Kalz, S. Liu and C. Felsmann, “Experimental analysis of model-based predictive optimal control for active and passive building thermal storage inventory,” HVAC&R Research, Vol. 11, No. 2, pp. 189-214, 2005.

[11] M. M. Gouda, C. P. Underwood and S. Danaher, “Modeling the robustness properties of HVAC plant under feedback control,” Building Services Engineering Research and Technology, Vol. 24, No. 4, pp. 271-280, 2003.

[12] R. E. Rink and N. Li, “Aggregation / disaggregation method for optimal control of multi-zone HVAC systems ,” Energy Conversion and Management, Vol. 36, No. 2, pp 79-86, 1995.

[13] N. N. Kota, J. M. House, J. S. Arora and T. F. Smith, “Optimal control of HVAC systems using DDP and NLP techniques,” Optimal Control Applications & Methods, Vol. 17, pp. 71-78, 1996.

[14] P. Sreedharan and P. Haves, “Comparison of chiller models for use in model-based fault detection,” Proceedings of International Conference of Enhanced Building Operations, Austin, 2001.

[15] J. L. Nizet, J. Lecomte and F. X. Litt, “Optimal control applied to air conditioning in buildings,” ASHRAE Transactions, Vol. 90, No. 1, pp. 587-600, 1984.

[16] A. M. Bassily and G. M. Colver, “Cost optimization of a conical electric heater,” International Journal of Energy Research, Vol. 29, No. 4, pp. 359-376, 2005.

[17] X. Xi, A. Poo and S. Chou, “Support vector regression model predictive control on a HVAC plant,” Control Engineering Practice, Vol. 15, No. 8, pp. 897-908 2007.

[18] M. Chow and J. Teeter, “Reduced-order functional link neural network for HVAC thermal system identification and modeling,” International Conference on Neural Networks, Houston, TX, USA, 1997.

[19] S. M. Namburu, M. S. Azam, J. Luo, K. Choi and K. R. Pattipati, “Data-driven modeling, fault diagnosis and optimal sensor selection for HVAC chillers,” IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, Vol. 4, No. 3, pp. 469-473, 2007.

[20] E.A. Koeppel, S.A. Klein, J.W. Mitchell and B.A. Flake, “Optimal supervisory control of an absorption chiller system,” HVAC&R Research. Vol. 1, No. 4, pp. 325–342, 1995.

[21] V. Congradac and F. Kulic, “HVAC system optimization with CO2 concentration control using genetic algorithms,” Energy and Buildings, Vol. 41, No. 5, pp. 571-577, 2009.

[22] C. H. Lo, P. T. Chan, Y. K. Wong, A. B. Rad and K. L. Cheung, “Fuzzy-genetic algorithm for automatic fault detection in HVAC systems,” Applied Soft Computing, Vol. 7, No. 2, pp. 554-560, 2007.

[23] K. F. Fong, V. I. Hanby and T. T. Chow, “HVAC system optimization for energy management by evolutionary programming,” Energy and Buildings, Vol. 38, No.3, pp. 220-231, 2006.

[24] J. C. Bezdek, “On the relationship between neural network, pattern recognition and intelligence,” The International Journal of Approximate Reasoning, Vol. 6, pp. 85-107, 1992.

[25] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanism, Spartan Press, Washington, 1961.

[26] M. Minsky and S. Pappert, Perceptrons: An Introduction to Computational Geometry, MIT Press, Cambridge, MA, 1969.

[27] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.

[28] L. J. Fogel, A. J. Owens and M. J. Walsh, Artificial Intelligence through Simulated Evolution, J. Wiley, Chichester, 1966.

[29] J. Kennedy and R.C. Eberhart, “Particle swarm optimization,” Proceedings of International Conference on Neural Networks, pp. 1942-1948, IV (Perth, Australia), Piscataway, NJ, IEEE, Service Center (1995).

[30] L. A. Zadeh, “Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transaction on Systems, Man, and Cybernetics,” Vol. 2, pp. 28-44, 1973.

[31] A. P. Engelbrecht, Computational Intelligence: An Introduction, NJ: John Wiley & Sons Inc, 2007.

[32] A. I. Dounis and C. Caraiscos, “Advanced control systems engineering for energy and comfort management in a building environment—A review,” Renewable and Sustainable Energy Reviews, Vol. 13, No. 6-7, pp. 1246-1261, 2009.

[33] S. Atthajariyakul and T. Leephakpreeda, “Neural computing thermal comfort index for HVAC systems,” Energy Conversion and Management, Vol. 46, No. 15-16, pp. 2553-2565, 2005.

[34] C. V. Altrock, H. O. Arend, B. Krause, C. Steffens and E. Behrens-Rommler, “Adaptive fuzzy control applied to home heating system,” Fuzzy Sets and Systems, Vol. 61, No. 1, pp. 29-35, 1994.

[35] E. H. Mathews, C. P. Botha, D. C. Arndt and A. Malan, "HVAC control strategies to enhance comfort and minimize energy usage," Energy and Buildings, vol. 33, no. 8, pp. 853-863, 2001.

[36] S. Cho, Y. Hong, W. Kim and M. Zaheer-uddin, "Multi-fault detection and diagnosis of HVAC systems: an experimental study," International Journal on Energy Research, vol. 29, no. 6, pp. 471-483, 2005.

[37] T. Salsbury and R. Diamond, "Performance validation and energy analysis of HVAC systems using simulation," Energy and Buildings, vol. 32, no. 1, pp. 5-17, 2000.

[38] Z. Hou, Z. Lian, Y. Yao and X. Yuan, "Data mining based sensor fault diagnosis and validation for building air conditioning system," Energy Conversion and Management, vol. 47, no. 15-16, pp. 2479-2490, 2006.

[39] J. Schein, S. T. Bushby, N. S. Castro and J. M. House, "A rule-based fault detection method for air handling units," Energy and Buildings, vol. 38, no. 12, pp. 1485-1492, 2006.

[40] W. H. Woodall, D. J. Spitzner, D. C. Montgomery and S. Gupta, "Using control charts to monitor process and product quality profile," Journal of Quality Technology, vol. 36, no. 3, pp. 309-320, 2004

[41] A. Mitra, Fundamentals of Quality Control and Improvement, 2nd ed., New Jersey: Prentice Hall, Upper Saddle River, 1998.

[42] D. C. Montgomery, Introduction to Statistical Quality Control, 5th ed., New York: John Wiley, 2005.

[43] L. Kang and S. L. Albin, "On-line monitoring when the process yields a linear profile," Journal of Quality Technology, vol. 32, no. 4, pp. 418-426, 2000.

[44] O. Mestek, J. Pavlik and M. Suchanek, "Multivariate control chart: control charts for calibration curves," Journal of Analytical Chemistry, vol. 350, no. 6, pp. 344-351, 1994.

[45] D. J. Wheeler, Understanding Variation: The Key to Managing Chaos, 2nd ed., Knoxville, TN: SPC Press, 2000.

[46] G. Casella and R. Berger, Statistical Inference, 2nd ed., Pacific Grove, CA: Brooks/Cole, 1990.

[47] C. M. Bishop, Neural Networks for Pattern Recognition, New York: Oxford University Press, 1995.

[48] P. Seidel, A. Seidel and O. Herbarth, "Multilayer perceptron tumor diagnosis based on chromatography analysis of urinary nucleoside," Neural Networks, vol. 20, no. 5, pp. 646-651, 2007.

[49] J. Espinosa, J. Vandewalle and V. Wertz, Fuzzy Logic, Identification and Predictive Control. London, UK: Springer-Verlag, 2005.

[50] A. J. Smola and B. Schoelkopf, "A tutorial on support vector regression," Statistics and Computing, vol. 14, no. 3, pp. 199-222, 2004.

[51] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya and K.R.K. Murthy, "Improvements to the SMO Algorithm for SVM Regression," IEEE Transactions on Neural Networks, vol. 11, pp. 1188-1193, 2000.

[52] P.N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Boston, MA: Pearson Education/Addison Wesley, 2006.

[53] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed., San Francisco, CA: Morgan Kaufmann, 2005.

[54] American Society of Heating, Refrigerating and Air-Conditioning Engineers. ANSI/ASHRAE 90.1-2001, Energy standard for buildings except low-rise residential buildings. SI Edition, Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2001.

[55] American Society of Heating, Refrigerating and Air-Conditioning Engineers. ANSI/ASHRAE 62-2004, Design for acceptable indoor air quality, Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2004.

[56] K. W. Mui and W. T. Chan, "Building calibration for IAQ management," Building and Environment, Vol. 41, No.7, pp. 877-886, 2006.

[57] B. L. Capehart, Encyclopedia of Energy Engineering and Technology, Boca Raton, FL: CRC Press, 2007.

[58] S. J. Emmerich, "Literature review on CO2-based demand-controlled ventilation," ASHRAE Transactions, Vol. 103, No.2, pp. 229-243, 1997.

[59] C. C. Federspiel, "On-demand ventilation control: A new approach to demand-controlled ventilation," Proceedings of INDOOR AIR '96, Vol. 3, 935-940, 1996.

[60] Y. P. Ke, S.A. Mumma, "Using carbon dioxide measurements to determine occupancy for ventilation controls, " ASHRAE Transactions, Vol. 103, No.2, pp. 365-374, 1997.

[61] M. M. Gouda, "Fuzzy ventilation control for zone temperature and relative humidity," 2005 American Control Conference, pp. 507-512, 2005.

[62] Y. P. Ke and S. A. Mumma, "Optimized supply-air temperature in variable-air-volume systems," Energy, Vol. 22, No. 6, pp. 601-614, 1997.

[63] Y. C. Chang and W. H. Chen, "Optimal chilled water temperature calculation of multiple chiller systems using Hopfield neural network for saving energy," Energy, Vol. 34, No. 4, pp. 448-456, 2009.

[64] G. R. Zheng and M. Zaheer-Uddin, "Optimization of thermal processes in a variable air volume HVAC system ," Energy, Vol. 21, No. 5, pp. 407-420, 1996.

[65] M. Mossolly, K. Ghali and N. Ghaddar, "Optimal control strategy for a multi-zone air conditioning system using a genetic algorithm," Energy, Vol. 34, No. 1 pp. 58-66, 2009.

[66] Y. C. Chang, "An innovative approach for demand side management—optimal chiller loading by simulated annealing," Energy, Vol. 31, No. 12, pp. 1883-1896, 2006.

[67] M. A. Abido, “Multiobjective evolutionary algorithms for electric power dispatch problem,” IEEE Transaction on Evolutoinary Computation, Vol. 10, No. 3, pp. 315-329, 2006.

[68] C. M. Fonseca and P. J. Fleming, “An overview of evolutionary algorithms in multiobjective optimization,” Evolutionary Computation, Vol. 3, No. 1, pp. 1-16, 1995.

[69] C. A. C. Coello and A. D. Christiansen, “MOSES: A multiobjective optimization tool for engineering design,” Engineering Optimization, Vol. 31, No. 3, pp. 337-368, 1999.

[70] J. D. Schaffer, “Multiple objective optimization with vector evaluated genetic algorithms,” in Proc. Int. Conf. Genetic Algorithms and Their Application, Pittsburgh, PA, Jul. 24-26, 1985, pp. 93-100.

[71] J. Lis and A. E. Eiben, “A multi-sexual genetic algorithm for multi-objective optimizatoin,” in Proc. IEEE 1996 Int. Conf. Evol. Comput., Nagoya, Japan, 1996, pp. 59-64.

[72] C. A. C. Coello, F. S. Hernandez, and F. A. Farrera, “Optimal design of reinforce concrete beams using genetic algorithms,” Int. J. Exp. Syst. Appl., Vol. 12, No. 1, pp. 101-108, 1997.

[73] J. Horn, N. Nafpliotis, and D. E. Goldberg, “A niched Pareto genetic algorithm for multiobjective optimization,” in Proc. 1st IEEE Conf. Evol. Comput., IEEE World Congr. Comput. Intell., Vol. 1, 1994, pp. 67-72

[74] N. Srinivas and K. Deb, “Multiobjective function optimization using nondominated sorting genetic algorithms,” Evolutionary Computation, Vol. 2, No. 3, pp. 221-248, 1994.

[75] E. Zitzler and L. Thiele, “An evolutionary algorithm for multiobjective optimization: The strength Pareto approach,” TIK-Rep. 43, 1998.

[76] M. A. Abido, “Two-level of nondominated solutions approach to multiobjective particle swarm optimization,” Genetic And Evolutionary Computation Conference 2007,pp. 726-733

[77] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, 1st Ed., Chichester, England: John Wiley, 2001.

[78] E. Zitzler and L. Thiele, "Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach", IEEE Transactions on Evolutionary Computation, Vol. 3, No.4, pp. 257-271, 1999.

[79] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing, Berlin; New York: Springer, 2003.

[80] A. Persily and W. S. Dols, "The relation of CO2 concentration to office building ventilation," ASTM Standard Technical Publication, 1997.

[81] W.J. Fisk, and A.T. De Almeida, "Sensor-based demand-controlled ventilation: A Review, " Energy and Buildings, Vol. 29, No.1, pp. 35-45, 1998.

[82] W. J. Fisk, “Health and productivity gains from better indoor environments and their relationship with building energy efficiency”, Annual Review of Energy and the Environment, Vol. 25, No.1, pp. 537-566, 2000.

[83] I. N. Potter and W. B. Booth, CO2 Controlled Mechanical Ventilation Systems, Technical Note 12/94.1. BSRIA; 1994.

[84] N. Nassif and S. Moujaes, "A cost-effective operating strategy to reduce energy consumption in a HVAC system," International Journal of Energy Research, Vol. 32, No.6, pp. 543-558, 2008.

[85] L. Lu, W. Cai, L. Xie, S. Li, and Y. C. Soh, "HVAC system optimization—in-building section," Energy and Buildings, Vol. 37, No.1, pp. 11-22, 2005.

[86] S. Ari, H.E. Khalifa, J.F. Dannenhoffer, P. Wilcoxen and C. Isik, "Fuzzy logic and neural network approximation to indoor comfort and energy optimization", IEEE NAFIPS Conference, pp.692-695, 2006.

[87] M. Kintner-Meyer and A.F. Emery, "Optimal control of an HVAC system using cold storage and building thermal capacitance", Energy and Buildings, Vol. 23, No.1, pp. 19-31, 1995.

[88] B. H. Gebreslassie, G. Guillen-Gosalbaz, L. Jimenez and D. Boer, " Design of environmentally conscious absorption cooling systems via multi-objective optimization and life cycle assessment", Applied Energy, Available online 25 December 2008.

[89] E. H. Mathews, D. C. Arndt, C. B. Piani and E. van Heerdan, "Developing cost efficient control strategies to ensure optimal energy use and sufficient indoor comfort", Applied Energy, Vol. 66, No.2, pp. 135-159, 2000.

[90] J. F. Kreider and A. Rabl, Heating and Cooling of Buildings: Design for Efficiency, 2nd Ed., New York: McGraw-Hill, 2002.

[91] Y. C. Chang, J. K. Lin and M. H. Chuang, "Optimal chiller loading by genetic algorithm for reducing energy consumption," Energy and Buildings, Vol. 37, No.2, pp. 147-155, 2005.

[92] M. Nadeem, "Evaluation of overall chiller performance characteristics," Air Conditioning and Refrigeration Journal, No.7-9, 2001.

[93] F. W. Yu and K. T. Chan, "Part load performance of air-cooled centrifugal chillers with variable speed condenser fan control," Building and Environment, Vol. 42, No.11, pp. 3816-3829, 2007.

[94] Z. Song and A. Kusiak, "Optimization of temporal processes: a model predictive control approach," IEEE Transactions on Evolutionary Computation, DOI: 10.1109/TEVC.2008.920680.

[95] J. Wang, Data Mining: Opportunities and Challenges, Idea Group Inc, 2003.

[96] R. Kohavi and G.H. John, "Wrappers for Feature Subset Selection", Artificial Intelligence, Vol. 97, No. 1-2, pp. 273-324, 1997.

[97] J. Friedman, "Stochastic gradient boosting," Stanford University Statistics Department, 1999.

[98] T. Hastie, R. Tibshirani and J.H. Firedman, The Elements of Statistical Learning, New York: Springer, 2001.

[99] J. A. McDonald and F. H. Sheehan, "Ventriculogram segmentation using boosted decision trees", Proceedings of SPIE Medical Imaging, Vol. 5370, pp. 1804-1814, 2004.

[100] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," The Annals of Statistics, Vol. 29, No.5, pp. 1189-1232, 2001.

[101] L. Breiman, "Random Forests," Machine Learning, Vol. 45, No.1, pp. 5-32, 2001.

[102] V. N. Vapnik, The Nature of Statistical Learning Theory, New York, Springer, 2000.

[103] L. Lu, W. Cai, Y. S. Chai and L. Xie, "Global optimization for overall HVAC systems—Part I problem formulation and analysis", Energy Conversion & Management, Vol. 46, No.7-8, pp. 999-1014, 2005.

[104] I. H. Yang, M. S. Yeo and K. W. Kin, "Application of artificial neural network to predict the optimal start time for heating system in building", Energy Conversion & Management, Vol. 44, No.17, pp. 2791-2809, 2003.

[105] X. C. Xi, A. N. Poo and S. K. Chou, "Support vector regression model predictive control on a HVAC plant", Control Engineering Practice, Vol. 15, No.8, pp. 897-908, 2007.

[106] K. F. Fong, V. I. Hanby and T. T. Chow, "System optimization for HVAC energy management using the robust evolutionary algorithm", Applied Thermal Engineering, Vol. 29, No.11-12, pp. 2327-2334, 2009.

[107] E. F. Camacho and C. Bordons, Model Predictive Control, London: Springer, 2004.

[108] R. Findeisen, F. Allgower and L. T. Biegler, Assessment and Future Directions of Nonlinear Model Predictive Control, NY: Springer-Verlag, 2007.

[109] M. A. Abido, “Optimal Design of Power System Stabilizers: Using Particle Swarm Optimization,” IEEE Trans. on Energy Conversion, Vol. 17, No. 3, September 2002, pp. 406-413.

[110] M. P. Wachowiak, R. Smolíková, Y. Zheng, J. M. Zurada, and A. S. Elmaghraby, “An Approach to Multimodal Biomedical Image Registration Utilizing Particle Swarm Optimization,” IEEE Transactions on Evolutionary Computation, Vol. 8, No. 3, June 2004, pp. 289-301.

[111] J. A. Hertz, A. Krogh and R. G. Palmer, Introduction to the Theory of Neural Computation, Boulder, CO: Westview Press, 1999.

[112] S. Haykin, Neural Networks: A Comprehensive Foundation, Englewood Cliffs, NJ: Prentice Hall, 1998.