Adaptive Operation Decisions for a System of Smart Buildings

by

Mengqi Hu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2012 by the
Graduate Supervisory Committee:

Teresa Wu, Co-Chair
Jeffery Weir, Co-Chair
Jin Wen
John Fowler
Dan Shunk

ARIZONA STATE UNIVERSITY

August 2012

ABSTRACT

Buildings (approximately half commercial and half residential) consume over 70% of the electricity among all the consumption units in the United States. Buildings are also responsible for approximately 40% of CO_2 emissions, which is more than any other industry sectors. As a result, the initiative smart building which aims to not only manage electrical consumption in an efficient way but also reduce the damaging effect of greenhouse gases on the environment has been launched. Another important technology being promoted by government agencies is the smart grid which manages energy usage across a wide range of buildings in an effort to reduce cost and increase reliability and transparency. As a great amount of efforts have been devoted to these two initiatives by either exploring the smart grid designs or developing technologies for smart buildings, the research studying how the smart buildings and smart grid coordinate thus more efficiently use the energy is currently lacking.

In this dissertation, a “system-of-system” approach is employed to develop an integrated building model which consists a number of buildings (building cluster) interacting with smart grid. The buildings can function as both energy consumption unit as well as energy generation/storage unit. Memetic Algorithm (MA) and Particle Swarm Optimization (PSO) based decision framework are developed for building operation decisions. In addition, Particle Filter (PF) is explored as a mean for fusing online sensor and meter data so adaptive decision could be made in responding to dynamic environment. The dissertation is divided into three inter-connected research components. First, an integrated building energy model including building consumption, storage, generation sub-systems for the building cluster is developed. Then a bi-level Memetic Algorithm (MA) based decentralized decision framework is developed to identify the Pareto optimal operation strategies for the building cluster. The Pareto solutions not only enable multiple dimensional tradeoff analysis, but also provide valuable insight for determining pricing mechanisms and power grid capacity. Secondly, a multi-objective PSO based decision framework is developed to reduce the computational effort of the MA based decision framework without scarifying accuracy. With the improved performance, the decision time scale could be refined to make it capable for hourly operation decisions. Finally, by integrating the multi-objective PSO based decision framework with PF, an adaptive framework is developed for adaptive operation decisions for smart building cluster. The adaptive framework not only enables me to develop a high fidelity decision model but also enables the building cluster to respond to the dynamics and uncertainties inherent in the system.

DEDICATION

I dedicate this dissertation to my lovely dad, mom and Qing.

ACKNOWLEDGMENTS

I want to thank my advisors – Dr. Teresa Wu and Dr. Jeffery Weir for their continuing mentoring, guidance and support through the past four years. Not only being patient and inspiring, they always create great research opportunities for me. Their caring for students, dedictation to research, hard-working spirit and optimistic attitude will keep encouraging me in future career development.

Thanks go as well to Dr. Jin Wen, Dr. John Fowler, Dr. Dan Shunk, who gave me valuable advices during comprehensive exam and when I have research questions. I also want to thank Dr. William Pavlicek for being an encouraging supervisor during my internship at Mayo Clinic and for being an exampler of excellent leader, lecturer and scientist.

I would also like to thank my friends and lab mates, especially Shanshan Wang, Houtao Deng, and Min Zhang, with whom that I spend good time and am always amazed with their talents and kindness. I am grateful to have met so many wonderful people since I came to Arizona.

Last but not least, I give my most special thanks to Qing Huang, for her endless love, understanding and help throughout my Ph.D. study.

TABLE OF CONTENTS

Page

CHAPTER Page

Appendix

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

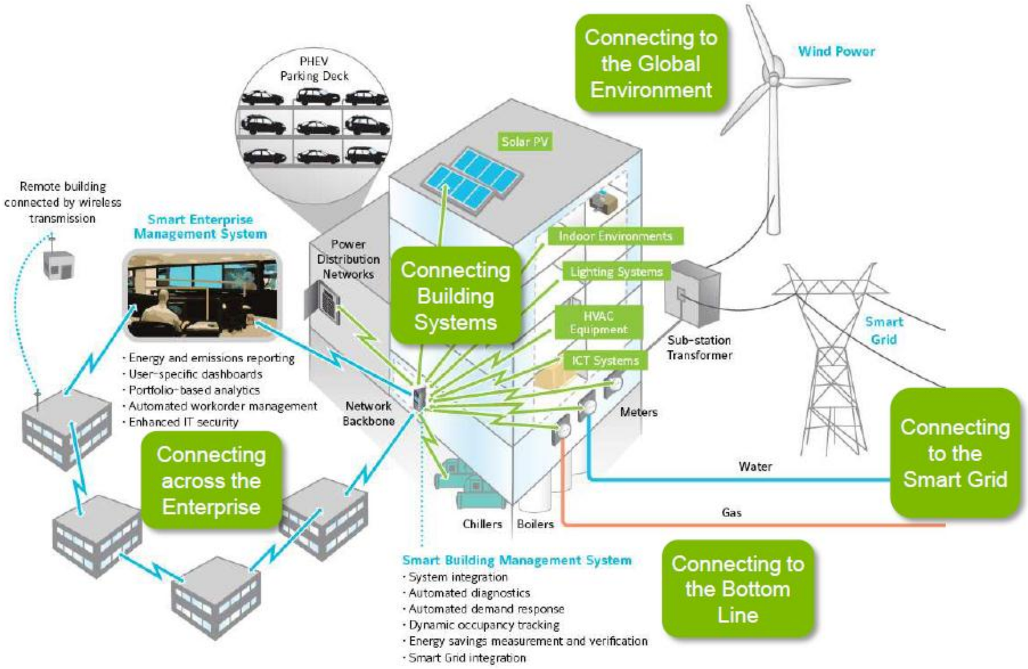
INTRODUCTION

1.1 Motivation

In the United States, buildings are responsible for over 70% of electricity consumption - approximately half commercial and half residential (Friedman, 2009). According to McKinsey Global Institute, the largest opportunities for saving are in the residential sector. The fact is between 4 and 20% of energy used for Heating, Ventilating and Air Conditioning (HVAC), lighting and refrigeration in building is wasted due to the problems with system operation. Therefore, a new concept, net-zero energy (smart) building aiming to reduce building primary energy consumption is being promoted by the United States Department of Energy. Net-zero energy (smart) buildings refer to the buildings that generate as much energy as they consume through efficient technologies and on-site power generation. Legislation was recently made establishing targets for all commercial buildings for net-zero energy by 2050. As a result, increasing numbers of buildings have adopted on-site energy generation devices and increasingly complex sensor and control systems.

Another important technology that is being promoted by many government agencies is smart grid technology, which is a network of computers and power infrastructures that monitor and manage energy usage, reduce cost and increase reliability and transparency. The smart grid technology movement provides the infrastructure (two-way flow of electricity and information) for distributed management of power distribution systems, allowing buildings to be more interactive with the power grid.

A significant amount of available and alternative energy technologies can be directly used to develop smart buildings and implement smart grid concepts. Both initiatives urge the building industry to improve their energy efficiency and to have better capabilities to interact with the power grid, and drive research moving from centralized operation decisions on a single building to decentralized decisions on *a system of buildings*, termed *building cluster* which shares energy resources locally and globally. Traditionally, buildings have been viewed as mere energy consumers with no negotiation power for determining energy price. Today, with the new power grid infrastructure and distributed energy resources, buildings can not only consume energy, but they can also output energy. With this as my motivation, this dissertation anticipates that next generation building systems (Figure 1) will be able to utilize smart grid for the exchange of information and to freely form cluster, amongst which buildings (even small residential buildings) can share and exchange site-generated energy. This cluster will thereby be more resilient to power disturbances, reduce energy cost and energy consumption, and improve energy efficiency and environment sustainability.



Connecting to the Global Environment
Wind Power
PHEV Parking Deck
Solar PV
Remote building connected by wireless transmission
Smart Enterprise Management System
Power Distribution Networks
Connecting Building Systems
Indoor Environments
Lighting Systems
HVAC Equipment
ICT Systems
Sub-station Transformer
Smart Grid
· Energy and emissions reporting
· User-specific dashboards
· Portfolio-based analytics
· Automated workorder management
· Enhanced IT security
Network Backbone
Meters
Connecting to the Smart Grid
Water
Connecting across the Enterprise
Gas
Chillers
Boilers
Smart Building Management System
· System integration
· Automated diagnostics
· Automated demand response
· Dynamic occupancy tracking
· Energy savings measurement and verification
· Smart Grid integration
Connecting to the Bottom Line

Figure 1 System architecture for next generation building systems (adopted from (Perez & Farnham, 2010))

1.2 Research Overview

The objective of this dissertation is to develop an adaptive decision framework to derive adaptive operation decisions for the next generation building systems to guarantee the buildings could quickly respond to the dynamic environment and reduce energy cost. In order to achieve these goals, several critical research issues have been identified in this dissertation.

Firstly, most existing literature focuses on operation strategies for one subsystem only, that is, HVAC, energy storage or energy generation (see Chapter 2). Considering a building is an integrated system as a whole, studying the interactions among the subsystems is necessary. So the first challenge issue is:

1) How to develop an integrated building energy model including building consumption, storage, generation sub-systems for the building cluster to make the decentralized decision making possible?

In this dissertation, the using of agent based simulation is explored to study the important aspects of a building system which including building consumption, storage and generation sub-systems.

Nowadays, commercial buildings are increasingly using sophisticated energy management and control systems (EMCSs) to monitor and control building systems, yet building systems routinely fail to perform as designed (Hicks & von Neida, 2000). Although the EMCSs are sophisticated, they lack the tools necessary to detect and diagnose faults arising in building systems. Furthermore, building designers and operators generally overlook the symptoms because of a lack of proper understanding of the operation strategies and the symptoms related to system failures. This leads to the manual override of operation strategies and gradual erosion of proper system performance. Research has demonstrated that energy and operation costs can be greatly reduced by adopting good operation strategies, which utilize building passive and active thermal storages, demand response measures, and the dynamic relationship between sub-systems from a whole-building point of view (see Chapter 2). The significant cost savings drive research to explore ways to develop appropriate operation strategy. Some related challenging questions are:

2) Considering each building consists of energy consumption, storage, generation sub-systems, when to charge or discharge the storage system or leave it as dormant? What is the optimal strategy for a generation system (e.g., power the building vs. charge its storage system vs. sell back to the power grid)? How do the buildings adjust their HVAC set-point temperature and coordinate with each other on the shared energy splitting strategy to reach the win-win goal?

To address these questions, a bi-level decision framework for building cluster operation decisions is developed. A Memetic Algorithm (MA) based model (Hu et al., 2012) is employed to identify the Pareto optimal operation strategies for the building cluster. However, the MA based framework cannot be applied to derive hourly operation decisions for the building energy system due to its computational issues. So, another research challenging is:

3) How to improve the computational performance of the decentralized decision framework to ensure it is capable to make hourly operation decisions without losing solution accuracy?

Particle Swarm Optimization (PSO) which is capable of deriving good results with very low computational cost is employed to improve the performance of the decentralized decision framework (Reyes-Sierra & Coello Coello, 2006). In this dissertation, two novel particle swarm optimization algorithms are developed to improve PSO’s performance on a diverse set of optimization problems with different properties, and improve PSO's robustness to its parameter settings. Then, a multi-objective PSO algorithm is developed based on these two novel PSO algorithms to improve the computational performance of the decentralized decision framework, and study decentralized operation decisions for the building cluster.

Due to the dynamics and complexity of the building energy system, a good operation strategy first requires an accurate model for building system energy usage which is currently lacking. The optimal operation decisions should be made in responding to the dynamics and uncertainties inherent in the system. Due to the variety and diversity of uncertainties and noises in the building system, environment, sensor and meter, less research is conducted on developing building operation strategies under uncertainty and noise though the importance of this research topic has long existing. The research question is:

4) What is the appropriate approach to handle uncertainties and noises existing in the building, environment, sensors and meters?

This dissertation considers the uncertainties in building energy consumption, temperature and solar radiation, noises in sensors and meters, and employs the Gaussian mixture sigma point particle filter (GMSPPF) (van der Merwe, 2004) algorithm to calibrate the building cluster model with noise measurement data collected from sensors and meters. The GMSPPF is integrated with the multi-objective PSO based decentralized decision framework to derive adaptive operation decisions for the building cluster to guarantee the building cluster could quickly respond to its dynamic environment.

1.3 Research Contributions

This section summarizes all the original research contributions this dissertation has achieved by addressing all the research issues stated in section 1.2.

1) An agent based building cluster model which is less studied in the existing literature is developed. This building cluster model enables me to study the interactions between multiple buildings, multiple subsystems, shared energy resources, and power grid.

2) A bi-level decentralized decision framework is developed and applied for deriving operation decisions for the building cluster model. The derived Pareto operation decisions are able to reduce energy cost and consumption, improve energy efficiency and environment sustainability, and make the buildings be more resilient to power disturbances.

3) A computationally efficient particle swarm optimization (PSO) is developed. The newly developed PSO algorithm performs well on a diverse set of optimization problems with different properties. In addition, a generalized intelligent multiple search methods selection strategy which could be used to assess multiple search methods is developed. Furthermore, two adaptive optimization techniques (e.g., adaptive sub-gradient method and Cauchy mutation operator) based on particle’s velocity information are developed.

4) An adaptive parameter tuning mechanism is developed to improve particle swarm optimization’s robustness to the parameter settings which is a common issue for most of the existing PSO algorithms. The parameter tuning mechanism could be plugged into other PSO algorithms to improve their robustness.

5) A novel multi-objective particle swarm optimization algorithm is developed which is demonstrated to outperform some of the existing multi-objective PSO algorithms and multi-objective evolutionary algorithms.

6) A multi-objective PSO based decentralized decision framework is developed which could improve the computational performance of the decentralized decision

framework without losing of solution quality. The hourly operation decisions derived from this computationally efficient framework could tremendously reduce energy cost and consumption.

7) An adaptive decision framework which integrates the Gaussian mixture sigma point particle filter algorithm and multi-objective PSO based decision framework is developed to study adaptive operation decisions for the dynamic building energy systems. The adaptive decision framework could accurately calibrate the building model, and the adaptive operation decisions derived from this framework enable buildings to quickly respond to the dynamic environment.

Finally, the adaptive decision framework can be applied not only in building energy system, but also for a wide range of complex dynamic systems decision support, such as healthcare delivery management and disease control.

1.4 Dissertation Organization

The interconnection of the remainder chapters and questions to be answered are shown in Figure 2.

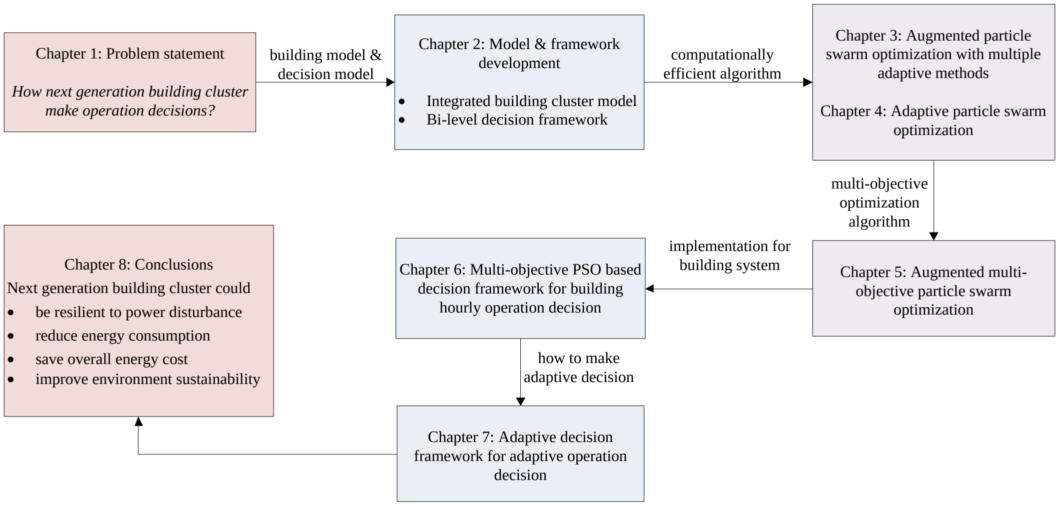

Chapter 1: Problem statement
How next generation building cluster make operation decisions?
building model & decision model
Chapter 2: Model & framework development
• Integrated building cluster model
• Bi-level decision framework
computationally efficient algorithm
Chapter 3: Augmented particle swarm optimization with multiple adaptive methods
Chapter 4: Adaptive particle swarm optimization
multi-objective optimization algorithm
Chapter 5: Augmented multi-objective particle swarm optimization
implementation for building system
Chapter 6: Multi-objective PSO based decision framework for building hourly operation decision
how to make adaptive decision
Chapter 7: Adaptive decision framework for adaptive operation decision
Chapter 8: Conclusions
Next generation building cluster could
• be resilient to power disturbance
• reduce energy consumption
• save overall energy cost
• improve environment sustainability

Figure 2 Dissertation organization

The remainder of this dissertation is organized as follows:

Chapter 2 presents the agent based building cluster model and develops a bi-level decentralized decision framework. This chapter also includes the literature reviews for most of the existing research in the building energy system.

Chapter 3 reviews most of the existing particle swarm optimization algorithms and develops a computationally efficient particle swarm optimization algorithm which is termed as PSO-MAM. PSO-MAM is demonstrated to significantly outperform 10 published PSO algorithms on at least 36 out of 43 test functions.

Chapter 4 studies how to adaptively change parameter settings of the PSO algorithm to improve its robustness. Most of the existing parameter tuning techniques is reviewed in this chapter. The adaptive PSO algorithm which is termed as BLOSSM-APSO is demonstrated to be able to improve the robustness of the PSO algorithm and make it insensitive to its parameter settings.

Chapter 5 extends the single objective PSO algorithms studied in Chapter 3 and 4 to a multi-objective optimization algorithm. The developed multi-objective PSO algorithm which is termed as AMOPSO is demonstrated to significantly outperform the 7 out of 8 published multi-objective optimization techniques.

Chapter 6 improves the computational performance of the decentralized decision framework by using AMOPSO presented in Chapter 5. The AMOPSO based decision framework is able to derive hourly operation decisions, and the operation decisions could reduce energy cost and consumption.

Chapter 7 develops an adaptive decision framework by integrating the Gaussian mixture sigma point particle filter (GMSPPF) algorithm with the AMOPSO based decision framework presented in Chapter 6. GMSPPF is demonstrated to be able to accurately calibrate the building cluster model and buildings could respond to the dynamic environment by using the adaptive decision framework.

Chapter 8 summarizes the research works in this dissertation and points out some future directions in both the application and algorithm development research.

Chapter 2

DECENTRALIZED OPERATION STRATEGIES FOR AN INTEGRATED BUILDING ENERGY SYSTEM USING A MEMETIC ALGORITHM

The emerging technologies in smart building and smart grids drive research moving from centralized operation decisions on a single building to decentralized decisions on a system of buildings, termed a building cluster which shares energy resources locally and globally. However, current research has focused on developing an accurate simulation of single building energy usage which limits its application to building cluster as scenarios such as energy sharing and competition cannot be modeled and studied. This chapter hypothesizes that the study of energy usage for a group of buildings instead of one single building will result in a cost effective building system which in turn will be resilient to power disruption. To this end, this chapter develops a decision model based on a building cluster simulator with each building modeled by energy consumption, storage and generation sub modules. Assuming each building is interested in minimizing its energy cost, a bi-level operation decision framework based on a Memetic algorithm is developed to study the tradeoff in energy usage among the group of buildings. One additional metric, measuring the degree of dependencies on the power grid is introduced for the analysis. The experimental result demonstrates that the bi-level decision framework is capable of deriving the Pareto solutions for the building cluster in a decentralized manner. The Pareto solutions not only enable multiple dimensional tradeoff analysis, but also provide valuable insight for determining pricing mechanisms and power grid capacity.

2.1 Introduction

According to the Electric Power Research Institute (EPRI), the electricity consumption of the U.S. grew 1.7% annually from 1996 to 2006 with the expectation of total growth through 2030 being 26% (Parks, 2009). Buildings (approximately half commercial and half residential) consume over 70% of the electricity among all the consumption units (Friedman, 2009). The fact is between 4 and 20% of energy used for heating, ventilating and air conditioning (HVAC), lighting and refrigeration in buildings is wasted due to problems with system operation. Therefore, extensive researches in the past two decades have explored optimal operation strategies including on-site generation and storage for smart buildings to reduce energy cost and improve energy efficiency for building systems.

Research on HVAC has employed simulation and mathematical modeling for optimal strategies. For example, Fong et al. (2006) develop a simulation-evolutionary programming coupled approach to optimize the HVAC control which demonstrates a 7% cost savings compared with the existing control methods. Lu et al. (2005) formulate a mixed integer nonlinear programming problem and apply a modified genetic algorithm to derive the HVAC system optimal control strategy which significantly improves the HVAC performance. Wright et al. (2002) and Nassif et al. (2005) study the multi-objective genetic algorithm to optimize building thermal control and HVAC control aiming to minimize energy cost and maximize zone thermal comfort.

Other than HVAC, recent literature indicates the use of energy storage such as thermal mass control strategies can alleviate the energy load and thus potentially reduce the energy cost (Braun, 2003). As an example, Keeney and Braun (1996) successfully demonstrate pre-cooling of a building can reduce the peaking cooling load, electricity demand and energy cost. Hämäläinen and Mäntysaari (2002) employ dynamic goal programming to study the tradeoff between energy cost, energy consumption and living comfort for the residential house heating system. Braun (2007) and Sun et al. (2006) further develop a heuristic near-optimal control strategy for thermal storage systems with dynamic real-time electric rates. The results indicate that the annual cost under this control method is close to optimum (less than 2% higher than the minimal costs). Drees and Braun (1996) present a rule-based control strategy for a thermal storage system which outperforms the conventional control strategy such as chiller-priority and storage-priority strategies. The monthly electrical cost is near-optimal (less than 3% higher than optimum obtained from dynamic programming). Henze et al. (2005) further develop model-based predictive optimal control of active and passive building thermal storage and successfully achieve 18% and 7% of cost savings compared to the reference case and base case which are two testing cases studied in (Henze et al., 2005). Henze and Schoenmann (2003), Liu and Henze (2007) demonstrate that the model-free reinforcement learning control for the thermal storage system can achieve more cost savings than the conventional storage control strategies but less than the predictive optimal control strategies. In addition, Liu and Henze (2006a, 2006b) develop a hybrid reinforcement learning control approach combining model-based with model-free control to locate the optimal control for passive and active thermal storage which can achieve 8.3% cost savings compared to the base case studied in (Liu & Henze, 2006b). Lee et al. (2009) employ the particle swarm algorithm to optimize operations of the ice-storage air-conditioning system which can minimize the life cycle cost and reduce the CO_2 emission.

Another noteworthy emerging effort is energy generation and the use of energy generation on site. For example, Manolakos et al. (2001) develop a simulation-optimization program to design and control a hybrid energy system which consists of a battery, wind generator and photovoltaic module. Rong et al. (2008b) develop an efficient and near-optimal planning strategy for the tri-generation system which includes an electric power, heat, cooling and storage system using a Lagrangian relaxation based algorithm. García-González et al. (2007) optimize the short-term scheduling which is formulated as a mixed integer linear programming problem for hydroelectric generation units. Arun et al. (2009) adopt chance constraint programming to optimize the design of a battery-integrated diesel generator system and identify the optimum configuration. El-shatter et al. (2006) design a fuzzy logic control based management system to improve the energy efficiency for a hybrid wind/photovoltaic/fuel cell generation system. Henze and Dodier (2003) develop a model-free reinforcement learning algorithm which outperforms the conventional control strategy to adaptively control a grid-independent photovoltaic system which has a collector, storage and load. Rong and Lahdelma (2007) develop an envelope-based branch and bound algorithm to derive the long-term planning strategy for single-period combined heat and power system. Rong et al. (2008a) further study the multi-period combined heat and power system planning using a modified dynamic programming approach.

While promising, most literature focuses on operation strategies for one subsystem only, that is, HVAC, energy storage or energy generation. Considering a building is an integrated system as a whole, studying the interactions among the subsystems is necessary. Secondly, even though there exists research exploring a building as a system consisting of subsystems, a majority of the research formulates the decision problem for a single building only. Realizing the emerging technologies in multi-energy source building (Corrado et al., 2007), net-zero building (Torcellini et al., 2006) and smart grid (Parks, 2009), it is becoming urgent critical to develop a decentralized decision framework modeling the coordination among a cluster of buildings to obtain Pareto decisions which enable tradeoff analysis. There are notable efforts taken in this direction. For example, Kiesling (2009) investigates the decentralized coordination mechanism to increase energy efficiency through markets, technology and institutions. However, to my knowledge, decisions for buildings consisting of multiple interacting subsystems, which coordinate with other buildings and the power grid has been less explored. This is probably due to the complexity of the problem which involves both subsystems, and cluster of buildings. An even further level of complexity is the varied time scales, ranging from models running based on minutes (e.g., energy consumption subsystem), to hourly and possibly even daily. Given the complexities discussed above, this chapter demonstrates a methodology for modeling the coordination among a cluster of buildings. Specifically, with this methodology, decision makers can determine when to charge or discharge the storage system or leave it dormant. They can determine an optimal strategy for a generation system (e.g., power the building vs. charge its storage system vs. sell back to the power grid). Given competing owners in multiple buildings, HVAC set-point temperature strategies can be coordinated with each other on the shared energy splitting to reach a win-win goal. Finally, decision makers can use the methodology to determine how local energy pricing levels and power grid capacity can influence the operation of a building cluster.

This chapter extends the agent-based simulator developed by Hu et al. (2010), and develops a bi-level decision framework for building cluster operation decisions. A Memetic algorithm (MA) based model is employed to identify the Pareto optimal control strategies for the building cluster. The power grid dependency rate (PGDR) is utilized to evaluate the Pareto operation strategies. In addition, different pricing mechanisms and power capacities are studied to demonstrate the impacts on energy costs for the group of buildings.

This chapter is organized as follows: the building energy model is introduced in section 2.2; the decision model is formulated in section 2.3 followed by the detailed explanation on the bi-level decentralized framework in section 2.4; the MA based framework is discussed in section 2.5; the experimental results in section 2.6 demonstrate how the bi-level decision framework can be used for decentralized decision making. Finally, conclusions are drawn in section 2.7.

2.2 Integrated Building Energy System Simulator

Kosny et al. (2001) and Zhou et al. (2005) employ thermal mass concept which determines the building's capability to utilize its structural mass for thermal storage to differentiate heavy and light weight/mass buildings. The thermal mass can be modified by changing either the thickness or density of the wall material without altering the architectural and construction of the building model (Zhou et al., 2005). In this chapter, the density of the wall material is changed (Figure 3) to distinguish heavy and light mass buildings.

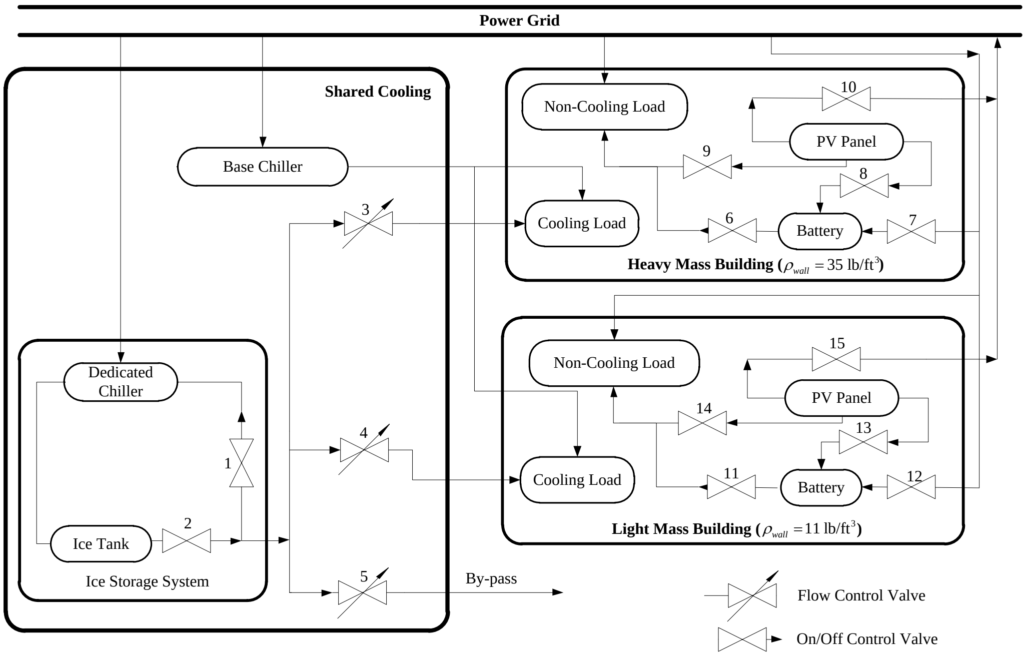


Figure 3 Overall schematic of the integrated building energy system

Table 1 Building system development

Modules/Parameters	Source
Building Consumption Model (module)	The building cooling load is from a building thermal model. Non-cooling load data from (Valenzuela et al., 2000) has been appropriately scaled down for this chapter.
Chiller Model (module)	The chiller model is adopted from (Sun et al., 2006).

Ice Storage Model (module)	The ice storage model is from (West & Braun, 1999). The parameters of the ice tank have been appropriately scaled down to make it applicable for this chapter.
Battery Model (module)	The battery model is from (Lu, 2004).
Photovoltaic Model (module)	The PV-panel model is from (Lu & Yang, 2004). The angular losses factor in (Martin & Ruiz, 2001) is employed to compute the absorbed solar energy.
Dry/Wet Bulb Temperature (parameter)	The hourly dry/wet bulb temperature T_{db} (℉), T_{wb} (℉) in Phoenix are obtained from (NCDC, 2010)
Solar Radiation on Inclined Surfaces (parameter)	The total hourly solar irradiance on the inclined surface with slope angle β (degree) and surface azimuth angle Υ (degree) is estimated using the model from (Lorenzo, 2003; Lu, 2004; NREL, 2010)
Real-time Pricing Rate (parameter)	Three pricing plans used by Salt River Project (SRP) Company (SRP, 2010) are considered in this chapter.

A simplified building cluster consisting of two different mass level - heavy mass (HM) and light mass (LM) buildings is then modeled. The two buildings, each having its own battery and photovoltaic (PV) panel, share one ice storage system and one base chiller. The ice storage system charged by a dedicated chiller is configured in parallel with the base chiller. During on-peak hours, the buildings cooling loads are met primarily by the ice storage system with the remaining cooling request satisfied by the base chiller. The overall schematic of the building energy system configuration is illustrated in Figure 3 with the arrows denoting the energy flow among each component in the system.

In this chapter, the PV panel for each building can be in only one of the following four states: charging battery, powering building, selling power to grid or being dormant. It is assumed that the extra electricity of the PV panel will be wasted when the PV panel is at the state of charging battery or powering building. If the electricity generated by the PV panel is not sufficient to charge the battery, energy from the power grid will be supplied. An integrated simulator using MATLAB® is developed which includes five subsystems modules: building consumption model, chiller model, ice storage model, battery model, and photovoltaic model. Three parameters are considered: dry/wet bulb temperature, solar radiation on inclined surfaces, and real-time pricing rate.

The modules are collected from the literature and validated from experiments and the parameters are collected from the literature and industry practices for the use of this chapter (Table 1). The simulator is a black-box which is used to evaluate the operation decisions. The detailed decision model is explained in the next section.

2.3 Formulation of Building Energy System Decision Model

In the decision model, three building operation modes (Liu & Henze, 2007) are considered for each day: (1) from midnight to the onset of the on-peak period (0am-1pm); (2) the on-peak period (1pm-8pm); and (3) from the end of on-peak period to midnight (8pm-0am). The building shares the same characteristics (e.g., set-point temperature, pricing rate structure, etc.) during the successive hours in each building operation mode.

2.3.1 Decision Variables

Each building will control its set-point temperature. The shared ice storage will decide when to be charged or discharged to cool the buildings, and how to distribute its discharged cooling energy to each building. The decisions will be made for the battery on when to be charged or discharged to provide electricity for its served building. The decisions for the photovoltaic collector are charging battery, powering building, selling power to grid. Let *M* be the number of buildings, *K* be the number of modes, Table 2 lists the decision variables for building *m (m*=1,…, *M*) at building operation mode *k* (*k*=1,…, *K*):

In this chapter, a group of binary intermediate variables *BI* for the last three state variables ($S_{is,k}$, $S^{m}_{bat,k}$, $S^{m}_{PV,k}$) in Table 2 is introduced to simplify the problem formulations which is defined as:

$$BI = \begin{cases} decimal2binary\left(2^{S-1}\right) & S > 0 \\ decimal2binary\left(0\right) & S \le 0 \end{cases} \tag{2.1}$$

where decimal2binary(.) is commonly available function used to transform the decimal number to a binary number, *S* denotes the three state variables ($S_{is,k}$, $S^m_{bat,k}$, $S^m_{PV,k}$). Taking $S^m_{PV,k}$ =3 (PV sells power to grid) as an example, *S*=3, and 2^{S-1}=4, by employing function decimal2binary(4), *BI* is 100. Thus, following Eq. (2.1), *BI*=000, 001, 010, 100 represents *S*=0, 1, 2, 3 respectively.

Table 2 List of decision variables

Decision Variables	**Description**	**Type**
$T^m_{sp,k}$	Temperature set-point	Continuous
η^m_k	Percentage of the energy from ice storage to building	Continuous
$S_{is,k}$	State of ice storage	Integer (0: dormant; 1: charging; 2: discharging)
$S^m_{bat,k}$	State of battery	Integer (0: dormant; 1: charging; 2: discharging)
$S^m_{PV,k}$	State of PV panel	Integer (0: dormant; 1: charging battery; 2: powering building; 3: selling power to grid)

2.3.2 Objective Functions

Let us assume each building has the objective to minimize its energy cost for one day, which is written as:

$$f_m = \sum_{k=1}^{K}\sum_{j=1}^{H_k}\left(R^m_{p,j}P^m_{p,j} - R^m_{s,j}P^m_{s,j}\right) \tag{2.2}$$

where H_k is the number of hours in the building operation mode *k* (*k*=1,…, *K*); $R^m_{p,j}$ and $R^m_{s,j}$ ($/kWh) are the energy purchase and selling price at time *j* for building *m* respectively; $P^m_{p,j}$ and $P^m_{s,j}$ (kW) are the purchase energy from power grid and selling energy back to the power grid at time *j* for building *m* respectively.

Please note from Figure 3, the power grid supplies energy to the shared cooling (base chiller and dedicated chiller) to satisfy the cooling load of each building, and to each building to satisfy the non-cooling load. Let us assume for building *m* at time *j*, the purchase energy $P_{p,j}^{m}$ is composed of: 1) $P_{bch,j}^{m}$ (kW) which represents the base chiller electrical consumption allocated to building (see Eq. (2.8)); 2) $P_{dch,j}^{m}$ (kW) which is electrical consumption for the dedicated chiller allocated to building (see Eq. (2.6)); 3) $P_{nc,j}^{m}$ (kW) which is the building non-cooling electrical consumption.

$$P_{p,j}^{m} = P_{nc,j}^{m} + P_{dch,j}^{m} + P_{bch,j}^{m} \tag{2.3}$$

The building non-cooling electrical consumption includes building itself and battery's electrical consumption supplied by the power grid, and is determined as

$$\begin{aligned} P_{nc,j}^{m} &= \max\left(P_{load,j}^{m} - P_{bat,j}^{m}\eta_{conv}BI_{bat,k}^{m}\left(2\right) - P_{PV,j}^{m}\eta_{inv}BI_{PV,k}^{m}\left(2\right),0\right) \\ &+ \max\left(P_{bat,j}^{m}BI_{bat,k}^{m}\left(1\right) - P_{PV,j}^{m}BI_{PV,k}^{m}\left(1\right),0\right)\big/\eta_{conv} \end{aligned} \tag{2.4}$$

where $P_{load,j}^{m}$ (kW) is the non-cooling electricity load for building *m* at time *j*; $P_{bat,j}^{m}$ (kW) is the charging/discharging power of the battery for building *m* at time *j*; η_{conv} is the battery AC/DC converter efficiency, which is 0.9 according to its specification in this chapter; $P_{PV,j}^{m}$ (kW) is the energy generated by the PV panel for building *m* at time *j*; η_{inv} is the PV panel inverter efficiency, which is 0.92 in this chapter (Lu, 2004); $BI_{PV,k}^{m}\left(2\right)$ is the second digit of $BI_{PV,k}^{m}$ (from right to left).

Considering dedicated chiller based ice storage and base chiller, there are two common control strategies: chiller-priority control where the base chiller is the primary cooling provider with the dedicated chiller based ice storage as the secondary, and storage-priority control where the dedicated chiller based ice storage is the primary with the base chiller being the secondary. Extensive researches have demonstrated that storage-priority control can successfully shift the energy cost from on-peak period to off-peak period when the price rate of the electricity changes thus save more energy costs (Braun, 2007; Henze, 2003a, 2003b; Henze, 2004; Henze et al., 2005; Henze et al., 2003;

Henze & Schoenmann, 2003; Liu & Henze, 2007; Zhou et al., 2005). Therefore, the storage-priority strategy is employed in this chapter.

The energy request from dedicated chiller (primary cooling provider) is determined by the states of the ice tank. That is, the dedicated chiller will request energy from the power grid if and only if the ice tank is in charge stage, and the ice tank will provide cooling energy for buildings if and only if the ice tank is in discharge stage. Here, to realistically model the system, decision variable η_k^m is introduced to control the percentage of cooling energy allocated to each building (flow control valve 3 and flow control valve 4). Considering the daily energy consumption thus cost, $\overline{\eta}_m$ is introduced to denote the daily average percentage of cooling energy allocated to building *m*. That is,

$$\overline{\eta}_m = \sum_{k=1}^{K} \sum_{j=1}^{H_k} \left(\eta_k^m u_j \right) \Big/ \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{j=1}^{H_k} \left(\eta_k^m u_j \right) \tag{2.5}$$

The energy request for building *m* at time *j* is:

$$P_{dch,j}^m = P_{dedicated,j} \overline{\eta}_m BI_{is,k}(1) \tag{2.6}$$

where $P_{dedicated,j}$ (kW) is the electrical consumption for the dedicated chiller at time *j*; u_j (Btu/h) is the discharging rate of the ice storage at time *j*.

As secondary cooling provider, the base chiller only provides the amount of cooling to the building when the ice tank’s supply is not sufficient. Thus, the cooling energy supplied by the base chiller for each building $Q_{b,j}^m$ (Btu/h) is determined by Eq. (2.7), and the electrical consumption proportional to the cooling request from the base chiller for each building $P_{bch,j}^m$ is computed in Eq. (2.8).

$$Q_{b,j}^m = \max\left(Q_{c,j}^m - u_j BI_{is,k}(2) \eta_k^m, 0 \right) \tag{2.7}$$

$$P_{bch,j}^m = P_{base,j} Q_{b,j}^m \Big/ \sum_{m=1}^{M} Q_{b,j}^m \tag{2.8}$$

where $Q_{c,j}^{m}$ (Btu/h) is the cooling load for building *m* at time *j*; u_j (Btu/h) is the charging/discharging rate of the ice storage at time *j*; $P_{base,j}$ (kW) is the electrical consumption for the base chiller at time *j*.

The electricity selling back to the power grid from building *m* at time *j* is computed as

$$P_{s,j}^{m} = P_{PV,j}^{m} \eta_{inv} BI_{PV,k}^{m} \left(3\right) \tag{2.9}$$

Other than energy cost, one additional metric to evaluate the decentralized decisions is introduced, which is power grid dependency rate (PGDR). The PGDR is a measure of the degree of dependencies of a building to the power grid. For the building (e.g., smart building) with on-site generation and storage capability, PGDR metric may reflect the resilience of the building to a power disruption. For building *m* at time *j*, PGDR is defined as

$$PGDR_{j}^{m} = \left(P_{p,j}^{m} - P_{s,j}^{m}\right) \Big/ P_{grid} \tag{2.10}$$

where P_{grid} (kW) is the power grid capacity.

2.3.3 Constraints

(1) Power Grid: The total electricity purchased from the power grid for *M* buildings cannot exceed the capacity of the power grid at each time *j*. That is,

$$\sum\nolimits_{m=1}^{M} P_{p,j}^{m} \le P_{grid} \tag{2.11}$$

(2) Building: The building should keep its indoor temperature at a comfort level at each time *j*.

$$T_{i}^{mL} \le \overline{T}_{i,j}^{m} \le T_{i}^{mU} \tag{2.12}$$

where $\overline{T}_{i,j}^{m}$ is the average indoor temperature for building *m* at time *j*; T_{i}^{mL} and T_{i}^{mU} are 74 °F and 81 °F in this chapter.

(3) Base chiller: The base chiller load cannot exceed its capacity at each time *j*.

$$\sum_{m=1}^{M} Q_{b,j}^{m} \le Q_{\max,j} \tag{2.13}$$

where $Q_{max,j}$ is the chiller capacity at time *j*.

(4) Ice Storage: At each building operation mode *k*, the ice storage system cannot be charged if the state of charge is at the maximum level (Eq. (2.14)) and cannot be discharged if the state of charge is at the minimum level (Eq. (2.15)). The summed percentage of cooling from ice storage to each building cannot exceed one (Eq. (2.16)) when the ice storage is in the discharging state.

$$BI_{is,k}\left(1\right) \le ceil\left(\max\left(0, SOC_{is,\max} - SOC_{is,k}\right)\right) \tag{2.14}$$

$$BI_{is,k}\left(2\right) \le ceil\left(\max\left(0, SOC_{is,k} - SOC_{is,\min}\right)\right) \tag{2.15}$$

$$\sum_{m=1}^{M} \eta_{k}^{m} \le BI_{is,k}(2) \tag{2.16}$$

where ceil(.) rounds the element to the nearest integer towards infinity; $SOC_{is,max}$ and $SOC_{is,min}$ are maximum and minimum state of charge for the ice storage; $SOC_{is,k}$ is the initial state of charge for ice storage at building operation mode *k*. The state of charge is a percentage value in this chapter.

(5) Battery: At each building operation mode *k*, the battery cannot be charged if the state of charge is at the maximum level (Eq. (2.17)) and cannot be discharged if the state of charge is at the minimum level (Eq. (2.18)).

$$BI_{bat,k}^{m}\left(1\right) \le ceil\left(\max\left(0, SOC_{bat,\max}^{m} - SOC_{bat,k}^{m}\right)\right) \tag{2.17}$$

$$BI_{bat,k}^{m}\left(2\right) \le ceil\left(\max\left(0, SOC_{bat,k}^{m} - SOC_{bat,\min}^{m}\right)\right) \tag{2.18}$$

where ceil(.) rounds the element to the nearest integer towards infinity; $SOC_{bat,\max}^{m}$ and $SOC_{bat,\min}^{m}$ are maximum and minimum state of charge for building *m*'s battery; $SOC_{bat,k}^{m}$

is the battery's initial state of charge for building *m* at building operation mode *k*. The state of charge is a percentage value in this chapter.

(6) PV-panel: The PV-panel can charge the battery only when the battery is in the charging state (Eq. (2.19)).

$$BI_{PV,k}^{m}\left(1\right)\leq BI_{bat,k}^{m}\left(1\right) \tag{2.19}$$

where *m=1,…, M* and *k=1,…, K*.

Thus, for *M* buildings, *M* decision models are introduced with each model having the objective function shown in Eq. (2.2) and constraints shown in Eqs. (2.11)-(2.19). A decentralized decision framework based on a Memetic algorithm is then introduced to ensure *M* decision models can converge to Pareto solutions.

2.4 Decentralized Decision Making Framework

In the bi-level decentralized framework, other than the building agents each representing one building with the decision model explained in the section 2.3, a facilitator agent is introduced aiming to coordinate the buildings to reach converged solutions. This is achieved by deriving a weighted-sum of the buildings' objectives as the function for the facilitator agent. The facilitator agent then classifies the decision variables from the derived objective function into local variables (**X**) which are controlled by each building and coupled variables (**Y**) which are jointly controlled by more than one building. Similarly, the constraints are classified into local constraints which apply for each building and system constraints which apply for the group of buildings. Artificial coupled variables **Z** are introduced to decompose the system constraints into separable pieces so that each building can solve fully independent sub-problems. Let us assume the *j^th^* coupled system constraint $b_j\left(\mathrm{X}_1,\ldots,\mathrm{X}_M,\mathrm{Y}\right)\leq h_j$ can be written as:

$$b_{1,j}\left(\mathrm{X}_1,\mathrm{Y}\right)+b_{2,j}\left(\mathrm{X}_2,\mathrm{Y}\right)+\cdots+b_{M,j}\left(\mathrm{X}_M,\mathrm{Y}\right)\leq h_j \tag{2.20}$$

where $b_{1,j}\left(\mathrm{X}_1,\mathrm{Y}\right), b_{2,j}\left(\mathrm{X}_2,\mathrm{Y}\right),\ldots, b_{M,j}\left(\mathrm{X}_M,\mathrm{Y}\right)$ are the constraints local to each building respectively. *M-1* artificial variables ($z_{1,j},\ldots,z_{M-1,j}$) can be introduced as:

$$\begin{gathered} b_{1,j}\left(\mathrm{X}_1,\mathrm{Y}\right)\le z_{1,j} \\ \cdots \\ b_{M,j}\left(\mathrm{X}_M,\mathrm{Y}\right)\le h_j-\sum_{m=1}^{M-1} z_{m,j} \end{gathered} \tag{2.21}$$

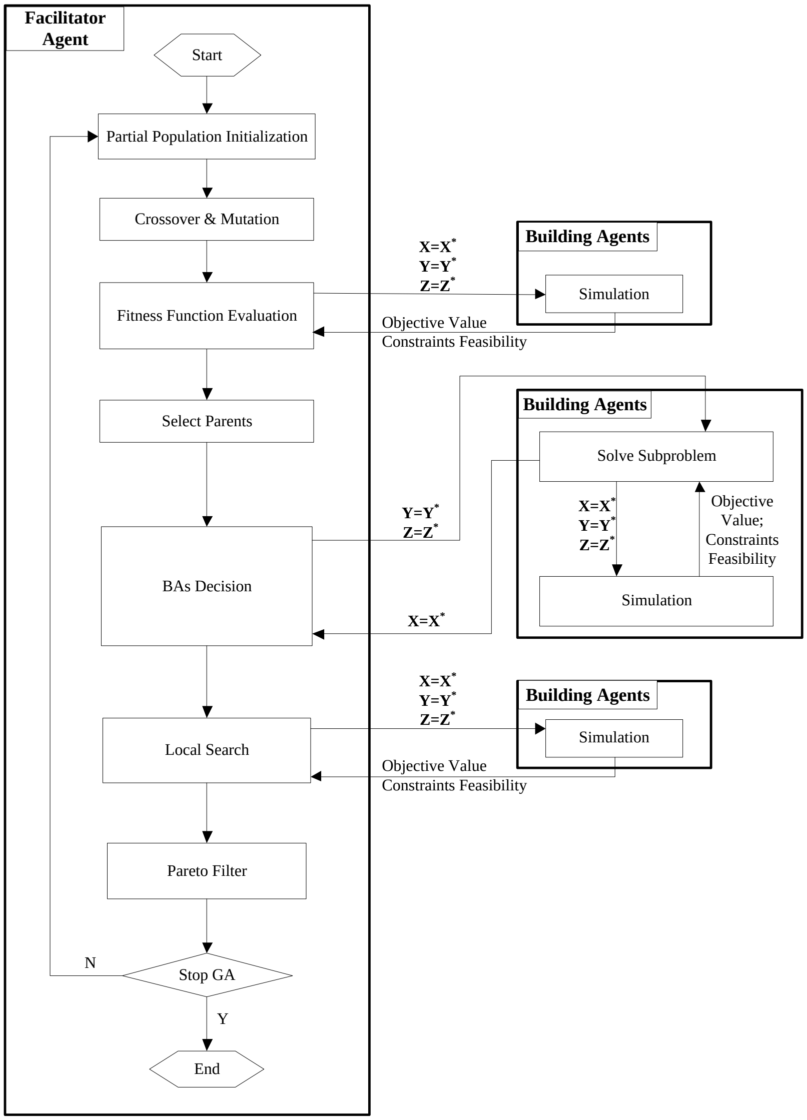


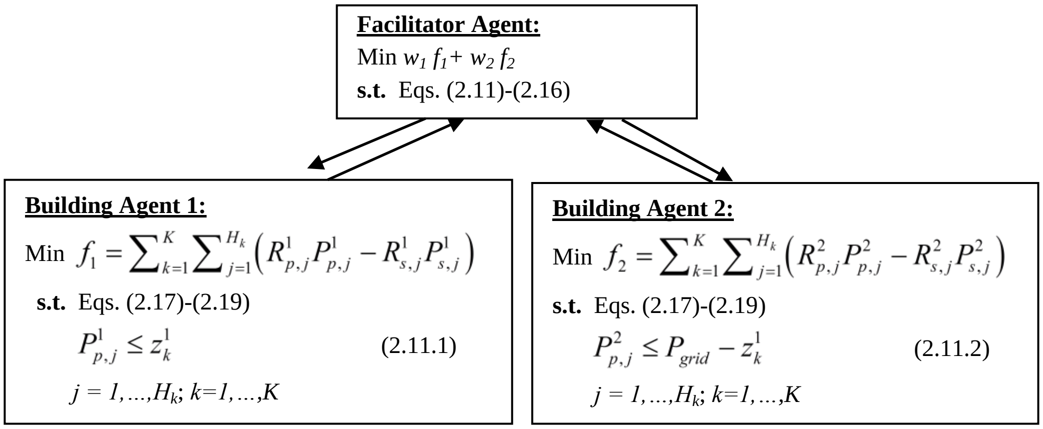
Figure 4 Bi-level decentralized framework based on MA

Thus, the coordination function of the facilitator agent has decision variables of **X**, **Y** and **Z**. It will employ genetic algorithm (GA) operators including crossover and mutation to explore the global decision space (**X**, **Y**, **Z**) followed by local search (LS) to exploit the coupled decision space (**Y**, **Z**). The updated decisions are passed to each building agent who attempts to "optimize" its own objective over the local variables (**X**) only and feeds the decisions on local variables back to the facilitator agent. At the end of each MA iteration, the Pareto filter (Loukil et al., 2007) is applied on the population to filter out the dominated solution. The Memetic algorithm based bi-level decision framework is illustrated in Figure 4.

2.5 Implementation of Decentralized Memetic Algorithm

2.5.1 Decentralized Decision Model

Based on the discussion in section 2.4, the decentralized decision model for the building cluster is constructed as shown in Figure 5.



Facilitator Agent:

Min $w_1 f_1 + w_2 f_2$

s.t. Eqs. (2.11)-(2.16)

Building Agent 1:

Min $f_1 = \sum_{k=1}^{K} \sum_{j=1}^{H_k} \left(R_{p,j}^1 P_{p,j}^1 - R_{s,j}^1 P_{s,j}^1 \right)$

s.t. Eqs. (2.17)-(2.19)

$P_{p,j}^1 \le z_k^1$ (2.11.1)

$j = 1,\ldots,H_k;\ k=1,\ldots,K$

Building Agent 2:

Min $f_2 = \sum_{k=1}^{K} \sum_{j=1}^{H_k} \left(R_{p,j}^2 P_{p,j}^2 - R_{s,j}^2 P_{s,j}^2 \right)$

s.t. Eqs. (2.17)-(2.19)

$P_{p,j}^2 \le P_{grid} - z_k^1$ (2.11.2)

$j = 1,\ldots,H_k;\ k=1,\ldots,K$

Figure 5 Decentralized decision model for building cluster

The decision variables $T_{sp,k}^m$, $S_{is,k}$, and η_k^m in Table 2 are the coupled variables Y, while the remaining variables in Table 2 are the local variables X_m. The constraints in Eqs. (2.11)-(2.16) are system constraints which are handled by the facilitator agent. Artificial coupled variables Z are employed to decompose the coupled system constraint in Eq. (2.11) as two constraints Eqs. (2.11.1) and (2.11.2) shown in Figure 5. Please note it is not necessary to decompose constraints in Eqs. (2.12)-(2.16) since they do not contain local variables. The constraints in Eqs. (2.17)-(2.19) are local constraints handled by each building agent.

2.5.2 Solution Representation and Population Initialization

Real code GA is used to encode the continuous variables and binary code for the binary variables. Researchers have demonstrated that utilizing the building thermal mass (pre-cooling building), and shifting the peaking load by using a storage system can significantly reduce the energy cost (Braun, 2007; Drees & Braun, 1996; Sun et al., 2006). In this chapter, the set-point temperature $T_{sp,k}^{m}$ for building *m* at building operation mode *k* is initialized as follows:

$$T_{sp,k}^{m} = \begin{cases} T_{sp,k}^{m\ L} + \left(T_{sp,k}^{m\ U} - T_{sp,k}^{m\ L}\right) \times r/2 & k \text{ is prepeak period} \\ T_{sp,k}^{m\ L} + \left(T_{sp,k}^{m\ U} - T_{sp,k}^{m\ L}\right) \times r & \text{otherwise} \end{cases} \tag{2.22}$$

where the uniform random number $r \in [0,1]$; $T_{sp,k}^{m\ L}$ and $T_{sp,k}^{m\ U}$ are 74 ℉ and 81 ℉ in this chapter.

The state of the ice storage system $S_{is,k}$ at peak building operation mode *k* is initialized as follows:

$$S_{is,k} = 2 \text{ if } r \le 0.8; \text{ otherwise } S_{is,k} = 0 \tag{2.23}$$

At off-peak building operation mode *k*, the state of the ice storage system is

$$S_{is,k} = 2 \text{ if } r \le 0.2;\ S_{is,k} = 1 \text{ if } 0.2 < r \le 0.7; \text{otherwise } S_{is,k} = 0 \tag{2.24}$$

Partial population initialization strategy (Figure 4) is used to balance the exploration and exploitation search capability. At each iteration *g*, the *n* worst solutions in the population are replaced with new solutions where *n* is computed as

$$n = round\left(\exp\left(-\left(g-1\right)^{2.6} / G^{2}\right) \times N_{P}\right) \tag{2.25}$$

where round(.) rounds the element to the nearest integer; *G* is the maximum iterations of MA; N_P is the population size.

The initial population is generated from the feasible solutions after building agents check the feasibility of the tentative solutions.

2.5.3 Fitness Function and Parents Selection

Different weights are assigned for each solution in the population to realize various search directions (Arroyo & Armentano, 2005). The weight combination is randomly generated as

$$w_m = r_m \big/ \left(r_1 + \cdots + r_M \right) \qquad m = 1, \cdots, M \tag{2.26}$$

where the uniform random number $r_m \in [0,1]$.

N_P weight combinations will be generated using Eq. (2.26). For each weight combination, the solution from the population with best value for the fitness function is selected which is defined as

$$\sum\nolimits_{m=1}^{M} w_m f_m \left(\mathrm{X}_m, \mathrm{Y} \right) \tag{2.27}$$

2.5.4 Crossover and Mutation

The 3-points crossover operator is applied which chooses 3 cut points randomly for the binary variables. Real-parameter crossover operators (Lozano et al., 2004), which take advantage of numerical values, are employed for continuous variables. Given two chromosomes $C_1 = \left(\mathrm{X}_1^1, \cdots, \mathrm{X}_M^1, \mathrm{Y}^1, \mathrm{Z}_1^1, \cdots, \mathrm{Z}_{M-1}^1 \right)$ and $C_2 = \left(\mathrm{X}_1^2, \cdots, \mathrm{X}_M^2, \mathrm{Y}^2, \mathrm{Z}_1^2, \cdots, \mathrm{Z}_{M-1}^2 \right)$, the offspring are generated through the following crossover operator as, $C_1' = \theta C_1 + (1-\theta) C_2$ and $C_2' = \theta C_2 + (1-\theta) C_1$, where $\theta \in [0,1]$.

Note the feasibility of the new generated offspring needs to be checked by each building agent. The mutation operation is triggered if a solution is not feasible where a new feasible solution is generated to replace the infeasible one.

2.5.5 Local Search (LS)

The facilitator agent applies LS over coupled variables (**Y**, **Z**) to improve the solutions. The LS adopted in this chapter is the simulated annealing algorithm with adaptive neighborhood (Zhao, 2011). As a part of Memetic algorithms, the simulated annealing algorithm works as a local optimizer to find the local optimal solutions of the weighted system problem.

First, each building agent evaluates the objective functions and constraints of its sub-problems *wrt* the coupled variables after the local variables are obtained by solving the sub-problem. For example, given the coupled variables $\mathbf{Y}=\mathbf{Y}^*$ and $\mathbf{Z}=\mathbf{Z}^*$, the building agents solve the sub-problems independently and obtain the optimal values of the local variables.

Let $G(\cdot)$ denote the value of the weighted objective function, S_0 denote the best solution found so far, S_i denote the current solution at iteration i, S_c denote the candidate solution, β_i is the cooling constant at iteration i, I is the maximum number of iterations of the simulated annealing; then the simulated annealing algorithm in the MA based decision framework is as follows:

Step 1. Facilitator agent sets i=1, $\beta_i=0.95$, and $S_i=S_0$

Step 2. A candidate solution S_c is generated according to the following steps.

Step 2.1. The state of the ice storage system is the most critical factor impacting the weighted system objective value. So here a uniform random number $r \in [0,1]$ is employed to control the convergence speed of the state of the ice storage system. The state of the ice storage system will be the same as the state in the best solution S_0 when $r \le 0.1^{(1-i/I)^5}$. Otherwise the state of ice storage system will be generated by Eqs. (2.23)-(2.24).

Step 2.2. The set-point temperature for building m at building operation mode k

$$T_{sp,k}^{m}\,' = \begin{cases} T_{sp,k}^{m} - \Delta\left(i, T_{sp,k}^{m} - T_{sp,k}^{m\ \ L}\right) & \begin{array}{l}\left(k \text{ is pre-peak period and } r \le 0.8\right) \\ \text{or } \left(k \text{ is on peak period and } r > 0.8\right)\end{array} \\ T_{sp,k}^{m} + \Delta\left(i, T_{sp,k}^{m\ \ U} - T_{sp,k}^{m}\right) & \begin{array}{l}\left(k \text{ is pre-peak period and } r > 0.8\right) \\ \text{or } \left(k \text{ is on peak period and } r \le 0.8\right)\end{array} \end{cases} \tag{2.28}$$

Step 2.3. The percentage of energy from the ice storage system to each building is generated as:

$$\eta_k^{m}\,' = \begin{cases} \eta_k^{m} + \Delta\left(i, 1 - \eta_k^{m}\right) & r \le w_m R_{p,k}^{m} \Big/ \sum_{m=1}^{M} w_m R_{p,k}^{m} \\ \eta_k^{m} - \Delta\left(i, \eta_k^{m}\right) & \text{otherwise} \end{cases} \tag{2.29}$$

where w_m is the weight for building m in the weighted system objective; and $R_{p,k}^{m}$ is the average power grid purchase price for the building m at building operation mode k.

The following function is adopted from (Zhao, 2011):

$$\Delta\left(i, y\right) = y \times \left(1 - \rho^{(1-i/I)^2}\right) \tag{2.30}$$

where ρ is a uniform random number from (0, 1).

Step 2.4. The artificial coupled variables **Z** are updated only when at least one of the constraints (2.11.1) and (2.11.2) is violated.

Step 3. Facilitator agent checks the feasibility for the system constraints (2.11)-(2.16). Each building agent checks the feasibility of constraints (2.17)-(2.19), (2.11.1) or (2.11.2), and returns its objective value and constraint feasibility information to the facilitator agent.

Step 4. If $G(S_c)<G(S_0)$, then set $S_0=S_c$. If $G(S_c)<G(S_i)$, then a move is made, setting $S_{i+1}=S_c$. If $G(S_c)\geq G(S_i)$, then a move is made to S_c with probability

$$P\left(S_i, S_c\right) = \exp\left(\left(G\left(S_i\right) - G\left(S_c\right)\right)\big/\beta_i\right) \tag{2.31}$$

If S_c is rejected then $S_{i+1}=S_i$.

Step 5. i is set to $i+1$, β_i is set to $0.95\beta_{i-1}$. Go to Step 2 until a stopping criterion is met.

2.6 Experimental Analysis

The MA based framework is applied to study a simple building cluster (two buildings) located in the Phoenix, Arizona area. Since Phoenix is known for hot summers when energy usage is critically important, July 21, 2009 is studied as an example day for the experiments with data from SRP (http://www.srpnet.com), a local electricity provider.

2.6.1 Analysis on Pareto Frontier for Decentralized Decision

The capacity of the power grid is assumed to be 15 kW. The heavy mass building applies the time-of-use (TOU) plan and the light mass building adopts the SRP EZ-3 option plan. In the EZ-3 plan, 3pm-6pm are the peak-hours where the price is much higher than the off-peak hours. In the TOU plan, 1pm-8pm are the peak-hours where the price is also higher (less than that of EZ-3) than the off-peak hours. During the off-peak hours, the price of the EZ-3 plan is relatively lower than that of the TOU plan. The following parameters of MA are applied: (1) the MA population size N_P is set to 40; (2) the maximal number of iteration G for MA is 30; (3) the maximal number of iterations I for the simulated annealing is 20.

The Pareto frontier in the single building energy cost performance space obtained by the MA based bi-level decision framework is shown in Figure 6. Five Pareto solutions (A, B, C, D, and E) are highlighted for energy costs for the heavy mass building vs. the light mass building, where A is (4.97, 9.96), B is (6.38, 8.40), C is (5.19, 9.05), D is (6.16, 8.44), E is (5.55, 8.88). A centralized decision model where one optimization problem aiming to minimize the summed energy cost of the two buildings is formulated. Solution from the centralized model is highlighted as F in Figure 6 where F is (5.27, 8.99). Solution G (5.13, 10.07) is obtained when the heavy mass building fully controls the ice storage which means the heavy mass building consumes all the energy from ice storage and the two buildings cooperate with each other to minimize its own energy cost. Solution H (6.93, 8.89) is the case when the light mass building fully controls the ice storage. Obviously, solutions G and H are dominated by the Pareto frontier and it will be more cost effective if the two buildings share ice storage. Two non-cooperative game theoretical approaches (leader/follower) solutions I (5.05, 10.59) and J (6.77, 8.42) are also obtained. In the leader/follower case, the leader will make the decisions first with the assumption that the follower's behavior is rational (Lewis & Mistree, 1998), and then the follower will solve its problem subject to the leader's decision. Solution I is the case when the heavy mass building is the leader and the light mass building is the follower and solution J is when the light mass building is the leader and the heavy mass building is the follower. Comparing solutions I and J with the Pareto solutions A-E, it is observed that these two solutions are dominated by the Pareto solutions, and two buildings will have more cost savings when they cooperate with each other to make decisions.

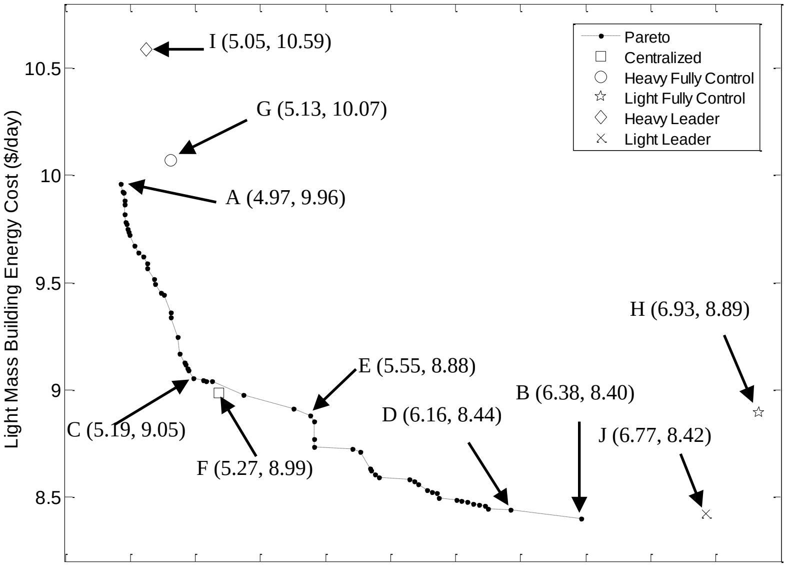


Figure 6 Pareto frontier on energy cost for each building

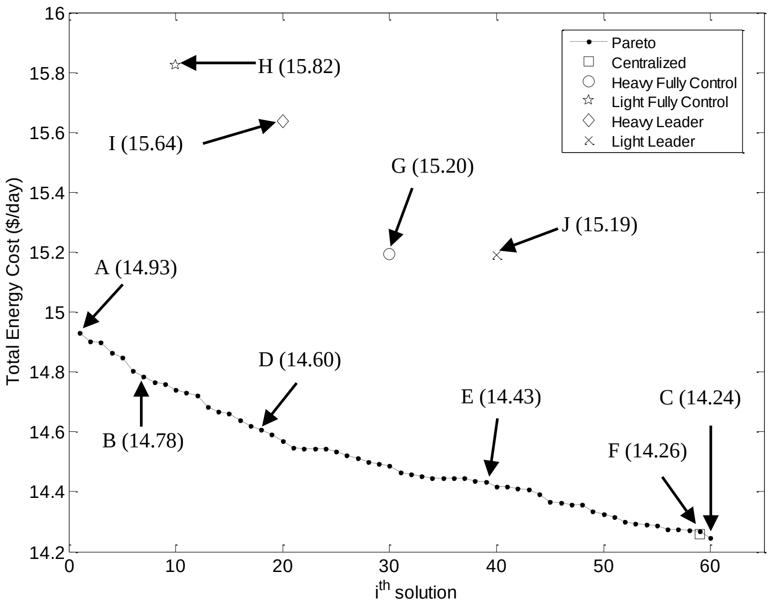


Figure 7 Overall energy cost of two buildings

In addition, the overall costs of the two buildings for all the Pareto solutions and solutions F-J (from Figure 6) are illustrated in Figure 7. Pareto solution C achieves minimum total energy cost. This demonstrates that a Pareto solution can achieve the same cost effectiveness for the whole system as centralized decision given an appropriate weight assigned to each building. Solution H (light mass building fully controls the ice storage system) is the least cost effective since the light mass building cannot utilize the ice storage efficiently.

From Figure 6, one may argue if the manager is keen on minimizing the energy cost of the heavy mass building (e.g., an apartment complex with more residents), Solution A is more preferable than C than E than D than B, and vice versa if the light mass building energy costs are to be minimized. Not surprisingly, Figure 7 indicates that a centralized decision F is more cost effective than some Pareto solutions for the whole system. The Pareto solutions outperform the solutions G-J on the total energy cost. If one building is heavily preferred over another, the solution is least cost effective for the whole system (e.g., solutions A, B). Although most Pareto solutions are inferior to the centralized optimal solution on the total cost metric, the Pareto analysis enables tradeoff analysis on other dimensions, for example, power grid dependency rate (PGDR) (see section 2.3.2 for definition).

The Pareto frontier on the average PGDR metric based on all the Pareto solutions on the single building energy cost space in Figure 6 are demonstrated in Figure 8. Since the ultimate goal of the smart building cluster is to be resilient to disturbance and robust to the power grid, it is assumed less dependence on the power grid will be more preferable for the smart building managers. The Pareto frontier on the PGDR performance space is denoted as the line in Figure 8. Please note that only solutions C and E are Pareto solutions on the PGDR space, and they dominate other Pareto solutions on the single building energy cost space. Obviously, solution C (16.53, 23.29) indicates the heavy mass building is least dependent on the power grid; and the light mass building is most independent from the power grid for solution G (17.65, 22.87).

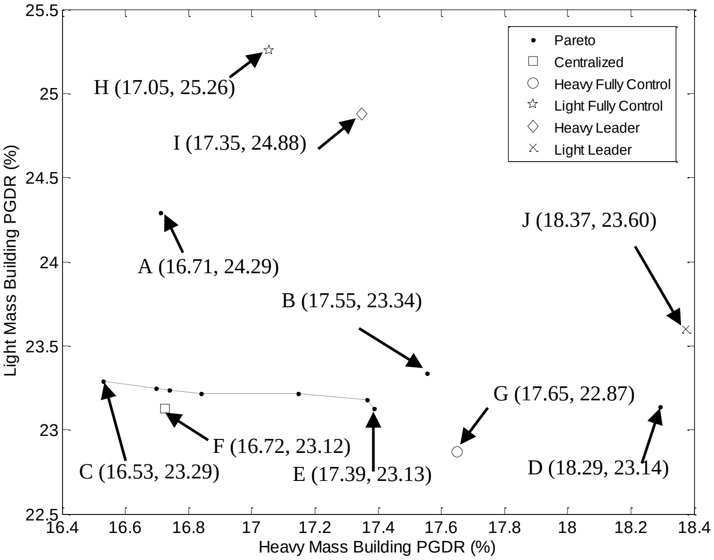


Figure 8 PGDR for two buildings

Table 3 Solutions value and decision makers' decisions

Solutions	Single building cost ($/day)		PGDR (%)		Total cost ($/day)
	HM	LM	HM	LM	
A	**4.97**	9.96	16.71	24.29	14.93

B	6.38	**8.40**	17.55	23.34	14.78
C	5.19	9.05	**16.53**	23.29	**14.24**
D	6.16	8.44	18.29	23.14	14.60
E	5.55	8.88	17.39	23.13	14.43
F	5.27	8.99	16.72	23.12	14.26
G	5.13	10.07	17.65	**22.87**	15.20
H	6.93	8.89	17.05	25.26	15.82
I	5.05	10.59	17.35	24.88	15.64
J	6.77	8.42	18.37	23.60	15.19
Recommended Decisions	A	B	C	G	C

Table 3 summarizes the values for the ten solutions A-J highlighted in Figure 6 on the three metrics (single building cost; PGDR and total cost). The decisions based on these individual metrics are suggested in the last row. It is concluded the MA based bi-level decision framework is capable of recommending decisions based on different performance metrics.

2.6.2 Decentralized Decision under Different Pricing Mechanisms

Next, this section studies the impact of different pricing mechanisms on the performance of the decentralized decisions, and explores how the decentralized decisions may assist building managers in choosing a pricing mechanism. Nowadays, it is common that electricity providers are implementing different pricing mechanisms for on-peak and off-peak hours. Taking SRP as an example, three pricing mechanisms have been implemented: basic plan, SRP EZ-3 plan and time-of-use (TOU) plan. In the basic plan, the price is constant. Details of EZ-3 and TOU plans are discussed in section 2.6.1. This pricing mechanism has been widely applied across the Phoenix metropolitan area and the detailed prices can be found at SRP website. Please note the building’s decision should be highly dependent on its pricing mechanism. That is, during the off-peak period in the basic plan, a building should not utilize the storage system and the PV panel should power the building. However, during the off-peak period in the EZ-3 plan and the TOU plan cost savings should be realized using a storage system and the energy from the PV panel being sold back to the grid.

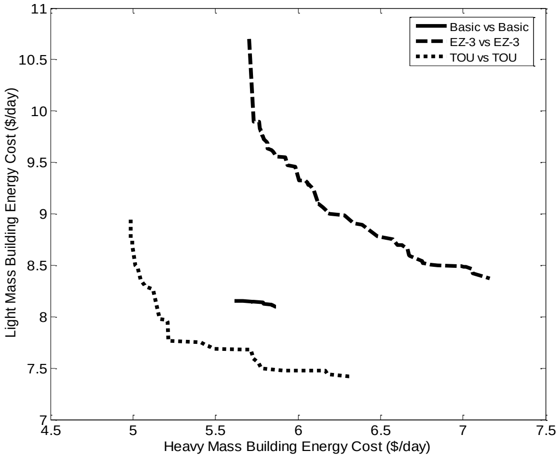


Figure 9 Two buildings use the same pricing mechanism

Different Pareto frontiers when the two buildings use the same pricing mechanism are explored in Figure 9. “Basic vs. Basic” in Figure 9 means that both the HM building and LM building use the basic plan. It is observed that the EZ-3 plan is the least effective, and it is dominated by the basic plan, and the basic plan is dominated by the TOU plan. This may be due to the fact that three operation modes are adopted (0am-1pm, 1pm-8pm and 8pm-0am) in this chapter. Given the peak building operation mode is from 1pm to 8pm which is 2 hours earlier than the peak price hours in the EZ-3 plan (3pm-6pm), the inefficient usage of the storage system under the EZ-3 plan makes the performance unfavorable. Since the pricing plans are set by the energy provider (e.g., SRP), one solution is to investigate more building operation modes (e.g., 24 modes on the hourly basis). The challenges though lie in the increased complexity of the decision model. This motivates me to improve the computational performance of the decentralized decision framework to ensure the model is able to generate decisions within the shortened time frame in the following chapters.

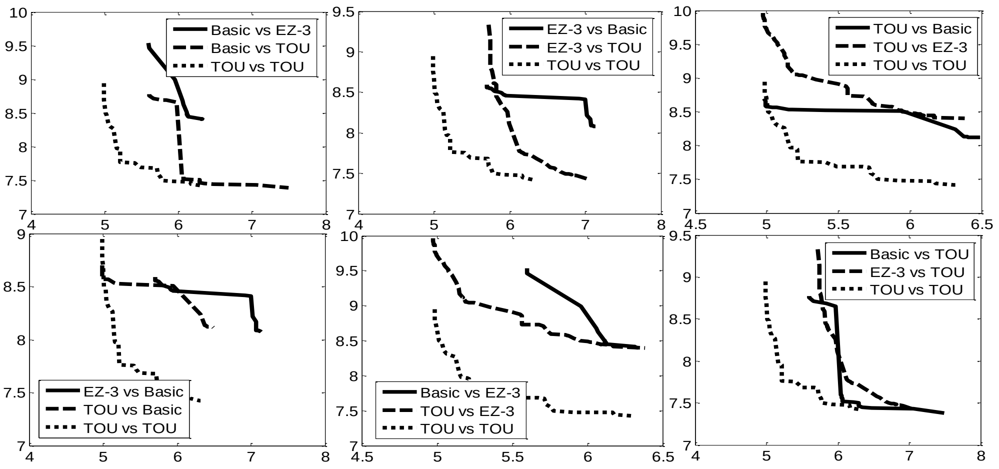


Figure 10 Two buildings use different pricing mechanisms

The case where the two buildings use different pricing mechanisms is demonstrated in Figure 10. “Basic vs. EZ-3” in Figure 10 means HM building uses the basic plan and LM building uses the EZ-3 plan, the horizontal and vertical axis represents daily energy cost of the HM building and LM building respectively. “TOU vs. TOU” is also presented in Figure 10 and it dominates all the other pricing mechanism combinations. As in Figure 9, Figure 10 also indicates that the EZ-3 plan is not preferred by the two buildings, and the TOU plan is the most preferred by the two buildings. Under the TOU plan, the two buildings can utilize the building thermal mass (pre-cooling building), and storage system to reduce the energy cost more effectively. In addition, it is observed from the experiments that the relative cooling consumption of the HM building to the LM building during the on-peak period (1pm-8pm) is much smaller than the pre-peak period (0am-1pm) which demonstrates that the HM building can shift more cooling consumption from the on-peak period to the pre-peak period by using the precooling strategy (Zhou et al., 2005). The results from this experiment may help the decision maker choose an appropriate pricing mechanism for different buildings (heavy mass vs. light mass). In addition, the results indicate the EZ-3 plan is not cost effective from a building perspective which may be a limitation of fewer building operation modes (*K*=3).

Therefore, it may be necessary to develop pricing mechanisms with more building operation modes which will be explored in the following chapters.

2.6.3 Decentralized Decision under Different Power Grid Capacities

In the first set of experiments, the energy from the power grid is assumed to be at an unlimited level (15 kW). However, in reality, such an assumption does not always hold. Therefore, the effect of three different power grid capacities on the performance of the decentralized decisions is further investigated. It is assumed the power capacity for the higher level is 12 kW, 10 kW for the medium level and 8 kW for the lower level. The three Pareto frontiers for these three capacities are plotted in Figure 11.

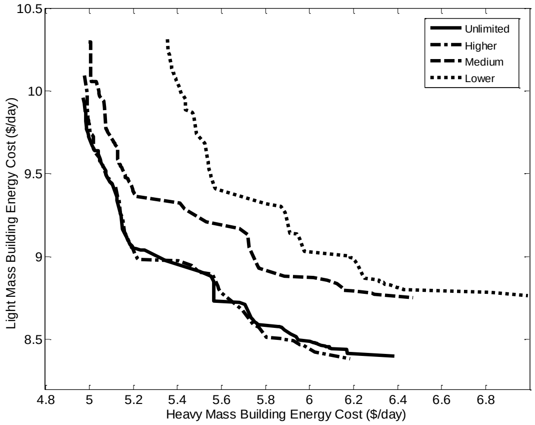


Figure 11 Pareto frontiers under four different power grid capacities

It is observed that the Pareto frontier under the higher level capacity is almost the same as the Pareto frontier under the unlimited level, and dominates the Pareto frontiers under the medium and lower level capacity. Pre-cooling a building and shifting the peak load to the storage system can significantly reduce the energy cost. However, the power grid capacity restricts the capability to pre-cool building and charge the storage system at the pre-peak period under the medium and lower capacity levels. Only two of the three storage systems (one ice storage and two batteries) can be charged simultaneously under the medium level capacity. Moreover, only one of the three storage systems can be charged under the lower level capacity. It is not surprising that Pareto solutions from the tighter power grid capacity are dominated by those with more capacity. Therefore, my immediate next step is to explore the Pareto solutions with three performance metrics (single building cost, PGDR, overall cost) for a tighter power grid capacity where tradeoff analysis is critical.

2.7 Conclusions

Energy usage has attracted national and even international attentions lately. Yet academic research and industry practices have focused on either improving the efficiency of one sub system of a building or one single building. This chapter takes a systematic approach with the hypothesis being a group of buildings jointly sharing energy resources locally will be both cost effective and resilient to a power disturbance. To test the hypothesis, a hybrid model for building systems which consists of energy consumption, storage and generation subsystems is developed. Based on this model, decentralized decisions on energy usage for a cluster of buildings are explored. To accomplish this, a bi-level decision framework based on MA is developed and different Pareto frontiers are generated. The derived Pareto frontier can assist building managers to: 1) make decisions on different metrics (e.g. single building cost; PGDR; total cost; etc.); 2) choose pricing mechanisms effectively; and 3) increase smart buildings' disaster resilience capability. With the development of new energy technology, a building cluster concept and decentralized energy operation decisions will attract more interest. This chapter makes the first attempt to address some important issues related to the research. The research outcomes could gain the insights for developing a building management system aiming to reduce energy costs and improve building energy efficiency.

While promising, the bi-level decision framework based on MA is computationally expensive due to the number of sub systems being modeled and the large number of decision variables. This may prohibit its application to real time operation decisions which are usually made on an hourly (or even less) basis. In the following chapters, the particle swarm optimization (PSO) algorithm which is capable of deriving good results with very low computational cost (Reyes-Sierra & Coello Coello, 2006) will be employed to improve the performance of the decentralized decision framework. With the improved performance, the building operation modes could be refined to make it capable of hourly decisions, and improve the performance for any pricing mechanisms. In addition, uncertainties can be introduced into the energy systems and an adaptive decentralized optimization algorithm under uncertainty should be investigated to guarantee buildings can respond to this dynamic environment.

Chapter 3

AN INTELLIGENT AUGMENTATION OF PARTICLE SWARM OPTIMIZATION WITH MULTIPLE ADAPTIVE METHODS

Over the last two decades, the newly developed optimization technique - Particle Swarm Optimization (PSO) has attracted great attention. Two common criticisms exist. First, most existing PSOs are designed for a specific search space thus an algorithm performing well on a diverse set of problems is lacking. Secondly, PSO suffers premature convergence. To address the first issue, the PSO is augmented via the fusion of multiple search methods. An intelligent selection mechanism is developed based on an effectiveness index to trigger appropriate search methods. In this chapter, two search techniques are studied: a non-uniform mutation-based method and an adaptive sub-gradient method. The augmented PSO is further improved using adaptive Cauchy mutation to prevent premature convergence. As a result, an augmented PSO with multiple adaptive methods (PSO-MAM) is developed. The performance of PSO-MAM is tested on 43 functions (uni-modal, multi-modal, non-separable, shifted, rotated, noisy and mis-scaled functions). The results are compared in terms of solution quality and convergence speed with 10 published PSO methods. The experimental results demonstrate PSO-MAM outperforms the comparison algorithms on 36 out of 43 functions. The conclusion is, while promising, there is still room for improving PSO-MAM on complex multi-modal functions (e.g., rotated multi-modal functions).

3.1 Introduction

Particle Swarm Optimization (PSO), which was developed by Kennedy and Eberhart in 1995 (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995), is a swarm intelligence which mimics a flock of birds that communicate together as they fly. The process in PSO involves both social interaction and intelligence so that birds learn from their own experience (local search) and from experiences of others around them (global search) (Kennedy & Eberhart, 2001). Although PSO was initially employed for balancing weights in neural networks (Kennedy & Eberhart, 1995), it soon became a popular global optimizer, mainly for problems in which the decision variables are real numbers (Kennedy & Eberhart, 2001).

Like most population-based algorithms, a pool of individuals which is termed a swarm in PSO is employed to search on the solution space. Each individual in the swarm is called a particle which moves on the search space directed by three components: 1) its previous velocity; 2) its best position (pBest) found so far; 3) the best position (gBest) found so far from its neighbors where the neighborhood is defined by the topology. Some common topologies include global, local, star network and tree network (Kennedy & Mendes, 2002). In this chapter, the global topology is studied which means gBest is the best position found so far among all the particles. Research on PSO algorithm development can be classified in two general categories. One focuses on the velocity update formulation for each particle in order to accelerate the convergence speed or maintain diversity of the swarm. For example, PSO with inertia weight (PSO-w) (Shi & Eberhart, 1998), PSO with constriction coefficient (PSO-cf) (Clerc & Kennedy, 2002), unified particle swarm optimizer (UPSO) (Parsopoulos & Vrahatis, 2004), and dynamic velocity-based PSO (PSO-c3dyn) (Garcia-Villoria & Pastor, 2009), just to name a few. The second category of PSO research is to investigate different learning strategies on exemplar (pBest and gBest) selection for the particle to quickly converge to near-optimum (if not optimum) solutions. Some examples are fully informed particle swarm (FIPS) (Mendes et al., 2004), fitness-distance-ratio-based PSO (FDR-PSO) (Peram et al., 2003), cooperative PSO (CPSO-H) (van den Bergh & Engelbrecht, 2004), dynamic multi-swarm PSO (DMS-PSO) (Liang & Suganthan, 2005), comprehensive learning PSO

(CLPSO) (Liang et al., 2006), generalized opposition-based learning PSO (GOPSO) (Wang et al., 2011), self-adaptive learning-based PSO (SLPSO) (Wang et al., 2010), example-based learning PSO (ELPSO) (Huang et al., 2010), and orthogonal learning PSO (OLPSO) (Zhan et al., 2010). Please note the methods reviewed above concentrate on PSO techniques only. Some recent research indicates that the performance of PSO (e.g., convergence rate, solution quality) could be much improved via a model fusion concept, that is, integrating PSO with other search techniques, such as 1) evolutionary operators: selection (Angeline, 1998), crossover (Chen et al., 2007), mutation (Andrews, 2006; Higashi & Iba, 2003; Thangaraj et al., 2009), etc.; 2) evolutionary algorithms: genetic algorithm (GA) (Kao & Zahara, 2008), Memetic algorithm (MA) (Petalas et al., 2007), cellular automata (CA) (Shi et al., 2010), etc.; 3) traditional optimization techniques: quasi-Newton sequential quadratic programming (SQP) (Plevris & Papadrakakis, 2010), Nelder-Mead (NM) simplex search method (Fan & Zahara, 2007), discrete Lagrange multipliers method (Mohammad Nezhad & Mahlooji, 2011).

The large amount of emerging literature implies that PSO has increasingly gained popularity. This is supported by extensive experimental studies (Elbeltagi et al., 2005; Hassan et al., 2005; Kennedy & Spears, 1998) which have demonstrated that PSO may outperform other population-based evolutionary algorithms including genetic algorithms (GA), Memetic algorithms (MA), differential evolution (DE), ant-colony optimization (ACO) and shuffled frog leaping (SFL) in terms of solution quality and computational efficiency on some optimization problems. While promising, it is noted most existing PSO algorithms are designed for a specific search space (e.g. multi-modal). For example, the comprehensive learning strategy implemented in CLPSO (Liang et al., 2006) achieves high-quality performance on complex multi-modal functions due to its effectiveness in avoiding local optima while its convergence rate on uni-modal functions is unsatisfactory. To my knowledge, there currently lacks a generalized PSO algorithm that performs well across diverse search spaces with different characteristics, such as uni-modal, multi-modal, non-separable, shifted, rotated, noisy, and mis-scaled. Secondly, extensive researches have investigated ways to increase the diversity of the swarm to eliminate premature convergence (Andrews, 2006; Angeline, 1998; Chen et al., 2007; Garcia-Villoria & Pastor, 2009; Higashi & Iba, 2003; Liang et al., 2006; Thangaraj et al., 2009; van den Bergh & Engelbrecht, 2004).

This chapter aims to develop a generalized PSO algorithm that is efficient for a diverse set of optimization problems. First an intelligent selection approach is developed to identify the appropriate search method to be used based on the quantitative measure of its performance. Two search techniques are studied: a non-uniform mutation-based method (Michalewicz, 1996) and an extension of a sub-gradient method (Boyd, 2010). Next, an extended Cauchy mutation operator (Andrews, 2006) is employed to maintain the diversity of the swarm to prevent premature convergence. As a result, a novel PSO termed augmented PSO with multiple adaptive methods (PSO-MAM) is developed. Extensive comparison experiments are conducted to demonstrate the efficacy of PSO-MAM.

This chapter is organized as follows: several existing PSO algorithms are briefly reviewed in section 3.2; followed by the detailed explanation on the PSO-MAM in section 3.3; the experimental results in section 3.4 demonstrate the effectiveness of the PSO-MAM algorithm. Finally, conclusions are drawn in section 3.5.

3.2 Literature Review

In the PSO with inertia weight, the velocity and position for particle *p* at iteration *i* are updated as (Shi & Eberhart, 1998),

$$\mathbf{v}_p^{i+1} = w\mathbf{v}_p^i + c_1 r_{1,p}^i \times \left(\mathbf{p}_p^i - \mathbf{x}_p^i\right) + c_2 r_{2,p}^i \times \left(\mathbf{p}_g^i - \mathbf{x}_p^i\right) \tag{3.1}$$

$$\mathbf{x}_p^{i+1} = \mathbf{x}_p^i + \mathbf{v}_p^{i+1} \tag{3.2}$$

where D-dimensional vector $\mathbf{v}_p^i$ is the velocity of the p^{th} particle ($\mathbf{v}_p^i \in \left[-\mathbf{V}_{\max}, +\mathbf{V}_{\max}\right]$), $\mathbf{V}_{\max}$ is used to constrain the velocity for each particle and is usually set between 0.1 and 1.0 times the search range of the solution space (Banks et al., 2007); D-dimensional vector $\mathbf{x}_p^i$ is the position of the p^{th} particle; $\mathbf{p}_p^i$ is the best position found so far by the p^{th} particle; $\mathbf{p}_g^i$ is the best position found so far by the swarm; $r_{1,p}^i$ and $r_{2,p}^i$ represent two independent random numbers uniformly distributed on [0, 1]; c_1 is the cognitive learning factor which represents the attraction that a particle has toward its own success $\mathbf{p}_p^i$; c_2 is the social learning factor which represents the attraction that a particle has toward the swarm's best position $\mathbf{p}_g^i$; w is the inertia weight. Over the last decade, many different PSO algorithms have been developed to improve performance of the PSO and are reviewed in the following sections.

3.2.1 PSO Variants

The first area of research concentrates on the PSO formulation (Eqs. (3.1)-(3.2)). Shi and Eberhart (1998) add a positive parameter termed as inertia weight w to the original version of PSO (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995) to balance the capability of local search and global search of PSO. Clerc and Kennedy (2002) introduce constriction coefficients to prevent explosion and guarantee convergence of particles. Different neighborhood topologies are studied in (Kennedy, 1999), and it is found that a large neighborhood may perform better on simple problems and small neighborhoods may be preferred by complex problems. Parsopoulos and Vrahatis (2004) develop a unified particle swarm optimizer (UPSO) by combining the local version PSO with the global version PSO. Mendes et al. (2004) assume that the particle will be affected by all particles in its neighborhood and develop a fully informed

PSO (FIPSO). In FIPSO, particle's velocity is updated using information for all the particles instead of the best one of its neighbors. In the fitness-distance-ratio-based PSO (FDR-PSO) (Peram et al., 2003), a new velocity component based on one additional selected particle which has higher fitness values and is closer to the updated particle is added in the velocity update equation. CPSO-H (van den Bergh & Engelbrecht, 2004) uses a one-dimensional swarm to search on each dimension separately and then employs a global swarm to integrate these *D* one-dimensional swarms together. DMS-PSO (Liang & Suganthan, 2005) divides the swarm into several small swarms dynamically, exchanges information between these swarms and uses various strategies to regroup them frequently.

Another area of focus is to explore the learning strategies for each particle. In CLPSO (Liang et al., 2006), a comprehensive learning strategy is developed to ensure that every particle's personal best position could be learned by other particles with probability. This learning strategy can keep the diversity of the swarm and eliminate premature convergence. Wang et al. (2011) investigate integrating a generalized opposition-based learning (GOBL) strategy with PSO where GOBL is employed to render diversified particles. Four PSO based search approaches are simultaneously utilized in SLPSO (Wang et al., 2010) with one being selected based on a probability derived from a self-adaptively improved probability model. ELPSO (Huang et al., 2010) may be one of the first few considering a set of (instead of one) global best particles. By doing so, particles can learn from different global best particles which helps avoid premature convergence. A first-in-first-out order strategy is employed to update the example set when it exceeds its capacity. The particles learn from the global best and personal best via orthogonal experimental design in OLPSO (Zhan et al., 2010).

In summary, most PSO variants reviewed above are uniquely designed for some specific complex problems (e.g. multi-modal functions (Huang et al., 2010; Liang et al., 2006; Liang & Suganthan, 2005; Wang et al., 2011)). For improved generalization, researchers are exploring the integration of PSO with other methods which are reviewed in the following section.

3.2.2 Hybrid PSO

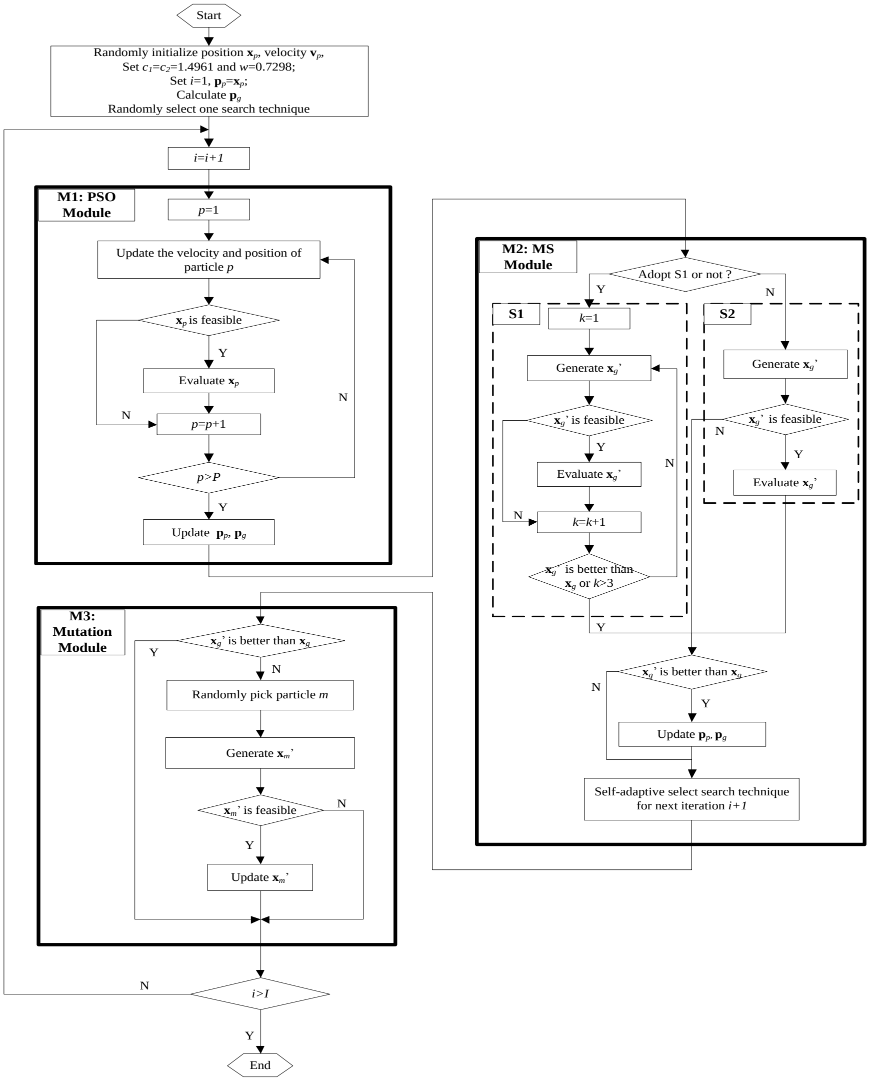
By employing Gaussian mutation into the PSO, Higashi and Iba (2003) observe that PSO with a mutation operator outperforms either Genetic Algorithm (GA) or PSO alone on uni-modal and multi-modal functions. Thangaraj et al. (2009) utilize Beta mutation to maintain the diversity of the swarm and improve the performance of PSO. Andrews (2006) studies the impact of different mutation operators on different test functions. Although a mutation operator can keep diversity of the swarm, selection of the mutation operator in PSO depends on the nature of the optimization problem.

Integration of other evolutionary algorithms and optimization techniques with PSO is also of interest. For example, Kao and Zahara (2008) develop a method which combines GA with PSO for multi-modal function optimization, and demonstrate the superiority of the hybrid method in terms of solution quality and convergence rate by using 17 multi-modal test functions. The cellular automata (CA) is integrated with PSO in the velocity update to avoid premature convergence in (Shi et al., 2010). Integrating PSO with a gradient-based quasi-Newton sequential quadratic programming (SQP), Plevris and Papadrakakis (2010) demonstrate this hybrid method outperforms other existing optimization techniques for global structural optimization. Fan and Zahara (2007) explore the integration of PSO with the Nelder-Mead (NM) simplex search method for unconstrained optimization. A method based on PSO and discrete Lagrange multipliers is implemented for nonlinear programming problems and is demonstrated to be very efficient and robust in (Mohammad Nezhad & Mahlooji, 2011).

Integrating PSO with other techniques has greatly strengthened PSO's capability for solving both uni-modal and multi-modal functions. Unfortunately, the performance on some complex problems (e.g., rotated, noisy, mis-scaled) is unsatisfactory (Fan & Zahara, 2007; Kao & Zahara, 2008; Thangaraj et al., 2009). This is probably due to the low convergence speed and/or poor exploitation capability of the techniques integrated with PSO. Another issue is integrating PSO with other techniques tends to be difficult for implementation, and is more computationally expensive compared to the PSO variants reviewed in section 3.2.1. Therefore, several computational efficient techniques are studied in this chapter including search techniques for good exploration and exploitation, and mutation to avoid premature convergence. The integration of these complement techniques will result in an improved PSO.

3.3 PSO-MAM Algorithm

As discussed in previous sections, PSO has two criticisms: 1) PSO and most of its variants are not guaranteed to perform well on a diverse set of optimization problems; 2) it suffers premature convergence. Due to the diversity of the search space for different optimization problems, the first issue commonly exists in most optimization algorithms. To address this issue, a model fusion approach is developed, that is, multiple search methods are applied and the one with good performance, measured by the effectiveness index (see section 3.3.1.3) will be triggered. In this chapter, for demonstration purpose, two single solution based (non-population based) search techniques are studied (detailed in section 3.3.1): a non-uniform mutation-based method which may be preferred by multi-modal functions due to its capability to explore the search space at the early stage (Michalewicz, 1996), and an adaptive sub-gradient method which may be effective for uni-modal functions due to its capability to quickly find a local optimum and fine tune the search space (Boyd, 2010). To address the second issue, the use of an extended Cauchy mutation operator (see section 3.3.2) is developed to prevent premature convergence in the intelligent multiple search methods enhanced PSO.



Start
Randomly initialize position $\mathbf{x}_p$, velocity $\mathbf{v}_p$,
Set c_1=c_2=1.4961 and w=0.7298;
Set i=1, $\mathbf{p}_p$=$\mathbf{x}_p$;
Calculate $\mathbf{p}_g$
Randomly select one search technique
i=i+1
M1: PSO Module
p=1
Update the velocity and position of particle p
$\mathbf{x}_p$ is feasible
Y
Evaluate $\mathbf{x}_p$
N
p=p+1
N
p>P
Y
Update $\mathbf{p}_p$, $\mathbf{p}_g$
M2: MS Module
Adopt S1 or not ?
Y
N
S1
k=1
Generate $\mathbf{x}_g$'
$\mathbf{x}_g$' is feasible
Y
Evaluate $\mathbf{x}_g$'
N
N
k=k+1
$\mathbf{x}_g$' is better than $\mathbf{x}_g$ or k>3
Y
S2
Generate $\mathbf{x}_g$'
$\mathbf{x}_g$' is feasible
N
Y
Evaluate $\mathbf{x}_g$'
$\mathbf{x}_g$' is better than $\mathbf{x}_g$
N
Y
Update $\mathbf{p}_p$, $\mathbf{p}_g$
Self-adaptive select search technique for next iteration i+1
M3: Mutation Module
$\mathbf{x}_g$' is better than $\mathbf{x}_g$
Y
N
Randomly pick particle m
Generate $\mathbf{x}_m$'
$\mathbf{x}_m$' is feasible
N
Y
Update $\mathbf{x}_m$'
N
i>I
Y
End

Figure 12 Flowchart of PSO-MAM ("S1": non-uniform mutation-based method; "S2": sub-gradient method)

As shown in Figure 12, the PSO-MAM has three modules: 1) PSO module: The swarm is randomly initialized with the PSO operator being employed to update the swarm. 2) Intelligent multiple search methods module: two search methods (non-uniform mutation-based method and sub-gradient method) are implemented. At each iteration, an appropriate search method will be triggered using the roulette wheel selection (see section 3.3.1.3). 3) Mutation module: after the further improvement on the best particle, the mutation operator is used to update one randomly selected particle. The algorithm will stop if the stopping criterion (such as the maximum number of PSO iterations, predefined solution accuracy) is satisfied.

3.3.1 Intelligent Multiple Search Methods

Like most of the existing optimization algorithms, PSO is not guaranteed to be effective for different optimization problems. Therefore, an intelligent multiple search methods module is employed to assist PSO and improve its effectiveness for different problems. After each PSO iteration, the multi-method search is adopted to improve particle $\mathbf{x}_g^i$ in the *current* swarm where *g* is the index of $\mathbf{p}_g^i$

$$g = \left\{ p : \mathbf{p}_p^i = \operatorname{argmin}_{p=1,\ldots,P} \left\{ f\left(\mathbf{p}_p^i\right) \right\} \right\} \tag{3.3}$$

The solution $\mathbf{x}_g^i$ will be replaced if it is improved by the multiple search methods module.

3.3.1.1 Non-uniform Mutation-based Method

In the multi-method search module, one search technique studied is the non-uniform mutation-based method (Michalewicz, 1996) which is good at searching the solution space uniformly (exploration) at the early stage and very locally (exploitation) at the later stage (Zhao, 2011). The non-uniform mutation-based method has been demonstrated to have the merits of large jumping (exploration) and fine-tuning (exploitation) (Zhao, 2011; Zhao et al., 2007). Due to its good balance between exploration and exploitation, non-uniform mutation-based method may be preferred by multi-modal functions.

In the non-uniform mutation-based method, the dth dimension of the solution $\mathbf{x}_g^i$ is randomly picked to be mutated to generate a new solution as

$$x_{g,d}^i{}' = \begin{cases} x_{g,d}^i + \Delta\left(i, U_d - x_{g,d}^i\right) & \text{if } r \geq 0.5 \\ x_{g,d}^i - \Delta\left(i, x_{g,d}^i - L_d\right) & \text{if } r < 0.5 \end{cases} \tag{3.4}$$

where i is the current iteration index of PSO; U_d and L_d are the upper and lower bounds of $x_{g,d}^i$; r is a uniform random number from (0, 1). The function $\Delta(i,y)$ is defined as

$$\Delta\left(i, y\right) = y \times \left(1 - \rho^{(1-i/I)^b}\right) \tag{3.5}$$

where ρ is a uniform random number from (0, 1); I is the maximum number of iterations for PSO; b is a system parameter determining the degree of dependency on iteration number (non-uniformity). In this chapter, b is set as 1.

3.3.1.2 Adaptive Sub-gradient Method

The sub-gradient method (Boyd, 2010) is extended with adaptive step size derived from the particle velocity information. The sub-gradient method for unconstrained problems is equivalent to the gradient based method when the objective function is differentiable. Like the gradient based method, the sub-gradient method could find a local optimum very fast and exhibit good exploitation capability (Plevris & Papadrakakis, 2010). Therefore, the sub-gradient method could strengthen PSO’s search capability for uni-modal functions. In the sub-gradient method, a new solution $\mathbf{x}_g^i{}'$ is generated as

$$\mathbf{x}_g^i{}' = \mathbf{x}_g^i - \alpha_i \boldsymbol{\gamma}_g^i \tag{3.6}$$

where $\boldsymbol{\gamma}_g^i$ is the sub-gradient of the objective function; α_i is the step size. The sub-gradient of the objective function is evaluated as the gradient of the function if the gradient is available, otherwise the sub-gradient will be approximated by the

Simultaneous Perturbation Stochastic Approximation (SPSA) method (Spall, 1992). The solution generated by Eq. (3.6) may not be effective, such as, $\boldsymbol{\gamma}_g^i$ may be too small at the early stage which could trap the solution in the local optimum too early (e.g., Griewank function, Salomon function), and $\boldsymbol{\gamma}_g^i$ may be too large at the later stage which will make the exploitation speed of sub-gradient method slower than PSO's exploitation speed (e.g., Schwefel P2.22 function). Therefore, the velocity $\mathbf{v}_g^i$ of particle $\mathbf{x}_g^i$ is adopted to constrain the sub-gradient $\boldsymbol{\gamma}_g^i$ to avoid this ineffectiveness and balance the sub-gradient method's exploration and exploitation capability. In this chapter, instead of using a pre-defined step size, the step size α_i according to $\mathbf{v}_g^i$ is defined, that is,

$$\alpha_i = \left\|\mathbf{v}_g^i\right\|_2 \Big/ \left\|\boldsymbol{\gamma}_g^i\right\|_2 \tag{3.7}$$

where $\|\mathbf{x}\|_2$ is the Euclidean norm of vector **x**.

3.3.1.3 Intelligent Selection Strategy

To implement the intelligent selection, the effectiveness index η_s^i is introduced which measures the performance of the s^{th} search method at iteration i:

$$\eta_s^i = \begin{cases} \left(f_g^i - f_g^i{}'\right) \big/ \left|f_g^i\right| & s^{th} \text{ search method is used at iteration } i \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

where f_g^i and $f_g^i{}'$ are the fitness value of $\mathbf{p}_g^i$ before and after the multi-method search respectively (please note a minimization problem is studied in this chapter). An execution probability $prob_s^i$ is assigned to each search method to determine the probability that the s^{th} method will be adopted for the following iteration i+1. The execution probability $prob_s^i$ is calculated as

$$prob_s^i\,' = \begin{cases} \sum_{ii=1}^{i} \eta_s^{ii} \big/ n_s & if\;\; n_s \geq N,\; \forall s = 1,\ldots,S \\ 1/S & otherwise \end{cases} \tag{3.9}$$

$$prob_s^i = prob_s^i\,' \Big/ \sum_{s=1}^{S} prob_s^i\,' \tag{3.10}$$

where n_s is the number of iterations that the s^{th} method is adopted; I is the maximum number of PSO iterations; S is the number of search methods implemented in PSO-MAM which is 2 in this chapter; N is the minimal required execution number for each search method which is set as 50 in this chapter. It is observed from Eq. (3.9) that each search method has an equal probability to be selected at the early stage, and the effective search method tends to be preferred iteration by iteration. The effective search method is selected by roulette wheel selection based on the search method's execution probability.

3.3.2 Cauchy Mutation

As shown in Figure 12, for the cases that a new solution is introduced by the search methods, the mutation module is ignored. Otherwise, extended Cauchy mutation is employed to increase the diversity of the swarm. The Cauchy mutation operator is studied here due to its capability in generating a larger range of jump steps compared to other operators, e.g., Gaussian mutation. In the extended Cauchy mutation operator, a randomly selected dimension d of a randomly selected particle m will be mutated as

$$x_{m,d}^i\,' = x_{m,d}^i + cauchy\left(\delta_i\right) \tag{3.11}$$

where δ_i is the scale parameter of Cauchy distribution. A notable issue of the constant scale parameter is that the mutation scale may be too large at the later stage which will impair the exploitation capability (Yao et al., 1999). Secondly, the mutation of the particle may not be consistent with the scale of the particle movement. Thus, in this chapter, δ_i should have two properties: 1) ensure the magnitude of the mutation is at the same scale as the particle movement; 2) enable larger moves at the earlier stage and smaller moves as the process evolves. At iteration *i*, the velocity vector $\mathbf{v}_p^i$ of each particle *p* is obtained. Let $\left\|\mathbf{v}_p^i\right\|_2$ denotes the moving distance for particle *p*, then a *P* dimensional moving vector is generated as $\left[\left\|\mathbf{v}_1^i\right\|_2,\ldots,\left\|\mathbf{v}_P^i\right\|_2\right]$. To satisfy the first property,

$$\delta_i \propto \left[\left\|\mathbf{v}_1^i\right\|_2,\ldots,\left\|\mathbf{v}_P^i\right\|_2\right] \tag{3.12}$$

To satisfy the second property,

$$k = \left(P \times \left(i/I\right)^{2-i/I}\right) \tag{3.13}$$

as *i* increases, *k* nonlinearly increases from 1 to *P*. δ_i is set as the k^{th} largest component from vector $\left[\left\|\mathbf{v}_1^i\right\|_2,\ldots,\left\|\mathbf{v}_P^i\right\|_2\right]$. As a result, at the current iteration *i*, the jump magnitude is at the same scale as the particles movement. Secondly, the use of *k* ensures the mutation jump is picked in accordance with *i*.

3.4 Experimental Analysis

To fully test the performance of PSO-MAM, 31 functions are collected from (Iwasaki et al., 2006; Liang et al., 2006; Salomon, 1996; Wang et al., 2010; Yao et al., 1999) and 10 PSO methods from the literature are implemented for comprehensive comparisons. The general formulas of these 31 test functions are listed in Appendix A where $\mathbf{z} = \mathbf{M} \times \left(\mathbf{x} - \mathbf{o}\right)$, **o** is employed to shift the global optimal solution of the original function from the center of the search range to a new location and the orthogonal rotated matrix (Salomon, 1996) **M** is used to increase the complexity of the function by changing separable functions to non-separable functions without altering the shape of the function. Vector **o** is 0 if the function is non-shifted and matrix **M** is a *D*-dimensional identity matrix if the function is non-rotated. The function name, search range, optimal solution and features are described in Table 4. For fair comparison, the complexity of the sub-gradient calculation is measured by the equivalent number of function evaluations for the sub-gradient calculation (SGFE) which is determined as the total number of floating point operations (FLOP) for the *D*-dimensional sub-gradient calculations divided by the floating point operations (FLOP) of the objective function evaluation.

$$SGFE = \left\lceil \sum\nolimits_{d=1}^{D} FLOP_{\gamma_{g,d}} \Big/ FLOP_f \right\rceil \tag{3.14}$$

where FLOP is the output of “flops” function in MATLAB®; $FLOP_{\gamma_{g,d}}$ is the number of floating point operations for calculating the sub-gradient of the objective function *f* on $x_{g,d}$; $FLOP_f$ is the number of floating point operations for calculating the objective function *f*.

The 31 test functions in Table 4 are divided into six groups (see section 3.4.3): 1) 6 uni-modal non-rotated functions (f_1~f_6); 2) 6 uni-modal rotated functions (f_7~f_{12}); 3) 11 multi-modal non-rotated functions (f_{13}~f_{23}); 4) 4 multi-modal rotated functions (f_{24}~f_{27}); 5) 2 noisy functions (f_{28}~f_{29}); 6) 2 mis-scaled functions (f_{30}~f_{31}).

Table 4 Features of the 31 test functions (Note: “Md” denotes “modality”; “U” denotes “uni-modal”; “M” denotes “multi-modal”; “Sp” denotes “separable”; “Sf” denotes “shifted”; “Rt” denotes “rotated”; “Ny” denotes “noisy”; “Ms” denotes “Mis-scaled”)

No	**Func. Name**	**Search**	**Opt.**	**Features**					
.		**Range**	**Solution z***	Md	Sp	Sf	Rt	Ny	Ms
f_1	Shifted Sphere	$[-100, 100]^D$	**0**	U	Y	Y	N	N	N
f_2	Shifted Schwefel P2.22	$[-10, 10]^D$	**0**	U	N	Y	N	N	N
f_3	Shifted Schwefel P1.2	$[-100, 100]^D$	**0**	U	N	Y	N	N	N
f_4	Shifted Schwefel P2.21	$[-100, 100]^D$	**0**	U	Y	Y	N	N	N
f_5	Shifted Rosenbrock	$[-100, 100]^D$	**1**	U	N	Y	N	N	N
f_6	Shifted Step	$[-100, 100]^D$	**[-0.5, 0.5)**	U	Y	Y	N	N	N
f_7	Shifted Rotated Sphere	$[-100, 100]^D$	**0**	U	N	Y	Y	N	N
f_8	Shifted Rotated Schwefel P2.21	$[-100, 100]^D$	**0**	U	N	Y	Y	N	N
f_9	Shifted Rotated Rosenbrock	$[-100, 100]^D$	**1**	U	N	Y	Y	N	N
f_{10}	Shifted Rotated Tablet	$[-100, 100]^D$	**0**	U	N	Y	Y	N	N
f_{11}	Shifted Rotated Ellipse	$[-100, 100]^D$	**0**	U	N	Y	Y	N	N
f_{12}	Shifted Rotated Diff Power	$[-100, 100]^D$	**0**	U	N	Y	Y	N	N
f_{13}	Schwefel	$[-500, 500]^D$	**420.9687**	M	Y	N	N	N	N
f_{14}	2^D minima	$[-5, 5]^D$	**-2.9035**	M	Y	N	N	N	N
f_{15}	Shifted Rastrigin	$[-5, 5]^D$	**0**	M	Y	Y	N	N	N
f_{16}	Shifted Noncontinuous Rastrigin	$[-5, 5]^D$	**0**	M	Y	Y	N	N	N
f_{17}	Shifted Ackley	$[-32, 32]^D$	**0**	M	N	Y	N	N	N
f_{18}	Shifted Griewank	$[-600, 600]^D$	**0**	M	N	Y	N	N	N
f_{19}	Weierstrass	$[-0.5, 0.5]^D$	**0**	M	Y	N	N	N	N

f_{20}	Shifted Salomon	$[-100, 100]^D$	**0**	M	N	Y	N	N	N
f_{21}	Schwefel P2.13	$[-\pi, \pi]^D$	**α**	M	N	N	N	N	N
f_{22}	Shifted Penalized 1	$[-50, 50]^D$	**-1**	M	N	Y	N	N	N
f_{23}	Shifted Penalized 2	$[-50, 50]^D$	**1**	M	N	Y	N	N	N
f_{24}	Rotated 2^D minima	$[-5, 5]^D$	**-2.9035**	M	N	N	Y	N	N
f_{25}	Shifted Rotated Griewank	$[-600, 600]^D$	**0**	M	N	Y	Y	N	N
f_{26}	Rotated Weierstrass	$[-0.5, 0.5]^D$	**0**	M	N	N	Y	N	N
f_{27}	Shifted Rotated Salomon	$[-100, 100]^D$	**0**	M	N	Y	Y	N	N
f_{28}	Shifted Noise Schwefel P1.2	$[-100, 100]^D$	**0**	U	N	Y	N	Y	N
f_{29}	Shifted Rotated Noise Quadric	$[-100, 100]^D$	**0**	U	N	Y	Y	Y	N
f_{30}	Shifted Rastrigin10	$[-5, 5]^D$	**0**	M	Y	Y	N	N	Y
f_{31}	Shifted Rastrigin100	$[-5, 5]^D$	**0**	M	Y	Y	N	N	Y

3.4.1 Parameter Settings for the Compared PSO Algorithms

Experiments are conducted to compare the performance of 10 existing PSO algorithms in the literature with PSO-MAM. The compared PSO algorithms and their parameter settings are:

1) PSO with inertia weight (PSO-w) (Shi & Eberhart, 1998): w=0.9-0.5i/I, c_1=c_2=2;

2) PSO with constriction factor (PSO-cf) (Clerc & Kennedy, 2002): w=0.729, c_1=c_2=1.49445;

3) Local version of PSO with inertia weight (PSO-w-local) (Kennedy & Mendes, 2002): w=0.9-0.5i/I, c_1=c_2=2;

4) Local version of PSO with constriction factor (PSO-cf-local) (Kennedy & Mendes, 2002): w=0.729, c_1=c_2=1.49445;

5) Unified PSO (UPSO) (Parsopoulos & Vrahatis, 2004): w=0.729, c_1=c_2=1.49445;

6) Weighted fully informed particle swarm (wFIPS) (Mendes et al., 2004): w=0.729, c_1=c_2=2;

7) Fitness-Distance-Ratio based PSO (FDR-PSO) (Peram et al., 2003): w=0.9-0.5i/I, f_1=f_2=1, f_3=2;

8) Cooperative PSO (CPSO-H) (van den Bergh & Engelbrecht, 2004): w=0.9-0.5i/I, c_1=c_2=1.49;

9) Comprehensive learning PSO (CLPSO) (Liang et al., 2006): *w*=0.9-0.7*i*/*I*, c_1=c_2=1.49445;

10) Dynamic multi-swarm PSO (DMS-PSO) (Liang & Suganthan, 2005): c_1=c_2=1.49445, *w*=0.729, *n*=6, *m*=5;

11) PSO with multiple adaptive methods (PSO-MAM): c_1=c_2=1.4961, *w*=0.7298; where *i* is the current iteration index, and *I* is the maximum number of iteration. Additional parameter settings for PSO-w, PSO-cf, PSO-w-local, PSO-cf-local, UPSO, wFIPS, FDR-PSO, CPSO-H and CLPSO are the same as (Liang et al., 2006).

3.4.2 Performance Metrics

To evaluate the overall performance in regards to both the solution quality and computing cost, the metrics success performance (SP) and success rate (SR) are adopted from (Auger & Hansen, 2005). For the cases where SPs are not available, the fitness value (Auger & Hansen, 2005) is used. The mean value of the fitness value, mean value of SP, and SR over 30 independent runs are recorded. A run during which the algorithm achieves a solution at the fixed accuracy level within the maximum number of function evaluations is considered to be successful. In this example, the accuracy level is set to be 10^{-5}. The success rate (SR) is defined as

$$SR = \#\text{ of successful runs/total \# of runs} \tag{3.15}$$

The success performance (SP) is the number of function evaluations for the algorithm to reach the fixed accuracy level. The mean of SP is defined as (Auger & Hansen, 2005)

$$\text{mean}\left(SP\right) = \left(\left(1-SR\right)/SR\right)FE_{\max} + \text{mean}\left(\#\text{ of func. eval. for successful runs}\right) \tag{3.16}$$

where FE_{max} is the maximum number of function evaluations.

3.4.3 Comparison Experiments for 30 Dimensional Functions

This section attempts to test PSO-MAM's capability for a diverse set of optimization problems. The 31 test functions studied in this section have 30 dimensions, the population size is set to be 30 and the maximum number of function evaluations is set to be 300,000. The statistical comparison of the PSO-MAM with the other ten PSO algorithms uses a two-tailed *t*-test with 58 (2x30-2) degrees of freedom at a 0.05 level of significance. The *t*-test of two compared algorithms is based on the success performance if at least one algorithm has 100% success rate. Otherwise the *t*-test is based on the fitness value. Values "+", "=" and "-" in the column "h" in Table 5~10 denote PSO-MAM performs significantly better than, almost the same as, and significantly worse than the compared algorithm, respectively.

3.4.3.1 Uni-modal Non-rotated Functions

In the first set of experiments, 6 uni-modal and non-rotated functions are studied (see Table 4). The optimization results are summarized in Table 5. Please note column SP is blank when there is no successful run among the 30 runs (SR=0).

Table 5 Optimization results for uni-modal non-rotated functions

Algorithms	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**
	Shifted Sphere (f_1)				Shifted Schwefel P2.22 (f_2)				Shifted Schwefel P1.2 (f_3)			
PSO-w	2.21E-28	176162	100	+	9.55E-16	176346	100	+	2.47E-02		0	+
PSO-cf	1.77E-27	14517	100	+	1.41E-13	28835	100	=	1.73E-22	103109	100	+
PSO-w-local	2.02E-27	219943	100	+	5.63E-16	221545	100	+	2.80E+03		0	+
PSO-cf-local	6.84E-30	24389	100	+	0.00E+00	31038	100	=	1.16E-09	198400	100	+
UPSO	0.00E+00	15708	100	+	**0.00E+00**	**20956**	**100**	-	5.47E-11	183693	100	+
wFIPS	5.21E-27	79319	100	+	2.46E-14	107719	100	+	1.91E+00		0	+
FDR-PSO	1.26E-30	99527	100	+	0.00E+00	101755	100	+	2.55E-18	186429	100	+
CPSO-H	2.42E-12	95478	100	+	2.71E-07	201064	100	+	4.96E+03		0	+
CLPSO	0.00E+00	92887	100	+	0.00E+00	107976	100	+	2.17E+02		0	+
DMS-PSO	7.15E-30	24082	100	+	0.00E+00	30186	100	=	1.10E+00		0	+
PSO-MAM	**0.00E+00**	**1823**	**100**		0.00E+00	28252	100		**3.58E-27**	**48344**	**100**	
	Shifted Schwefel P2.21 (f_4)				Shifted Rosenbrock (f_5)				Shifted Step (f_6)			
PSO-w	2.86E-01		0	+	4.97E+01		0	+	1.00E-01	186372	90	+
PSO-cf	3.08E-10	167606	100	+	9.74E+00		0	+	1.56E+01		0	+
PSO-w-local	1.18E+00		0	+	6.42E+01		0	+	0.00E+00	191646	100	+
PSO-cf-local	2.16E-09	186927	100	+	1.38E+01		0	+	0.00E+00	12408	100	+
UPSO	1.54E-05	571530	50	+	1.13E+01		0	+	0.00E+00	8943	100	+
wFIPS	4.97E-05		0	+	2.85E+01		0	+	0.00E+00	33422	100	+
FDR-PSO	4.11E-04		0	+	1.46E+00	8975940	3.33	+	3.00E-01	169571	76.7	+
CPSO-H	7.98E-05		0	+	2.95E+01		0	+	0.00E+00	8709	100	+
CLPSO	5.35E-01		0	+	3.71E+00		0	+	0.00E+00	55895	100	+
DMS-PSO	9.53E-12	154354	100	+	2.80E+01		0	+	0.00E+00	11696	100	+
PSO-MAM	**5.77E-16**	**5953**	**100**		**5.33E-28**	**57994**	**100**		**0.00E+00**	**853**	**100**	

It is observed that PSO-MAM achieves the best success rate and convergence speed (success performance) for 5 out of 6 the uni-modal non-rotated functions. Compared to other PSO algorithms, PSO-MAM is the most reliable algorithm (shown from 100% success rates for six functions) and can achieve a satisfactory result quickly (shown from small values of success performance). Starting with the good solution derived by the PSO module, the intelligent multiple search methods module can exploit the solution space of the uni-modal function to fine-tune the solution quickly. The performance of PSO-MAM on shifted Schwefel P2.22 (f_2) is comparable with UPSO.

3.4.3.2 Uni-modal Rotated Functions

In the second set of experiments, 6 uni-modal rotated functions are tested (see Table 4). The optimization results are summarized in Table 6.

Table 6 Optimization results for uni-modal rotated functions

Algorithms	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**
	Shifted Rotated Sphere (f_7)				Shifted Rotated Schwefel P2.21 (f_8)				Shifted Rotated Rosenbrock (f_9)			
PSO-w	1.96E-28	176633	100	+	4.72E-03		0	+	6.26E+02		0	+
PSO-cf	6.40E-29	14618	100	+	5.87E-13	128094	100	+	4.94E+02		0	=
PSO-w-local	2.46E-27	220891	100	+	1.45E-01		0	+	4.23E+02		0	+
PSO-cf-local	5.05E-30	24307	100	+	4.28E-13	135275	100	+	8.94E+01		0	+
UPSO	8.41E-31	15833	100	+	4.10E-08	196394	100	+	3.95E+01		0	+
wFIPS	4.98E-27	79141	100	+	1.16E-06	262428	100	+	5.67E+01		0	+
FDR-PSO	0.00E+00	99737	100	+	8.49E-07	237174	96.7	+	2.40E+01		0	+
CPSO-H	3.05E-12	93277	100	+	5.52E+00		0	+	2.54E+02		0	=
CLPSO	0.00E+00	93487	100	+	1.05E-01		0	+	2.99E+01		0	+
DMS-PSO	7.15E-30	24111	100	+	1.03E-12	143619	100	+	4.35E+01		0	+
PSO-MAM	**0.00E+00**	**1903**	**100**		**0.00E+00**	**5816**	**100**		**3.99E-01**	**83480**	**90**	
	Shifted Rotated Tablet (f_{10})				Shifted Rotated Ellipse (f_{11})				Shifted Rotated Diff Power (f_{12})			
PSO-w	4.47E+02		0	+	2.49E-04	2990170	10	+	7.78E-06	229806	96. 7	+
PSO-cf	2.87E+01		0	+	3.39E-25	62506	100	+	2.18E-13	26680	100	+
PSO-w-local	1.89E+03		0	+	4.87E+04		0	+	1.95E+06	864481	33.3	+
PSO-cf-local	2.87E+02		0	+	2.43E-17	129467	100	+	4.92E-13	39895	100	+
UPSO	1.06E+03		0	+	1.01E-21	108496	100	+	1.19E-13	30406	100	+
wFIPS	1.21E+03		0	+	8.83E-03		0	+	8.44E-11	102229	100	+
FDR-PSO	2.20E+02		0	+	1.22E-25	163049	100	+	3.91E-14	103330	100	+
CPSO-H	1.52E+04		0	+	4.53E+03		0	+	1.13E+07	1043833	23. 3	+
CLPSO	4.68E+02		0	+	1.08E+02		0	+	7.62E-09	173461	100	+
DMS-PSO	3.93E+01		0	+	4.15E-10	212431	100	+	4.03E-12	45216	100	+
PSO-MAM	**0.00E+00**	**2777**	**100**		**2.62E-26**	**24077**	**100**		**8.56E-16**	**5014**	**100**	

For uni-modal rotated functions, PSO-MAM is the most reliable algorithm (shown from large success rates) and can achieve a satisfactory result quickly (shown from small mean values of success performance) especially on the shifted rotated Rosenbrock function (f_9) and the shifted rotated Tablet function (f_{10}). PSO-MAM outperforms other PSO algorithms on all 6 of these functions both in terms of convergence speed and solution quality.

3.4.3.3 Multi-modal Non-rotated Functions

In the third set of experiments, 11 multi-modal non-rotated functions are explored (see Table 4). The optimization results are summarized in Table 7.

Table 7 Optimization results for multi-modal non-rotated functions

Algorithms	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**
	Schwefel (f_{13})				2^D minima (f_{14})				Shifted Rastrigin (f_{15})			
PSO-w	1.09E+03		0	+	4.71E+00		0	+	2.01E+01		0	+
PSO-cf	2.92E+03		0	+	1.12E+01		0	+	7.26E+01		0	+
PSO-w-local	5.16E+03		0	+	4.57E-10	200131	100	+	2.93E+01		0	+
PSO-cf-local	2.18E+03		0	+	7.01E+00		0	+	4.23E+01		0	+
UPSO	3.65E+03		0	+	8.07E+00		0	+	6.87E+01		0	+
wFIPS	2.37E+01	220819	80	+	4.08E-01	269664	60	+	2.80E+01		0	+
FDR-PSO	3.10E+03		0	+	1.06E+01		0	+	2.84E+01		0	+
CPSO-H	2.37E+02	8783760	3.33	+	4.57E-10	37136	100	+	9.95E-02	117335	90	+
CLPSO	1.70E-08	95772	100	+	4.57E-10	75820	100	+	0.00E+00	159419	100	+
DMS-PSO	2.43E+03		0	+	3.61E+00	4228670	6.67	+	7.16E+00		0	+
PSO-MAM	**1.70E-08**	**67656**	**100**		**4.57E-10**	**17094**	**100**		**0.00E+00**	**62187**	**100**	
	Shifted Noncontinuous Rastrigin (f_{16})				Shifted Ackley (f_{17})				Shifted Griewank (f_{18})			
PSO-w	7.57E+00	2968630	10	+	3.10E-14	192348	100	+	2.12E-02	1004104	26.7	+
PSO-cf	4.08E+01		0	+	1.53E+00	1222345	20	+	2.44E-02	715564	30	+
PSO-w-local	1.66E+01		0	+	2.34E-14	239791	100	+	5.91E-03	442035	56.7	+
PSO-cf-local	6.07E+00	715855	36.7	+	6.39E-15	34510	100	+	8.19E-03	257348	56.7	+
UPSO	7.09E+01		0	+	3.55E-15	22997	100	+	1.89E-03	85077	83.3	+
wFIPS	4.16E+01		0	+	2.52E-14	117343	100	+	0.00E+00	98894	100	+
FDR-PSO	7.63E+00		0	+	1.85E-14	109428	100	+	1.26E-02	625812	36.7	+
CPSO-H	2.33E-01	158412	80	+	3.03E-07	199111	100	+	2.07E-02	621219	36.7	+
CLPSO	0.00E+00	167351	100	+	8.05E-15	118336	100	+	0.00E+00	121255	100	+
DMS-PSO	3.80E+00	377190	63. 3	+	3.55E-15	34502	100	+	0.00E+00	29281	100	+
PSO-MAM	**0.00E+00**	**58468**	**100**		**0.00E+00**	**20411**	**100**		**0.00E+00**	**3971**	**100**	
	Weierstrass (f_{19})				Shifted Salomon (f_{20})				Schwefel P2.13 (f_{21})			
PSO-w	1.01E-01	308081	73.3	-	3.81E-01		0	+	8.65E+04		0	+
PSO-cf	7.61E+00		0	+	5.90E-01		0	+	1.61E+05		0	+
PSO-w-local	0.00E+00	238863	100	-	3.13E-01		0	+	7.02E+04		0	+
PSO-cf-local	7.27E-01	874227	26.7	=	2.33E-01		0	+	6.26E+04		0	+
UPSO	4.22E-01	635110	33.3	=	5.37E-01		0	+	1.09E+05		0	+
wFIPS	0.00E+00	167853	100	-	2.00E-01		0	+	1.22E+04		0	=
FDR-PSO	1.07E-01	950523	26.7	=	3.50E-01		0	+	1.38E+04		0	=
CPSO-H	9.88E-05		0	+	1.37E+00		0	+	2.74E+04		0	+
CLPSO	0.00E+00	142057	100	-	2.32E-01		0	+	1.01E+04		0	=
DMS-PSO	**0.00E+00**	**48357**	**100**	-	2.00E-01		0	+	1.01E+04		0	=
PSO-MAM	9.50E-05	553944	43.3		**0.00E+00**	**11539**	**100**		**9.41E+03**		**0**	
	Shifted Penalized 1 (f_{22})				Shifted Penalized 2 (f_{23})							
PSO-w	2.07E-02	249245	80	+	2.20E-03	257914	80	+				
PSO-cf	3.74E-01	361848	46.7	+	2.76E-01	317006	50	+				
PSO-w-local	6.91E-03	246538	93.3	+	1.83E-03	301082	83.3	+				
PSO-cf-local	1.21E-01	224849	60	+	2.17E-03	84526	83.3	+				

UPSO	2.76E-02	86255	83.3	+	7.32E-04	39932	93.3	=
wFIPS	1.08E-29	61000	100	+	3.42E-28	69452	100	+
FDR-PSO	3.46E-03	99463	96.7	+	1.10E-03	124916	90	+
CPSO-H	1.98E-14	38407	100	+	1.61E-13	60603	100	+
CLPSO	1.57E-32	83945	100	+	1.35E-32	89144	100	+
DMS-PSO	1.57E-32	21791	100	+	5.56E-32	24968	100	+
PSO-MAM	**1.57E-32**	**15377**	**100**		**1.35E-32**	**13517**	**100**	

PSO-MAM outperforms in terms of convergence speed for 10 out 11 multi-modal non-rotated functions. The performance of PSO-MAM on Weierstrass (*f_{19}*) is inferior to some algorithms (e.g., CLPSO, DMS-PSO, etc.). In addition, PSO-MAM achieves 100% success rate for all the test functions except functions Weierstrass (*f_{19}*) and Schwefel P2.13 (*f_{21}*) which have much more complex search spaces. It is argued that such search spaces require a specifically designed PSO for high-quality performance. Another observation from the experiment is that the step size defined in Eq. (3.7) is effective for the two multi-modal functions shifted Griewank (*f_{18}*) and shifted Salomon (*f_{20}*) since the original sub-gradient method is very easy to trap in the local optimum at the early stage. By using velocity to adaptively change the sub-gradient, the adaptive sub-gradient method is very effective in exploring the search space at early stages without damaging the sub-gradient's exploitation capability at later stages.

3.4.3.4 Multi-modal Rotated Functions

In the fourth set of experiments, 4 multi-modal rotated functions are studied (see Table 4). The optimization results are recorded in Table 8.

Table 8 Optimization results for multi-modal rotated functions

Algorithms	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**
	Rotated 2^D minima (f_{24})				Shifted Rotated Griewank (f_{25})			
PSO-w	6.71E+00		0	=	1.32E-02	1012572	26.7	+
PSO-cf	1.02E+01		0	+	1.86E-02	408508	43.3	+
PSO-w-local	3.75E+00		0	-	7.71E-03	483820	53.3	+
PSO-cf-local	9.19E+00		0	+	6.65E-03	327566	50	+
UPSO	9.75E+00		0	+	3.38E-03	154359	70	+
wFIPS	2.08E+00		0	-	0.00E+00	99049	100	+
FDR-PSO	1.10E+01		0	+	1.34E-02	708387	33.3	+
CPSO-H	6.81E+00		0	=	3.42E-02	810937	30	+
CLPSO	**1.89E+00**		**0**	-	4.20E-10	143041	100	+
DMS-PSO	6.20E+00		0	-	6.57E-04	58795	93.3	+
PSO-MAM	7.75E+00		0		**0.00E+00**	**3824**	**100**	
	Rotated Weierstrass (f_{26})				Shifted Rotated Salomon (f_{27})			

PSO-w	7.33E+00		0	-	3.93E-01		0	+
PSO-cf	1.25E+01		0	+	5.97E-01		0	+
PSO-w-local	2.76E+00		0	-	3.01E-01		0	+
PSO-cf-local	7.40E+00		0	-	2.53E-01		0	+
UPSO	1.48E+01		0	+	5.47E-01		0	+
wFIPS	**6.31E-02**	**421625**	**60**	-	1.97E-01		0	+
FDR-PSO	2.44E+00		0	-	3.43E-01		0	+
CPSO-H	1.24E+01		0	+	1.30E+00		0	+
CLPSO	1.37E+00		0	-	2.24E-01		0	+
DMS-PSO	1.17E-01		0	-	1.97E-01		0	+
PSO-MAM	9.46E+00		0		**0.00E+00**	**11822**	**100**	

For the multi-modal rotated functions, PSO-MAM only outperforms on 2 out of 4 functions (shifted rotated Griewank function (f_{25}) and shifted rotated Salomon function (f_{27})) which is mainly due to the step size defined in Eq. (3.7). For the other two functions, the rotated matrix increases the complexity of the function by changing separable functions to non-separable functions, which makes it difficult for the search methods implemented in PSO-MAM to search the rotated space instead of the original space. It is observed that the well performing algorithms on these functions are all designed for the specific search space thus the algorithms perform well on these functions and in general perform poorly on most of the other functions (see section 3.4.3.7).

3.4.3.5 Noisy Functions

In the fifth set of experiments, 2 noisy functions are tested (see Table 4). The optimization results are recorded in Table 9. PSO-MAM outperforms other algorithms on these two noisy functions in terms of solution quality.

Table 9 Optimization results for noisy functions

Algorithms	Fit. Value	SP	SR(%)	h	Fit. Value	SP	SR(%)	h
	Shifted Noise Schwefel P1.2 (f_{28})				Shifted Rotated Noise Quadric (f_{29})			
PSO-w	1.20E+03		0	+	9.24E-03		0	+
PSO-cf	7.43E+02		0	+	3.90E-03		0	+
PSO-w-local	1.38E+03		0	+	2.07E-02		0	+
PSO-cf-local	1.37E+02		0	=	2.37E-03		0	+
UPSO	2.69E+03		0	+	1.89E-02		0	+
wFIPS	2.09E+02		0	+	2.68E-03		0	+
FDR-PSO	2.52E+02		0	+	7.07E-03		0	+
CPSO-H	1.76E+04		0	+	1.45E-02		0	+
CLPSO	2.09E+03		0	+	4.15E-03		0	+
DMS-PSO	2.34E+02		0	+	3.84E-03		0	+
PSO-MAM	**7.07E+01**	**1084755**	**26.7**		**5.05E-04**		**0**	

3.4.3.6 Mis-scaled Functions

Two mis-scaled functions are studied in this set of experiments (see Table 4). The optimization results are recorded in Table 10. PSO-MAM outperforms on the 2 mis-scaled Rastrigin functions in terms of convergence speed.

Table 10 Optimization results for mis-scaled functions

Algorithms	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**
	Shifted Rastrigin10 (f_{30})				Shifted Rastrigin100 (f_{31})			
PSO-w	2.89E+01		0	+	2.58E+01		0	+
PSO-cf	1.06E+02		0	+	1.46E+02		0	+
PSO-w-local	3.92E+01		0	+	3.93E+01		0	+
PSO-cf-local	5.47E+01		0	+	6.40E+01		0	+
UPSO	8.91E+01		0	+	1.14E+02		0	+
wFIPS	5.32E+01		0	+	5.27E+01		0	+
FDR-PSO	3.61E+01		0	+	5.18E+01		0	+
CPSO-H	1.33E-01	123673	96.7	+	2.32E-01	243602	76. 7	+
CLPSO	0.00E+00	176403	100	+	0.00E+00	176792	100	+
DMS-PSO	9.65E+00	8914706	3.33	+	1.37E+01		0	+
PSO-MAM	**0.00E+00**	**65171**	**100**		**2.43E-15**	**105302**	**100**	

3.4.3.7 Conclusions on Comparison Experiments

It is observed that PSO-MAM in general outperforms other algorithms on most of the test functions. To assess the overall performance, the dominance index is introduced to quantitatively measure the PSO algorithm. Considering any two PSO algorithms, A and B, algorithm A dominates algorithm B on a function when 1) the measure of SP for algorithm A is better than algorithm B when SP is available, or 2) the fitness value for algorithm A is better than algorithm B when SP is not available. For example, the dominance index for PSO-MAM on the Shifted Sphere (f_1) function is 10 since it dominates the other 10 algorithms in terms of SP. For each algorithm, the total number of dominated algorithms on each function is obtained and then the dominance rate is computed as the cumulative number of dominated algorithms on the 31 functions divided by the ideal case which has 310 (31x10) cumulative dominated algorithms. The dominance rate and the overall comparisons between PSO-MAM and other PSO algorithms using the *t*-test are recorded in Table 11. It is observed from Table 11, PSO-MAM has the largest dominance rate which means PSO-MAM is the most generalized algorithm for diverse functions with different properties. This is also observed from comparison results between PSO-MAM and other PSO algorithms using the *t*-test.

Table 11 Dominance rate and overall comparisons between PSO-MAM and other algorithms

Algorithms Metrics		PSO-w	PSO-cf	PSO-w-local	PSO-cf-local	UPSO	wFIPS	FDR-PSO	CPSO-H	CLPSO	DMS-PSO	PSO-MAM
Dominance Rate (%)		31.9	36.1	24.8	52.6	45.5	56.1	42.6	36.8	57.1	72.6	**93.9**
t-test	+ (Better)	28	29	28	27	28	27	28	29	27	26	-
	= (Same)	1	2	0	3	2	1	2	2	1	2	-
	- (Worse)	2	0	3	1	1	3	1	0	3	3	-

3.4.4 Comparison Experiments for 100 Dimensional Functions

This section attempts to test PSO-MAM's applicability for high-dimensional optimization problems. The 12 test functions studied in this section are 100 dimensions, the population size is set to be 30 and the maximum number of function evaluations is set to be 600,000. Values "+", "=" and "-" in the column "h" in Table 12~13 denote PSO-MAM performs significantly better than, almost the same as, and significantly worse than the compared algorithm, respectively.

3.4.4.1 Uni-modal Functions

In the first set of experiments, 6 non-shifted and non-rotated uni-modal functions (f_1~f_6 in Table 4) are studied. The optimization results are summarized in Table 12.

Table 12 Optimization results for 100-*D* uni-modal functions

Algorithms	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**	**Fit. Value**	**SP**	**SR(%)**	**h**
	Sphere (f_1)				Schwefel P2.22 (f_2)				Schwefel P1.2 (f_3)			
PSO-w	1.03E-15	483521	100	+	2.89E-11	486954	100	+	2.61E+04		0	+
PSO-cf	9.86E-39	127805	100	+	1.19E-07	300117	100	+	8.68E+01		0	+
PSO-w-local	7.06E-08	564487	100	+	1.39E-07	559830	100	+	2.31E+04		0	+
PSO-cf-local	4.80E-46	113273	100	+	2.37E-29	138214	100	-	8.51E+02		0	+
UPSO	4.42E-104	54604	100	+	**1.07E-60**	**72416**	**100**	-	1.88E+02		0	+
wFIPS	4.39E-05		0	+	1.37E-04		0	+	8.41E+04		0	+
FDR-PSO	2.95E-43	294187	100	+	4.49E-21	311244	100	+	1.03E+02		0	+
CPSO-H	7.51E-08	379618	100	+	7.81E-05		0	+	3.39E+04		0	+
CLPSO	1.19E-24	265905	100	+	5.56E-16	301638	100	+	4.79E+04		0	+
DMS-PSO	1.86E-45	113156	100	+	1.03E-31	125985	100	-	1.27E+04		0	+
PSO-MAM	**0.00E+00**	**1925**	**100**		1.07E-21	205858	100		**1.27E-01**	**383083**	**93.3**	
	Schwefel P2.21 (f_4)				Rosenbrock (f_5)				Step (f_6)			
PSO-w	5.35E+01		0	+	2.31E+02		0	+	2.38E+01		0	+
PSO-cf	3.57E+01		0	+	7.82E+01		0	+	3.10E+03		0	+
PSO-w-local	4.61E+01		0	+	3.93E+02		0	+	2.03E+00	4452555	13.3	+
PSO-cf-local	3.83E+01		0	+	1.27E+02		0	+	0.00E+00	186739	100	+

UPSO	3.11E+01		0	+	1.06E+02		0	+	0.00E+00	70447	100	+
wFIPS	1.77E+01		0	+	2.41E+02		0	+	0.00E+00	294083	100	+
FDR-PSO	1.31E+01		0	+	8.82E+01	17962598	3.33	+	3.53E+01		0	+
CPSO-H	3.81E+01		0	+	1.44E+02		0	+	0.00E+00	37430	100	+
CLPSO	9.95E+00		0	+	8.04E+01		0	+	0.00E+00	170966	100	+
DMS-PSO	6.86E+00		0	+	1.57E+02		0	+	0.00E+00	106491	100	+
PSO-MAM	**1.50E-164**	**6199**	**100**		**1.06E-29**	**130107**	**100**		**0.00E+00**	**880**	**100**	

3.4.4.2 Multi-modal Functions

In the second set of experiments, 6 non-shifted and non-rotated multi-modal functions (f_{13}~f_{18} in Table 4) are studied. The optimization results are summarized in Table 13.

Table 13 Optimization results for 100-*D* multi-modal functions

Algorithms	Fit. Value	SP	SR(%)	h	Fit. Value	SP	SR(%)	h	Fit. Value	SP	SR(%)	h
	Schwefel (f_{13})				2^D minima (f_{14})				Rastrigin (f_{15})			
PSO-w	5.35E+03		0	+	9.67E+00		0	+	2.71E+02		0	+
PSO-cf	1.30E+04		0	+	1.20E+01		0	+	4.06E+02		0	+
PSO-w-local	2.43E+04		0	+	2.80E+00		0	+	1.35E+02		0	+
PSO-cf-local	1.17E+04		0	+	1.07E+01		0	+	3.42E+02		0	+
UPSO	1.42E+04		0	+	1.09E+01		0	+	3.69E+02		0	+
wFIPS	7.35E+03		0	+	9.52E-01		0	+	6.52E+02		0	+
FDR-PSO	1.53E+04		0	+	1.14E+01		0	+	1.82E+02		0	+
CPSO-H	7.26E+02		0	+	4.93E-10	121058	100	+	**1.87E-08**	**334370**	**100**	=
CLPSO	**8.69E+01**	**1036318**	**43.3**	=	4.57E-10	192621	100	+	1.74E-12	474926	100	+
DMS-PSO	1.57E+04		0	+	8.98E+00		0	+	1.09E+02		0	+
PSO-MAM	1.30E+02	1079515	43.3		**4.57E-10**	**83337**	**100**		0.00E+00	343494	100	
	Noncontinuous Rastrigin (f_{16})				Ackley (f_{17})				Griewank (f_{18})			
PSO-w	3.14E+02		0	+	4.90E-02	577046	96.7	+	1.45E-02	1380485	40	+
PSO-cf	2.00E+02		0	+	1.11E+01		0	+	3.45E-01	5547540	10	+
PSO-w-local	1.35E+02		0	+	1.56E-03		0	+	3.45E-03	745122	76.7	+
PSO-cf-local	1.32E+02		0	+	2.09E+00	8570790	6.67	+	3.01E-02	708736	50	+
UPSO	4.36E+02		0	+	2.72E+00		0	+	2.51E-02	236835	76.7	+
wFIPS	6.50E+02		0	+	9.36E-04		0	+	3.66E-05		0	+
FDR-PSO	1.14E+02		0	+	8.55E-02	377887	93.3	+	1.44E-02	891283	50	+
CPSO-H	1.90E-08	323361	100	=	3.94E-05		0	+	8.18E-03	707520	63.3	+
CLPSO	2.67E-01	649019	80	+	2.14E-13	324157	100	+	0.00E+00	276899	100	+
DMS-PSO	6.32E+01		0	+	1.40E-14	165089	100	+	5.11E-03	300265	76.7	+
PSO-MAM	**0.00E+00**	**308480**	**100**		**0.00E+00**	**71993**	**100**		**4.81E-17**	**5505**	**100**	

3.4.4.3 Conclusions on Comparison Experiments

It is observed that PSO-MAM in general outperforms other algorithms on most of the 100-dimensional test functions.

Table 14 Overall comparisons between PSO-MAM and other algorithms on *t*-tests

Algorithms ***t*-test Results**	PSO-w	PSO-cf	PSO-w-local	PSO-cf-local	UPSO	wFIPS	FDR-PSO	CPSO-H	CLPSO	DMS-PSO
+ (Better)	12	12	12	11	11	12	12	10	11	11
= (Same)	0	0	0	0	0	0	0	2	1	0
- (Worse)	0	0	0	1	1	0	0	0	0	1

From Table 14 which summarizes the overall comparisons between PSO-MAM and other PSO algorithms using the *t*-test, it demonstrates that PSO-MAM is significantly better than other algorithms on 9 functions, is comparable on 2 functions, and is inferior to PSO-cf-local, UPSO, and DMS-PSO on 1 function.

3.5 Conclusions

An augmented PSO algorithm with multiple adaptive search methods, PSO-MAM, is developed in an attempt to overcome the following disadvantages of PSO: 1) PSO and most of its variants are not guaranteed to perform well on a diverse set of optimization problems; 2) it suffers premature convergence. By intelligently selecting the effective search method to enhance PSO, the intelligent multi-method search module is effective in improving solution quality for both uni-modal and multi-modal functions. The extended Cauchy mutation operator is efficient in guaranteeing diversity of the swarm. The experiment conducted in this chapter demonstrates that PSO-MAM is superior to the existing PSO algorithms in terms of solution quality and convergence speed for not only diverse problems with different properties but also high-dimensional problems. My contributions lay into three aspects: 1) develop a generalized PSO algorithm which performs well on a diverse set of optimization problems; 2) develop a generalized intelligent multiple search methods selection strategy which could be used to assess multiple search methods; 3) develop an adaptive sub-gradient method and Cauchy mutation operator based on a particle’s velocity information.

Comparing with CLPSO and DMS-PSO, the performance of PSO-MAM on functions which have very complex search spaces (e.g. rotated multi-modal functions) should be improved. In the future, PSO-MAM will be comprehensively compared with some competitive PSO algorithms (e.g., ELPSO (Huang et al., 2010), SLPSO (Wang et al., 2010)) on more test functions (e.g., mis-scaled, noisy). In addition, like other evolutionary algorithms, the performance of PSO also depends on its parameter settings. Enhancing particle swarm optimization with an adaptive parameter tuning mechanism to improve its robustness is the second approach to improve particle swarm optimization’s performance which will be studied in the next chapter.

Chapter 4

A BI-LOCAL SEARCHES AND MUTATION BASED ADAPTIVE PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) has attracted much attention and has been applied to many scientific and engineering applications in the last decade. However, inherited from PSO, the performance of PSO heavily depends on its three parameters: the two learning factors and inertia weight. In this chapter, firstly, a bi-local searches and mutation based PSO algorithm (BLOSSM-PSO) is developed by using the multiple methods concept from the chapter 3, and then a parameter tuning mechanism is developed to adaptively change the three parameters to improve PSO's robustness. A new PSO algorithm, BLOSSM-**Adaptive** PSO (BLOSSM-APSO) is developed which is expected to be more robust than BLOSSM-PSO. The performance of BLOSSM-APSO is comprehensively evaluated on 31 functions and it is compared with ten published PSO methods and BLOSSM-PSO. The conclusions are (1) BLOSSM-APSO outperforms the ten PSO methods on 23 functions, has comparable performance for 4 functions, and underperforms for 4 functions on solution quality and/or convergence speed; (2) BLOSSM-APSO improves BLOSSM-PSO on solution quality and/or convergence speed for 29 out of 31 functions, and is more robust than BLOSSM-PSO.

4.1 Introduction

Inspired by the social cooperative and competitive behavior of bird flocking and fish schooling, Kennedy and Eberhart developed a new optimization technique called Particle Swarm Optimization (PSO) in 1995 (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995). Similar to other evolutionary algorithms (EAs), PSO is a stochastic and population based meta-heuristic algorithm which is particularly effective on optimization problems that are partially irregular, noisy, stochastic and dynamic. The main differences between PSO and other EAs are twofold. First, most EAs such as genetic algorithms and Memetic algorithms use explicit selection functions, PSO adopts leaders to guide the search of each particle in the swarm (Reyes-Sierra & Coello Coello, 2006). Secondly, most EAs employ the competitive strategy, that is, individuals compete with each other on a 'survival of the fittest' basis for inheritance. In PSO, the particles cooperate with each other and explore the search space directed by a combination of the swarm's previous best (gBest) and their own previous best (pBest), with an additional stochastic element (Banks et al., 2007). The differences have allowed PSO to be successfully applied to various industry applications (Engelbrecht, 2006).

In general, the performance of PSO is affected by swarm size (Poli et al., 2007) and swarm neighborhood topology (Janson & Middendorf, 2005; Kennedy, 1999; Kennedy & Mendes, 2002; Liang & Suganthan, 2005; Parsopoulos & Vrahatis, 2004; Reyes-Sierra & Coello Coello, 2006). For example, Liang and Suganthan (2005) dynamically divide the swarm into several small swarms which can interact with each other. A dynamic hierarchy is used to define the neighborhood topology in (Janson & Middendorf, 2005). The local and global topology structures are combined together in the unified particle swarm optimizer (UPSO) (Parsopoulos & Vrahatis, 2004). Recently, notable efforts are devoted to studying the impact of exemplar learning strategy (selection of gBest and pBest) on the performance of PSO (Huang et al., 2010; Liang et al., 2006; Mendes et al., 2004; Peram et al., 2003; van den Bergh & Engelbrecht, 2004; Wang et al., 2011; Wang et al., 2010). In FIPS (Mendes et al., 2004), the particles are allowed to learn from all the particles instead of the best one in its neighborhood. A third particle other than the personal best and swarm best is selected to guide the movement of a particle based on the fitness-distance-ratio (Peram et al., 2003). Considering a multi-dimensional search space problem, CPSO-H first locates and searches the exemplar for each dimension independently, the joint solution is evaluated for the multi-dimensional space (van den Bergh & Engelbrecht, 2004). In CLPSO (Liang et al., 2006), the particle learns from other particle's personal best and no swarm best is used. Two learning strategies - standard PSO learning and generalized opposition-based learning (GOBL) are alternatively applied based on a dynamically updated probability (Wang et al., 2011). The particles can self-adaptively learn from four PSO based search approaches simultaneously in SLPSO (Wang et al., 2010). A set of multiple swarm best particles are selected to guide the movement of particles in ELPSO (Huang et al., 2010).

A common issue for PSO in general, is that its performance is heavily dependent on the three parameters: cognitive learning factor c_1, social learning factor c_2 and inertia weight w (Clerc & Kennedy, 2002; Fernandez Martınez & Garcıa Gonzalo, 2008; Jiang et al., 2007; Kadirkamanathan et al., 2006; Ozcan & Mohan, 1998; Ozcan & Mohan, 1999; van den Bergh, 2002). Realizing a number of researchers has successfully studied different parameter tuning mechanisms to improve the performance (Chatterjee & Siarry, 2006; Eberhart & Shi, 2001; Jiao et al., 2008; Juang et al., 2010; Ratnaweera et al., 2004; Shi & Eberhart, 1999; Shi & Eberhart, 2001; Shu & Li, 2009; Yamaguchi & Yasuda, 2006; Zhan et al., 2009), here the multiple methods concept is adopted from the chapter 3 to develop a bi-local searches and mutation based PSO algorithm (BLOSSM-PSO), and then develop an adaptive parameter tuning mechanism to enhance BLOSSM-PSO. The new algorithm is termed as BLOSSM-APSO. The basic idea is to pull one randomly selected particle close to the gBest to reduce the large jumps effect from the Cauchy mutation. Aiming to minimize the distance between the selected particle and the gBest, the parameter tuning problem is formulated as a convex optimization and the sub-gradient method (Boyd, 2010) is employed to adaptively change the parameters (see section 4.3.3). It is expected that BLOSSM-APSO is more robust than BLOSSM-PSO as it is less sensitive to the initial parameter settings.

This chapter is organized as follows: several existing parameter tuning mechanisms are briefly reviewed in section 4.2; followed by the detailed explanation on the BLOSSM-APSO in section 4.3; the experimental results in section 4.4 demonstrate the effectiveness of the BLOSSM-APSO. Finally, conclusions are drawn in section 4.5.

4.2 Literature Review

In the PSO with inertia weight, the velocity and position for particle *j* at iteration *i* are updated as (Shi & Eberhart, 1998)

$$\mathbf{v}_j^{i+1} = w\mathbf{v}_j^i + c_1 r_{1,j}^i \times \left(\mathbf{p}_j^i - \mathbf{x}_j^i\right) + c_2 r_{2,j}^i \times \left(\mathbf{p}_g^i - \mathbf{x}_j^i\right) \tag{4.1}$$

$$\mathbf{x}_j^{i+1} = \mathbf{x}_j^i + \mathbf{v}_j^{i+1} \tag{4.2}$$

where *j* denotes the *j*th particle in the swarm; *D*-dimensional vector $\mathbf{v}_j^i$ is the velocity of the *j*th particle ($\mathbf{v}_j^i \in \left[-\mathbf{v}_{\max}, +\mathbf{v}_{\max}\right]$), $\mathbf{v}_{\max}$ is used to constraint the velocity for each particle and is usually set between 0.1 and 1.0 times the search range of the solution space (Banks et al., 2007); *D*-dimensional vector $\mathbf{x}_j^i$ is the position of the *j*th particle; $\mathbf{p}_j^i$ is the best position found so far by the *j*th particle; $\mathbf{p}_g^i$ is the best position found so far by the swarm; $r_{1,j}^i$ and $r_{2,j}^i$ represent two independent random numbers uniformly distributed on [0, 1]; c_1 is the cognitive learning factor which represents the attraction that a particle has toward its own success $\mathbf{p}_j^i$; c_2 is the social learning factor which represents the attraction that a particle has toward its neighbors' best position $\mathbf{p}_g^i$; *w* is the inertia weight. Cognitive learning factor c_1 impacts the local search ability while the global search ability is influenced by the social learning factor c_2. Large inertia weight *w* enables particles to move at a high velocity and perform extensive exploration, and small inertia weight enhances the exploitation ability (Poli et al., 2007).

Over the last decade, two different theoretical models (deterministic and stochastic) have been developed to study the impacts of PSO parameters on the performance of PSO. In the deterministic model, the stochastic components r_1 and r_2 in Eq. (4.1) are ignored. By using the deterministic model, Clerc and Kennedy (2002) prove that the particle will converge to a stable point

$$\mathbf{x}_j = \left(c_1\mathbf{p}_j + c_2\mathbf{p}_g\right) \Big/ \left(c_1 + c_2\right) \tag{4.3}$$

when c_1+c_2<4 for the PSO algorithm without inertia weight w. Different from (Clerc & Kennedy, 2002), van den Bergh (2002) studies the PSO with inertia weight and concludes that the particles will converge to the stable point in Eq. (4.3) when

$$0 \le w < 1 \quad \text{and} \quad w > \left(c_1 + c_2\right) \big/ 2 - 1 \tag{4.4}$$

And the particles will become divergent if the conditions in Eq. (4.4) are not satisfied.

Kadirkamanathan et al. (2006) employ the Lyapunov stability analysis method to study the stability of particles with stochastic components and conclude that the sufficient condition for stability of the particles is

$$0 \le w < 1 \quad \text{and} \quad c_1 + c_2 < 2\left(1 - w\right)^2 \big/ \left(1 + w\right) \tag{4.5}$$

Due to the restriction of the Lyapunov function, this condition is conservative (Kadirkamanathan et al., 2006). Using stochastic process theory to analyze the particle trajectory of the stochastic model, Jiang et al. (2007) demonstrate that the expectation of particle position is guaranteed to converge to the stable point in Eq. (4.3) when

$$0 \le w < 1 \quad \text{and} \quad 0 < c_1 + c_2 < 4\left(w + 1\right) \tag{4.6}$$

The theoretical results for the two models provide general guidelines on parameter settings for PSO. The actual settings of the parameters however still rely on empirical studies with respect to the optimization problem formulation. To reduce the sensitivity of PSO to its parameter settings, researches (Chatterjee & Siarry, 2006; Eberhart & Shi, 2001; Jiao et al., 2008; Juang et al., 2010; Ratnaweera et al., 2004; Shi & Eberhart, 1999; Shi & Eberhart, 2001; Shu & Li, 2009; Yamaguchi & Yasuda, 2006; Zhan et al., 2009) have also attempted to develop parameter tuning rules which can be classified into two categories: 1) simple rule based parameter tuning where linear, non-linear functions or fuzzy rules are explored; 2) self-learning adaptive parameter tuning which considers the current evolutionary state in the parameter tuning. Each mechanism is reviewed in the following sections.

4.2.1 Simple Rule based Parameter Tuning

Shi and Eberhart (1999) show that the performance of PSO can be greatly improved by linearly decreasing the inertia weight as

$$w = w_{\max} - \left(w_{\max} - w_{\min}\right) i / I \tag{4.7}$$

where w_{max} and w_{min} are usually fixed as 0.9 and 0.4; i is the current iteration number; and I is the maximum number of iterations. In addition, Jiao et al. (2008), Chatterjee and Siarry (2006) improve performance of PSO by tuning the inertia weight according to non-linear functions which are expressed as

$$w = w_0 u^{-i}, \quad w_0 \in [0,1], \quad u \in [1.001, 1.005] \tag{4.8}$$

$$w = w_{\min} + \left(w_{\max} - w_{\min}\right)\left(I - i\right)^n \big/ I^n \tag{4.9}$$

where n is the non-linear modulation index (Chatterjee & Siarry, 2006). A random function is implemented in (Eberhart & Shi, 2001) to tune the inertia weight as

$$w = 0.5 + random\left(0,1\right) \big/ 2 \tag{4.10}$$

A fuzzy rule based tuning mechanism is developed in (Shi & Eberhart, 2001) to modify the inertia weight. Ratnaweera et al. (2004) update the learning factors c_1 and c_2 using the same functions as Eq. (4.7).

Although a simple rule based parameter tuning mechanism can improve performance of PSO for several problem instances, the performance on a broader spectrum of problems is unsatisfactory (Chatterjee & Siarry, 2006; Eberhart & Shi, 2001; Jiao et al., 2008; Ratnaweera et al., 2004; Shi & Eberhart, 1999; Shi & Eberhart, 2001). Another criticism as in (Chatterjee & Siarry, 2006; Jiao et al., 2008), is that additional parameters may be needed for this mechanism.

4.2.2 Self-learning Adaptive Parameter Tuning

Juang et al. (2010) develop an adaptive fuzzy PSO termed as AFPSO where the inertia weight is altered according to Eq. (4.7) and the two learning factors c_1 and c_2 are changed according to the three fuzzy rules: let *df* denote the difference between the best fitness for two consecutive iterations, c_1 and c_2 are changed as: 1) c_1 is big and c_2 is small when *df* is small; 2) c_1 is medium and c_2 is medium when *df* is medium; 3) c_1 is small and c_2 is big when *df* is big. Zhan et al. (2009) employ the evolutionary state estimation method to identify the evolutionary states of the swarm as exploration, exploitation, convergence and jumping out, and develop an adaptive parameter tuning method as: 1) increase c_1 and decrease c_2 at the exploration state; 2) increase slightly c_1 and decrease slightly c_2 at the exploitation state; 3) increase slightly c_1 and increase slightly c_2 at the convergence state; and 4) decrease c_1 and increase c_2 at the jumping out state. The inertia weight is modified as

$$w = 1 \big/ \left(1 + 1.5e^{-2.6f}\right) \in \left[0.4, 0.9\right] \quad \forall f \in \left[0,1\right] \tag{4.11}$$

where f is the evolutionary factor implemented in (Zhan et al., 2009). The adaptive particle swarm optimization is demonstrated to outperform other existing PSO algorithms for most problem instances.

Please note the methods reviewed above (Juang et al., 2010; Zhan et al., 2009) provide the interval values with a fuzzy guideline (e.g., slightly increase, slightly decrease) for the changes. To explicitly quantify the parameter tunings, Yamaguchi and Yasuda (2006) assign c_1 and c_2 for each particle, and update c_1 and c_2 using the following equations

$$c_{1,j}^{i+1} = c_{1,j}^{i} + \alpha_j^i \left(cbest_1^i - c_{1,j}^i \right) \tag{4.12}$$

$$c_{2,j}^{i+1} = c_{2,j}^{i} + \alpha_j^i \left(cbest_2^i - c_{2,j}^i \right) \tag{4.13}$$

where $cbest_1^i$ and $cbest_2^i$ are parameters for the global best particles at iteration i; α_j^i is selected from two values 0 and $2/I$. Shu and Li (2009) introduce a piecewise function $F(.)$, called the adjustment function, to adaptively change parameters w, c_1 and c_2 as

$$w_j^{i+1} = w_1 \times F\left(\eta_j^i\right) + w_2 \tag{4.14}$$

$$\eta_j^i = f_g^{i-1} \big/ f_j^{i-1} \tag{4.15}$$

where f_g^{i-1} and f_j^{i-1} are the fitness value for the best particle and particle j at iteration i-1. Experiments though show that (Shu & Li, 2009; Yamaguchi & Yasuda, 2006) are effective on only a few instances (<4 functions). In this chapter, the parameters will be adaptively and explicitly tuned to improve PSO's performance for diverse functions.

4.3 BLOSSM-APSO Algorithm

Two common criticisms for PSO and its variants are: 1) PSO is not effective for high dimensional functions especially for multi-modal functions, and most improved PSOs are not guaranteed to perform well on functions with different properties; 2) it suffers premature convergence. To address these issues, the multiple methods concept from the chapter 3 is adopted to develop a bi-local searches and mutation based PSO, termed (BLOSSM-PSO) (see sections 4.3.1 and 4.3.2). First, two local searches: a non-uniform mutation based method (Michalewicz, 1996) which may be preferred by uni-modal functions and a sub-gradient method (Boyd, 2010) which may be preferred by multi-modal functions are studied. Next, a Cauchy mutation operator is incorporated to prevent premature convergence. As discussed in (Clerc & Kennedy, 2002; Fernandez Martınez & Garcıa Gonzalo, 2008; Jiang et al., 2007; Kadirkamanathan et al., 2006; Ozcan & Mohan, 1998; Ozcan & Mohan, 1999; van den Bergh, 2002), PSO's performance is highly impacted by its three parameters: the cognitive learning factor c_1, the social learning factor c_2 and the inertia weight w. Therefore, an adaptive parameter tuning mechanism is developed to enhance the robustness and the performance of BLOSSM-PSO. The BLOSSM-APSO is shown in Figure 13 which has four modules: 1) PSO module: The swarm is randomly initialized with the PSO operator being employed to update the swarm. 2) Bi-Local searches module: two local search methods (non-uniform mutation based method and sub-gradient method) are implemented. At each iteration, an appropriate local search method will be triggered based on the dynamic selection criteria. The initial local search method is the non-uniform mutation based method (ls_indicator=0) which may perform well at the early search in PSO. 3) Mutation module: after the local searches on the best particle, the mutation operator is used to update one randomly selected particle. 4) Parameter tuning module: the three parameters for one randomly selected particle will be changed by the adaptive parameter tuning mechanism (see section 4.3.3). The algorithm will stop if the stopping criterion (such as the maximum number of PSO iterations, predefined solution accuracy) is satisfied.

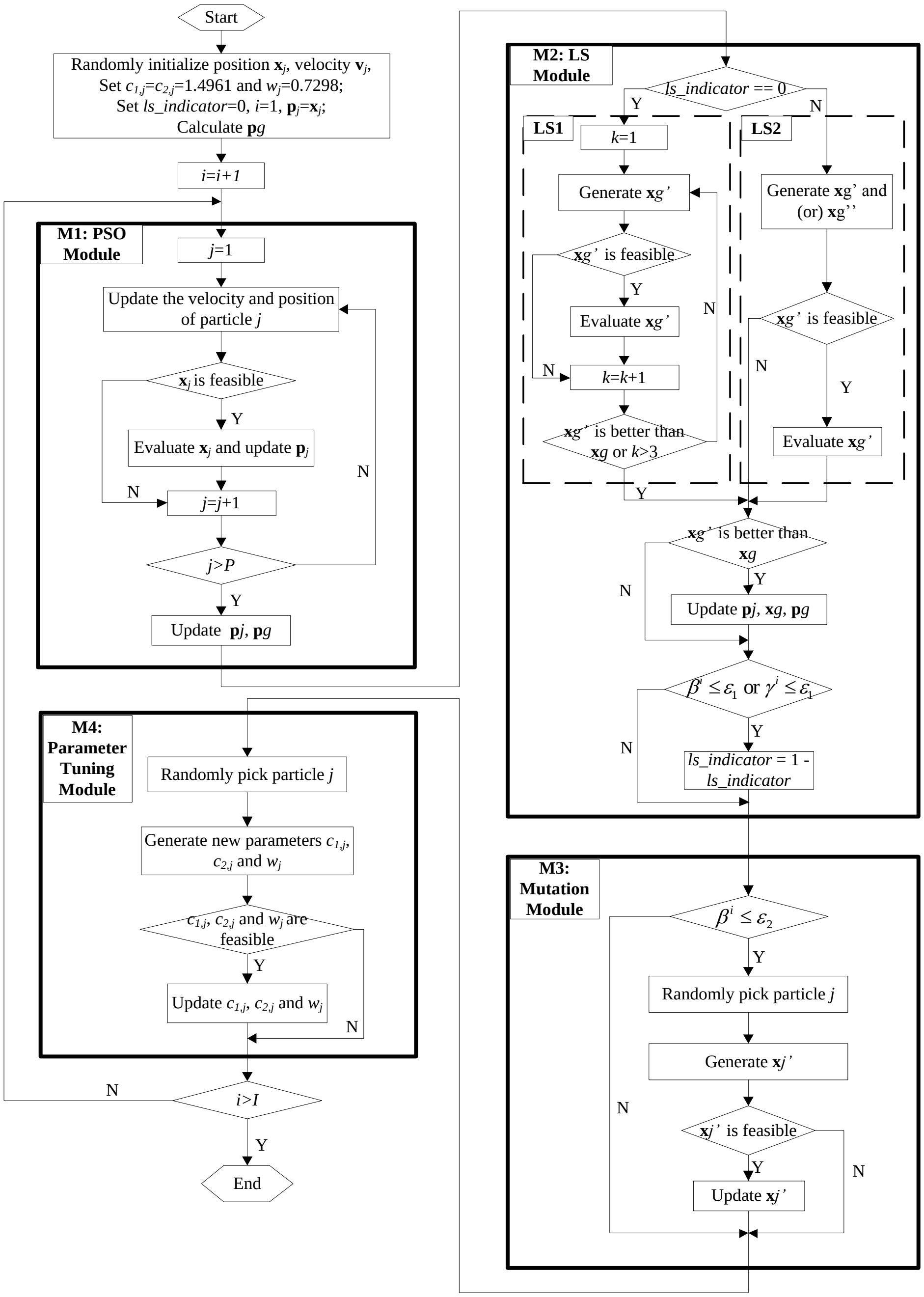

Start
Randomly initialize position $\mathbf{x}_j$, velocity $\mathbf{v}_j$,
Set $c_{1,j}=c_{2,j}=1.4961$ and $w_j=0.7298$;
Set $ls\_indicator=0$, $i=1$, $\mathbf{p}_j=\mathbf{x}_j$;
Calculate $\mathbf{p}g$
$i=i+1$
M1: PSO Module
$j=1$
Update the velocity and position of particle j
$\mathbf{x}_j$ is feasible
Y
Evaluate $\mathbf{x}_j$ and update $\mathbf{p}_j$
N
$j=j+1$
N
$j>P$
Y
Update $\mathbf{p}j$, $\mathbf{p}g$
M4: Parameter Tuning Module
Randomly pick particle j
Generate new parameters $c_{1,j}$, $c_{2,j}$ and w_j
$c_{1,j}$, $c_{2,j}$ and w_j are feasible
Y
Update $c_{1,j}$, $c_{2,j}$ and w_j
N
N
$i>I$
Y
End
M2: LS Module
$ls\_indicator == 0$
Y
N
LS1
$k=1$
Generate $\mathbf{x}g'$
$\mathbf{x}g'$ is feasible
Y
Evaluate $\mathbf{x}g'$
N
N
$k=k+1$
$\mathbf{x}g'$ is better than $\mathbf{x}g$ or $k>3$
Y
LS2
Generate $\mathbf{x}$g' and (or) $\mathbf{x}$g''
$\mathbf{x}g'$ is feasible
N
Y
Evaluate $\mathbf{x}g'$
$\mathbf{x}g'$ is better than $\mathbf{x}g$
Y
N
Update $\mathbf{p}j$, $\mathbf{x}g$, $\mathbf{p}g$
$\beta^i \le \varepsilon_1$ or $\gamma^i \le \varepsilon_1$
Y
N
$ls\_indicator$ = 1 - $ls\_indicator$
M3: Mutation Module
$\beta^i \le \varepsilon_2$
Y
Randomly pick particle j
Generate $\mathbf{x}j'$
N
$\mathbf{x}j'$ is feasible
Y
N
Update $\mathbf{x}j'$

Figure 13 Flowchart of BLOSSM-APSO (“LS1”: non-uniform mutation-based method; “LS2”: sub-gradient method)

4.3.1 Local Searches

In the bi-local searches, let the current best solution $\mathbf{x}_g^i$ be

$$\mathbf{x}_g^i = \operatorname{argmin}_{1\le j\le P}\left\{ f\left(\mathbf{x}_j^i\right)\right\} \tag{4.16}$$

Two local search approaches which combine the non-uniform mutation based method (Michalewicz, 1996) and sub-gradient based method (Boyd, 2010) are studied. The non-uniform mutation based method (Michalewicz, 1996) is good at searching the solution space uniformly (exploration) at the early stage and locally (exploitation) at the later stage (Zhao, 2011). The non-uniform mutation based method has been demonstrated to have the merits of large jumping (exploration) and fine-tuning (exploitation) (Zhao, 2011; Zhao et al., 2007). In addition, the non-uniform mutation based method does not require the problem instance to have analytical functions. The second local search method is the sub-gradient method (Boyd, 2010) which is an iterative method for solving convex minimization problems and is also applicable for non-convex problems. Like gradient based methods, the sub-gradient method exhibits good exploitation capability around the neighborhood of the local or global optimum (Plevris & Papadrakakis, 2010). The sub-gradient method for unconstrained problems is equivalent to the gradient based method when the objective function is differentiable.

In the non-uniform mutation based method, the d^{th} dimension of the current best solution $\mathbf{x}_g^i$ is randomly picked to be mutated to generate a new solution as

$$x_{g,d}^i{}' = \begin{cases} x_{g,d}^i + \Delta\left(i, U_d - x_{g,d}^i\right) & \text{if } r \geq 0.5 \\ x_{g,d}^i - \Delta\left(i, x_{g,d}^i - L_d\right) & \text{if } r < 0.5 \end{cases} \tag{4.17}$$

where i is the current iteration number of PSO; U_d and L_d are the upper and lower bounds of $x_{g,d}^i$; r is a uniform random number from (0, 1). The function $\Delta(i,y)$ is defined as

$$\Delta\left(i, y\right) = y \times \left(1 - \rho^{(1-i/I)}\right) \tag{4.18}$$

where ρ is a uniform random number from (0, 1); I is the maximum number of iterations for PSO. In the sub-gradient method, a new solution $\mathbf{x}_g^i{}'$ is generated as

$$\mathbf{x}_g^i{}' = \mathbf{x}_g^i - \alpha_i \mathbf{g}_g^i \tag{4.19}$$

where $\mathbf{g}_g^i$ is the sub-gradient of the objective function; $\alpha_i = 1/i^{0.6}$ is the step size used in this chapter. Solution $\mathbf{x}_g^i{}'$ generated by Eq. (4.19) may be ineffective (infeasible or local exploitation) at the beginning of PSO iterations. Therefore, one additional solution $\mathbf{x}_g^i{}''$ is generated by

$$\mathbf{x}_g^i{}'' = \mathbf{x}_g^i - \alpha_i \frac{\mathbf{g}_g^i}{\left\|\mathbf{g}_g^i\right\|_2} \frac{\left(\mathbf{U}_g - \mathbf{L}_g\right)}{4} \tag{4.20}$$

if $r \le \exp\left(-i^{2.8}/I^2\right)$, where $\mathbf{U}_g$ and $\mathbf{L}_g$ are the upper and lower bounds of $\mathbf{x}_g^i$; r is a uniform random number from (0, 1); i is the current iteration number of PSO; I is the maximum number of iterations for PSO.

The exploration ability of the non-uniform mutation based method benefits BLOSSM-APSO on multi-modal functions, but may slow the convergence speed on uni-modal functions. The sub-gradient method is good at exploiting the search space, but tends to be trapped in the local optimum for some functions (e.g. Schwefel, 2^D minima function). Therefore, a dynamic selection mechanism is introduced to balance the exploration and exploitation capability of BLOSSM-APSO. Let f_g^i be the fitness value for $\mathbf{p}_g^i$ at current iteration i, the fitness evolutionary state β and solution evolutionary state γ are introduced which are defined as

$$\beta^i = \left(f_g^{i-1} - f_g^i\right) / f_g^{i-1} \tag{4.21}$$

$$\gamma^i = \min_{d=1,\ldots,D} \left\{ \left| \left(x_{g,d}^i - x_{g,d}^{i-1}\right) / x_{g,d}^{i-1} \right| \right\} \tag{4.22}$$

When the changes on either state are small for one local search method, the alternative local search method is triggered for the next iteration.

4.3.2 Cauchy Mutation

To keep the diversity of the swarm, in hopes of accelerating the converging speed (non-premature), the Cauchy mutation operator is adopted which is demonstrated to be able to assist the particle by having a large jump out of its local optimum (Andrews, 2006). At iteration *i*, a particle *j* is randomly picked to be mutated if $\beta^i \le \varepsilon_2$ where the mutation threshold $\varepsilon_2=10^{-4}$. The dimension *d* which has a minimum value defined in Eq. (4.22) is mutated as

$$x_{j,d}^{i}\,' = x_{j,d}^{i} + cauchy\left(0.1\right)\times\eta^{i}\times\left(U_d - L_d\right) \tag{4.23}$$

where U_d and L_d are the upper and lower bounds of $x_{j,d}^{i}$; and η^i is the mutation scale parameter which is defined as

$$\eta^{i} = \begin{cases} \exp\left(-i^{2.61}/I^{2}\right) & \gamma^{i} > \varepsilon_2 \\ \max\left(\exp\left(-i^{2.61}/I^{2}\right), 0.1\right) & \gamma^{i} \le \varepsilon_2 \end{cases} \tag{4.24}$$

4.3.3 Adaptive Tuning

As discussed in (Yao et al., 1999), the large jumps from Cauchy mutation may be detrimental when the current search position is close to the neighborhood of the global optimum. In order to minimize this effect, this section attempts to pull some particles close to the best solution found so far (gBest) and minimizes the distance between these particles and gBest. Instead of using the same parameters for all particles, each particle is allowed to adjust its parameter. Therefore, different parameters may be adopted by different particles. In order to reduce the computational time spent on parameter tuning, at iteration *i*, the distance between one randomly selected particle $\mathbf{x}_j^i$ and the gBest $\mathbf{p}_g^i$ is minimized. Taking $w_j^i, c_{1,j}^i$ and $c_{2,j}^i$ as decision variables, a *convex optimization* problem is formulated as:

$$\begin{aligned}&\min\ f_{dist}^{i-1} = \left\|\mathbf{x}_j^i - \mathbf{p}_g^i\right\|_2^2 \\ &= \left\|\mathbf{x}_j^{i-1} + w_j^{i-1}\mathbf{v}_j^{i-1} + c_{1,j}^{i-1}r_{1,j}^{i-1} \times \left(\mathbf{p}_j^{i-1} - \mathbf{x}_j^{i-1}\right) + c_{2,j}^{i-1}r_{2,j}^{i-1} \times \left(\mathbf{p}_g^{i-1} - \mathbf{x}_j^{i-1}\right) - \mathbf{p}_g^i\right\|_2^2 \\ &\text{s.t.}\quad 0.5 \le c_1 \le 2.5,\quad 0.5 \le c_2 \le 2.5,\quad 0.4 \le w \le 0.9\end{aligned} \tag{4.25}$$

It is intuitive that the three constraints in Eq. (4.25) satisfy the particle stability conditions expressed in Eq. (4.6). The sub-gradient method (Boyd, 2010) is employed to solve the convex optimization problem formulated in Eq. (4.25). Taking w_j^i as an example, it can updated as described in the following equation

$$w_j^i = w_j^{i-1} - \alpha_j^{i-1} g_{w_j}^{i-1} \tag{4.26}$$

where α_j^{i-1} and $g_{w_j}^{i-1}$ are the step size and sub-gradient of the objective function in Eq. (4.25) at iteration *i* for particle *j*. Since the objective function in Eq. (4.25) is derivable, the derivative of f_{dist}^{i-1} evaluated at w_j^{i-1} is used as $g_{w_j}^{i-1}$. The optimal step size when the optimal value f_{dist}^* of the convex objective function is known is Polyak's step size (Boyd, 2010) which is computed as

$$\alpha_j^{i-1} = \left(f_{dist}^{i-1} - f_{dist}^*\right) \Big/ \left\{\left(g_{w_j}^{i-1}\right)^2 + \left(g_{c_{1,j}}^{i-1}\right)^2 + \left(g_{c_{2,j}}^{i-1}\right)^2\right\} \tag{4.27}$$

Since the optimal value f_{dist}^* is always 0, the parameters *w*, *c₁* and *c₂* are updated using Eqs. (4.26), (4.28) and (4.29)

$$c_{1,j}^i = c_{1,j}^{i-1} - \alpha_j^{i-1} g_{c_{1,j}}^{i-1} \tag{4.28}$$

$$c_{2,j}^i = c_{2,j}^{i-1} - \alpha_j^{i-1} g_{c_{2,j}}^{i-1} \tag{4.29}$$

where

$$g_{w_j}^{i-1} = \sum_{d=1}^{D} 2\left(x_{j,d}^i - p_{g,d}^i\right) \times v_{j,d}^{i-1} \tag{4.30}$$

$$g_{c_{1,j}}^{i-1} = \sum_{d=1}^{D} 2\left(x_{j,d}^i - p_{g,d}^i\right) \times r_{1,j}^{i-1} \times \left(p_{j,d}^{i-1} - x_{j,d}^{i-1}\right) \tag{4.31}$$

$$g_{c_{2,j}}^{i-1} = \sum_{d=1}^{D} 2\left(x_{j,d}^{i} - p_{g,d}^{i}\right) \times r_{2,j}^{i-1} \times \left(p_{g,d}^{i-1} - x_{j,d}^{i-1}\right) \qquad (4.32)$$

and $x_{j,d}^{i}$ is the d^{th} component of $\mathbf{x}_j^i$.

4.4 Experimental Analysis

Thirty-one test functions are collected from the literature (Iwasaki et al., 2006; Liang et al., 2006; Salomon, 1996; Wang et al., 2010; Yao et al., 1999). The general formulas of these 31 test functions are listed in Appendix A where $\mathbf{z} = \mathbf{M} \times (\mathbf{x} - \mathbf{o})$, **o** is employed to shift the global optimal solution of the original function from the center of the search range to a new location and the orthogonal rotated matrix (Salomon, 1996) **M** is used to increase the complexity of the function by changing separable functions to non-separable functions without altering the shape of the function. Vector **o** is 0 if the function is non-shifted and matrix **M** is a *D*-dimensional identity matrix if the function is non-rotated. The function name, corresponding general formula, search range on the **x**-space and the optimal solution **z*** on the **z**-space of each test function is listed in Table 15. The global optimal objective values for these 31 test functions are 0. The equivalent number of function evaluations for sub-gradient calculation, SGFE, is determined as the total number of floating point operations (FLOP) for the *D*-dimensional sub-gradient calculations divided by the floating point operations (FLOP) of the objective function evaluation.

$$SGFE = ceil\left(\sum_{d=1}^{D} FLOP_{g_d} / FLOP_f\right) \qquad (4.33)$$

where ceil(.) rounds the element to the nearest integer towards infinity; FLOP is the output of the “flops” function in MATLAB®.

The 31 test functions in Table 15 are divided into six groups: 1) 6 uni-modal non-rotated functions (scaled and non-noisy); 2) 6 uni-modal rotated functions (scaled and non-noisy); 3) 11 multi-modal non-rotated functions (scaled and non-noisy); 4) 4 multi-modal rotated functions (scaled and non-noisy); 5) 2 noisy functions; 6) 2 mis-scaled functions.

Table 15 Thirty-one benchmark functions

Category	No.	Name	Formula	Search Range	Optimal Solution z*
Uni-modal Non-rotated Functions	f_1	Shifted Sphere	F_{Sphere}	$[-100, 100]^D$	**0**
	f_2	Shifted Schwefel P2.22	$F_{Schwefel2.22}$	$[-10, 10]^D$	**0**
	f_3	Shifted Schwefel P1.2	$F_{Schwefel1.2}$	$[-100, 100]^D$	**0**
	f_4	Shifted Schwefel P2.21	$F_{Schwefel2.21}$	$[-100, 100]^D$	**0**
	f_5	Shifted Rosenbrock	$F_{Rosenbrock}$	$[-100, 100]^D$	**1**
	f_6	Shifted Step	F_{Step}	$[-100, 100]^D$	**[-0.5, 0.5)**
Uni-modal Rotated Functions	f_7	Shifted Rotated Sphere	F_{Sphere}	$[-100, 100]^D$	**0**
	f_8	Shifted Rotated Schwefel P2.21	$F_{Schwefel2.21}$	$[-100, 100]^D$	**0**
	f_9	Shifted Rotated Rosenbrock	$F_{Rosenbrock}$	$[-100, 100]^D$	**1**
	f_{10}	Shifted Rotated Tablet	F_{Tablet}	$[-100, 100]^D$	**0**
	f_{11}	Shifted Rotated Ellipse	$F_{Ellipse}$	$[-100, 100]^D$	**0**
	f_{12}	Shifted Rotated Diff Power	$F_{Diffpower}$	$[-100, 100]^D$	**0**
Multi-modal Non-rotated Functions	f_{13}	Schwefel	$F_{Schwefel}$	$[-500, 500]^D$	**420.9687**
	f_{14}	2^D minima	$F_{2Dminima}$	$[-5, 5]^D$	**-2.9035**
	f_{15}	Shifted Rastrigin	$F_{Rastrigin}$	$[-5, 5]^D$	**0**
	f_{16}	Shifted Noncontinuous Rastrigin	$F_{NoncRastrigin}$	$[-5, 5]^D$	**0**
	f_{17}	Shifted Ackley	F_{Ackley}	$[-32, 32]^D$	**0**
	f_{18}	Shifted Griewank	$F_{Griewank}$	$[-600, 600]^D$	**0**
	f_{19}	Weierstrass	$F_{Weierstrass}$	$[-0.5, 0.5]^D$	**0**
	f_{20}	Shifted Salomon	$F_{Salomon}$	$[-100, 100]^D$	**0**
	f_{21}	Schwefel P2.13	$F_{Schwefel2.13}$	$[-\pi, \pi]^D$	**α**
	f_{22}	Shifted Penalized 1	$F_{Penalized1}$	$[-50, 50]^D$	**-1**
	f_{23}	Shifted Penalized 2	$F_{Penalized2}$	$[-50, 50]^D$	**1**
Multi-modal Rotated Functions	f_{24}	Rotated 2^D minima	$F_{2Dminima}$	$[-5, 5]^D$	**-2.9035**
	f_{25}	Shifted Rotated Griewank	$F_{Griewank}$	$[-600, 600]^D$	**0**
	f_{26}	Rotated Weierstrass	$F_{Weierstrass}$	$[-0.5, 0.5]^D$	**0**
	f_{27}	Shifted Rotated Salomon	$F_{Salomon}$	$[-100, 100]^D$	**0**
Noisy Functions	f_{28}	Shifted Noise Schwefel P1.2	$F_{NoiseSch1.2}$	$[-100, 100]^D$	**0**
	f_{29}	Shifted Rotated Noise Quadric	$F_{NoiseQuadric}$	$[-100, 100]^D$	**0**
Mis-scaled Functions	f_{30}	Shifted Rastrigin10	$F_{Rastrigin10}$	$[-5, 5]^D$	**0**
	f_{31}	Shifted Rastrigin100	$F_{Rastrigin100}$	$[-5, 5]^D$	**0**

4.4.1 Performance Metrics

In the experiments, the population size is set to be 30 and the maximum number of function evaluations is set to be 300,000. For all test functions, the algorithms carry out 30 independent runs. To evaluate the overall performance in regards to both the solution quality and computing cost, the metrics success performance (SP), success rate (SR), and fitness value are adopted from (Auger & Hansen, 2005). A run during which the algorithm achieves the fixed accuracy level within the maximum number of function evaluations is considered to be successful. In this example, the accuracy level is set to be 10^{-5}. The success rate (SR) is defined as

$$SR = \#\text{ of successful runs}/\text{total \# of runs} \tag{4.34}$$

The mean of the success performance (SP) is defined as (Auger & Hansen, 2005)

$$\text{mean}\left(SP\right)=\left(\left(1-SR\right)/SR\right)FE_{\max}+\text{mean}\left(\text{\# of func. eval. for successful runs}\right) \quad (4.35)$$

4.4.2 Comparison between BLOSSM-APSO and 11 PSO Algorithms

Experiments are conducted to compare twelve PSO algorithms including ten collected from literature, BLOSSM-PSO and the BLOSSM-APSO on 31 test functions with 30 dimensions using two metrics, fitness value and success performance (SP). The compared PSO algorithms and their parameter settings are:

1) PSO with inertia weight (PSO-w) (Shi & Eberhart, 1998): w=0.9-0.5i/I, c_1=c_2=2;

2) PSO with constriction factor (PSO-cf) (Clerc & Kennedy, 2002): w=0.729, c_1=c_2=1.49445;

3) Local version of PSO with inertia weight (PSO-w-local) (Kennedy & Mendes, 2002): w=0.9-0.5i/I, c_1=c_2=2;

4) Local version of PSO with constriction factor (PSO-cf-local) (Kennedy & Mendes, 2002): w=0.729, c_1=c_2=1.49445;

5) Unified PSO (UPSO) (Parsopoulos & Vrahatis, 2004): w=0.729, c_1=c_2=1.49445;

6) Weighted fully informed particle swarm (wFIPS) (Mendes et al., 2004): w=0.729, c_1=c_2=2;

7) Fitness-Distance-Ratio based PSO (FDR-PSO) (Peram et al., 2003): w=0.9-0.5i/I, f_1=f_2=1, f_3=2;

8) Cooperative PSO (CPSO-H) (van den Bergh & Engelbrecht, 2004): w=0.9-0.5i/I, c_1=c_2=1.49;

9) Comprehensive learning PSO (CLPSO) (Liang et al., 2006): w=0.9-0.7i/I, c_1=c_2=1.49445;

10) Dynamic multi-swarm PSO (DMS-PSO) (Liang & Suganthan, 2005): w=0.729, c_1=c_2=1.49445, n=6, m=5;

11) Bi-Local searches and mutation based PSO (BLOSSM-PSO): w=0.7298, c_1=c_2=1.4961;

12) Bi-Local searches and mutation based adaptive PSO (BLOSSM-APSO): adaptively change w, c_1, c_2;

where i is the current iteration number, and I is the maximum number of iteration. Additional parameter settings for PSO-w, PSO-cf, PSO-w-local, PSO-cf-local, UPSO, wFIPS, FDR-PSO, CPSO-H and CLPSO are the same as (Liang et al., 2006).

The statistical comparison of the BLOSSM-APSO with the other eleven PSO algorithms uses a two-tailed t-test with 58 degrees of freedom at a 0.05 level of significance. Values "+", "=" and "-" in the column "h" in Table 16~21 denote BLOSSM-APSO performs significantly better than, almost the same as, and significantly worse than the compared algorithm, respectively. The first value in column "h" is the t-test result on the fitness value and the second value is on the success performance.

4.4.2.1 Uni-modal Non-rotated Functions

In the first set of experiments, 6 uni-modal and non-rotated functions are studied (Table 15). The optimization results are summarized in Table 16. Please note column SP is blank when there are no successful runs among the 30 runs (SR=0).

Table 16 Optimization results for uni-modal non-rotated functions

Algorithms	Fitness Value	SP	SR (%)	h	Fitness Value	SP	SR (%)	h
	Shifted Sphere (f_1)				Shifted Schwefel P2.22 (f_2)			
PSO-w	2.21E-28	176162	100	-/+	9.55E-16	176346	100	-/+
PSO-cf	1.77E-27	14517	100	=/+	1.41E-13	28835	100	-/-
PSO-w-local	2.02E-27	219943	100	=/+	5.63E-16	221545	100	-/+
PSO-cf-local	6.84E-30	24389	100	-/+	**0.00E+00**	31038	100	-/-
UPSO	**0.00E+00**	15708	100	-/+	**0.00E+00**	**20956**	100	-/-
wFIPS	5.21E-27	79319	100	+/+	2.46E-14	107719	100	-/+
FDR-PSO	1.26E-30	99527	100	-/+	**0.00E+00**	101755	100	-/+
CPSO-H	2.42E-12	95478	100	+/+	2.71E-07	201064	100	+/+
CLPSO	**0.00E+00**	92887	100	-/+	**0.00E+00**	107976	100	-/+
DMS-PSO	7.15E-30	24082	100	-/+	**0.00E+00**	30186	100	-/-
BLOSSM-PSO	5.65E-24	3063	100	=/+	7.00E-07	89113	100	+/+
BLOSSM-APSO	1.02E-27	**1978**	100		1.62E-08	44613	100	

	Shifted Schwefel P1.2 (f_3)				Shifted Schwefel P2.21 (f_4)			
PSO-w	2.47E-02		0	+/+	2.86E-01		0	+/+
PSO-cf	1.73E-22	103109	100	+/+	3.08E-10	167606	100	+/+
PSO-w-local	2.80E+03		0	+/+	1.18E+00		0	+/+
PSO-cf-local	1.16E-09	198400	100	+/+	2.16E-09	186927	100	+/+
UPSO	5.47E-11	183693	100	+/+	1.54E-05	571530	50	+/+
wFIPS	1.91E+00		0	+/+	4.97E-05		0	+/+
FDR-PSO	2.55E-18	186429	100	+/+	4.11E-04		0	+/+
CPSO-H	4.96E+03		0	+/+	7.98E-05		0	+/+
CLPSO	2.17E+02		0	+/+	5.35E-01		0	+/+
DMS-PSO	1.10E+00		0	+/+	9.53E-12	154354	100	+/+
BLOSSM-PSO	5.00E-14	**45998**	100	+/-	1.61E-08	**25400**	100	+/-
BLOSSM-APSO	**4.76E-26**	53849	100		**4.81E-16**	32949	100	
	Shifted Rosenbrock (f_5)				Shifted Step (f_6)			
PSO-w	4.97E+01		0	+/+	1.00E-01	186372	90	=/+
PSO-cf	9.74E+00		0	=/+	1.56E+01		0	+/+
PSO-w-local	6.42E+01		0	+/+	**0.00E+00**	191646	100	=/+
PSO-cf-local	1.38E+01		0	=/+	**0.00E+00**	12408	100	=/+
UPSO	1.13E+01		0	=/+	**0.00E+00**	8943	100	=/+
wFIPS	2.85E+01		0	+/+	**0.00E+00**	33422	100	=/+
FDR-PSO	**1.46E+00**	8975940	3.33	=/+	3.00E-01	169571	76.7	+/+
CPSO-H	2.95E+01		0	+/+	**0.00E+00**	8709	100	=/+
CLPSO	3.71E+00		0	=/+	**0.00E+00**	55895	100	=/+
DMS-PSO	2.80E+01		0	+/+	**0.00E+00**	11696	100	=/+
BLOSSM-PSO	1.14E+01	119612	83.3	=/=	**0.00E+00**	1271	100	=/=
BLOSSM-APSO	9.08E+00	**118691**	83.3		**0.00E+00**	**1260**	100	

BLOSSM-APSO outperforms the other 10 algorithms on 5 out of 6 functions except the shifted Schwefel P2.22 function (f_2) in terms of convergence speed (success performance). BLOSSM-APSO converges faster than the 10 algorithms on shifted Sphere function (f_1) and shifted Rosenbrock (f_5) function with comparable solution quality (fitness value). Comparing BLOSSM-APSO with BLOSSM-PSO, it is observed that BLOSSM-APSO could improve solution quality for 5 out of 6 functions and convergence speed for 4 out of 6 functions. Particularly, BLOSSM-APSO tremendously improves the convergence speed for shifted Schwefel P2.22 (f_2) function whose search space is fully coupled. And the success performance value is comparable with the best value (20956) which is obtained by UPSO algorithm. For shifted Schwefel P1.2 (f_3) and shifted Schwefel P2.21 (f_4) functions, BLOSSM-APSO improves the solution quality within a comparable convergence speed.

4.4.2.2 Uni-modal Rotated Functions

In the second set of experiments, 6 uni-modal rotated functions are tested (Table 15). The optimization results are summarized in Table 17.

Table 17 Optimization results for uni-modal rotated functions

Algorithms	Fitness Value	SP	SR (%)	h	Fitness Value	SP	SR (%)	h
	Shifted Rotated Sphere (f_7)				Shifted Rotated Schwefel P2.21 (f_8)			
PSO-w	1.96E-28	176633	100	-/+	4.72E-03		0	+/+
PSO-cf	6.40E-29	14618	100	-/+	5.87E-13	128094	100	=/+
PSO-w-local	2.46E-27	220891	100	+/+	1.45E-01		0	+/+
PSO-cf-local	5.05E-30	24307	100	-/+	4.28E-13	135275	100	+/+
UPSO	8.41E-31	15833	100	-/+	4.10E-08	196394	100	=/+
wFIPS	4.98E-27	79141	100	+/+	1.16E-06	262428	100	+/+
FDR-PSO	**0.00E+00**	99737	100	-/+	8.49E-07	237174	96.7	=/+
CPSO-H	3.05E-12	93277	100	+/+	5.52E+00		0	=/+
CLPSO	**0.00E+00**	93487	100	-/+	1.05E-01		0	+/+
DMS-PSO	7.15E-30	24111	100	-/+	1.03E-12	143619	100	+/+
BLOSSM-PSO	1.50E-22	2874	100	=/+	2.64E-08	**27601**	100	+/-
BLOSSM-APSO	9.77E-28	**2041**	100		**4.32E-16**	32847	100	
	Shifted Rotated Rosenbrock (f_9)				Shifted Rotated Tablet (f_{10})			
PSO-w	6.26E+02		0	+/+	4.47E+02		0	+/+
PSO-cf	4.94E+02		0	=/+	2.87E+01		0	+/+
PSO-w-local	4.23E+02		0	+/+	1.89E+03		0	+/+
PSO-cf-local	8.94E+01		0	+/+	2.87E+02		0	+/+
UPSO	3.95E+01		0	+/+	1.06E+03		0	+/+
wFIPS	5.67E+01		0	+/+	1.21E+03		0	+/+
FDR-PSO	2.40E+01		0	=/+	2.20E+02		0	+/+
CPSO-H	2.54E+02		0	=/+	1.52E+04		0	+/+
CLPSO	2.99E+01		0	+/+	4.68E+02		0	+/+
DMS-PSO	4.35E+01		0	+/+	3.93E+01		0	+/+
BLOSSM-PSO	1.60E+01	209194	66.7	=/=	2.23E-23	105678	100	+/+
BLOSSM-APSO	**9.76E+00**	**132059**	80		**2.23E-25**	**75981**	100	
	Shifted Rotated Ellipse (f_{11})				Shifted Rotated Diff Power (f_{12})			
PSO-w	2.49E-04	2990170	10	=/+	7.78E-06	229806	96.7	=/+
PSO-cf	3.39E-25	62506	100	=/+	2.18E-13	26680	100	-/+
PSO-w-local	4.87E+04		0	+/+	1.95E+06	864481	33.3	+/+
PSO-cf-local	2.43E-17	129467	100	+/+	4.92E-13	39895	100	-/+
UPSO	1.01E-21	108496	100	+/+	1.19E-13	30406	100	-/+
wFIPS	8.83E-03		0	+/+	8.44E-11	102229	100	+/+
FDR-PSO	**1.22E-25**	163049	100	=/+	**3.91E-14**	103330	100	-/+
CPSO-H	4.53E+03		0	+/+	1.13E+07	1043833	23.3	=/+
CLPSO	1.08E+02		0	+/+	7.62E-09	173461	100	+/+
DMS-PSO	4.15E-10	212431	100	+/+	4.03E-12	45216	100	-/+
BLOSSM-PSO	2.42E-16	46759	100	=/+	6.73E-11	16850	100	+/+
BLOSSM-APSO	8.33E-23	**46053**	100		1.46E-11	**14632**	100	

BLOSSM-APSO outperforms the 10 algorithms on all six functions in terms of convergence speed, and 3 out of 6 functions in terms of solution quality. The solution quality is comparable with the 10 algorithms on shifted rotated Sphere (f_7), shifted rotated Ellipse (f_{11}) and shifted rotated Diff Power (f_{12}). Comparing BLOSSM-APSO with BLOSSM-PSO, BLOSSM-APSO improves solution quality for all the uni-modal rotated functions, and improves convergence speed for 5 out of 6 functions. BLOSSM-APSO is more reliable than BLOSSM-PSO on the shifted rotated Rosenbrock (f_9) function.

4.4.2.3 Multi-modal Non-rotated Functions

In the third set of experiments, 11 multi-modal non-rotated functions are explored (Table 15). The optimization results are summarized in Table 18.

Table 18 Optimization results for multi-modal non-rotated functions

Algorithms	**Fitness Value**	**SP**	**SR (%)**	**h**	**Fitness Value**	**SP**	**SR (%)**	**h**
	Schwefel (f_{13})				2^D minima (f_{14})			
PSO-w	1.09E+03		0	+/+	4.71E+00		0	+/+
PSO-cf	2.92E+03		0	+/+	1.12E+01		0	+/+
PSO-w-local	5.16E+03		0	+/+	**4.57E-10**	200131	100	=/+
PSO-cf-local	2.18E+03		0	+/+	7.01E+00		0	+/+
UPSO	3.65E+03		0	+/+	8.07E+00		0	+/+
wFIPS	2.37E+01	220819	80	+/+	4.08E-01	269664	60	+/+
FDR-PSO	3.10E+03		0	+/+	1.06E+01		0	+/+
CPSO-H	2.37E+02	8783760	3.33	+/+	**4.57E-10**	37136	100	=/+
CLPSO	**1.70E-08**	95772	100	=/+	**4.57E-10**	75820	100	=/+
DMS-PSO	2.43E+03		0	+/+	3.61E+00	4228670	6.67	+/+
BLOSSM-PSO	**1.70E-08**	86946	100	=/=	**4.57E-10**	18163	100	=/=
BLOSSM-APSO	**1.70E-08**	**77792**	100		**4.57E-10**	**17731**	100	
	Shifted Rastrigin (f_{15})				Shifted Noncontinuous Rastrigin (f_{16})			
PSO-w	2.01E+01		0	+/+	7.57E+00	2968630	10	+/+
PSO-cf	7.26E+01		0	+/+	4.08E+01		0	+/+
PSO-w-local	2.93E+01		0	+/+	1.66E+01		0	+/+
PSO-cf-local	4.23E+01		0	+/+	6.07E+00	715855	36.7	+/+
UPSO	6.87E+01		0	+/+	7.09E+01		0	+/+
wFIPS	2.80E+01		0	+/+	4.16E+01		0	+/+
FDR-PSO	2.84E+01		0	+/+	7.63E+00		0	+/+
CPSO-H	9.95E-02	117335	90	=/=	2.33E-01	158412	80	+/=
CLPSO	**0.00E+00**	159419	100	=/+	**0.00E+00**	167351	100	=/+
DMS-PSO	7.16E+00		0	+/+	3.80E+00	377190	63.3	+/+
BLOSSM-PSO	**0.00E+00**	119597	100	=/=	**0.00E+00**	137889	100	=/=
BLOSSM-APSO	**0.00E+00**	**108706**	100		**0.00E+00**	**122212**	100	
	Shifted Ackley (f_{17})				Shifted Griewank (f_{18})			
PSO-w	3.10E-14	192348	100	=/+	2.12E-02	1004104	26.7	+/+
PSO-cf	1.53E+00	1222345	20	+/+	2.44E-02	715564	30	+/+
PSO-w-local	2.34E-14	239791	100	=/+	5.91E-03	442035	56.7	+/+
PSO-cf-local	6.39E-15	34510	100	=/-	8.19E-03	257348	56.7	+/+
UPSO	**3.55E-15**	**22997**	100	=/-	1.89E-03	85077	83.3	+/+
wFIPS	2.52E-14	117343	100	=/+	**0.00E+00**	98894	100	-/+
FDR-PSO	1.85E-14	109428	100	=/+	1.26E-02	625812	36.7	+/+
CPSO-H	3.03E-07	199111	100	+/+	2.07E-02	621219	36.7	+/+
CLPSO	8.05E-15	118336	100	=/+	**0.00E+00**	121255	100	-/+
DMS-PSO	**3.55E-15**	34502	100	=/-	**0.00E+00**	29281	100	-/+
BLOSSM-PSO	8.74E-07	88067	100	+/+	7.59E-15	12577	100	+/+
BLOSSM-APSO	8.55E-12	56538	100		1.73E-15	**8473**	100	
	Weierstrass (f_{19})				Shifted Salomon (f_{20})			
PSO-w	1.01E-01	308081	73.3	=/+	3.81E-01		0	+/+
PSO-cf	7.61E+00		0	+/+	5.90E-01		0	+/+
PSO-w-local	**0.00E+00**	238863	100	=/+	3.13E-01		0	+/+
PSO-cf-local	7.27E-01	874227	26.7	+/+	2.33E-01		0	+/+
UPSO	4.22E-01	635110	33.3	+/+	5.37E-01		0	+/+
wFIPS	**0.00E+00**	167853	100	=/+	2.00E-01		0	+/+
FDR-PSO	1.07E-01	950523	26.7	=/+	3.50E-01		0	+/+
CPSO-H	9.88E-05		0	+/+	1.37E+00		0	+/+
CLPSO	**0.00E+00**	142057	100	=/+	2.32E-01		0	+/+
DMS-PSO	**0.00E+00**	**48357**	100	=/-	2.00E-01		0	+/+
BLOSSM-PSO	1.24E-05	530459	53.3	+/+	3.08E-11	24194	100	+/=
BLOSSM-APSO	**0.00E+00**	105011	100		**0.00E+00**	**21162**	100	
	Schwefel P2.13 (f_{21})				Shifted Penalized 1 (f_{22})			
PSO-w	8.65E+04		0	+/+	2.07E-02	249245	80	+/+
PSO-cf	1.61E+05		0	+/+	3.74E-01	361848	46.7	+/+
PSO-w-local	7.02E+04		0	+/+	6.91E-03	246538	93.3	=/+
PSO-cf-local	6.26E+04		0	+/+	1.21E-01	224849	60	+/+
UPSO	1.09E+05		0	+/+	2.76E-02	86255	83.3	=/=
wFIPS	1.22E+04		0	+/+	1.08E-29	61000	100	-/+

FDR-PSO	1.38E+04		0	+/+	3.46E-03	99463	96.7	=/+
CPSO-H	2.74E+04		0	+/+	1.98E-14	38407	100	=/=
CLPSO	1.01E+04		0	+/+	**1.57E-32**	83945	100	-/+
DMS-PSO	1.01E+04		0	=/+	**1.57E-32**	**21791**	100	-/-
BLOSSM-PSO	9.04E+03	**8793708**	3.33	=/=	7.24E-18	35015	100	+/=
BLOSSM-APSO	**6.33E+03**	8800622	3.33		2.50E-27	34605	100	
	Shifted Penalized 2 (f_{23})							
PSO-w	2.20E-03	257914	80	+/+				
PSO-cf	2.76E-01	317006	50	=/+				
PSO-w-local	1.83E-03	301082	83.3	+/+				
PSO-cf-local	2.17E-03	84526	83.3	+/+				
UPSO	7.32E-04	39932	93.3	=/=				
wFIPS	3.42E-28	69452	100	+/+				
FDR-PSO	1.10E-03	124916	90	=/+				
CPSO-H	1.61E-13	60603	100	+/+				
CLPSO	**1.35E-32**	89144	100	-/+				
DMS-PSO	5.56E-32	24968	100	-/+				
BLOSSM-PSO	2.41E-28	12612	100	=/=				
BLOSSM-APSO	2.29E-28	**11687**	100					

Comparing BLOSSM-APSO with the other 10 algorithms, BLOSSM-APSO is inferior to DMS-PSO on shifted Ackley (f_{17}), Weierstrass (f_{19}) and shifted Penalized 1 (f_{22}) functions. Comparing BLOSSM-APSO with BLOSSM-PSO, BLOSSM-APSO is superior to BLOSSM-PSO on 7 out of 11 functions in terms of solution quality, and 10 out of 11 functions on success performance. The success performance on Schwefel P2.13 (f_{21}) function is almost the same. Taking functions shifted Ackley (f_{17}) and Weierstrass (f_{19}) as an example, the performance of BLOSSM-PSO on these two functions is poorer than the other 10 algorithms. It is observed that BLOSSM-APSO converges much faster than BLOSSM-PSO. The success performance (56538) on shifted Ackley (f_{17}) function is comparable to the best value (22997) obtained by UPSO algorithm. BLOSSM-APSO achieves 100% success rate on Weierstrass (f_{19}) and raises its rank from 6th to 2nd among the 10 algorithms.

4.4.2.4 Multi-modal Rotated Functions

In the fourth set of experiments, 4 multi-modal rotated functions are studied (Table 15). The optimization results are summarized in Table 19.

Table 19 Optimization results for multi-modal rotated functions

Algorithms	**Fitness Value**	**SP**	**SR (%)**	**h**	**Fitness Value**	**SP**	**SR (%)**	**h**
	Rotated 2^D minima (f_{24})				Shifted Rotated Griewank (f_{25})			
PSO-w	6.71E+00		0	=/=	1.32E-02	1012572	26.7	+/+
PSO-cf	1.02E+01		0	+/=	1.86E-02	408508	43.3	+/+

PSO-w-local	3.75E+00		0	-/=	7.71E-03	483820	53.3	+/+
PSO-cf-local	9.19E+00		0	+/=	6.65E-03	327566	50	+/+
UPSO	9.75E+00		0	+/=	3.38E-03	154359	70	+/+
wFIPS	2.08E+00		0	-/=	**0.00E+00**	99049	100	-/+
FDR-PSO	1.10E+01		0	+/=	1.34E-02	708387	33.3	+/+
CPSO-H	6.81E+00		0	=/=	3.42E-02	810937	30	+/+
CLPSO	**1.89E+00**		0	-/=	4.20E-10	143041	100	=/+
DMS-PSO	6.20E+00		0	=/=	6.57E-04	58795	93.3	=/+
BLOSSM-PSO	7.57E+00		0	=/=	9.38E-15	19916	100	+/=
BLOSSM-APSO	6.89E+00		0		3.83E-15	**9284**	100	
	Rotated Weierstrass (f_{26})				Shifted Rotated Salomon (f_{27})			
PSO-w	7.33E+00		0	=/=	3.93E-01		0	+/+
PSO-cf	1.25E+01		0	+/=	5.97E-01		0	+/+
PSO-w-local	2.76E+00		0	-/=	3.01E-01		0	+/+
PSO-cf-local	7.40E+00		0	=/=	2.53E-01		0	+/+
UPSO	1.48E+01		0	+/=	5.47E-01		0	+/+
wFIPS	**6.31E-02**	**421625**	60	-/-	1.97E-01		0	+/+
FDR-PSO	2.44E+00		0	-/=	3.43E-01		0	+/+
CPSO-H	1.24E+01		0	+/=	1.30E+00		0	+/+
CLPSO	1.37E+00		0	-/=	2.24E-01		0	+/+
DMS-PSO	1.17E-01		0	-/=	1.97E-01		0	+/+
BLOSSM-PSO	8.06E+00		0	=/=	8.76E-11	24670	100	+/=
BLOSSM-APSO	8.02E+00		0		**0.00E+00**	**21791**	100	

Comparing BLOSSM-APSO with BLOSSM-PSO, BLOSSM-APSO outperforms BLOSSM-PSO on these four multi-modal rotated functions in terms of both solution quality and convergence speed. However, the improvements on rotated 2^D minima (f_{24}) and rotated Weierstrass (f_{26}) are very small, and the performance of BLOSSM-APSO on these two complex functions is still inferior to other algorithms designed for the specific search space (multi-modal rotated space).

4.4.2.5 Noisy Functions

In the fifth set of experiments, 2 noisy functions are tested (Table 15). The optimization results are summarized in Table 20.

Table 20 Optimization results for noisy functions

Algorithms	**Fitness Value**	**SP**	**SR (%)**	**h**	**Fitness Value**	**SP**	**SR (%)**	**h**
	Shifted Noise Schwefel P1.2 (f_{28})				Shifted Rotated Noise Quadric (f_{29})			
PSO-w	1.20E+03		0	+/=	9.24E-03		0	+/=
PSO-cf	7.43E+02		0	+/=	3.90E-03		0	-/=
PSO-w-local	1.38E+03		0	+/=	2.07E-02		0	+/=
PSO-cf-local	1.37E+02		0	=/=	**2.37E-03**		0	-/=
UPSO	2.69E+03		0	+/=	1.89E-02		0	+/=
wFIPS	2.09E+02		0	+/=	2.68E-03		0	-/=
FDR-PSO	2.52E+02		0	+/=	7.07E-03		0	=/=
CPSO-H	1.76E+04		0	+/=	1.45E-02		0	+/=
CLPSO	2.09E+03		0	+/=	4.15E-03		0	-/=
DMS-PSO	2.34E+02		0	+/=	3.84E-03		0	-/=
BLOSSM-PSO	**3.31E+00**		0	-/=	4.89E-03		0	-/=
BLOSSM-APSO	8.98E+01		0		6.68E-03		0	

Unlike other functions, BLOSSM-APSO is inferior to BLOSSM-PSO on the two noisy functions. The parameter tuning mechanism accelerates the convergence speed of the particles to the global best solution which may be the worst particle among the swarm without considering noise. Although BLOSSM-APSO underperforms BLOSSM-PSO on noisy functions, it is still the most effective algorithm among the 10 algorithms on shifted noise Schwefel P1.2 (f_{28}) function.

4.4.2.6 Mis-scaled Functions

Two mis-scaled functions are studied in this set of experiments (Table 15). The optimization results are summarized in Table 21.

Table 21 Optimization results for mis-scaled functions

Algorithms	**Fitness Value**	**SP**	**SR (%)**	**h**	**Fitness Value**	**SP**	**SR (%)**	**h**
	Shifted Rastrigin10 (f_{30})				Shifted Rastrigin100 (f_{31})			
PSO-w	2.89E+01		0	+/+	2.58E+01		0	+/+
PSO-cf	1.06E+02		0	+/+	1.46E+02		0	+/+
PSO-w-local	3.92E+01		0	+/+	3.93E+01		0	+/+
PSO-cf-local	5.47E+01		0	+/+	6.40E+01		0	+/+
UPSO	8.91E+01		0	+/+	1.14E+02		0	+/+
wFIPS	5.32E+01		0	+/+	5.27E+01		0	+/+
FDR-PSO	3.61E+01		0	+/+	5.18E+01		0	+/+
CPSO-H	1.33E-01	123673	96.7	=/+	2.32E-01	243602	76.7	+/+
CLPSO	**0.00E+00**	176403	100	=/+	**0.00E+00**	176792	100	-/+
DMS-PSO	9.65E+00	8914706	3.33	+/+	1.37E+01		0	+/+
BLOSSM-PSO	4.60E-09	119764	100	=/+	8.54E-08	161883	100	+/+
BLOSSM-APSO	7.40E-14	**98376**	100		2.18E-14	**101300**	100	

BLOSSM-APSO outperforms BLOSSM-PSO and the 10 PSO algorithms on the 2 mis-scaled Rastrigin functions in terms of convergence speed, and is comparable with CLPSO on solution quality.

4.4.2.7 Conclusions on Comparison between BLOSSM-APSO and 11 PSO Algorithms

It is observed the BLOSSM-APSO in general outperforms the other algorithms on most of the test functions. The dominance relation (Reyes-Sierra & Coello Coello, 2006) is employed to comprehensively compare BLOSSM-APSO with the 11 PSO algorithms. The dominance relation for two algorithms A and B is defined as: algorithm A is said to dominate algorithm B when A is significantly better than B on at least one performance metric without sacrificing on any other metrics. Algorithm A is dominated by algorithm B when A is significantly worse than B on at least one performance metric without improvement on any other metrics. Algorithm A and B are Pareto when A significantly outperforms B on at least one performance metric and significantly underperforms B on at least one other metric. The overall comparison between BLOSSM-APSO and other 11 algorithms is summarized in Table 22. It is observed that BLOSSM-APSO is better for improvement on convergence speed than solution quality. From the dominance relation, it is observed that BLOSSM-APSO is the most effectiveness algorithm among the 12 algorithms.

Table 22 Overall comparisons between BLOSSM-APSO and other 11 algorithms

Algorithms Metrics (t-test results)		PSO-w	PSO-cf	PSO-w-local	PSO-cf-local	UPSO	wFIPS	FDR-PSO	CPSO-H	CLPSO	DMS-PSO	BLOSSM-PSO
Fitness Value	+ (Better)	21	21	22	21	21	21	17	22	11	16	14
	= (Same)	7	6	6	5	6	3	9	9	10	6	15
	- (Worse)	3	4	3	5	4	7	5	0	10	9	2
Success Performance (SP)	+ (Better)	27	26	27	25	23	27	27	24	27	23	11
	= (Same)	4	4	4	4	6	3	4	7	4	4	17
	- (Worse)	0	1	0	2	2	1	0	0	0	4	3
Dominance Relation	+ (Dominates)	26	27	28	23	24	24	25	28	21	19	15
	= (Same)	5	2	1	5	5	4	5	3	7	6	14
	- (Dominated)	0	2	2	3	2	3	1	0	3	6	2

To quantitatively evaluate the effectiveness and generality of BLOSSM-APSO, the dominance rate for each algorithm is employed which is computed as the cumulative number of dominated functions by this algorithm comparing with the other 11 algorithms divided by the ideal case which has 341 (31x11) cumulative dominated functions. The total number of functions where algorithm B is dominated by algorithm A is recorded in the column A row B in Table 23. It is observed from Table 23 that BLOSSM-APSO has the largest dominance rate, and is the most generalized algorithm for diverse functions with different properties. BLOSSM-APSO improves the generality of BLOSSM-PSO by increasing the dominance rate from 71.0% to 76.2%.

Table 23 Dominance relations for twelve algorithms

Algorithms Algorithms	PSO-w	PSO-cf	PSO-w-local	PSO-cf-local	UPSO	wFIPS	FDR-PSO	CPSO-H	CLPSO	DMS-PSO	BLOSSM-PSO	BLOSSM-APSO

PSO-w	-	12	7	20	16	20	19	11	25	27	26	26
PSO-cf	15	-	15	18	11	19	15	13	18	20	25	27
PSO-w-local	16	13	-	21	18	23	19	14	29	29	26	28
PSO-cf-local	8	7	8	-	10	14	10	11	16	19	24	23
UPSO	14	12	12	15	-	16	14	10	17	20	26	24
wFIPS	7	10	4	13	11	-	11	7	12	20	22	24
FDR-PSO	8	7	8	16	12	14	-	11	17	24	26	25
CPSO-H	13	15	10	18	16	20	15	-	22	21	26	28
CLPSO	5	9	1	10	11	13	9	6	-	20	20	21
DMS-PSO	2	6	2	7	9	5	3	7	8	-	19	19
BLOSSM-PSO	1	1	3	3	2	5	3	0	4	7	-	15
BLOSSM-APSO	0	2	2	3	2	3	1	0	3	6	2	-
Dominance Rate (%)	26.1	27.6	21.1	42.2	34.6	44.6	34.9	26.4	50.1	62.5	71.0	**76.2**

4.4.3 Robustness Comparison

In the experiments conducted in section 4.4.2, it is observed that BLOSSM-APSO outperforms BLOSSM-PSO and the other 10 algorithms for most of the 31 functions. As discussed in the previous sections, the adaptive parameter tuning mechanism implemented in BLOSSM-APSO is capable of reducing the parameter effect and improving BLOSSM-PSO's robustness. Therefore, seven different initial parameter settings are selected which are listed in Table 24 to test the robustness of BLOSSM-APSO. For comparison, the robustness of BLOSSM-PSO is also tested. In the experiments, the population size is set to be 30 and maximum number of function evaluations is set to be 300,000. For all test functions, the algorithms carry out 30 independent runs, and the mean and standard deviation of fitness value, success performance, and success rate are recorded for *t*-test.

Table 24 Seven different initial parameter settings for robustness testing

No.	**Parameter Settings**
IPS1	w=0.7298, c_1=c_2=1.4961
IPS2	w=0.7298, c_1=c_2=2
IPS3	w=0.7298, c_1=2.5, c_2=0.5
IPS4	w=0.7298, c_1=0.5, c_2=2.5
IPS5	w=0.9, c_1=c_2=1.4961
IPS6	w=0.4, c_1=c_2=1.4961
IPS7	random initial

A two-tailed *t*-test with 58 degrees of freedom at a 0.05 level of significance is employed to test the difference of BLOSSM-APSO on these seven different initial parameter settings, and the number of functions that BLOSSM-APSO performs significantly the same in terms of solution quality and success performance on different initial parameter settings is recorded in Table 25. For example, “30” in the row “IPS2” and column “IPS5” in Table 25 denotes that BLOSSM-APSO under the second parameter setting performs the same as BLOSSM-APSO under the fifth parameter setting for 30 out of 31 functions. The robustness testing result for BLOSSM-PSO is recorded in Table 26. It is observed that the performance of BLOSSM-APSO under different initial parameter settings is almost the same. BLOSSM-APSO increases the robustness of BLOSSM-PSO and reduces the sensitivity of BLOSSM-APSO to its parameter setting.

Table 25 Robustness of BLOSSM-APSO

	Fitness Value							**Success Performance (SP)**						
	IPS1	IPS2	IPS3	IPS4	IPS5	IPS6	IPS7	IPS1	IPS2	IPS3	IPS4	IPS5	IPS6	IPS7
IPS1	-	31	31	31	31	31	31	-	31	31	31	31	31	31
IPS2	31	-	31	31	30	31	31	31	-	31	31	29	30	31
IPS3	31	31	-	31	29	31	31	31	31	-	30	31	30	31
IPS4	31	31	31	-	30	31	31	31	31	30	-	30	31	31
IPS5	31	30	29	30	-	30	30	31	29	31	30	-	31	31
IPS6	31	31	31	31	30	-	31	31	30	30	31	31	-	31
IPS7	31	31	31	31	30	31	-	31	31	31	31	31	31	-

Table 26 Robustness of BLOSSM-PSO

	Fitness Value							**Success Performance (SP)**						
	IPS1	IPS2	IPS3	IPS4	IPS5	IPS6	IPS7	IPS1	IPS2	IPS3	IPS4	IPS5	IPS6	IPS7
IPS1	-	15	16	18	6	14	19	-	11	12	16	21	12	10
IPS2	15	-	18	17	9	12	20	11	-	16	22	21	16	15
IPS3	16	18	-	25	12	15	20	12	16	-	15	21	15	16
IPS4	18	17	25	-	9	18	20	16	22	15	-	21	18	11
IPS5	6	9	12	9	-	7	18	21	21	21	21	-	20	23
IPS6	14	12	15	18	7	-	20	12	16	15	18	20	-	9
IPS7	19	20	20	20	18	20	-	10	15	16	11	23	9	-

4.5 Conclusions

In this chapter, a bi-local searches and mutation based adaptive particle swarm optimization (BLOSSM-APSO) is developed which is demonstrated to be effective for diverse functions with different properties. The adaptive parameter tuning mechanism implemented in BLOSSM-APSO is able to change the three parameters in PSO, and attempt to pull one randomly selected particle close to the gBest. The experiments conducted in this chapter demonstrate that the BLOSSM-APSO is able to improve the performance of bi-local searches and mutation based particle swarm optimization

(BLOSSM-PSO) on 29 out of 31 of the test functions. The BLOSSM-APSO is more robust to the settings of the three parameters in PSO than BLOSSM-PSO.

Due to the stochastic elements in noisy functions, BLOSSM-APSO may pull the particle close to a pseudo gBest and underperform BLOSSM-PSO on the noisy function. Therefore, the parameter tuning mechanism should be more intelligent to avoid this issue. Although BLOSSM-APSO outperforms BLOSSM-PSO on multi-modal rotated functions, its performance still could be improved comparing it with CLPSO and DMS-PSO. Some complex engineering problems such as product design, building energy system operation decisions, transportation problems should be used to test the effectiveness of BLOSSM-APSO.

Chapter 5

AN AUGMENTED PARTICLE SWARM OPTIMIZATION FOR MULTI-OBJECTIVE OPTIMIZATION

Particle Swarm Optimization (PSO) has achieved great attentions over the last decade due to its commendable performance on diverse applications, majority of which are constructed for single objective problems. When applying PSO to multi-objective problems though, the performance is less satisfactory. One reason may be that the PSO algorithms are not generalized enough to simultaneously handle diverse functions with different properties which commonly exist in multi-objective optimization (MOO). Therefore, an augmented PSO algorithm for MOO is developed, termed AMOPSO which employs bi-local searches to handle diverse functions, a crowding distance based archiving technique to maintain non-dominated solutions found during the search process, Cauchy mutation to prevent premature convergence, and a parameter tuning mechanism to adaptively change parameter settings of PSO and improve robustness of PSO. The performance of AMOPSO is evaluated on 19 problems by comparing with 4 representative multi-objective evolutionary algorithms and 4 published multi-objective PSOs using three metrics - generational distance (GD) which measures Pareto solution accuracy, maximum spreading (MS) which measures diversity of the Pareto frontier and spacing (S) which measures the distribution of Pareto solutions on Pareto frontier. Comparing AMOPSO with the 8 algorithms, the conclusions are (1) AMOPSO performs well on GD and MS measures, but less satisfactory on S. (2) AMOPSO strongly outperforms the existing 4 multi-objective PSOs and 3 MOEAs, and moderately outperforms MOCell (cellular genetic algorithm).

5.1 Introduction

Particle swarm optimization (PSO) which mimics a flock of birds that communicate together as they fly is a newly developed bio-inspired algorithm (Eberhart & Shi, 1998). PSO has been demonstrated to outperform other evolutionary algorithms (e.g., genetic algorithm, Memetic algorithm) in terms of solution quality and computational efficiency on some *single objective optimization problems* (Elbeltagi et al., 2005; Hassan et al., 2005; Kennedy & Spears, 1998). A large number of PSO variants have further been developed for some special single objective problems (e.g., multi-modal, non-separable, etc.).

During the last ten years, extensive researches are conducted to study PSO for multi-objective optimization (MOO) problems (Alvarez-Benitez et al., 2005; Bartz-Beielstein et al., 2003; Coello Coello & Lechuga, 2002; Coello Coello et al., 2004; Daneshyari & Yen, 2011; Fieldsend, 2004; Fieldsend & Singh, 2002; Goh et al., 2010; Ho et al., 2005; Hu et al., 2003; Huang et al., 2006; Janson & Merkle, 2005; Leong & Yen, 2008; Li, 2003; Lian, 2010; Liu et al., 2007; Liu et al., 2008; Mostaghim & Teich, 2003a, 2003b, 2004; Nebro et al., 2009a; Omkar et al., 2008; Pulido & Coello Coello, 2004; Raquel & Naval Jr., 2005; Ray & Liew, 2002; Reyes-Sierra & Coello Coello, 2005; Salazar-Lechuga & Rowe, 2005; Srinivasan & Seow, 2003; Tripathi et al., 2007; Villalobos-Arias et al., 2005; Wang & Yang, 2009, 2010; Yapicioglu et al., 2007; Yen & Leong, 2009; Zhao & Suganthan, 2011; Zielinski & Laur, 2007) due to its simplicity for implementation and good performance. As a result, a number of multi-objective PSO algorithms are developed. In general, the algorithms can be classified in two categories. The first category employs effective approaches (e.g., archive technique, Pareto ranking approach, etc.) which are utilized in existing multi-objective evolutionary algorithms (MOEAs) to study MOO problems (Alvarez-Benitez et al., 2005; Bartz-Beielstein et al., 2003; Coello Coello & Lechuga, 2002; Coello Coello et al., 2004; Fieldsend & Singh, 2002; Goh et al., 2010; Li, 2003; Mostaghim & Teich, 2003b; Omkar et al., 2008; Raquel & Naval Jr., 2005; Ray & Liew, 2002; Reyes-Sierra & Coello Coello, 2005; Salazar-Lechuga & Rowe, 2005; Srinivasan & Seow, 2003; Wang & Yang, 2009). For example, non-dominated sorting PSO (NSPSO) (Li, 2003) adopts a fast non-dominated sorting and sharing approach to maintain non-dominated solutions which is used in non-dominated sorting genetic algorithm (NSGA-II) (Deb et al., 2002); MOPSO developed in (Coello Coello et al., 2004) employs a hyper-cubes based adaptive grid technique to produce a well-distributed Pareto frontier; crowding distance approach is used in MOPSO-CD (Raquel & Naval Jr., 2005) to select global best solution and maintain non-dominated solution set; vector evaluated PSO (VEPSO) (Omkar et al., 2008) which is inspired by vector evaluated GA (VEGA) utilizes M (# of objectives) sub-swarms to search the M objectives separately; the multi-objective PSO algorithm developed in (Wang & Yang, 2009) uses a preference order scheme to identify the best compromise which is more efficient than Pareto ranking scheme; CCPSO (Goh et al., 2010) adopts competitive and cooperative co-evolutionary technique which explicitly models the co-evolution of competing and cooperating species to solve the multi-objective optimization problems. The second category is to augment PSO methods with focuses being on the exemplar learning, parameter tuning so it can be applied to MOO problems (Daneshyari & Yen, 2011; Ho et al., 2005; Hu et al., 2003; Huang et al., 2006; Janson & Merkle, 2005; Leong & Yen, 2008; Lian, 2010; Liu et al., 2007; Mostaghim & Teich, 2004; Nebro et al., 2009a; Pulido & Coello Coello, 2004; Tripathi et al., 2007; Yen & Leong, 2009; Zhao & Suganthan, 2011; Zielinski & Laur, 2007). For example, a modified dynamic neighborhood topology is used in the multi-objective PSO algorithm which is utilized in (Hu et al., 2003); the exemplar of each particle is selected based on a set of sub-swarms

(Janson & Merkle, 2005; Pulido & Coello Coello, 2004); CLMOPSO (Huang et al., 2006) is an extension of comprehensive learning PSO (CLPSO (Liang et al., 2006)) for multi-objective problems; time variant MOPSO (TV-MOPSO) (Tripathi et al., 2007) adaptively changes PSO's parameter settings; dynamic population multiple-swarm is used in MOPSO (Leong & Yen, 2008); cultural-based MOPSO (Daneshyari & Yen, 2011) adaptively changes PSO parameter settings using a cultural framework; two local bests are selected in MOPSO (2LB-MOPSO) (Zhao & Suganthan, 2011), just to name a few.

While great efforts have spent on exploring the application of PSO and extensions for multi-objective problems, the dominating performance of PSO on single objective problems does not extend to multi-objective PSOs. Indeed, it is observed that some multi-objective PSOs underperform other MOEAs (Goh et al., 2010; Nebro et al., 2009a; Wang & Yang, 2010; Zhao & Suganthan, 2011). This may be due to the facts that: (1) it is most likely multi-objective problems consist of different objectives with different properties, and the PSO algorithms used in multi-objective PSOs may not be generalized to handle diverse functions; (2) effective diversity preserved mechanisms for PSO are currently lacking. Such mechanisms are not only able to eliminate premature convergence issue in PSO but also able to keep diversity in Pareto frontier; (3) the performance of multi-objective optimization techniques may not only depend on their solution pools (particles in PSO, chromosomes in GA, etc.) but also some other issues, such as non-dominated solutions retaining and spreading approaches (Reyes-Sierra & Coello Coello, 2006).

To address the issues, an augmented PSO for multi-objective problem is developed, termed AMOPSO which consists of four modules: (1) bi-local searches: non-uniform mutation based method (Michalewicz, 1996) and sub-gradient method (Boyd,

2010) are employed to accelerate the convergence speed and improve solution quality for diverse functions; (2) an archiving strategy is implemented to maintain the non-dominated solutions found during the evolution process and the crowding distance method is employed to guarantee the non-dominated solutions are well spread along the Pareto frontier; (3) Cauchy mutation (Andrews, 2006) is adopted to avoid premature convergence; and (4) a parameter tuning mechanism is developed to adaptively change the parameters and thus improve the robustness of PSO. Extensive experiments are conducted to test the performance of AMOPSO and compare its performance with existing methods collected from literature.

This chapter is organized as follows: related background on MOO, PSO, and multi-objective PSO are briefly reviewed in section 5.2; followed by the detailed explanation on the AMOPSO in section 5.3; the experimental results in section 5.4 demonstrate the effectiveness of the AMOPSO algorithm, and conclusions are drawn in section 5.5.

5.2 Background Information

This section briefly reviews the backgrounds related to this chapter. The generalized formulation of multi-objective optimization is introduced in section 5.2.1, followed by PSO presented in section 5.2.2. The issues and the corresponding solution techniques for multi-objective PSOs are reviewed in section 5.2.3.

5.2.1 Multi-objective Optimization and Pareto Optimality

The general multi-objective optimization problem is formulated as follows:

$$\begin{aligned} &\min_{\mathbf{x}} \ \mathbf{f}(\mathbf{x}) = \left[f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}) \right]^T \\ &\text{s.t.} \ \ \mathbf{x} \in \mathbf{X} \end{aligned} \tag{5.1}$$

where **x** is a *D*-dimensional decision variable, **X** is the feasible decision space and *M* is the number of objective functions.

MOO aims to find a set of optimal tradeoff solutions termed as Pareto optimal set since it is impossible to find a single solution which optimizes all the objective functions simultaneously. A solution $\mathbf{x}^* \in \mathbf{X}$ is Pareto optimal iff there does not exist another solution, $\mathbf{x} \in \mathbf{X}$, such that $\mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}^*)$, and $f_m(\mathbf{x}) < f_m(\mathbf{x}^*)$ for at least one function. The Pareto optimal set is the set of all Pareto optimal solutions, and the corresponding set of objective vectors is the Pareto optimal frontier.

5.2.2 Particle Swarm Optimization

In single objective PSO (Shi & Eberhart, 1998), the velocity and position for each particle *j* at iteration *i* are updated according to the following equations

$$\mathbf{v}_j^{i+1} = w\mathbf{v}_j^i + c_1 r_{1,j}^i \times \left(\mathbf{p}_j^i - \mathbf{x}_j^i\right) + c_2 r_{2,j}^i \times \left(\mathbf{p}_{g,j}^i - \mathbf{x}_j^i\right) \tag{5.2}$$

$$\mathbf{x}_j^{i+1} = \mathbf{x}_j^i + \mathbf{v}_j^{i+1} \tag{5.3}$$

where *j* denotes the *j*th particle in the swarm; *D*-dimensional vector $\mathbf{v}_j^i$ is the velocity of the *j*th particle ($\mathbf{v}_j^i \in [-\mathbf{v}_{\max}, +\mathbf{v}_{\max}]$), $\mathbf{v}_{\max}$ is used to constraint the velocity for each particle and is usually set between 0.1 and 1.0 times the search range of the solution space (Banks et al., 2007); *D*-dimensional vector $\mathbf{x}_j^i$ is the position of the *j*th particle; $\mathbf{p}_j^i$ (pBest) is the best position found so far by the *j*th particle; $\mathbf{p}_{g,j}^i$ (gBest) is the best position found so far by particle *j*'s neighbors; $r_{1,j}^i$ and $r_{2,j}^i$ represent two independent random numbers uniformly distributed on [0, 1]; c_1 is the cognitive learning factor which represents the attraction that a particle has toward its own success $\mathbf{p}_j^i$; c_2 is the social learning factor which represents the attraction that a particle has toward its neighbors' best position $\mathbf{p}_{g,j}^i$; *w* is the inertia weight. Cognitive learning factor c_1 impacts the local search ability while the global search ability is influenced by the social learning factor c_2.

Large inertia weight *w* enables particles to move in a high velocity and perform extensive exploration, while small inertia weight enhances the exploitation ability (Poli et al., 2007).

5.2.3 Multi-objective Particle Swarm Optimization

Although PSO has been demonstrated to be effective for single objective optimization, three main issues should be considered when extending PSO to multi-objective optimization (Reyes-Sierra & Coello Coello, 2006):

- Non-dominated solutions retaining and spreading: how to maintain all the non-dominated solutions found during the search process and guarantee the solutions are uniformly and smoothly spread on the Pareto frontier?
- Leader selection: how to select pBest and gBest when considering Pareto dominance?
- Diversity keeping: how to maintain the diversity in swarm and diversity in Pareto optimal set?

The solution techniques employed to address these three issues in the existing literatures are reviewed in the following sections.

5.2.3.1 Non-dominated Solutions Retaining and Spreading

In the population based multi-objective optimization technique, multiple Pareto optimal solutions will be obtained at each iteration. The most commonly used approach to maintain the non-dominated solutions found during the search process is to use an external archive (Goh et al., 2010; Huang et al., 2006; Janson & Merkle, 2005; Li, 2003; Liu et al., 2007; Mostaghim & Teich, 2003a, 2003b; Nebro et al., 2009a; Raquel & Naval Jr., 2005; Salazar-Lechuga & Rowe, 2005; Tripathi et al., 2007; Villalobos-Arias et al., 2005; Wang & Yang, 2009, 2010). The non-dominated solutions will be added in the archive, and the dominated solutions in the archive will be removed at the end of each iteration. Some multi-objective PSOs use a secondary external archive for each particle to store the particles' non-dominated solutions found so far (Fieldsend & Singh, 2002; Ho et al., 2005; Reyes-Sierra & Coello Coello, 2005).

Since the external archive will increase quickly and it will be computationally prohibitive to update the external archive at the end of each iteration, some techniques attempting to keep less crowded solutions by deleting crowded solutions are employed to prune the archive when its size exceeds the capacity. These techniques are not only able to bound the archive but also make the non-dominated solutions evenly spread on the Pareto frontier. The most commonly used techniques include: 1) adaptive grid (Bartz-Beielstein et al., 2003; Coello Coello & Lechuga, 2002; Coello Coello et al., 2004; Fieldsend & Singh, 2002): the objective space based on all the non-dominated solutions in the archive is evenly divided into several regions (hypercubes) and solutions in the hypercubes which have less number of solutions will be preferred. This method may be computationally expensive especially when the objective space should be frequently divided. 2) clustering technique (Mostaghim & Teich, 2003a): the non-dominated solutions in the archive are divided into several clusters and the archive is filled by selecting a representative individual per cluster. The size and quality of the archive depend on the number of clusters. 3) ε-dominance (Mostaghim & Teich, 2003b; Reyes-Sierra & Coello Coello, 2005): a set of boxes with size ε is defined on the objective space and only one non-dominated solution is retained for each box based on ε-dominance. The ε-dominance has been demonstrated to be computationally efficient than clustering techniques with a comparable convergence and diversity for pruning the archive (Mostaghim & Teich, 2003b). However, the size of the archive is impacted by the user defined parameter ε. 4) niche count (Goh et al., 2010; Li, 2003; Liu et al., 2007; Salazar-Lechuga & Rowe, 2005): a neighborhood (niche) with radius σ_{share} for each solution in the archive is defined, and the solution has less neighbors in its neighborhood will be preferred. The spread of the final archive depends on the user defined parameter σ_{share}. 5) crowding distance (Huang et al., 2006; Nebro et al., 2009a; Raquel & Naval Jr., 2005; Zhao & Suganthan, 2011): the non-dominated solutions are sorted according to each objective function, and the crowding distance for a solution is the accumulate value of the distance between the solution and its two neighbors on each objective space. The solutions with large crowding distance values will be kept in the archive.

In summary, the performances of the first four techniques depend on some additional parameters. Therefore, in this chapter, the crowding distance method is adopted due to its simplicity for implementation and independency on the additional parameters.

5.2.3.2 Leaders (gBest and pBest) Selection

Different from single objective optimization, there is a set of leaders could be selected by the particles as their pBest and gBest. In the multi-objective PSO, the gBest is usually selected from the non-dominated solution set (external archive) by several commonly used methods: 1) crowding distance (Li, 2003; Nebro et al., 2009a; Raquel & Naval Jr., 2005; Ray & Liew, 2002; Reyes-Sierra & Coello Coello, 2005; Tripathi et al., 2007; Zhao & Suganthan, 2011): the non-dominated solution with large crowding distance is preferred; 2) niche count (Daneshyari & Yen, 2011; Li, 2003; Liu et al., 2007; Salazar-Lechuga & Rowe, 2005): the non-dominated solution with small niche count will be selected; 3) sigma method (Mostaghim & Teich, 2003a, 2003b, 2004): the sigma method aims to let the particles to fly directly towards the Pareto frontier; 4) randomly selection (Alvarez-Benitez et al., 2005; Huang et al., 2006; Janson & Merkle, 2005; Pulido & Coello Coello, 2004; Yapicioglu et al., 2007; Zielinski & Laur, 2007). Comparing with the first three methods, the randomly selection does not need additional calculation, but its performance may be inferior to the first three methods. In order to reduce the computational cost on gBest selection, the gBest is selected either randomly or according to a randomly generated weighted objective function in this chapter. The approaches used for gBest selection are also applicable for pBest selection when there is a secondary archive to store the particle’s non-dominated solutions found so far. The pBest will be updated based on Pareto dominance relation between the previous pBest and the current particle when no secondary archive available.

5.2.3.3 Swarm and External Archive Diversity Keeping

Other than keeping diversity in the swarm to prevent premature convergence, the diversity of the external archive should be kept to cover the true Pareto frontier as larger range as possible. Most of existing literature adopts mutation or turbulence operator to prevent premature convergence (Alvarez-Benitez et al., 2005; Coello Coello & Lechuga, 2002; Coello Coello et al., 2004; Daneshyari & Yen, 2011; Fieldsend & Singh, 2002; Ho et al., 2005; Li, 2003; Liu et al., 2007; Liu et al., 2008; Mostaghim & Teich, 2003a, 2003b, 2004; Nebro et al., 2009a; Raquel & Naval Jr., 2005; Reyes-Sierra & Coello Coello, 2005; Srinivasan & Seow, 2003; Tripathi et al., 2007; Villalobos-Arias et al., 2005; Wang & Yang, 2009, 2010). The diversity of the swarm could be kept through the selection of leaders as discussed in section 5.2.3.2, topology definition, PSO parameter tuning (Reyes-Sierra & Coello Coello, 2006), and some techniques discussed in section 5.2.3.1.

5.3 AMOPSO Algorithm

It is observed that the basis (PSO algorithm) in existing multi-objective PSOs may not be able to simultaneously handle diverse functions with different properties and may suffer from the premature convergence. This may be a reason why multi-objective PSO is not guaranteed to be more effective than other MOEAs. In this chapter, an augmented PSO algorithm for multi-objective optimization is developed, termed AMOPSO. In the AMOPSO, two local searches are stuided: a non-uniform mutation based method (Michalewicz, 1996) which has good exploration capability may be preferred by multi-modal functions and a sub-gradient method (Boyd, 2010) may be preferred by uni-modal functions due to its good exploitation and quickly finding local optimum capability. Next, Cauchy mutation operator (Andrews, 2006) is incorporated to prevent premature convergence. Furthermore, an adaptive parameter tuning is developed to enhance the robustness of PSO and further improve its performance in terms of solution quality and/or convergence speed. Therefore, it is expected that extending this generalized and robust PSO to MOPSO which may outperform and/or be comparable with other MOPSOs/MOEAs.

The AMOPSO (shown in Figure 14) has five modules: (1) PSO module: The swarm is randomly initialized with the PSO operator being employed to update the swarm. (2) Bi-Local searches module (see section 5.3.1): two local search methods (non-uniform mutation based method and sub-gradient method) are implemented. At each iteration, an appropriate local search method will be triggered based on the dynamic selection criteria. The initial local search method is the non-uniform mutation based method (ls_indicator=0). (3) Archive module (see section 5.3.2): the external archive is updated and followed by updating pBest and gBest for each particle. (4) Mutation module (see section 5.3.3): the mutation operator is used to update one randomly selected particle. (5) Parameter tuning module (see section 5.3.4): the three parameters for one randomly selected particle will be changed by the adaptive parameter tuning mechanism. The algorithm will stop if the stopping criterion is satisfied.

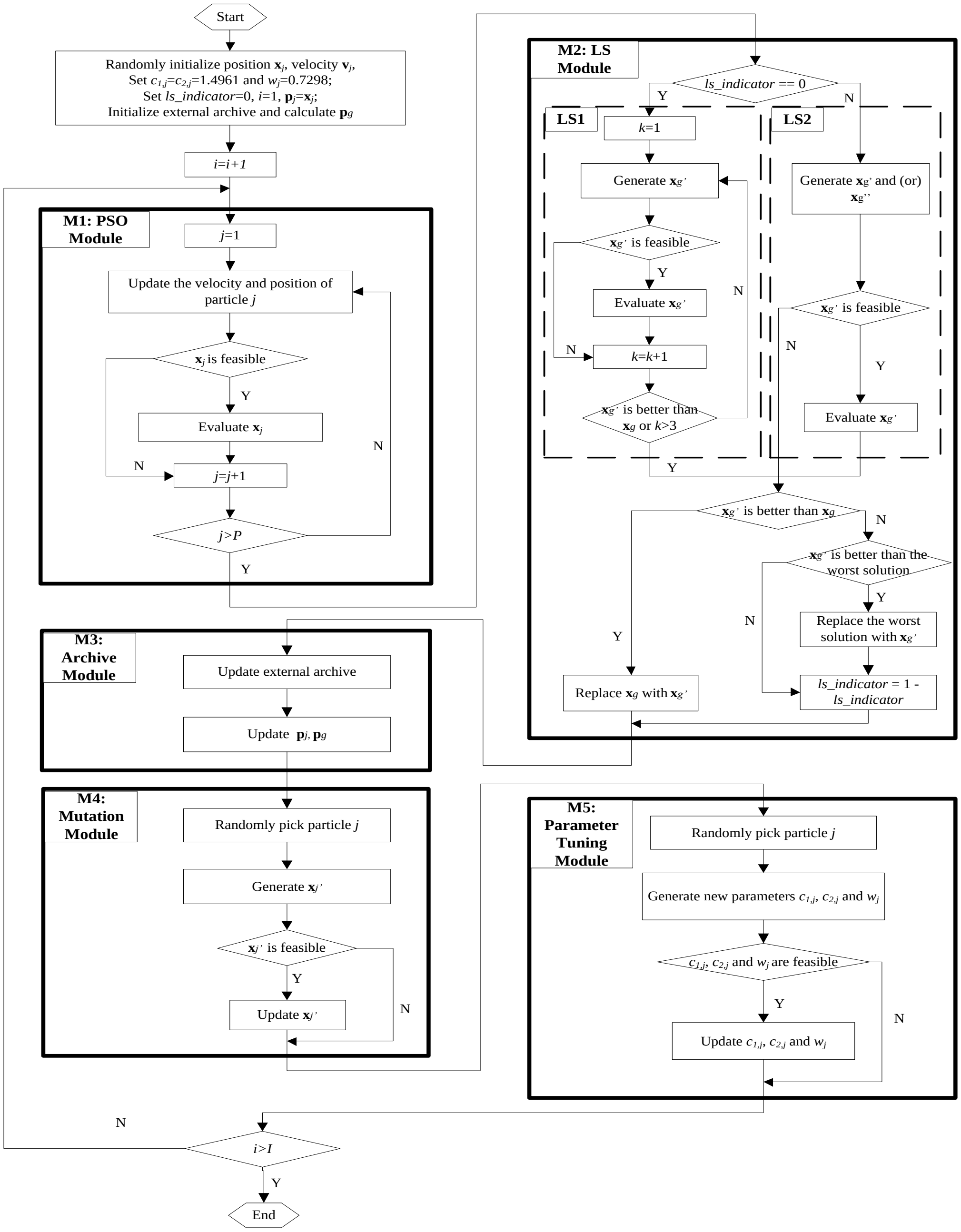

Start
Randomly initialize position xj, velocity vj,
Set c1,j=c2,j=1.4961 and wj=0.7298;
Set ls_indicator=0, i=1, pj=xj;
Initialize external archive and calculate pg
i=i+1
M1: PSO Module
j=1
Update the velocity and position of particle j
xj is feasible
Y
Evaluate xj
N
j=j+1
N
j>P
Y
M2: LS Module
ls_indicator == 0
Y
N
LS1
k=1
Generate xg'
xg' is feasible
Y
Evaluate xg'
N
k=k+1
N
xg' is better than xg or k>3
Y
LS2
Generate xg' and (or) xg''
xg' is feasible
N
Y
Evaluate xg'
xg' is better than xg
N
xg' is better than the worst solution
Y
Replace the worst solution with xg'
N
Y
Replace xg with xg'
ls_indicator = 1 - ls_indicator
M3: Archive Module
Update external archive
Update pj, pg
M4: Mutation Module
Randomly pick particle j
Generate xj'
xj' is feasible
Y
Update xj'
N
M5: Parameter Tuning Module
Randomly pick particle j
Generate new parameters c1,j, c2,j and wj
c1,j, c2,j and wj are feasible
Y
Update c1,j, c2,j and wj
N
N
i>I
Y
End

Figure 14 Flowchart of AMOPSO ("LS1": non-uniform mutation-based method; "LS2": sub-gradient method)

5.3.1 Local Searches

A weighted sum objective function f is randomly generated to select a solution to be improved by the bi-local searches method.

$$f = \sum_{m=1}^{M} w_m f_m \tag{5.4}$$

where

$$w_m = \begin{cases} r_m \Big/ \sum_{m=1}^{M} r_m & r' \leq 0.5 \\ 0 & r' > 0.5 \text{ and } \mathrm{randi}(M) \neq m \\ 1 & r' > 0.5 \text{ and } \mathrm{randi}(M) = m \end{cases} \tag{5.5}$$

r_m and *r'* are a uniform random number from (0, 1) and m=1,…,M, randi(M) generates a integer number between 1 to M. The weights generated by Eq. (5.5) could balance the search ability on both the boundary and middle region of the Pareto frontier, and thus promote diversity in the Pareto frontier. Two local search approaches are studied to improve the current best solution $\mathbf{x}_g^i$ which is defined as

$$\mathbf{x}_g^i = \mathrm{argmin}_{1\leq j\leq P} \left\{ f\left(\mathbf{x}_j^i\right)\right\} \tag{5.6}$$

The non-uniform mutation based method (Michalewicz, 1996) is good at searching the solution space uniformly (exploration) at the early stage and very locally (exploitation) at the later stage (Zhao, 2011). The non-uniform mutation based method has been demonstrated to have the merits of large jumping (exploration) and fine-tuning (exploitation) (Zhao, 2011; Zhao et al., 2007). In addition, the non-uniform mutation based method does not require the problem instance to have analytical functions. The second local search method is the sub-gradient method (Boyd, 2010) which is an iterative method for solving convex minimization problems and is also applicable for non-convex problems. Like gradient based methods, the sub-gradient method exhibits good exploitation capability around the neighborhood of the local or global optimum (Plevris & Papadrakakis, 2010). The sub-gradient method for unconstrained problems is equivalent to the gradient based method when the objective function is differentiable.

In the non-uniform mutation based method, the d^{th} dimension of the current best solution $\mathbf{x}_g^i$ is randomly picked to be mutated to generate a new solution as

$$x_{g,d}^{i}\,' = \begin{cases} x_{g,d}^{i} + \Delta\left(i, U_d - x_{g,d}^{i}\right) & \text{if } r \geq 0.5 \\ x_{g,d}^{i} - \Delta\left(i, x_{g,d}^{i} - L_d\right) & \text{if } r < 0.5 \end{cases} \tag{5.7}$$

where i is the current iteration number of PSO; U_d and L_d are the upper and lower bounds of $x_{g,d}^{i}$; r is a uniform random number from (0, 1). The function $\Delta(i,y)$ is defined as

$$\Delta\left(i, y\right) = y \times \left(1 - \rho^{(1-i/I)}\right) \tag{5.8}$$

where ρ is a uniform random number from (0, 1); I is the maximum number of iterations for PSO. In the sub-gradient method, a new solution $\mathbf{x}_g^i\,'$ is generated as

$$\mathbf{x}_g^i\,' = \mathbf{x}_g^i - \alpha_i \mathbf{g}_g^i \tag{5.9}$$

where $\mathbf{g}_g^i$ is the sub-gradient of the function f; $\alpha_i = 1/i^{0.6}$ is the step size used in this chapter. Solution $\mathbf{x}_g^i\,'$ generated by Eq. (5.9) may be ineffective (infeasible or local exploitation) at the beginning of PSO iterations. Therefore, one additional solution $\mathbf{x}_g^i\,''$ is generated by

$$\mathbf{x}_g^i\,'' = \mathbf{x}_g^i - \alpha_i \frac{\mathbf{g}_g^i}{\left\|\mathbf{g}_g^i\right\|_2} \frac{\left(\mathbf{U}_g - \mathbf{L}_g\right)}{4} \tag{5.10}$$

if $r \leq \exp\left(-i^{2.8}/I^2\right)$, where $\mathbf{U}_g$ and $\mathbf{L}_g$ are the upper and lower bounds of $\mathbf{x}_g^i$; r is a uniform random number from (0, 1); i is the current iteration number of PSO; I is the maximum number of iterations for PSO.

The exploration ability of the non-uniform mutation based method benefits AMOPSO on multi-modal functions, but may slow the convergence speed on uni-modal functions. The sub-gradient method is good at exploiting the search space, but tends to be trapped in the local optimum for some functions. Therefore, a dynamic selection mechanism is introduced to balance the exploration and exploitation capability of

AMOPSO. When the current local search method is not able to improve $\mathbf{x}_g^i$, the alternative local search method is triggered for the next iteration. The final solution from local search replaces the current best solution $\mathbf{x}_g^i$ if it is better than $\mathbf{x}_g^i$ in terms of Eq. (5.4), otherwise it replaces the current worst solution evaluated by Eq. (5.4) if it is better than the worst solution.

5.3.2 Archive and Leader Update

At iteration *i*, the external archive and particles' leaders will be updated after local search. Particle $\mathbf{x}_j^i$ is discarded if it is dominated by any solutions in the external archive. Otherwise it will be added into the external archive and all the solutions in the external archive which are dominated by $\mathbf{x}_j^i$ should be removed from the external archive. The first N_{max} (capacity of the external archive) non-dominated solutions in the external archive which have large crowding distance values will be kept in the archive when the size of the external archive exceeds N_{max}. The pBest ($\mathbf{p}_j^i$) for particle *j* is updated as

$$\mathbf{p}_j^{i+1} = \begin{cases} \mathbf{p}_j^i & \begin{array}{l} \left(\mathbf{x}_j^i \text{ is dominated by } \mathbf{p}_j^i\right) \text{ or} \\ \left(\mathbf{x}_j^i \text{ and } \mathbf{p}_j^i \text{ are non-dominated each other, and } r > 0.5\right) \end{array} \\ \mathbf{x}_j^i & \begin{array}{l} \left(\mathbf{x}_j^i \text{ dominates } \mathbf{p}_j^i\right) \text{ or} \\ \left(\mathbf{x}_j^i \text{ and } \mathbf{p}_j^i \text{ are non-dominated each other, and } r \le 0.5\right) \end{array} \end{cases} \tag{5.11}$$

The gBest ($\mathbf{p}_{g,j}^i$) for particle *j* is updated as

$$\mathbf{p}_{g,j}^{i+1} = \begin{cases} \text{random}\left(\mathbf{A}\right) & r \le 0.5 \\ \text{argmin}_{1 \le n \le N} \left\{ f'\left(\mathbf{A}_n\right) \right\} & r > 0.5 \end{cases} \tag{5.12}$$

where

$$f' = \sum_{m=1}^{M} \left\{ \left(r_m \Big/ \sum_{m=1}^{M} r_m \right) f_m \right\} \tag{5.13}$$

random(**A**) means randomly selecting a solution from archive **A**; *f'* is a randomly generated weighted sum objective function; *N* is the archive size; r_m is a uniform random number from (0, 1) and *m*=1,…,*M*. Please note all the *P* particles in the swarm use the same *f'* function to select gBest.

5.3.3 Cauchy Mutation

To keep the diversity of swarm, in hopes of accelerating the converging speed (non-premature), the Cauchy mutation operator is adopted which is demonstrated to be able to assist the particle by having a large jump out of its local optimum (Andrews, 2006). At iteration *i*, the *d*th dimension of a randomly selected particle *j* will be mutated as

$$x_{j,d}^{i}\,' = x_{j,d}^{i} + cauchy\left(0.1\right) \times \eta^{i} \times \left(U_d - L_d\right) \tag{5.14}$$

where U_d and L_d are the upper and lower bounds of $x_{j,d}^{i}$; and η^i is the mutation scale parameter which is defined as

$$\eta^{i} = \max\left(\exp\left(-i^{2.61}/I^{2}\right), 0.1\right) \tag{5.15}$$

5.3.4 Adaptive Tuning

As discussed in (Yao et al., 1999), the large jumps from Cauchy mutation may be detrimental when the current search position is close to the neighborhood of the global optimum. Therefore, the distance between one randomly selected particle $\mathbf{x}_j^i$ and its gBest $\mathbf{p}_{g,j}^i$ is minimized. Taking w_j^i, $c_{1,j}^i$ and $c_{2,j}^i$ as decision variables, a *convex optimization* problem is formulated as:

$$\begin{aligned}
&\min\ f_{dist}^{i-1} = \left\|\mathbf{x}_j^{i} - \mathbf{p}_{g,j}^{i}\right\|_2^2 \\
&= \left\|\mathbf{x}_j^{i-1} + w_j^{i-1}\mathbf{v}_j^{i-1} + c_{1,j}^{i-1}r_{1,j}^{i-1} \times \left(\mathbf{p}_j^{i-1} - \mathbf{x}_j^{i-1}\right) + c_{2,j}^{i-1}r_{2,j}^{i-1} \times \left(\mathbf{p}_{g,j}^{i-1} - \mathbf{x}_j^{i-1}\right) - \mathbf{p}_{g,j}^{i}\right\|_2^2 \\
&\text{s.t.}\quad 0.5 \le c_1 \le 2.5, \quad 0.5 \le c_2 \le 2.5, \quad 0.4 \le w \le 0.9
\end{aligned} \tag{5.16}$$

The sub-gradient method (Boyd, 2010) is employed to solve the convex optimization problem formulated in Eq. (5.16). w_j^i, $c_{1,j}^i$, and $c_{2,j}^i$ could be updated as described in the following equations

$$w_j^i = w_j^{i-1} - \alpha_j^{i-1} g_{w_j}^{i-1} \tag{5.17}$$

$$c_{1,j}^i = c_{1,j}^{i-1} - \alpha_j^{i-1} g_{c_{1,j}}^{i-1} \tag{5.18}$$

$$c_{2,j}^i = c_{2,j}^{i-1} - \alpha_j^{i-1} g_{c_{2,j}}^{i-1} \tag{5.19}$$

where α_j^{i-1} and $g_{w_j}^{i-1}$ are the step size and sub-gradient of the objective function in Eq. (5.16) at iteration *i* for particle *j*. Since the objective function in Eq. (5.16) is derivable, the derivative of f_{dist}^{i-1} evaluated at w_j^{i-1} is used as $g_{w_j}^{i-1}$. The optimal step size when the optimal value f_{dist}^* of the convex objective function is known is Polyak's step size (Boyd, 2010) which is computed as

$$\alpha_j^{i-1} = \left(f_{dist}^{i-1} - f_{dist}^* \right) \Big/ \left\{ \left(g_{w_j}^{i-1} \right)^2 + \left(g_{c_{1,j}}^{i-1} \right)^2 + \left(g_{c_{2,j}}^{i-1} \right)^2 \right\} \tag{5.20}$$

where the optimal value f_{dist}^* is always 0.

5.4 Experimental Analysis

Nineteen test problems are collected from the literature (Huband et al., 2006) including five ZDT (Zitzler-Deb-Thiele) problems, seven DTLZ (Deb-Thiele-Laumanns-Zitzler)-2D problems and seven DTLZ-3D problems. The formulations for these nineteen test problems are listed in Appendix B. Please note both ZDT and DTLZ problems are widely used as benchmark multi-objective problems in the EA literature (Huband et al., 2006).

In the experiments, the problem name, dimension (*D*), search range, property of Pareto frontier, and objective property are listed in Table 27~29. SGFE is determined as the total number of floating point operations (FLOP) for the *D*-dimensional sub-gradient calculations divided by the floating point operations (FLOP) of all the objective function evaluation.

$$SGFE = ceil\left(\sum\nolimits_{m=1}^{M}\sum\nolimits_{d=1}^{D} FLOP_{g_{m,d}} \Big/ \sum\nolimits_{m=1}^{M} FLOP_{f_m}\right) \tag{5.21}$$

where ceil(.) rounds the element to the nearest integer towards infinity; FLOP is the output of “flops” function in MATLAB®.

5.4.1 Parameter Settings for the Comparison Study

The detailed algorithms with parameter settings are:

1) NSGA-II (Deb et al., 2002): population size is 100; crossover probability is 0.9; mutation probability is 1/*D*; crossover operator is simulated binary crossover (SBX); mutation operator is polynomial mutation; selection operator is binary tournament selection; distribution indexes for crossover and mutation operators are $\eta_c = 20$ and $\eta_m = 20$.

2) PAES (Knowles & Corne, 1999): population size is 1; mutation probability is 1/*D*; mutation operator is polynomial mutation; the number of subdivisions of the space in the grid is 5.

3) SPEA2 (Zitzler et al., 2001): population size is 30; crossover probability and operator, mutation probability and operator, selection operator, and distribution indexes are the same as NSGA-II (Deb et al., 2002).

4) MOCell (Nebro et al., 2009b): population size is 30; feedback number is 20; crossover probability and operator, mutation probability and operator, selection operator, and distribution indexes are the same as NSGA-II (Deb et al., 2002).

5) MOPSO (Coello Coello et al., 2004): population size is 30; 10 divisions for the adaptive grid; c_1=c_2=1.4962; *w*=0.7298.

6) OMOPSO (Reyes-Sierra & Coello Coello, 2005): population size is 30; c_1, c_2=random(1.5, 2.0); w=random(0.1, 0.5).

7) MOCLPSO (Huang et al., 2006): population size is 30; learning probability P_c is 0.1; elitism probability P_m is 0.4; c_1=c_2=2.05; w=0.729.

8) 2LB-MOPSO (Zhao & Suganthan, 2011): population size is 30; # of bins is 10; # of count is 5; c_1=c_2=2.05; w=0.729.

9) AMOPSO: population size is 30; adaptively change c_1, c_2, and w.

Additional parameter settings for the eight MOO algorithms are the same as (Coello Coello et al., 2004; Deb et al., 2002; Huang et al., 2006; Knowles & Corne, 1999; Nebro et al., 2009b; Reyes-Sierra & Coello Coello, 2005; Zhao & Suganthan, 2011; Zitzler et al., 2001).

5.4.2 Performance Metrics

In the experiments, the maximum number of function evaluations is set to be 30,000. For all test problems, the algorithms carry out 30 independent runs. Three metrics are adopted from (Goh & Tan, 2007) to evaluate the performance. All these three metrics are calculated based on normalized objective value and a set of reference points which is available on (Durillo & Nebro, 2011) is used as true Pareto frontier (PF*).

1) Proximity Indicator: generational distance (GD) is a metric to measure the gap between the true Pareto frontier (PF*) and the approximated Pareto frontier (PF). GD is computed as

$$GD = \left(\frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d_i^2 \right)^{1/2} \tag{5.22}$$

where n_{PF} is the number of solutions in PF; d_i is the Euclidean distance (in objective space) between the i^{th} member of PF and its nearest member of PF*. A low value of GD is preferred, which reflects a small gap between PF and PF*.

2) Diversity Indicator: a modified maximum spread (MS) is a metric to measure how well the PF* is covered by PF. MS is computed as

$$MS = \left\{ \frac{1}{M} \sum_{m=1}^{M} \left[\frac{\max\left(\min\left(f_m^{\max}, F_m^{\max} \right) - \max\left(f_m^{\min}, F_m^{\min} \right), 0 \right)}{F_m^{\max} - F_m^{\min}} \right]^2 \right\}^{1/2} \tag{5.23}$$

Where $F_m^{\max}$ and $F_m^{\min}$ is the maximum and minimum value of the mth objective in PF*; $f_m^{\max}$ and $f_m^{\min}$ is the maximum and minimum value of the mth objective in PF. A large value of MS is preferred, which reflects that a large area of PF* is covered by PF.

3) Distribution Indicator: spacing (S) is a metric to measure how evenly the non-dominated solutions are distributed along the PF. S is computed as

$$S = \frac{1}{n_{PF}} \left[\frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} \left(d_i - \bar{d} \right)^2 \right]^{1/2}, \text{ where } \bar{d} = \frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d_i \tag{5.24}$$

where n_{PF} is the number of solutions in PF; d_i is the Euclidean distance (in objective space) between the ith member of PF and its nearest member of PF. A small value of spacing is preferred, which reflects that the members in PF are uniformly distributed.

The value is the smaller the better for generational distance (GD) and spacing (S); the value is the larger the better for maximum spread (MS).

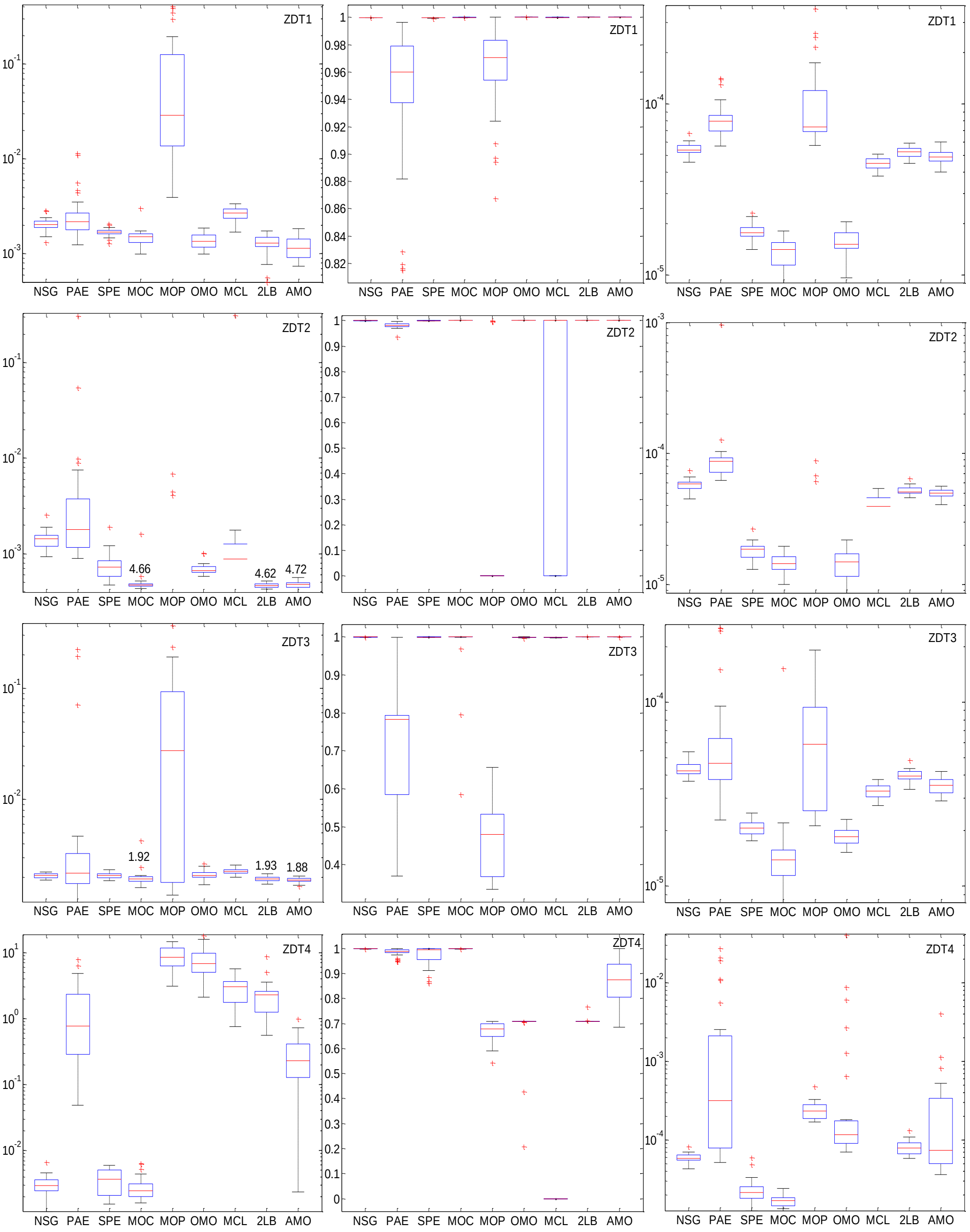
5.4.3 Comparison Experiments

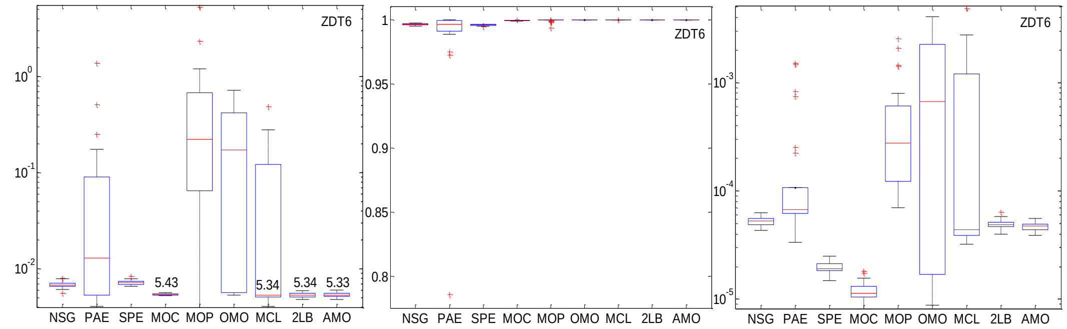
5.4.3.1 ZDT Problems

In the first set of experiments, five two-objective ZDT problems are studied (Table 27). The algorithm performances on the three metrics for ZDT problems are shown in Figure 15. The horizontal axis is the name of each algorithm (the first three characters are used to represent the algorithm), and the vertical axis is the value of each metric. Please note “U” and “M” in the last column of Table 27 means uni-modal and multi-modal respectively.

Table 27   ZDT problems

Name	Dimension D	Search range	Frontier property	Objective property
ZDT1	30	$[0,1]^D$	Convex	f_1: U; f_2: U
ZDT2	30	$[0,1]^D$	Concave	f_1: U; f_2: U
ZDT3	30	$[0,1]^D$	Disconnected	f_1: U; f_2: M
ZDT4	10	[0,1] for x_1, [-5,5] for $x_{2,\ldots,D}$	Convex	f_1: U; f_2: M
ZDT6	10	$[0,1]^D$	Concave	f_1: M; f_2: M





(a) (b) (c)

Figure 15 Algorithm performance in (a) GD, (b) MS, (c) S for ZDT problems

In terms of generational distance (GD) which measures solution accuracy, the five multi-objective PSOs except MOPSO are comparable with the four MOEAs on ZDT1, ZDT2, ZDT3 and ZDT6, and the five multi-objective PSOs underperform the four MOEAs on ZDT4. MOPSO is the worst algorithm among these nine algorithms. AMOPSO outperforms the four MOEAs on ZDT1, ZDT3, and ZDT6, is comparable with MOCell on ZDT2, and is inferior to three MOEAs on ZDT4 problem. AMOPSO outperforms the other four multi-objective PSOs on ZDT1, ZDT3, ZDT4 and ZDT6, and is comparable with 2LB-MOPSO on ZDT2. The higher solution accuracy of AMOPSO is benefited by the more generalized and robust PSO basis.

On the metric of maximum spread (MS) which measures diversity in the Pareto frontier, the five multi-objective PSOs except MOPSO are comparable with the four MOEAs on ZDT1, ZDT2, ZDT3 and ZDT6, and the five multi-objective PSOs underperform the four MOEAs on ZDT4. MOPSO is the worst algorithm among these nine algorithms. AMOPSO is comparable with the four MOEAs on four ZDT problems except ZDT4. AMOPSO is the best algorithm among the five MOPSOs on all the five ZDT problems. As discussed in section 5.3.1, the weights generated by Eq. (5.5) and effectiveness of the local search methods may balance the search on both the boundary and middle region of the Pareto frontier, and thus promote diversity in the Pareto frontier.

For spacing (S) which measures the distribution of non-dominated solutions on the Pareto frontier, the five multi-objective PSOs are comparable with NSGA-II and PAES, and underperform SPEA2 and MOCell. Both MOCell and SPEA2 outperform AMOPSO on all the five problems. This may be the consequence of the replacement and feedback procedure between population and archive in MOCell, and a more complicated archive truncation and refill strategy implemented in SPEA2. The simple leader selection approach and non-dominated retaining and spreading technique implemented in AMOPSO could not guarantee a good performance on the metric of spacing. AMOPSO underperforms OMOPSO on the ZDT1, ZDT2 and ZDT3.

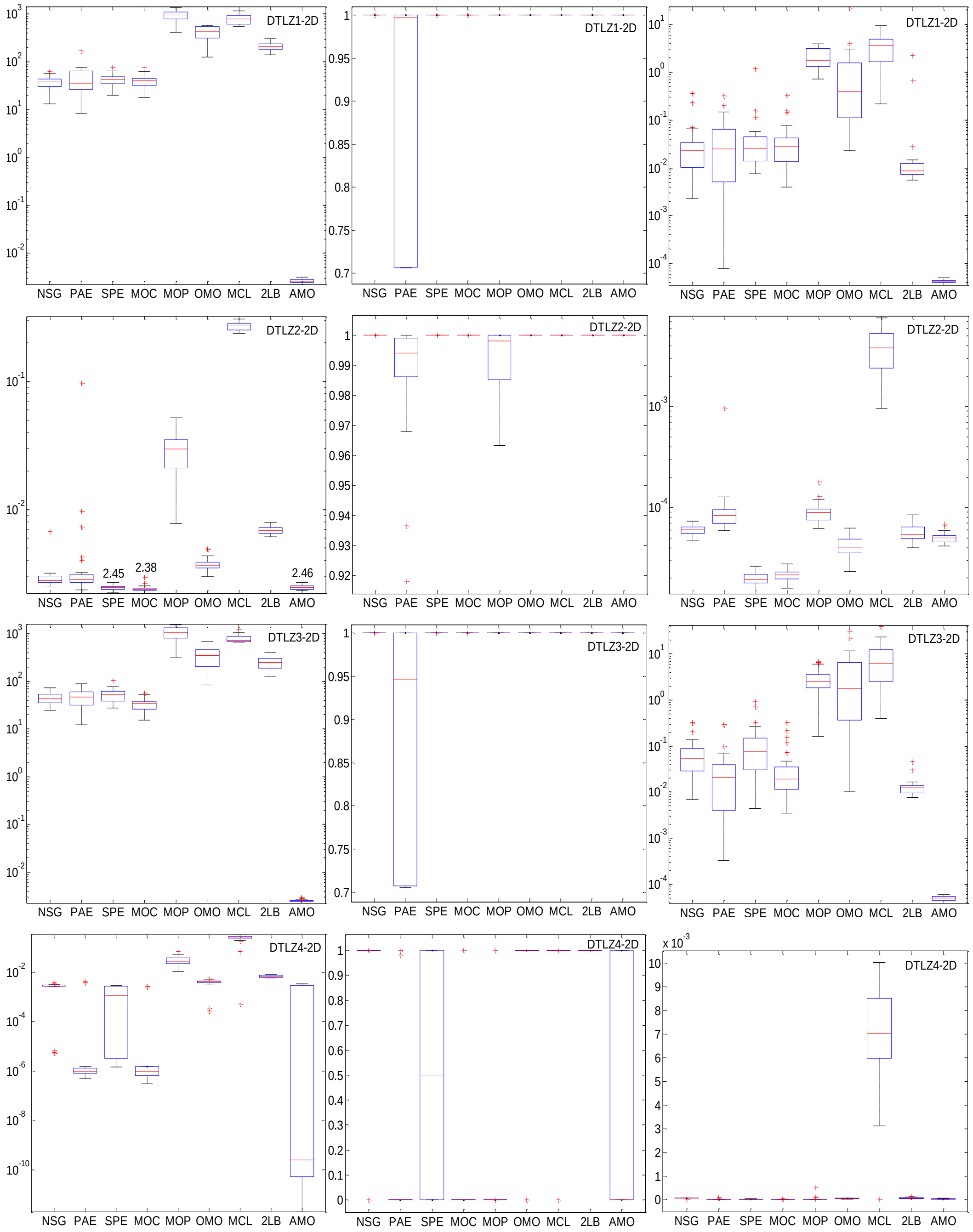
ZDT4 problem has a severe multi-modal landscape and large number of local fronts, and is proved to be very difficult for solving. The five multi-objective PSOs are inferior to the four MOEAs on this problem, which may be due to PSO's fast convergence speed.

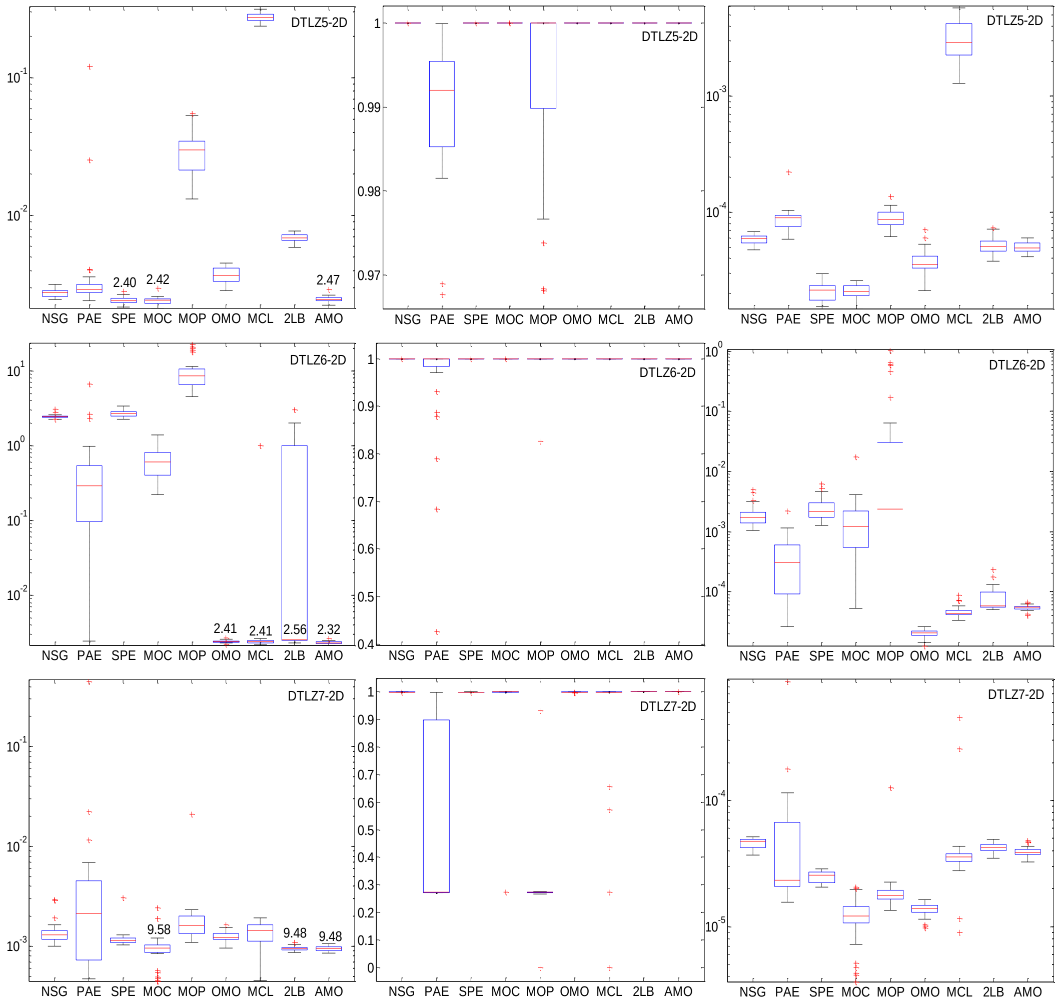
5.4.3.2 DTLZ-2D Problems

In the second set of experiments, seven two-objective DTLZ-2D problems are studied (Table 28). The algorithm performances on the three metrics for DTLZ-2D problems are shown in Figure 16. The horizontal axis is the name of each algorithm (the first three characters are used to represent the algorithm), and the vertical axis is the value of each metric. Again, the value is the smaller the better for generational distance (GD) and spacing (S); the value is the larger the better for maximum spread (MS).

Table 28 DTLZ-2D problems

Name	**Dimension *D***	**Search range**	**Frontier property**	**Objective property**
DTLZ1-2D	30	$[0,1]^D$	Linear	f_1: M; f_2: M
DTLZ2-2D	30	$[0,1]^D$	Concave	f_1: U; f_2: U
DTLZ3-2D	30	$[0,1]^D$	Concave	f_1: M; f_2: M
DTLZ4-2D	30	$[0,1]^D$	Concave	f_1: U; f_2: U
DTLZ5-2D	30	$[0,1]^D$	Concave	f_1: U; f_2: U
DTLZ6-2D	30	$[0,1]^D$	Concave	f_1: U; f_2: U
DTLZ7-2D	30	$[0,1]^D$	Disconnected	f_1: U; f_2: M


DTLZ1-2D
10^3
10^2
10^1
10^0
10^-1
10^-2
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ1-2D
1
0.95
0.9
0.85
0.8
0.75
0.7
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ1-2D
10^1
10^0
10^-1
10^-2
10^-3
10^-4
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ2-2D
10^-1
10^-2
2.45
2.38
2.46
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ2-2D
1
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0.92
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ2-2D
10^-3
10^-4
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ3-2D
10^3
10^2
10^1
10^0
10^-1
10^-2
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ3-2D
1
0.95
0.9
0.85
0.8
0.75
0.7
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ3-2D
10^1
10^0
10^-1
10^-2
10^-3
10^-4
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ4-2D
10^-2
10^-4
10^-6
10^-8
10^-10
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
DTLZ4-2D
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
NSG PAE SPE MOC MOP OMO MCL 2LB AMO
x 10^-3
DTLZ4-2D
10
9
8
7
6
5
4
3
2
1
0
NSG PAE SPE MOC MOP OMO MCL 2LB AMO



(a) (b) (c)

Figure 16 Algorithm performance in (a) GD, (b) MS, (c) S for DTLZ-2D problems

In terms of generational distance (GD) which measures solution accuracy, the four multi-objective PSOs except AMOPSO underperforms the four MOEAs on six DTLZ problems except DTLZ6. AMOPSO especially outperforms others on DTLZ1 and DTLZ3 which have two multi-modal objective functions. AMOPSO outperforms the four MOEAs on five DTLZ problems, and is comparable with SPEA2 and MOCell on DTLZ2 and DTLZ5. AMOPSO is the best algorithm among the five multi-objective PSOs. MOPSO and MOCLPSO are inferior to the other three multi-objective PSOs. This may be due to the fact that the standard PSO algorithm in MOPSO is not able to effectively handle multi-dimensional and multi-modal functions, and CLPSO in MOCLPSO is developed for multi-modal functions.

On the metric of maximum spread (MS) which measures diversity in the Pareto frontier, seven algorithms except PAES and MOPSO are comparable for six problems except DTLZ4. NSGA-II, OMOPSO, MOCLPSO and 2LB-MOPSO cover 100% of the true Pareto frontier for DTLZ4. AMOPSO achieves 100% coverage for six problems except DTLZ4.

For spacing (S) which measures the distribution of non-dominated solutions on the Pareto frontier, the five multi-objective PSOs underperform and/or are comparable with the four MOEAs on six DTLZ problems except DTLZ6. AMOPSO outperforms the four MOEAs on DTLZ1, DTLZ3 and DTLZ6. AMOPSO underperforms MOCell and SPEA2, and is comparable with NSGA-II and PAES on the other four problems. AMOPSO is the best algorithm among the five multi-objective PSOs on DTLZ1, DTLZ3 and DTLZ4, and is inferior to OMOPSO on the rest four problems. The ε-dominance based archive maintaining and crowding distance based leader selection strengthens OMOPSO in terms of spacing metric.

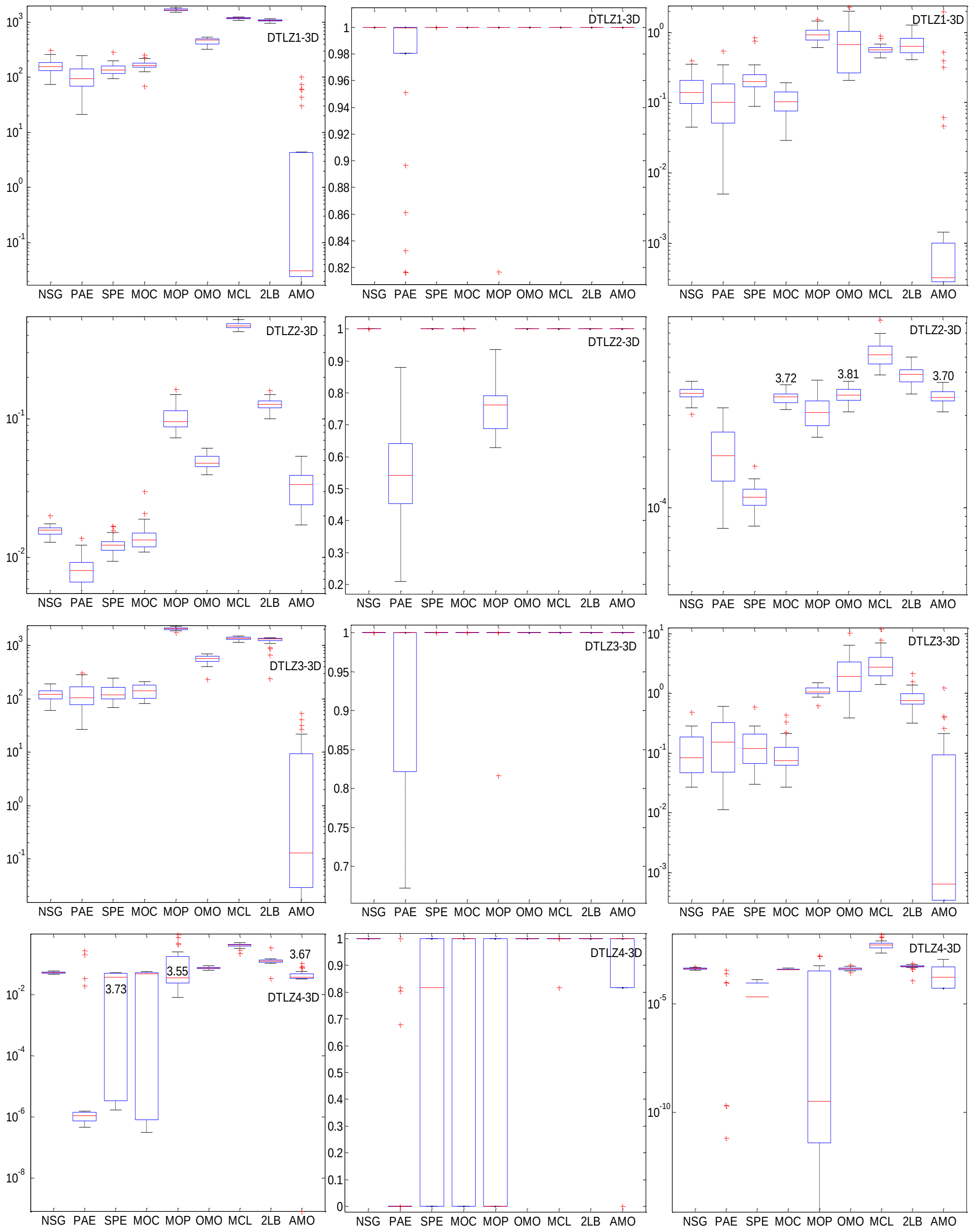
5.4.3.3 DTLZ-3D Problems

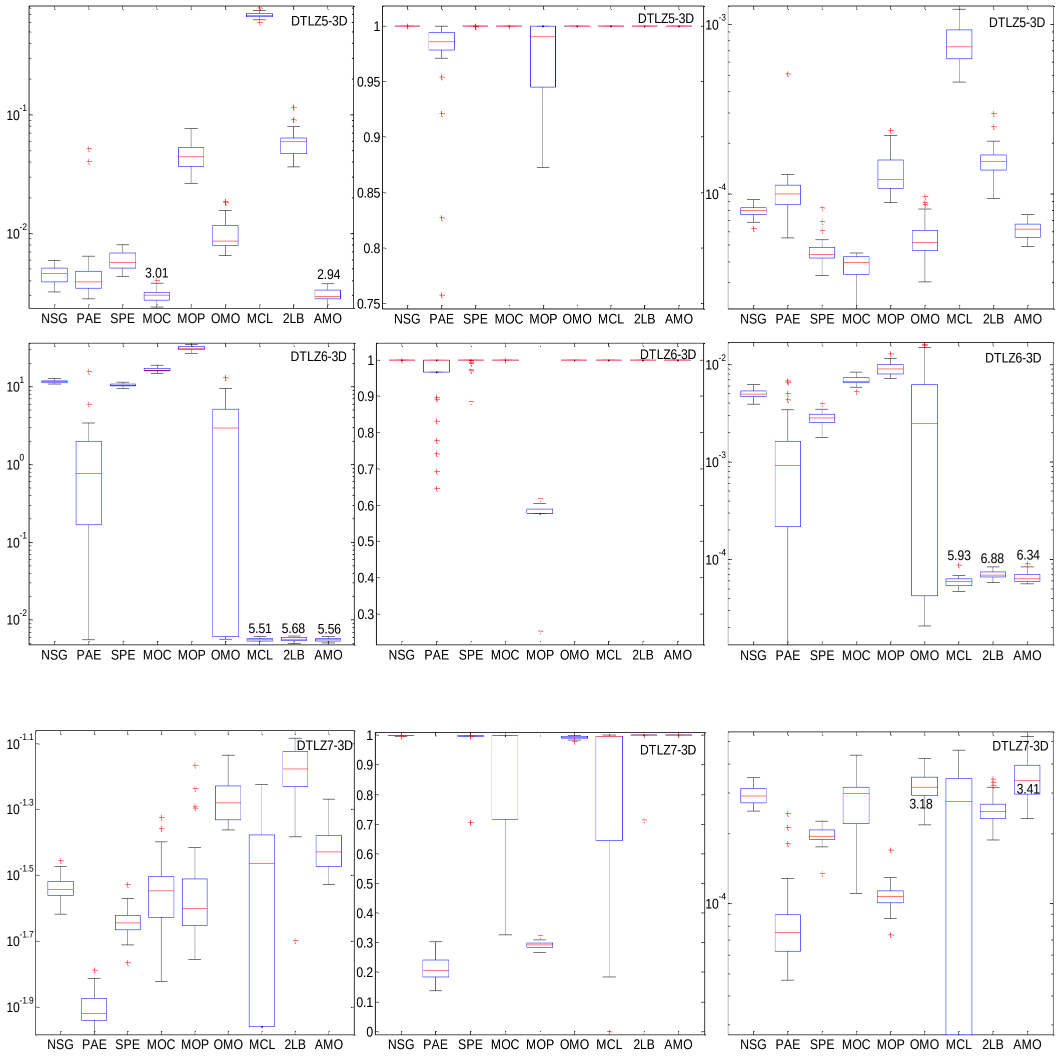
In the third set of experiments, seven three-objective DTLZ-3D problems are studied (Table 29). The algorithm performances on the three metrics for DTLZ-3D problems are shown in Figure 17. Again, the value is the smaller the better for generational distance (GD) and spacing (S); the value is the larger the better for maximum spread (MS).

Table 29 DTLZ-3D problems

Name	**Dimension *D***	**Search range**	**Frontier property**	**Objective property**
DTLZ1-3D	30	$[0,1]^D$	Linear	f_1: M; f_2: M; f_3: M
DTLZ2-3D	30	$[0,1]^D$	Concave	f_1: U; f_2: U; f_3: U
DTLZ3-3D	30	$[0,1]^D$	Concave	f_1: M; f_2: M; f_3: M

DTLZ4-3D	30	$[0,1]^D$	Concave	f_1: U; f_2: U; f_3: U
DTLZ5-3D	30	$[0,1]^D$	Concave	f_1: U; f_2: U; f_3: U
DTLZ6-3D	30	$[0,1]^D$	Concave	f_1: U; f_2: U; f_3: U
DTLZ7-3D	30	$[0,1]^D$	Disconnected	f_1: U; f_2: U; f_3: M





(a) (b) (c)

Figure 17 Algorithm performance in (a) GD, (b) MS, (c) S for DTLZ-3D problems

In terms of generational distance (GD) which measures solution accuracy, the four multi-objective PSOs except AMOPSO generally underperform the four MOEAs on six DTLZ problems except DTLZ6. AMOPSO outperforms the four MOEAs on four DTLZ problems, underperforms the four MOEAs on DTLZ2 and DTLZ7, and is comparable with three MOEAs except PAES on DTLZ4. It is observed that AMOPSO is the most accurate algorithm among the five MOPSOs for all the seven problems except DTLZ4 (underperforms MOPSO) and DTLZ7 (underperforms MOPSO, MOCLPSO).

On the metric of maximum spread (MS) which measures diversity in the Pareto frontier, AMOPSO performs better than and/or the same as other algorithms on six problems. AMOPSO is inferior to NSGA-II, OMOPSO, MOCLPSO, 2LB-MOPSO on DTLZ4.

For spacing (S) which measures the distribution of non-dominated solutions on the Pareto frontier, the four multi-objective PSOs except AMOPSO generally underperforms the four MOEAs on six DTLZ problems except DTLZ6. AMOPSO outperforms the four MOEAs on DTLZ1, DTLZ3 and DTLZ6, underperforms SPEA2 and MOCell on the rest four problems. AMOPSO outperforms and/or is comparable with the other four multi-objective PSOs on the seven problems except underperforms MOPSO on DTLZ4 and DTLZ7.

5.5 Conclusions

In this chapter, an augmented PSO algorithm for multi-objective optimization termed as AMOPSO is developed. The bi-local searches method is employed to handle diverse functions with different properties. The diversity of swarm is maintained by a Cauchy mutation operator. By incorporating an adaptive parameter tuning, the AMOPSO is able to improve the robustness and performance of PSO. The experiments conducted in this chapter demonstrate that AMOPSO significantly outperforms the existing 4 multi-objective PSOs and 3 representative MOEAs, and outperforms MOCell.

In this chapter, it is observed that AMOPSO performs poor in terms of spacing, which may be due to its simple leader selection and non-dominated retaining and spreading approaches. A more complicated leader selection and archive maintenance strategy should be investigated to enhance AMOPSO. Some complex mathematical multi-objective optimization problems such as shifted, rotated, noisy problems, and complex engineering problems such as product design, building energy system operation decisions (Hu et al., 2012), transportation problems should be used to test the effectiveness of AMOPSO.

Chapter 6

DECENTRALIZED OPERATION DECISIONS FOR SMART BUILDING CLUSTER USING A PARTICLE SWARM OPTIMIZATION

It is becoming urgent critical to develop a decentralized decision framework modeling the coordination among a cluster of buildings to obtain Pareto decisions as the emerging technology in smart building and smart grid. The smart grid enables bi-directional communication between the power gird and smart buildings, and the building could use other buildings which are connected with this building by the smart grid as a local energy buffer. Therefore, in chapter 2, a Memetic algorithm (MA) based decision framework is developed which is demonstrated to be capable of deriving the Pareto solutions for the building cluster in a decentralized manner, and thus reduce energy cost for the building cluster. This chapter attempts to improve the computational performance of the decision framework by replacing the MA with a multi-objective particle swarm optimization (PSO) presented in chapter 5. The experimental result demonstrates that the AMOPSO based decision framework is able to improve the computational performance of the decision framework. The multi-objective PSO based framework is capable for hourly operation decisions which are able to improve energy efficiency and thus achieve more energy cost savings for the smart buildings.

6.1 Introduction

In the United States, buildings use approximately 70% of total electricity usage and emit approximately 40% of greenhouse gases (GHG) annually (Kleissl & Agarwal, 2010). The building industry attempts to design an intelligent building termed as “smart building” (Hoffmann, 2009) which is able to meet the environment sustainability goal, keep occupants safe and comfortable, reduce the energy consumption and cost (Chen et al., 2009; Hoffmann, 2009). Although energy efficient building materials and appliances in the smart buildings are capable for energy demand reduction, it is still not sufficient to satisfy requirements of smart buildings due to ineffective operation strategies for those efficient appliances (Chen et al., 2009; Hoffmann, 2009). Therefore, intelligent and effective operation strategies which could achieve greatest energy efficiency are urgently needed for smart buildings.

The initial study of building operation and control research focuses on utilizing building thermal mass to achieve cost savings. Pre-cooling building through optimally controlling building temperature set-points can significantly reduce energy cost (Braun, 2003; Braun et al., 2001; Chen, 2001; Henze et al., 2010; Keeney & Braun, 1996). For example, the optimal strategy for building thermal mass determined by a dynamic programming and on-line simulation based technique is able to significantly reduce energy consumption and operating cost (Chen, 2001). A comprehensive review on building thermal mass operation strategy research is provided in (Braun, 2003). A near-optimal building thermal mass control is derived based on full factorial analyses of the important parameters impacting the building thermal mass control (Henze et al., 2010).

As the development in thermal storage techniques, extensive researches investigate utilizing both the building thermal mass and thermal storage to reduce energy cost. Most of the operation strategies which are derived by mathematical programming, simulation and reinforcement learning approaches are demonstrated to outperform the conventional control strategy such as chiller-priority and storage-priority strategies (Braun, 2007; Drees & Braun, 1996; Henze et al., 2005; Henze & Schoenmann, 2003; Lee et al., 2009; Liu & Henze, 2006a, 2006b; Liu & Henze, 2007; Sun et al., 2006). For example, the heuristic near-optimal control strategy developed in (Braun, 2007; Sun et al., 2006) is based on the optimal operation strategy obtained by dynamic programming. A closed-loop optimization technique is employed to derive optimal control strategy in

(Henze et al., 2005). Some meta-heuristic algorithms (e.g., particle swarm optimization (Lee et al., 2009)) are used to obtain optimal operation strategy. The rule based near-optimal control strategy is determined from monthly simulation of cooling system (Drees & Braun, 1996). The model-free reinforcement learning control strategy is studied in (Henze & Schoenmann, 2003; Liu & Henze, 2007) and the hybrid reinforcement learning control approaches combining model-based with model-free method are presented in (Liu & Henze, 2006a, 2006b).

Recent research is interested in using optimization techniques to study energy generation system design, planning and control, and expects that integrating the energy generation system to building could significantly reduce energy consumption and cost. For example, the chance constraint programming is employed to optimize the battery-integrated diesel generator system design (Arun et al., 2009). A mixed integer linear programming problem is solved to derive short-term scheduling for hydroelectric generation units (García-González et al., 2007). A hybrid energy system which consists of a battery, wind generator and photovoltaic module is designed and controlled by a simulation-optimization program (Manolakos et al., 2001). The long-term planning strategy for single-period combined heat and power system is derived by a branch and bound algorithm in (Rong & Lahdelma, 2007), and a modified dynamic programming approach is applied on multi-period combined heat and power system planning (Rong et al., 2008a). The short-term production plans for hydropower system are developed using a multi-stage mixed-integer linear stochastic programming (Fleten & Kristoffersen, 2008). Several heuristic algorithms are studied in (Kjeldsen & Chiarandini, 2012) to derive long-term strategic planning for cogeneration plants.

While promising, it is observed that in the smart buildings, there are several sub-systems (e.g., heating, ventilating and air conditioning (HVAC), energy storage system, energy generation system) integrated together. Secondly, the smart grid (Parks, 2009) enables bi-directional communication between the power gird and smart buildings, and the building could use other buildings which are connected with this building by the smart grids as a local energy buffer (SIEMENS, 2009). Therefore, there is an urgent need to develop a decentralized decision framework modeling the coordination among a cluster of buildings to obtain Pareto decisions which enable tradeoff analysis. In chapter 2, a decision model based on a building cluster simulator with each building modeled by energy consumption, storage and generation sub modules is developed. Assuming each building is interested in minimizing its energy cost, a bi-level operation decision framework based on a Memetic algorithm (MA) is developed to study the tradeoff in energy usage among the group of buildings. The MA based framework is capable of deriving the Pareto solutions for the building cluster in a decentralized manner. However, it is not able to study the hourly (or even less) basis operation decisions due to its computational issue. In this chapter, a multi-objective particle swarm optimization (PSO) (see chapter 5) based decentralized decision framework is developed to improve computational performance of the decision framework, and study hourly operation decisions for the integrated smart building cluster.

This chapter is organized as follows: section 6.2 formally defines the decision problem; the description of PSO and the AMOPSO based decision framework is presented in section 6.3; followed by the detail implementation of the decision framework in section 6.4. Section 6.5 reports the experimental results, and conclusions are drawn in section 6.6.

6.2 Problem Definition

The section briefly reviews the integrated building energy system simulator and decision model. The simulator is introduced in section 6.2.1, followed by the decision model presented in section 6.2.2.

6.2.1 Integrated Building Energy System Simulator

A simplified building cluster consisting of two different mass level - heavy mass (HM) and light mass (LM) buildings is modeled. The two buildings, each having its own battery and photovoltaic (PV) panel, share one ice storage system and one base chiller. The ice storage system charged by a dedicated chiller is configured in parallel with the base chiller. During on-peak hours, the buildings cooling loads are met primarily by the ice storage system with the remaining cooling request satisfied by the base chiller. The overall schematic of the building energy system configuration is illustrated in Figure 3 with the arrows denoting the energy flow among each component in the system. The simulator is a black-box which is used to evaluate the operation decisions. The decision model is explained in the following section.

6.2.2 Building Energy System Decision Model

In the decision model, the set-point temperature is controlled by each building. The shared ice storage will decide when to be charged or discharged to cool the buildings, and how to distribute its discharged cooling energy to each building. The decisions will be made for the battery on when to be charged or discharged to provide electricity for its served building. The decisions for the photovoltaic collector are charging battery, powering building, selling power to grid. The objective for each building is to minimize its daily energy cost subject to several constraints: 1) power grid capacity constraint (see Eq. (6.2)); 2) building comfort level constraint (see Eq. (6.3)); 3) base chiller capacity constraint (see Eq. (6.4)); 4) ice storage state constraint (see Eqs.

(6.5)-(6.7)); 5) battery state constraint (see Eqs. (6.8)-(6.9)); 6) PV panel state constraint (see Eq. (6.10)).

Let *M* be the number of buildings, *K* be the number of building operation modes (Liu & Henze, 2007), H_k be the number of hours in the k^{th} building operation mode, then the decisions for shared energy provider or building *m* (*m*=1,…,*M*) at building operation mode *k* (*k*=1,…,*K*) are expressed by: 1) a set of continuous variables $T_{sp,k}^{m}$ for set-point temperature; 2) a set of integer variables $S_{is,k}$ for ice storage state (0: dormant; 1: charging; 2: discharging); 3) a set of continuous variables η_k^m for percentage of energy from ice storage to building *m*; 4) a set of integer variables $S_{bat,k}^{m}$ for battery state (0: dormant; 1: charging; 2: discharging); 5) a set of integer variables $S_{PV,k}^{m}$ for PV panel state (0: dormant; 1: charging battery; 2: powering building; 3: selling power to grid). The objective function for each building (see Eq. (6.1)) and all the constraints (see Eqs. (6.2)-(6.10)) are written as

$$\min\ f_m = \sum\nolimits_{k=1}^{K} \sum\nolimits_{j=1}^{H_k} \left(R_{p,j}^{m} P_{p,j}^{m} - R_{s,j}^{m} P_{s,j}^{m} \right),\ \ m = 1,\ldots,M \tag{6.1}$$

$$\sum\nolimits_{m=1}^{M} P_{p,j}^{m} \le P_{grid} \tag{6.2}$$

$$T_i^{mL} \le \overline{T}_{i,j}^{m} \le T_i^{mU} \tag{6.3}$$

$$\sum\nolimits_{m=1}^{M} Q_{b,j}^{m} \le Q_{\max,j} \tag{6.4}$$

$$BI_{is,k}\left(1\right) \le ceil\left(\max\left(0, SOC_{is,\max} - SOC_{is,k}\right)\right) \tag{6.5}$$

$$BI_{is,k}\left(2\right) \le ceil\left(\max\left(0, SOC_{is,k} - SOC_{is,\min}\right)\right) \tag{6.6}$$

$$\sum\nolimits_{m=1}^{M} \eta_k^m \le BI_{is,k}(2) \tag{6.7}$$

$$BI_{bat,k}^{m}\left(1\right) \le ceil\left(\max\left(0, SOC_{bat,\max}^{m} - SOC_{bat,k}^{m}\right)\right) \tag{6.8}$$

$$BI_{bat,k}^{m}\left(2\right)\leq ceil\left(\max\left(0,SOC_{bat,k}^{m}-SOC_{bat,\min}^{m}\right)\right) \tag{6.9}$$

$$BI_{PV,k}^{m}\left(1\right)\leq BI_{bat,k}^{m}\left(1\right) \tag{6.10}$$

where $R_{p,j}^{m}$ and $R_{s,j}^{m}$ (\$/kWh) are the energy purchase and selling price at hour *j* for building *m* respectively; $P_{p,j}^{m}$ and $P_{s,j}^{m}$ (kW) are the purchase energy from power grid and selling energy back to the power grid at hour *j* for building *m* respectively; P_{grid} (kW) is the power grid capacity; $\overline{T}_{i,j}^{m}$ is the average indoor temperature for building *m* at hour *j*; T_i^{mL} and T_i^{mU} are 74 ℉ and 81 ℉ in this chapter; $Q_{b,j}^{m}$ (Btu/h) is the cooling energy supplied by the base chiller for building *m* at time *j*; $Q_{max,j}$ is the chiller capacity at time *j*; *BI* is a group of binary intermediate variables to denote the three state variables ($S_{is,k}$, $S_{bat,k}^{m}$, $S_{PV,k}^{m}$); ceil(.) rounds the element to the nearest integer towards infinity; $SOC_{is,max}$ and $SOC_{is,min}$ are maximum and minimum state of charge for the ice storage; $SOC_{is,k}$ is the initial state of charge for ice storage at building operation mode *k*; $SOC_{bat,\max}^{m}$ and $SOC_{bat,\min}^{m}$ are maximum and minimum state of charge for building *m*'s battery; $SOC_{bat,k}^{m}$ is the battery's initial state of charge for building *m* at building operation mode *k*. Please refer to chapter 2 for the detail explanation of the decision model.

6.3 Multi-objective PSO based Decision Framework

Particle Swarm Optimization (PSO) has attracted much attention and has been applied to many engineering and optimization problems in the last decade, for example, probabilistic traveling salesman problem (Marinakis & Marinaki, 2010), vehicle routing problem (Ai & Kachitvichyanukul, 2009), scheduling problem (Allahverdi & Al-Anzi, 2006; Liao et al., 2007), sequential ordering problem (Anghinolfi et al., 2011), just to name a few. In the PSO with inertia weight (Shi & Eberhart, 1998), the velocity and position for each particle *j* at iteration *i* are updated according to the following equations

$$\mathbf{v}_j^{i+1} = w\mathbf{v}_j^i + c_1 r_{1,j}^i \times \left(\mathbf{p}_j^i - \mathbf{x}_j^i\right) + c_2 r_{2,j}^i \times \left(\mathbf{p}_g^i - \mathbf{x}_j^i\right) \tag{6.11}$$

$$\mathbf{x}_j^{i+1} = \mathbf{x}_j^i + \mathbf{v}_j^{i+1} \tag{6.12}$$

where j denotes the j^{th} particle in the swarm; D-dimensional vector $\mathbf{v}_j^i$ is the velocity of the j^{th} particle ($\mathbf{v}_j^i \in \left[-\mathbf{v}_{\max}, +\mathbf{v}_{\max}\right]$), $\mathbf{v}_{\max}$ is used to constraint the velocity for each particle and is usually set between 0.1 and 1.0 times the search range of the solution space (Banks et al., 2007); D-dimensional vector $\mathbf{x}_j^i$ is the position of the j^{th} particle; $\mathbf{p}_j^i$ is the best position found so far by the j^{th} particle; $\mathbf{p}_g^i$ is the best position found so far by the swarm; $r_{1,j}^i$ and $r_{2,j}^i$ represent two independent random numbers uniformly distributed on [0, 1]; c_1 is the cognitive learning factor which represents the attraction that a particle has toward its own success $\mathbf{p}_j^i$; c_2 is the social learning factor which represents the attraction that a particle has toward its neighbors' best position $\mathbf{p}_g^i$; w is the inertia weight. Cognitive learning factor c_1 impacts the local search ability while the global search ability is influenced by the social learning factor c_2. Large inertia weight w enables particles to move in a high velocity and perform extensive exploration, and small inertia weight enhances the exploitation ability (Poli et al., 2007).

In the bi-level decentralized framework, other than the building agents each representing one building with the decision model explained in the section 6.2.2, a facilitator agent is introduced aiming to coordinate the buildings to reach converged Pareto solutions. Firstly, the facilitator agent classifies the decision variables into local variables (**X**) which are controlled by each building and coupled variables (**Y**) which are jointly controlled by more than one building. Similarly, the constraints are classified into local constraints which apply for each building and system constraints which apply for the group of buildings. Artificial coupled variables **Z** are introduced to decompose the decomposable system constraints into separable pieces so that each building can solve fully independent sub-problems.

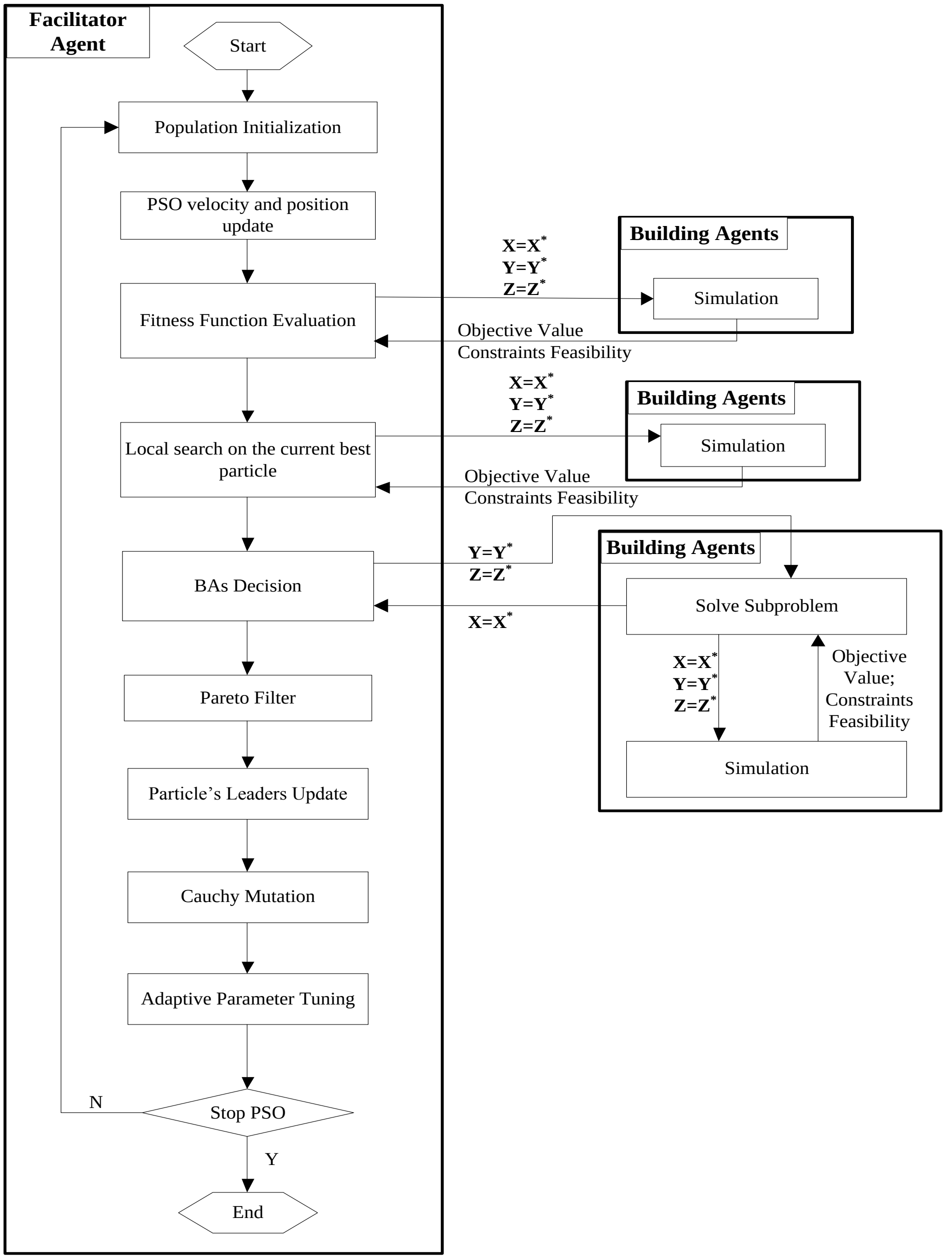


Figure 18 Bi-level decentralized framework based on AMOPSO

Then a Pareto-based multi-objective particle swarm optimization algorithms presented in chapter 5 is employed to search on the coupled decision space (**Y**, **Z**). The local search module is employed to improve the current best particle in the PSO swarm.

The updated decisions are passed to each building agent who attempts to "optimize" its own objective over the local variables (**X**) only and feeds the decisions on local variables back to the facilitator agent. Then, the Pareto filter (Loukil et al., 2007) is applied on the population to filter out the dominated solution, and the particles' leaders ($\mathbf{p}_j^i$, $\mathbf{p}_g^i$) are updated. The Cauchy mutation is employed to keep diversity of the swarm and an adaptive parameter tuning module is used to adaptively change the three parameters in PSO. The AMOPSO based decision framework is illustrated in Figure 18.

6.4 Implementation of Decentralized Particle Swarm Optimization

The augmented multi-objective particle swarm optimization (AMOPSO) in chapter 5 has been demonstrated to outperform existing representative multi-objective evolutionary algorithms (e.g., NSGA-II (Deb et al., 2002), PAES (Knowles & Corne, 1999), SPEA2 (Zitzler et al., 2001), MOCell (Nebro et al., 2009b)) and multi-objective particle swarm optimization algorithms (e.g., MOPSO (Coello Coello et al., 2004), OMOPSO (Reyes-Sierra & Coello Coello, 2005), MOCLPSO (Huang et al., 2006), 2LB-MOPSO (Zhao & Suganthan, 2011)) is employed in the decision framework to locate Pareto optimal solutions. The detail implementations of AMOPSO algorithm for building operation decisions are provided in the following sections.

6.4.1 Velocity and Position Update in Particle Swarm Optimization

Researchers have demonstrated that utilizing the building thermal mass (pre-cooling building), and shifting the peaking load by using a storage system can significantly reduce the energy cost (Braun, 2007; Drees & Braun, 1996; Sun et al., 2006). In this chapter, the set-point temperature $T_{sp,k}^{m}$ for building *m* at building operation mode *k* is initialized as follows:

$$T_{sp,k}^{m} = \begin{cases} T_{sp,k}^{m\ L} + \left(T_{sp,k}^{m\ U} - T_{sp,k}^{m\ L}\right) \times r/2 & k \text{ is pre-peak period and } r' \le 0.9 \\ T_{sp,k}^{m\ L} + \left(T_{sp,k}^{m\ U} - T_{sp,k}^{m\ L}\right) \times r & k \text{ is pre-peak period and } r' > 0.9 \\ T_{sp,k}^{m\ U} - \left(T_{sp,k}^{m\ U} - T_{sp,k}^{m\ L}\right) \times r/2 & \text{otherwise} \end{cases} \tag{6.13}$$

where the uniform random number $r, r' \in [0,1]$; $T_{sp,k}^{m\ L}$ and $T_{sp,k}^{m\ U}$ are 74 ℉ and 81 ℉ in this chapter.

The state of the ice storage system $S_{is,k}$ at building operation mode k is initialized as follows:

$$S_{is,k} = \begin{cases} 1 & k \text{ is pre-peak period and } r \le 0.9 \\ 2 & k \text{ is on-peak period and } r \le 0.9 \\ 0 & \text{otherwise} \end{cases} \tag{6.14}$$

The percentage of energy η_k^m from ice storage to building m is initialized as follows:

$$\eta_k^m = \begin{cases} r_m \Big/ \sum_{m=1}^{M} r_m & \text{if } S_{is,k} = 2 \\ 0 & \text{otherwise} \end{cases} \tag{6.15}$$

The initial population is generated from the feasible solutions after building agents check the feasibility of the tentative solutions.

The set-point temperature $T_{sp,k}^{m}$ and percentage of energy η_k^m are updated according to Eqs. (6.11)~(6.12). The state of the ice storage system is the most critical factor impacting the cost for each building. So here a uniform random number $r \in [0,1]$ is employed to control the convergence speed of the state of the ice storage system. The state of the ice storage system will be the same as the state in $\mathbf{p}_g^i$ when $r \le \exp\left(-i^5 / I^2\right)$. Otherwise the state of ice storage system will be generated by Eq. (6.14).

6.4.2 Local Search

A weighted sum objective function f is randomly generated to select a solution be improved by the local search.

$$f = \sum\nolimits_{m=1}^{M} w_m f_m \tag{6.16}$$

where

$$w_m = \begin{cases} r_m \Big/ \sum\nolimits_{m=1}^{M} r_m & r' \le 0.5 \\ 0 & r' > 0.5 \text{ and } \mathrm{randi}(M) \ne m \\ 1 & r' > 0.5 \text{ and } \mathrm{randi}(M) = m \end{cases} \tag{6.17}$$

r_m and *r'* are uniform random number from (0, 1) and m=1,…,M, randi(M) generates a integer number between 1 to M. In AMOPSO, two local search methods (non-uniform mutation based method (Michalewicz, 1996) and sub-gradient method (Boyd, 2010)) are studied to improve the current best solution $\mathbf{x}_g^i$ which is defined as

$$\mathbf{x}_g^i = \mathrm{argmin}_{1 \le j \le P} \left\{ f\left(\mathbf{x}_j^i\right) \right\} \tag{6.18}$$

Instead of using bi-local searches in this chapter, these two local search methods are combined together. The search direction in the non-uniform mutation based method is determined by the gradient descent direction. One dimension of set-point temperature $T_{sp,k}^m$ or percentage of energy η_k^m in the current best solution $\mathbf{x}_g^i$ is randomly picked to be mutated to generate a new solution. The set-point temperature for building m at building operation mode k is updated as

$$T_{sp,k}^m{}' = \begin{cases} T_{sp,k}^m - \Delta\left(i, T_{sp,k}^m - T_{sp,k}^{m\ L}\right) & \text{if case 1} \\ T_{sp,k}^m + \Delta\left(i, T_{sp,k}^{m\ U} - T_{sp,k}^m\right) & \text{if case 2} \end{cases} \tag{6.19}$$

where case 1 includes: 1) building comfort level constraint at building operation mode k is infeasible; or 2) building operation mode k is pre-peak period and a random number $r \in (0,1) > 0.5$; or 3) building operation mode k is not pre-peak period and a random number $r \in (0,1) > 0.9$; case 2 includes: 1) power grid capacity constraint or base chiller capacity constraint at building operation mode k is infeasible; or 2) building operation mode *k* is pre-peak period and a random number $r \in (0,1) \le 0.5$; or 3) building operation mode *k* is not pre-peak period and a random number $r \in (0,1) \le 0.9$. The percentage of energy from the ice storage system to each building is generated as:

$$\eta_k^m{}' = \begin{cases} \eta_k^m + \Delta\left(i, 1-\eta_k^m\right) & r \le w_m R_{p,k}^m \Big/ \sum_{m=1}^{M} w_m R_{p,k}^m \\ \eta_k^m - \Delta\left(i, \eta_k^m\right) & \text{otherwise} \end{cases} \tag{6.20}$$

where w_m is the weight for building *m* in the weighted objective; and $R_{p,k}^m$ is the average power grid purchase price for the building *m* at building operation mode *k*. The following function is adopted from (Zhao, 2011):

$$\Delta(i, y) = y \times \left(1 - \rho^{(1-i/I)^2}\right) \tag{6.21}$$

where ρ is a uniform random number from (0, 1). The final solution from local search replaces the current best solution $\mathbf{x}_g^i$ if it is better than $\mathbf{x}_g^i$ in terms of Eq. (6.16), otherwise it replaces the current worst solution evaluated by Eq. (6.16) if it is better than the worst solution.

6.4.3 Archive and Leader Update

At iteration *i*, the external archive and particles' leaders will be updated after local search. Particle $\mathbf{x}_j^i$ is discarded if it is dominated by any solution in the external archive. Otherwise it will be added into the external archive and all the solutions in the external archive which are dominated by $\mathbf{x}_j^i$ should be removed from the external archive. The first N_{max} (capacity of the external archive) non-dominated solutions in the external archive which have large crowding distance values will be kept in the archive when the size of the external archive exceeds N_{max}. The pBest ($\mathbf{p}_j^i$) for particle *j* is updated as

$$\mathbf{p}_j^{i+1} = \begin{cases} \mathbf{p}_j^i & \begin{array}{l} \left(\mathbf{x}_j^i \text{ is dominated by } \mathbf{p}_j^i\right) \text{ or} \\ \left(\mathbf{x}_j^i \text{ and } \mathbf{p}_j^i \text{ are non-dominated each other, and } r > 0.5\right) \end{array} \\ \mathbf{x}_j^i & \begin{array}{l} \left(\mathbf{x}_j^i \text{ dominates } \mathbf{p}_j^i\right) \text{ or} \\ \left(\mathbf{x}_j^i \text{ and } \mathbf{p}_j^i \text{ are non-dominated each other, and } r \le 0.5\right) \end{array} \end{cases} \tag{6.22}$$

The gBest ($\mathbf{p}_{g,j}^i$) for particle *j* is updated as

$$\mathbf{p}_{g,j}^{i+1} = \begin{cases} \text{random}\left(\mathbf{A}\right) & r \le 0.5 \\ \text{argmin}_{1 \le n \le N} \left\{ f'\left(\mathbf{A}_n\right)\right\} & r > 0.5 \end{cases} \tag{6.23}$$

where

$$f' = \sum_{m=1}^{M} \left\{ \left(r_m \middle/ \sum_{m=1}^{M} r_m \right) f_m \right\} \tag{6.24}$$

random(**A**) means randomly selecting a solution from archive **A**; *f'* is a randomly generated weighted sum objective function; *N* is the archive size; r_m is a uniform random number from (0, 1) and *m*=1,…,*M*. Please note all the *P* particles in the swarm use the same *f'* function to select gBest.

6.4.4 Cauchy Mutation

To keep the diversity of swarm, the Cauchy mutation operator is adopted which is demonstrated to be able to assist the particle by having a large jump out of its local optimum (Andrews, 2006). At iteration *i*, the *d*th dimension of the set-point temperature $T_{sp,k}^m$ or percentage of energy η_k^m from a randomly selected particle *j* will be mutated as

$$x_{j,d}^{i}{}' = x_{j,d}^{i} + cauchy\left(0.1\right) \times \eta^i \times \left(U_d - L_d\right) \tag{6.25}$$

where U_d and L_d are the upper and lower bounds of $x_{j,d}^i$; and η^i is the mutation scale parameter which is defined as

$$\eta^i = \max\left(\exp\left(-i^{2.61} / I^2\right), 0.1\right) \tag{6.26}$$

6.4.5 Adaptive Tuning

As large jumps from Cauchy mutation may be detrimental when the current search position is close to the neighborhood of the global optimum. Therefore, the distance between one randomly selected particle $\mathbf{x}_j^i$ and its gBest $\mathbf{p}_{g,j}^i$ is minimized. Taking w_j^i, $c_{1,j}^i$, and $c_{2,j}^i$ as decision variables, a *convex optimization* problem is formulated as:

$$\begin{aligned}&\min\ f_{dist}^{i-1} = \left\| \mathbf{x}_j^i - \mathbf{p}_{g,j}^i \right\|_2^2 \\ &= \left\| \mathbf{x}_j^{i-1} + w_j^{i-1}\mathbf{v}_j^{i-1} + c_{1,j}^{i-1} r_{1,j}^{i-1} \times \left(\mathbf{p}_j^{i-1} - \mathbf{x}_j^{i-1}\right) + c_{2,j}^{i-1} r_{2,j}^{i-1} \times \left(\mathbf{p}_{g,j}^{i-1} - \mathbf{x}_j^{i-1}\right) - \mathbf{p}_{g,j}^i \right\|_2^2 \\ &\text{s.t.}\quad 0.5 \le c_1 \le 2.5, \quad 0.5 \le c_2 \le 2.5, \quad 0.4 \le w \le 0.9\end{aligned} \tag{6.27}$$

The sub-gradient method (Boyd, 2010) employed to solve the convex optimization problem formulated in Eq. (6.27). w_j^i, $c_{1,j}^i$, and $c_{2,j}^i$ could be updated as described in the following equations

$$w_j^i = w_j^{i-1} - \alpha_j^{i-1} g_{w_j}^{i-1} \tag{6.28}$$

$$c_{1,j}^i = c_{1,j}^{i-1} - \alpha_j^{i-1} g_{c_{1,j}}^{i-1} \tag{6.29}$$

$$c_{2,j}^i = c_{2,j}^{i-1} - \alpha_j^{i-1} g_{c_{2,j}}^{i-1} \tag{6.30}$$

where α_j^{i-1} and $g_{w_j}^{i-1}$ are the step size and sub-gradient of the objective function in Eq. (6.27) at iteration *i* for particle *j*. Since the objective function in Eq. (6.27) is derivable, the derivative of f_{dist}^{i-1} evaluated at w_j^{i-1} is used as $g_{w_j}^{i-1}$. The optimal step size when the optimal value f_{dist}^* of the convex objective function is known is Polyak's step size (Boyd, 2010) which is computed as

$$\alpha_j^{i-1} = \left(f_{dist}^{i-1} - f_{dist}^* \right) \Big/ \left\{ \left(g_{w_j}^{i-1}\right)^2 + \left(g_{c_{1,j}}^{i-1}\right)^2 + \left(g_{c_{2,j}}^{i-1}\right)^2 \right\} \tag{6.31}$$

where the optimal value f_{dist}^* is always 0.

6.5 Experimental Analysis

The AMOPSO based framework is applied to study a simple building cluster (two buildings) located in the Phoenix, Arizona area. Since Phoenix is known for hot summers when energy usage is critically important, July 21, 2009 is studied as an example day for the experiments with data from SRP (http://www.srpnet.com), a local electricity provider.

6.5.1 Comparison between MA based Framework and AMOPSO based Framework

In this experiment, the computational performance of AMOPSO based framework and MA based framework is compared. Three building operation modes are considered: 1) from midnight to the onset of the on-peak period (0am-1pm); 2) the on-peak period (1pm-8pm); and 3) from the end of on-peak period to midnight (8pm-0am). The capacity of the power grid is assumed to be 15 kW. The heavy mass building applies the time-of-use (TOU) plan and the light mass building adopts the SRP EZ-3 option plan. In the EZ-3 plan, 3pm-6pm are the peak-hours where the price is much higher than the off-peak hours. In the TOU plan, 1pm-8pm are the peak-hours where the price is also higher (less than that of EZ-3) than the off-peak hours. During the off-peak hours, the price of the EZ-3 plan is relatively lower than that of the TOU plan. Other settings are the same as chapter 2. The following parameters of PSO are applied: 1) the PSO population size N_P is set to 30; 2) the maximal number of iteration I for PSO is 60; 3) the archive capacity is 50.

The Pareto frontier in the single building energy cost performance space obtained by the AMOPSO based framework is shown in Figure 19. The MA based Pareto frontier is also presented in Figure 19 for comparison. In the decision model, most of the computational time is spent on calling simulator to compute building cooling load, therefore, the number of simulator call is adopted to measure the computational performance of the MA and AMOPSO based framework. The computational time (minutes) on a computer (Intel Core i5 3.1 GHz CPU and 4GB memory) is also recorded. Three metrics are adopted from (Goh & Tan, 2007; Pulido & Coello Coello, 2004) to evaluate the solution performance of the MA and AMOPSO based framework in Table 30. All these three metrics are calculated based on normalized objective value, and all the non-dominated solutions in the MA based Pareto set and AMOPSO based Pareto set are used to approximate the true Pareto frontier (PF*).

1) Error Ratio: error ratio (ER) is a metric to measure the percentage of solutions in the approximated Pareto frontier (PF) that are not members of the true Pareto frontier (PF*). ER is computed as

$$ER = \sum_{i=1}^{n_{PF}} e_i \Big/ n_{PF} \tag{6.32}$$

where e_i=0 if solution i is a member of PF*, and e_i=1 otherwise; n_{PF} is the number of solutions in PF. A low value of ER is preferred.

2) Proximity Indicator: generational distance (GD) is a metric to measure the gap between the true Pareto frontier (PF*) and the approximated Pareto frontier (PF). GD is computed as

$$GD = \left(\frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d_i^2 \right)^{1/2} \tag{6.33}$$

where n_{PF} is the number of solutions in PF; d_i is the Euclidean distance (in objective space) between the i^{th} member of PF and its nearest member of PF*. A low value of GD is preferred, which reflects a small gap between PF and PF*.

3) Diversity Indicator: a modified maximum spread (MS) is a metric to measure how well the PF* is covered by PF. MS is computed as

$$MS=\left\{\frac{1}{M}\sum_{m=1}^{M}\left[\frac{\max\left(\min\left(f_m^{\max},F_m^{\max}\right)-\max\left(f_m^{\min},F_m^{\min}\right),0\right)}{F_m^{\max}-F_m^{\min}}\right]^2\right\}^{1/2} \tag{6.34}$$

where $F_m^{\max}$ and $F_m^{\min}$ is the maximum and minimum value of the mth objective in PF*; $f_m^{\max}$ and $f_m^{\min}$ is the maximum and minimum value of the mth objective in PF. A large value of MS is preferred, which reflects that a large area of PF* is covered by PF.

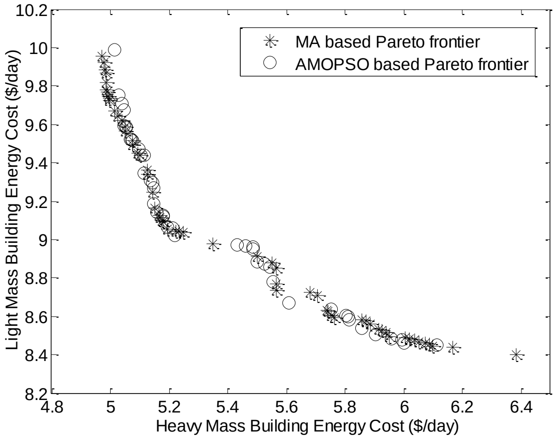


Figure 19 Pareto frontier obtained by the AMOPSO based decision framework

Table 30 Comparisons between MA based framework and AMOPSO based framework

	# of simulator calls	Computational time (minutes)	Error ratio (ER)	Generational distance (GD)	Maximum spread (MS)
MA	19200	~168	**0.3000**	0.0145	**1.0000**
PSO	**1883**	**~16**	0.3778	**0.0107**	0.8767

It is observed from Table 30 that AMOPSO based decision framework significantly reduces the computational cost (lower number of simulator call) and is more accurate than MA based decision framework (smaller value of generational distance). However, the AMOPSO based decision framework is poor on metrics of error ratio and maximum spread. This is also demonstrated in Figure 19 that AMOPSO based decision framework performs poor on the boundary of the Pareto frontier. It is due to the fact that small number of iterations reduces the chance of local search method to fine tune the search space, and thus to improve the solutions on the boundary.

6.5.2 Hourly Decentralized Decision

In section 6.5.1, it is demonstrated that AMOPSO based decision framework is able to significantly improve the computational performance without losing solution accuracy. This efficient decision framework enables me to explore hourly operation decisions. The Pareto frontiers in the single building energy cost performance space obtained by the AMOPSO based decision framework for hourly operation decisions and the three modes operation decisions (Liu & Henze, 2007) are demonstrated in Figure 20. It is observed that the Pareto frontier under three modes is dominated by the Pareto frontier of hourly operation decisions. Refining the decision time scale to hourly basis allows buildings to use the storage system more effectively, which is able to significantly reduce energy cost, and thus achieve more cost savings.

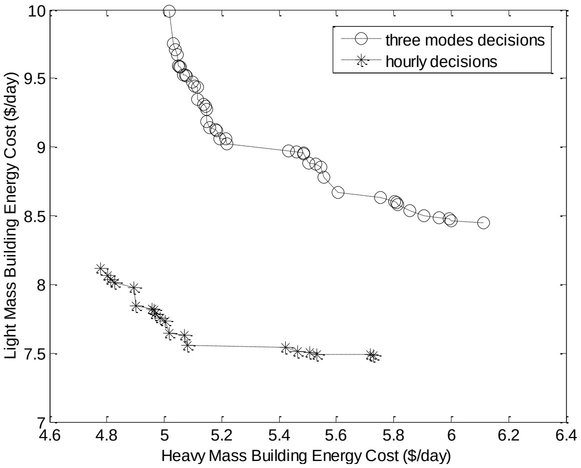


Figure 20 Hourly decentralized operation decisions

The minimal single building energy cost and total energy cost under three modes decisions and hourly decisions are recorded in Table 31. The cost for the two buildings is significantly reduced under hourly decisions.

Table 31 Energy costs for three modes decisions and hourly decisions

Decision types	HM building cost ($/day)	LM building cost ($/day)	Total cost ($/day)
Three modes decisions	5.0161	8.4502	14.2399

Hourly decisions	**4.7764**	**7.4813**	**12.6390**

6.6 Conclusions

The bi-level decision framework based on Memetic algorithm (MA) presented in chapter 2 is demonstrated to be capable of deriving the Pareto solutions for the building cluster in a decentralized manner. While promising, the decision framework based on MA is computationally expensive, which prohibits its application to hourly (or even less) basis operation decisions. In this chapter, a bi-level decision framework based on multi-objective particle swarm optimization (PSO) is developed which is capable of deriving good results with low computational cost to improve the computational performance of the decision framework. The hourly operation decision obtained by the AMOPSO based decision framework enables the buildings to utilize the storage system in a more efficient way, reduce energy waste and improve energy efficiency, and thus reduce the energy cost.

Although the AMOPSO based decision framework is capable for hourly decisions for building cluster, the decision framework is under deterministic assumption which means the uncertainties exist in weather (e.g., temperature, solar radiation) and building models, noises in sensors and meters are not considered. In the next chapter, some statistical analysis will be conducted to investigate properties of these uncertainties and noises, and then the model calibration techniques (e.g., particle filter (Chen, 2003)) are integrated with the decision framework to handle these uncertainties and noises. The integrated decision framework is expected to accurately calibrate the decision model by using on-line data collected from sensors and meters, and guarantee buildings could respond to their dynamic environments.

Chapter 7

ADAPTIVE OPERATION DECISIONS FOR SMART BUILDING CLUSTER

Due to the complexity of uncertainty analysis, the variety and diversity of uncertainties and noises in the building system and environment, less research is conducted on developing building operation strategies under uncertainty and noise though the importance of this research topic has long existing. In this chapter, an adaptive decision framework is developed to derive operation decisions for the smart building cluster in responding to dynamic environments by considering uncertainties exist in the building (e.g., non-cooling load) and environment (e.g., temperature, solar radiation), noises exist in sensors and meters. Let *L* be the decisions generation time length, *l* be the execution time scale, *l'* be the calibration frequency, the adaptive decision framework has three stages: 1) decisions generation stage: the decision framework presented in chapter 6 is employed to obtain the operation decisions for the next future *L* hours; 2) execution stage: the first *l* hours of the obtained operation decisions will be implemented; 3) calibration stage: a Gaussian mixture sigma point particle filter (GMSPPF) algorithm is launched to calibrate the building cluster model in a *l'* hour frequency. The experimental result demonstrates that GMSPPF algorithm is able to accurately calibrate the building cluster model, and the adaptive decision framework is able to derive adaptive operation decisions for the building cluster which could make the buildings quickly responding to the dynamic environment, and achieve more cost savings.

7.1 Introduction

In the United States, buildings use approximately 70% of total electricity usage and emit approximately 40% of greenhouse gases (GHG) annually (Kleissl & Agarwal, 2010). Building systems routinely fail to perform as designed (Hicks & von Neida, 2000), despite their sophisticated energy management and control systems. Between 4 to 20% of energy used for HVAC (Heating, Ventilating and Air Conditioning), lighting, and refrigeration in a building is wasted due to problems associated with systems operations. It is estimated that proper building energy load control and operation can lead to up to 40% utility cost savings (Braun, 1990).

Extensive researches have been conducted to develop efficient operation strategies for building system to reduce energy cost and improve energy efficiency. Pre-cooling building through optimally controlling building temperature set-points can significantly reduce energy cost (Braun, 2003; Braun et al., 2001; Chen, 2001; Henze et al., 2010; Keeney & Braun, 1996). As the development in thermal storage technique, extensive researches investigate utilizing both the building thermal mass and thermal storage to reduce energy cost. Most of the operation strategies which are derived by mathematical programming, simulation and reinforcement learning approaches are demonstrated to outperform the conventional control strategy such as chiller-priority and storage-priority strategies (Braun, 2007; Drees & Braun, 1996; Henze et al., 2005; Henze & Schoenmann, 2003; Lee et al., 2009; Liu & Henze, 2006a, 2006b; Liu & Henze, 2007; Sun et al., 2006). Recent research is interested in using optimization techniques to study energy generation system design, planning and control, and expects that integrating the energy generation system to building could significantly reduce energy consumption and cost (Arun et al., 2009; Fleten & Kristoffersen, 2008; Manolakos et al., 2001). Hu et al. (2012) investigate the operation decisions for the integrated building systems using a Memetic algorithm which could reduce energy cost and improve energy sustainability.

Due to the complexity and highly dynamics of building energy system, the challenges to develop an adaptive building operation strategy are:

1) *How to develop an accurate and high fidelity building model for decisions?*

Past researches show that the accuracy of the building model highly impacts the quality of the optimal operation strategies (Liu & Henze, 2004). Liu and Henze (2004) investigate the impact of five categories of building modeling mismatch on the performance of model-based predictive optimal control of thermal storage with perfect prediction of weather conditions, and demonstrate that the mismatch of internal heat gain, building construction and energy system efficiency can lead to a significant deviation in the optimal operation strategies. Henze et al. (2005) demonstrate that the cost savings for the calibrated model are substantial even with imperfect weather forecasts and imperfect match building models. The building operation strategy highly depends on the accuracy and robustness of the building models.

Nowadays, three models are implemented to model building energy system: 1) “white-box” model: requires specification of many physical parameters (Al-Homoud, 2001; Katipamula & Lu, 2006); 2) “black-box” model: requires a significant amount of training data and may not always reflect the actual physical behavior (Aydinalp et al., 2004; Dong et al., 2005; Ekici & Aksoy, 2009; Mihalakakou et al., 2002; Ozturk et al., 2004); 3) “gray-box” model: constructs a simplified model with online parameter/state estimation to represent the physical behavior of energy system (Braun & Chaturvedi, 2002; Wen, 2003; Zhou et al., 2008). The accuracy of the “white-box” model is highly dependent on the model parameters. Although the accuracy of “white-box” model can be improved by real time calibration, it still has more than 20% error in energy consumption prediction due to the measurement noise and model imprecision (Pan et al., 2007). The quality of “black-box” model is impacted by the accuracy of the training data and operation condition coverage of the training data. Although “gray-box” model is robust under different operation conditions, its accuracy is also impacted by the measurement noises.

2) *How to predict and quantify uncertainties existing in weather conditions, price rate, building parameters and measurements?*

The cost savings and on-peak electrical demand reductions are substantial when the weather forecasts are perfect (Henze et al., 2004). Henze et al. (2004) demonstrates that the bin predictor gives the smallest errors among TMY2 predictor, same-as-yesterday predictor, unbiased random walk predictor and SARIMA predictor. Henze et al. (1997; 2004) demonstrate that the predictive optimal control strategy outperforms the traditional control strategies. Henze et al. (2003b; 1997) summarize that the more accurate the real time pricing rate prediction is, the greater cost savings will be achieved. Several researches have demonstrated that the solar irradiation follows a Beta distribution (Mefti, 2003; Youcef Ettoumi et al., 2002), and the wind speed follows a Weibull distribution (Lu et al., 2002).

3) *How to develop an effective and efficient algorithm to locate the operation strategy under uncertainties?*

The fuzzy theory based mathematical programming (Mavrotas et al., 2008), simulation based optimization (Manolakos et al., 2001), chance-constrained programming (Arun et al., 2009; Cai et al., 2009), reinforcement learning (Henze & Schoenmann, 2003; Liu & Henze, 2006a, 2006b; Liu & Henze, 2007), stochastic dynamic programming (Livengood & Larson, 2009) have been employed to derive operation strategies under uncertainties. However, these techniques may have some issues. For example, the solution quality for the fuzzy theory based mathematical programming may be poor; the computation cost for the simulation based optimization is high; chance-constrained programming is restricted on the assumption of normal distribution for random variables; reinforcement learning is time consuming and its accuracy highly depends on the quality of data; stochastic dynamic programming cannot overcome the ‘curse of dimensionality’ problem.

To address these issues, this chapter develops an adaptive decision framework to study the adaptive operation decisions for the smart building cluster. First, the augmented multi-objective particle swarm optimization (AMOPSO) based decision framework presented in chapter 6 is employed to study hourly operation decisions for the smart building cluster. After implementation of the operation decision, a data fusion technique - Gaussian-Mixture Sigma-Point Particle Filter (GMSPPF) (van der Merwe, 2004) is launched to calibrate the building model with measurement data obtained from the sensors and meters. Then the AMOPSO based decision framework will obtain the operation decisions using the calibrated high-fidelity building model.

This chapter is organized as follows: section 7.2 reviews several existing data fusion techniques; the adaptive decision framework is presented in section 7.3; followed by the detail descriptions of the building model in section 7.4. Section 7.5 reports the experimental results, and conclusions are drawn in section 7.6.

7.2 Data Fusion Techniques

The section briefly reviews dynamic state-space model (DSSM) and several date fusion techniques for probabilistic inference in DSSM. The DSSM is introduced in section 7.2.1, followed by the Kalman filter presented in section 7.2.2, and particle filter presented in section 7.2.3.

7.2.1 Dynamic State-Space Model (DSSM)

This chapter focuses on the estimation of the states of a discrete-time dynamic system given noisy or incomplete measurements. This type of problem could be described by a dynamic state-space model (DSSM) which is formulated as

$$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k\right) \tag{7.1}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{n}_k) \tag{7.2}$$

where $\mathbf{x}_k$ is a *n*-dimension system state vector (unknown) at time *k* with initial probability density $p(\mathbf{x}_0)$; $\mathbf{u}_k$ is a *l*-dimension system input vector (known) at time *k*; $\mathbf{v}_k$ is the stochastic process noise at time *k*; *f* is the state transition function which relates the state at the current time *k* to the next time *k+1*; $\mathbf{y}_k$ is a *m*-dimension measurement vector (known) at time *k;* $\mathbf{n}_k$ is a *p*-dimension measurement noise vector (known) at time *k*; *h* is the measurement function which shows how the current system state relates to the measurement. The system states evolve over time as an indirect or partially observed first order Markov process according to the conditional probability density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and the measurement $\mathbf{y}_k$ is conditionally independent given the state according to the conditional probability density $p(\mathbf{y}_k|\mathbf{x}_k)$. The two main data fusion techniques for the system state estimation will be reviewed in the following sections.

7.2.2 Kalman Filter (KF)

Kalman filter is a well-known and often-used tool for stochastic state estimation from noisy measurements (Kalman, 1960). Kalman filter is an optimal, recursive data processing or filtering algorithm if both the state transition function *f* and measurement function *h* are linear, and the process noise $\mathbf{v}_k$ and measurement noise $\mathbf{n}_k$ are Guassian distributions. Under the linear and Guassian assumptions, the state transition and measurement functions could be simplified as

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{v}_k \tag{7.3}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k \tag{7.4}$$

where matrix **A** is the system state matrix that relates the state at the previous time *k*-1 to the current time *k*; matrix **B** relates the control input at previous time *k*-1 to the current time *k*; matrix **H** relates the system state to the measurement; the random variables $\mathbf{v}_k$ and

$\mathbf{n}_k$ are assumed to be white noises with independent normal distributions: $p(\mathbf{v}_k) \sim N(0, \mathbf{Q}_k)$ and $p(\mathbf{n}_k) \sim N(0, \mathbf{R}_k)$.

Kalman filter estimates the system states in two stages (Figure 21): 1) time update stage: *a priori* estimate of the system state for the next time is obtained by projecting forward the current system states and error covariance; 2) measurement update stage: *a posteriori* estimate of the system states at the current time is obtained by incorporating a new measurement into the *a priori* estimate.

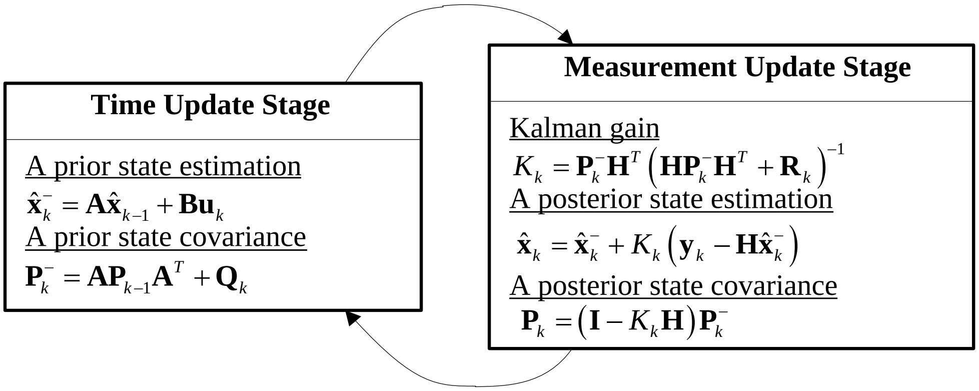

Time Update Stage
A prior state estimation
$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k$
A prior state covariance
$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}_k$
Measurement Update Stage
Kalman gain
$K_k = \mathbf{P}_k^- \mathbf{H}^T \left(\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k\right)^{-1}$
A posterior state estimation
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k\left(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right)$
A posterior state covariance
$\mathbf{P}_k = \left(\mathbf{I} - K_k\mathbf{H}\right)\mathbf{P}_k^-$

Figure 21 Two stages in Kalman filter

Several variants of Kalman filter have been developed to extend the Kalman filter for the nonlinear dynamic systems. For example, the extended Kalman filter (EKF) (Welch & Bishop, 1995) linearizes the state transition function *f* and measurement function *h*, and approximates the matrix **A** and **B** as the Jacobian matrix of partial derivatives of *f* and *h* with respect to **x**. The linearization in EKF may lead to poor error covariance updates and in some cases unstable growth of error covariance matrix (Evensen, 1992). Instead of linearization, ensemble Kalman filter (EnKF) (Evensen, 2003) uses Monte Carlo simulation to approximate the nonlinear state transition function *f*. Unlike KF and EKF where the estimation error is analytically propagated from time *k*-1 to *k*, a group of instances, called an ensemble, is used to track the evolution of the system state in EnKF. In EnKF, the mean and covariance of the ensemble, derived from samples, are taken as state estimation and error covariance. Other than linearization and sampling techniques, Julier et al. (1995) develop the unscented Kalman filter (UKF) which utilizes the unscented transformation to estimate the system states and error covariance. UKF is demonstrated to consistently outperform the EKF in terms of state estimation accuracy and estimate consistency for the same computational cost (Julier et al., 1995; Julier & Uhlmman, 1997). Ito and Xiong (2000), Norgaard et al. (2000) develop the central difference Kalman filter (CDKF) which is based on the Stirling's interpolations formula to estimate the system states and error covariance. Both the UKF and CDKF belong to the category of sigma point Kalman filter (SPKF) (van der Merwe, 2004). UKF and CDKF may be computationally ineffective for high-dimensional dynamic system states estimation (van der Merwe, 2004).

7.2.3 Particle Filter (PF)

The Gaussian assumption for Kalman filter and its variants does not hold in the non-linear non-Gaussian dynamic systems. The particle filter (PF) which uses the sequential importance sampling (SIS) technique is developed to handle non-linear non-Gaussian dynamic system (Carpenter et al., 1999; Gordon et al., 1993). Without assumptions for linear and Gaussian, the posterior system states and error covariance are approximated by a set of weighted samples in PF. A common problem with SIS particle filter is the rapid degeneracy of weights for samples which means only very few particles have non-zero importance weights after some iterations (Arulampalam et al., 2002). To address this issue, several resampling methods (e.g., sampling-importance resampling, residual resampling, etc.) have been implemented to avoid samples with low importance weights and multiply samples with high importance weights (van der Merwe, 2004). The pseudo-code description of the particle filter with resampling is presented in Table 32.

Table 32 Pseudo-code of particle filter with resampling

```
For time steps $k$=0,1,2,…
  For particle $p$=1,…,$N_p$, draw the particle $\mathbf{x}_k^p$ from the proposal distribution $\pi\left(\mathbf{x}_k^p \mid \mathbf{x}_{0:k-1}^p, \mathbf{y}_{0:k}\right)$
  For particle $p$=1,…,$N_p$, calculate the importance weight for particle $\mathbf{x}_k^p$ as
      $w_k^p = w_{k-1}^p \frac{p\left(\mathbf{y}_k \mid \mathbf{x}_k^p\right) p\left(\mathbf{x}_k^p \mid \mathbf{x}_{k-1}^p\right)}{\pi\left(\mathbf{x}_k^p \mid \mathbf{x}_{0:k-1}^p, \mathbf{y}_{0:k}\right)}$    (7.5)
  For particle $p$=1,…,$N_p$, normalize the importance weight for particle $\mathbf{x}_k^p$ as
      $\tilde{w}_k^p = w_k^p \Big/ \sum_{p=1}^{N_p} w_k^p$    (7.6)
  Multiply/suppress samples $\mathbf{x}_k^p$ with high/low importance weight $\tilde{w}_k^p$ respectively
  For particle $p$=1,…,$N_p$, reset $w_k^p = \tilde{w}_k^p = 1/N_p$
  Output the posterior distribution and system state as
      $\hat{p}\left(\mathbf{x}_k \mid \mathbf{y}_{1:k}\right) = \sum_{p=1}^{N_p} w_k^p \delta\left(\mathbf{x}_k - \mathbf{x}_k^p\right)$    (7.7)
      $\hat{\mathbf{x}}_k = E\left[\mathbf{x}_k \mid \mathbf{y}_{1:k}\right] \approx \sum_{p=1}^{N_p} w_k^p x_k^p$    (7.8)
```

The particle filter's performance depends on the selection of the proposal distribution $\pi\left(\mathbf{x}_k^p \mid \mathbf{x}_{0:k-1}^p, \mathbf{y}_{0:k}\right)$. Recently, extensive researches focus on improving the accuracy of the proposal distribution to improve particle filter's performance. For example, van der Merwe (2004) develops a sigma point particle filter (SPPF) which uses sigma point Kalman filter (SPKF) to accurately estimate the mean and covariance of the Gaussian proposal distribution for each particle. For each particle *p* at iteration *k*, a SPKF algorithm is launched to update the Gaussian proposal distribution for this particle with measurement data, and a new particle $\mathbf{x}_k^p$ is sampled from the updated proposal distribution. The computational cost of SPPF may be high especially for high-dimensional and complex system states estimation (van der Merwe, 2004). In order to reduce the computational cost for the SPPF, van der Merwe (2004) develops a Gaussian mixture sigma point particle filter (GMSPPF) which employs a finite Gaussian mixture model (GMM) to approximate the posterior density for the system states. At each time *k*, the SPKF is launched to update the mean and covariance of each Gaussian component in the Gaussian mixture model with updated measurement data. A group of new particles is sampled from the updated Gaussian mixture model to calibrate the system states at current time *k* and predict the states at next time *k*+1. The detail description and implementation of the SPPF and GMSPPF are provided in (van der Merwe, 2004). Due to its good performance in terms of accuracy and computational effectiveness, the GMSPPF algorithm is adopted as the data fusion technique in the adaptive decision framework.

7.3 Adaptive Decision Framework

In this chapter, an adaptive decision framework is developed to study the operation decisions for the smart building cluster. Let *L* be the decisions generation time length, *l* be the execution time scale, *l'* be the calibration frequency, the adaptive decision framework has three stages: 1) decisions generation stage: the AMOPSO based decentralized decision framework presented in chapter 6 is employed to derive Pareto operation decisions for the next future *L* hours; 2) execution stage: one of the Pareto operation decisions is selected based on a predefined performance metric (e.g. total energy cost), and the first *l* ($l \leq L$) hours operation decision is implemented; 3) calibration stage: the GMSPPF is employed to calibrate the building model with measurement data collected in a *l'* ($l' \leq l$) hour frequency. The updated building model will be incorporated into the decision model at the beginning of each decisions generation stage. The three stages process in the adaptive decision framework is presented in Figure 22.

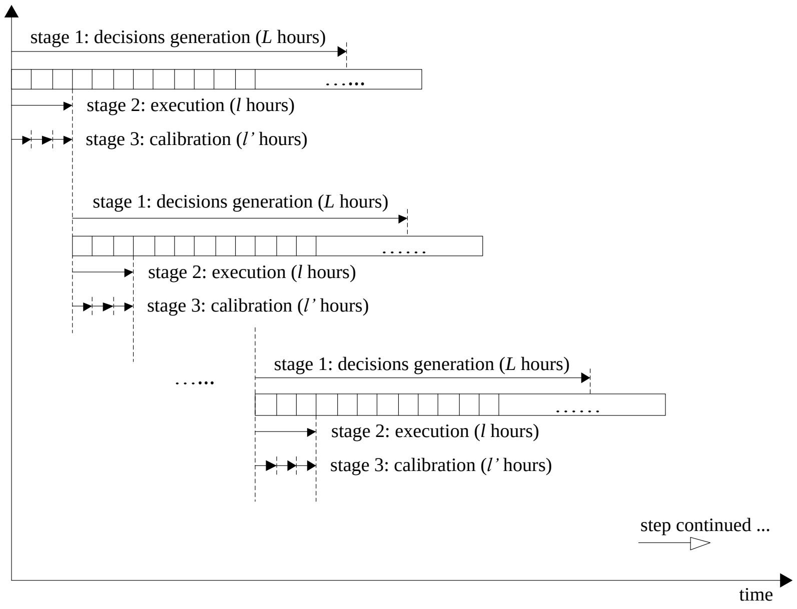

stage 1: decisions generation (*L* hours)
……
stage 2: execution (*l* hours)
stage 3: calibration (*l'* hours)
stage 1: decisions generation (*L* hours)
……
stage 2: execution (*l* hours)
stage 3: calibration (*l'* hours)
……
stage 1: decisions generation (*L* hours)
……
stage 2: execution (*l* hours)
stage 3: calibration (*l'* hours)
step continued ...
time

Figure 22 Three stages process in the adaptive decision framework

7.3.1 Decisions Generation Stage

In the last decade, Particle Swarm Optimization (PSO) has attracted much attention and has been applied to many engineering and optimization problems. In this chapter, the multi-objective particle swarm optimization based bi-level decentralized decision framework presented in chapter 6 is adopted to derive Pareto operation decisions for the building system. The decision framework presented in chapter 6 has been demonstrated to be able to derive hourly operation decisions for the building cluster and reduce energy cost for the building cluster. In this chapter, the decision problem studied in the decisions generation stage is deterministic. The uncertainties are considered in the calibration stage (see section 7.3.3).

7.3.2 Execution Stage

In this stage, the operation strategies of first *l* hours over the *L*-hour period in the decisions generation stage will be implemented. The solutions obtained in the decisions generation stage are a set of Pareto solutions, so some utility functions should be adopted in this stage to locate one Pareto solution from the Pareto set as the final decision for the first *l* hours. In this chapter, the utility function is defined as

$$F = \sum_{m=1}^{M} f_m \tag{7.9}$$

where *M* is the number of buildings studied, f_m is the energy cost of the *L*-hour period for the m^{th} building. In the execution stage, the solution which minimizes the total energy cost (utility function *F*) for the whole system will be implemented.

7.3.3 Calibration Stage

In this stage, the measurement data from sensors and meters will be collected every *l'*-hour to calibrate the current system states and predict the system states for the next time. The predicted system states are used as the initial states for the dynamic system in the decisions generation stage. The GMSPPF algorithm (van der Merwe, 2004) is adopted in the calibration stage. In the GMSPPF, the posterior state density at time *k*-1 is approximated by a G-component Gaussian mixture model (GMM)

$$\tilde{p}\left(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}\right) = \sum_{g=1}^{G} \tilde{\alpha}_{k-1}^{g} \, \mathrm{N}\left(\mathbf{x}_{k-1}; \tilde{\boldsymbol{\mu}}_{k-1}^{g}, \tilde{\mathbf{P}}_{k-1}^{g}\right) \tag{7.10}$$

and the process and measurement noise densities are approximated by the following *I* and *J* component GMMs respectively.

$$\tilde{p}\left(\mathbf{v}_{k-1}\right) = \sum_{i=1}^{I} \beta_{k-1}^{i} \, \mathrm{N}\left(\mathbf{v}_{k-1}; \boldsymbol{\mu}_{v,k-1}^{i}, \mathbf{Q}_{k-1}^{i}\right) \tag{7.11}$$

$$\tilde{p}\left(\mathbf{n}_{k}\right) = \sum_{j=1}^{J} \gamma_{k}^{j} \, \mathrm{N}\left(\mathbf{n}_{k}; \boldsymbol{\mu}_{n,k}^{j}, \mathbf{R}_{k}^{j}\right) \tag{7.12}$$

At each time k, the SPKF will be employed to update the mean and covariance for the prior density $p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}\right)$ and posterior density $p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k}\right)$. The detail implementation of GMSPPF is provided in (van der Merwe, 2004).

7.4 Integrated Building Model and Calibration Model

7.4.1 Integrated Building Model

Same as chapter 2, a simplified building cluster consisting of two different mass level - heavy mass (HM) and light mass (LM) buildings is modeled. The two buildings, each having its own battery and photovoltaic (PV) panel, share one ice storage system and one base chiller. The ice storage system charged by a dedicated chiller is configured in parallel with the base chiller. During on-peak hours, the buildings cooling loads are met primarily by the ice storage system with the remaining cooling request satisfied by the base chiller. The overall schematic of the building energy system configuration is illustrated in Figure 3 with the arrows denoting the energy flow among each component in the system.

7.4.2 Building Energy System Calibration Model

The building energy system is a complex dynamic system, and there exist many different types of uncertainties in the building system and environment. In this chapter, the process uncertainties from three aspects are considered: 1) dry bulb temperature T_{db} (℉) which is calculated as

$$T_{db} = \tilde{T}_{db} + \varepsilon_T \tag{7.13}$$

where $\tilde{T}_{db}$ (℉) is the forecast temperature from the weather station; ε_T is the error between the actual temperature and forecast temperature, and is assumed to follow a standard normal distribution. 2) clearness index k_T which follows a beta distribution (Mefti, 2003; Youcef Ettoumi et al., 2002). Given clearness index k_T, the total hourly solar irradiance on the horizontal surfaces G_{th} (W/m^2) is calculated as

$$G_{th} = G_0 k_T \tag{7.14}$$

where G_0 (W/m^2) is the extraterrestrial hourly solar irradiance on the horizontal surfaces. 3) P_{load}: building's non-cooling electricity load which follows a normal distribution (Valenzuela et al., 2000).

In this chapter, the system states studied in the building energy system are summarized in Table 33, and all the available measurements are summarized in Table 34, where m=1,…,M and M is the number of buildings studied. It is observed that there are 18 states, and 8 measurements in this chapter.

Table 33   System states in the building energy system

System States	**Descriptions**
T_i^m (°F)	air temperature inside the building
T_{ew}^m (°F)	east wall temperature
T_{sw}^m (°F)	south wall temperature
T_{ww}^m (°F)	west wall temperature
T_{nw}^m (°F)	north wall temperature
T_{roof}^m (°F)	roof temperature
Q_c^m (Btu/h)	building total cooling load
P_{PV}^m (kWh)	total energy generated by PV panel
P_{nc}^m (kWh)	building total non-cooling energy requested from power grid

Table 34   Measurements in the building energy system

Measurements	**Descriptions**
$\tilde{T}_i^m$ (°F)	air temperature inside the building
$\tilde{Q}_c^m$ (Btu/h)	building total cooling load
$\tilde{P}_{PV}^m$ (kWh)	total energy generated by PV panel
$\tilde{P}_{nc}^m$ (kWh)	building total non-cooling energy requested from power grid

The state transition functions (Eqs. (7.15)~(7.18)) and measurement functions (Eqs. (7.19)~(7.22)) are defined as

$$T_{j,k}^m = f_j\left(T_{i,k-1}^m, T_{ew,k-1}^m, T_{sw,k-1}^m, T_{ww,k-1}^m, T_{nw,k-1}^m, T_{roof,k-1}^m, T_{db,k-1}\right) \tag{7.15}$$

$$Q_{c,k}^{m} = Q_{c,k-1}^{m} + f_{Q}\left(T_{i,k-1}^{m}, T_{ew,k-1}^{m}, T_{sw,k-1}^{m}, T_{ww,k-1}^{m}, T_{nw,k-1}^{m}, T_{roof,k-1}^{m}, T_{db,k-1}\right) \tag{7.16}$$

$$P_{PV,k}^{m} = P_{PV,k-1}^{m} + f_{PV}^{m}\left(G_{th,k-1}, T_{db,k-1}\right) \times \eta_{inv} \tag{7.17}$$

$$\begin{aligned} P_{nc,k}^{m} = P_{nc,k-1}^{m} &+ \max\left(P_{bat,k}^{m} BI_{bat,k}^{m}\left(1\right) - f_{PV}^{m}\left(G_{th,k-1}, T_{db,k-1}\right)\eta_{inv} BI_{PV,k}^{m}\left(1\right), 0\right) \Big/ \eta_{conv} \\ &+ \max\left(P_{load,k}^{m} - P_{bat,k}^{m}\eta_{conv} BI_{bat,k}^{m}\left(2\right) - f_{PV}^{m}\left(G_{th,k-1}, T_{db,k-1}\right)\eta_{inv} BI_{PV,k}^{m}\left(2\right), 0\right) \end{aligned} \tag{7.18}$$

$$\tilde{T}_{i,k}^{m} = T_{i,k}^{m} + \eta_{T}^{m} \tag{7.19}$$

$$\tilde{Q}_{c,k}^{m} = Q_{c,k}^{m} \times \left(1 + \eta_{Q}^{m}\right) \tag{7.20}$$

$$\tilde{P}_{PV,k}^{m} = P_{PV,k}^{m} \times \left(1 + \eta_{PV}^{m}\right) \tag{7.21}$$

$$\tilde{P}_{nc,k}^{m} = P_{nc,k}^{m} \times \left(1 + \eta_{nc}^{m}\right) \tag{7.22}$$

where *j=i, ew, sw, ww, nw, roof*; f_j(.) and f_Q(.) are derived from the simulation model of the cooling load module, and f_{PV}(.) is derived from the PV panel module in Figure 3. $\eta_{T}^{m} \sim N\left(0,1\right)$, $\eta_{Q}^{m} \sim N\left(0, 0.01^{2}\right)$, $\eta_{PV}^{m} \sim N\left(0, 0.01^{2}\right)$, and $\eta_{nc}^{m} \sim N\left(0, 0.01^{2}\right)$ are measurement noises which are assumed to follow normal distributions; $P_{bat,k}^{m}$ (kW) is the charging/discharging power of the battery for building *m* at time *k*; η_{conv} is the battery AC/DC converter efficiency, which is 0.9 according to its specification in this chapter; η_{inv} is the PV panel inverter efficiency, which is 0.92 in this chapter; $BI_{bat,k}^{m}$ and $BI_{PV,k}^{m}$ are the states of battery and PV panel for building *m* at time *k*, which are two decision variables in the decisions generation stage.

7.5 Experimental Analysis

The adaptive decision framework is applied to study a simple building cluster (two buildings) located in the Phoenix, Arizona area. Since Phoenix is known for hot summers when energy usage is critically important, July 20, 2012 is studied as an example day for the experiments with data from SRP (http://www.srpnet.com), a local electricity provider. The heavy mass building applies the time-of-use (TOU) plan and the light mass building adopts the SRP EZ-3 option plan. In the EZ-3 plan, 3pm-6pm are the peak-hours where the price is much higher than the off-peak hours. In the TOU plan, 1pm-8pm are the peak-hours where the price is also higher (less than that of EZ-3) than the off-peak hours. During the off-peak hours, the price of the EZ-3 plan is relatively lower than that of the TOU plan.

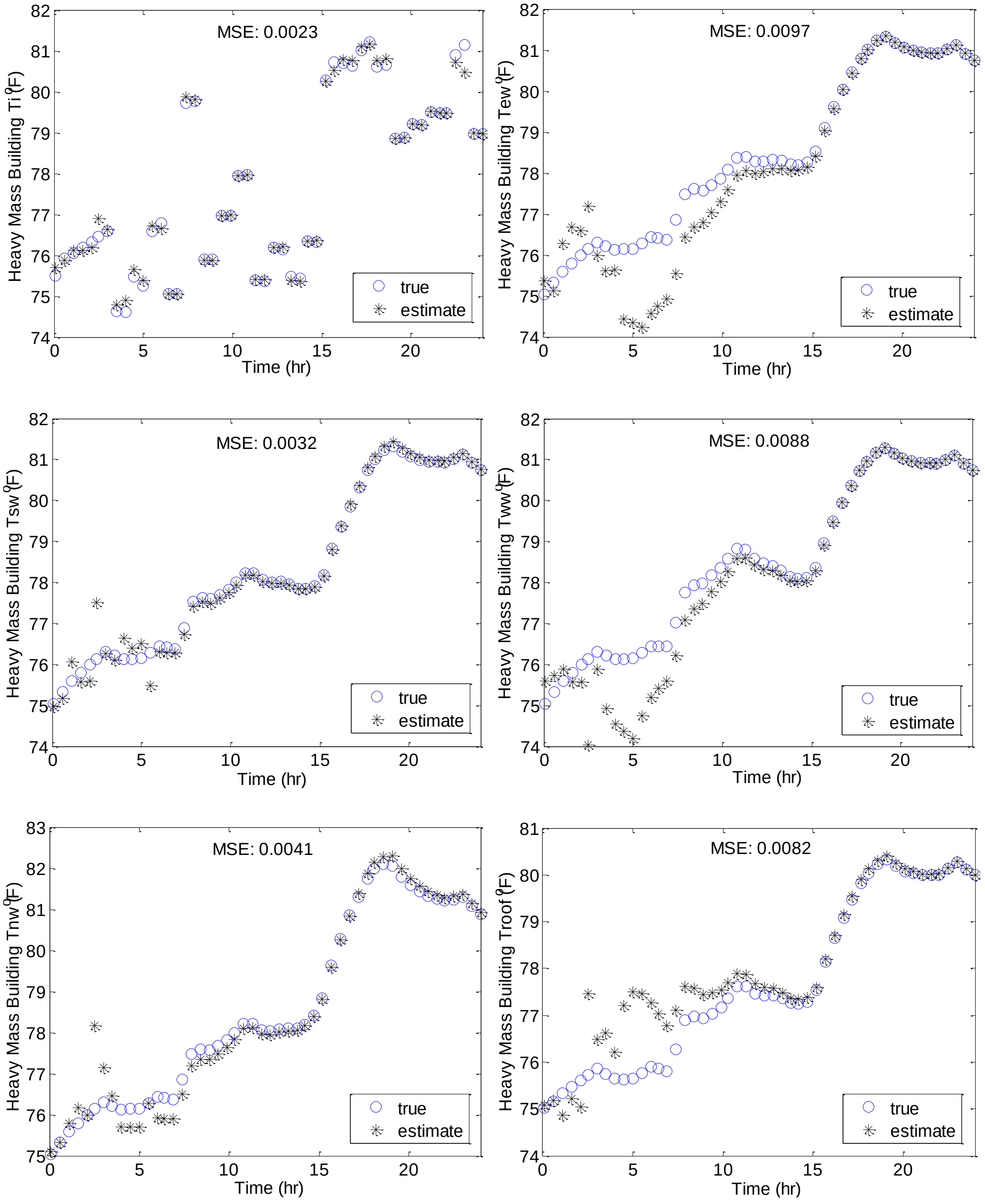
The parameter settings in the adaptive decision framework are: 1) decisions generation stage: decisions generation time length *L* is 24 hours; the PSO population size is set to 30; the maximal number of iteration for PSO is 80; and the archive capacity is 50; 2) execution stage: execution time scale *l* is 1 hour; 3) calibration stage: calibration time frequency *l'* is 0.1 hour; sample size for the GMSPPF is 200; number of components for the state, process noise and measurement noise GMM are 3, 2, and 1 respectively; the initial indoor temperature, wall temperature and roof temperature are 75 °F. In this chapter, an emulator is implemented to represent the real case and collect measurement data.

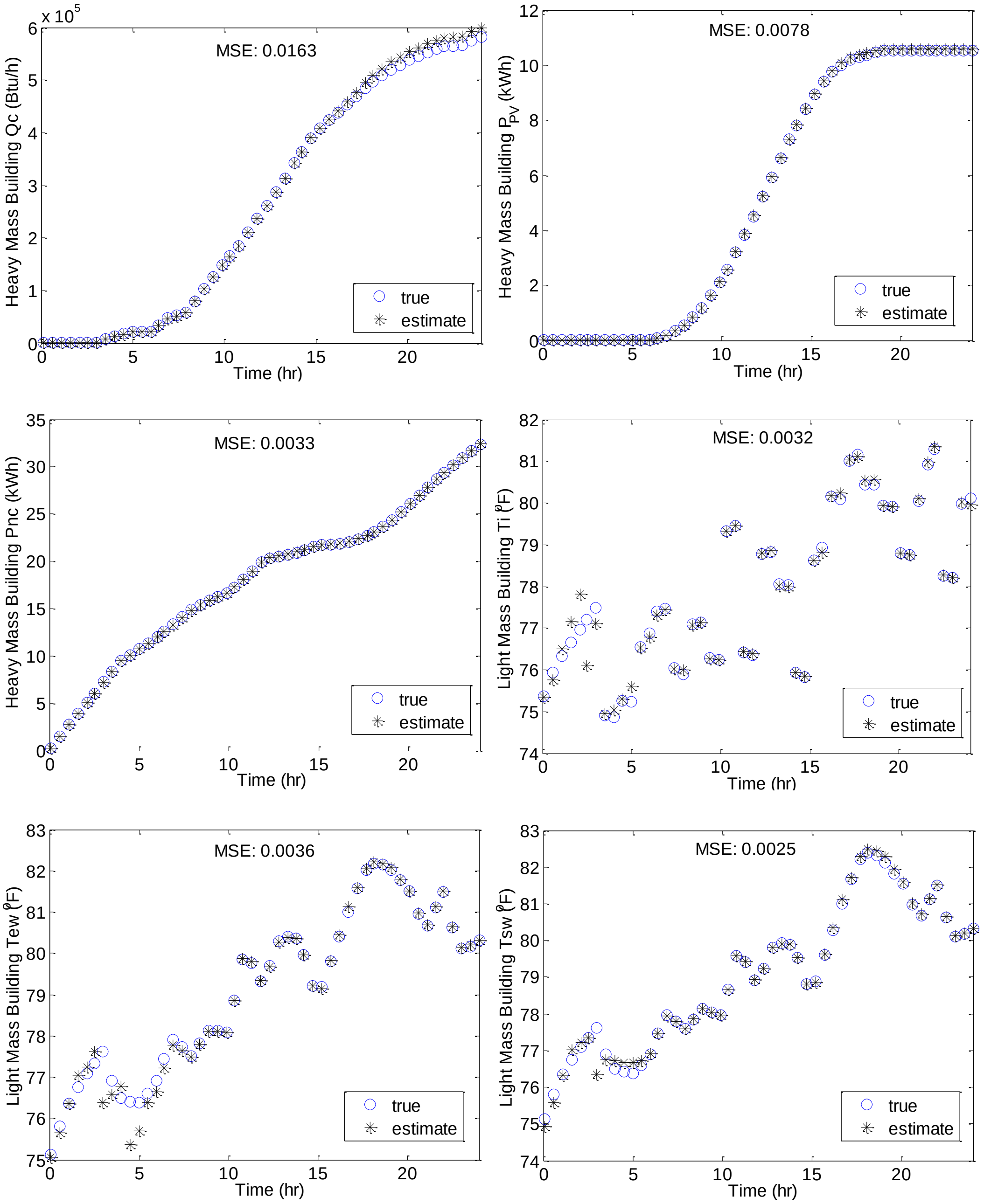
7.5.1 Calibration Result Analysis

In this experiment, the calibration time frequency is 0.1 hour. The mean square error (MSE) is adopted to evaluate the solution performance of GMSPPF for the building energy system calibration. The mean square error is calculated based on normalized value, which is defined as

$$MSE = \sqrt{\frac{\sum_{k=1}^{K}\left(\left(\hat{\mathbf{x}}_k - \mathbf{x}_k\right)/\mathbf{x}_k\right)^2}{K}} \tag{7.23}$$

where $\mathbf{x}_k$ is the true state vector, and $\hat{\mathbf{x}}_k$ is the posterior state vector at time *k*. The calibration results for the 18 states in Table 33 are demonstrated in Figure 23, and the MSE for each state is also presented. It is observed from Figure 23 that most of the states are close to the true states. However, the errors in T_{ew}^{m}, T_{sw}^{m}, T_{ww}^{m}, T_{nw}^{m} and T_{roof}^{m} may be large since no measurement data related to these five states is collected.




x 10^5
Heavy Mass Building Qc (Btu/h)
MSE: 0.0163
true
estimate
Time (hr)
Heavy Mass Building P_PV (kWh)
MSE: 0.0078
Heavy Mass Building Pnc (kWh)
MSE: 0.0033
Light Mass Building Ti (ºF)
MSE: 0.0032
Light Mass Building Tew (ºF)
MSE: 0.0036
Light Mass Building Tsw (ºF)
MSE: 0.0025

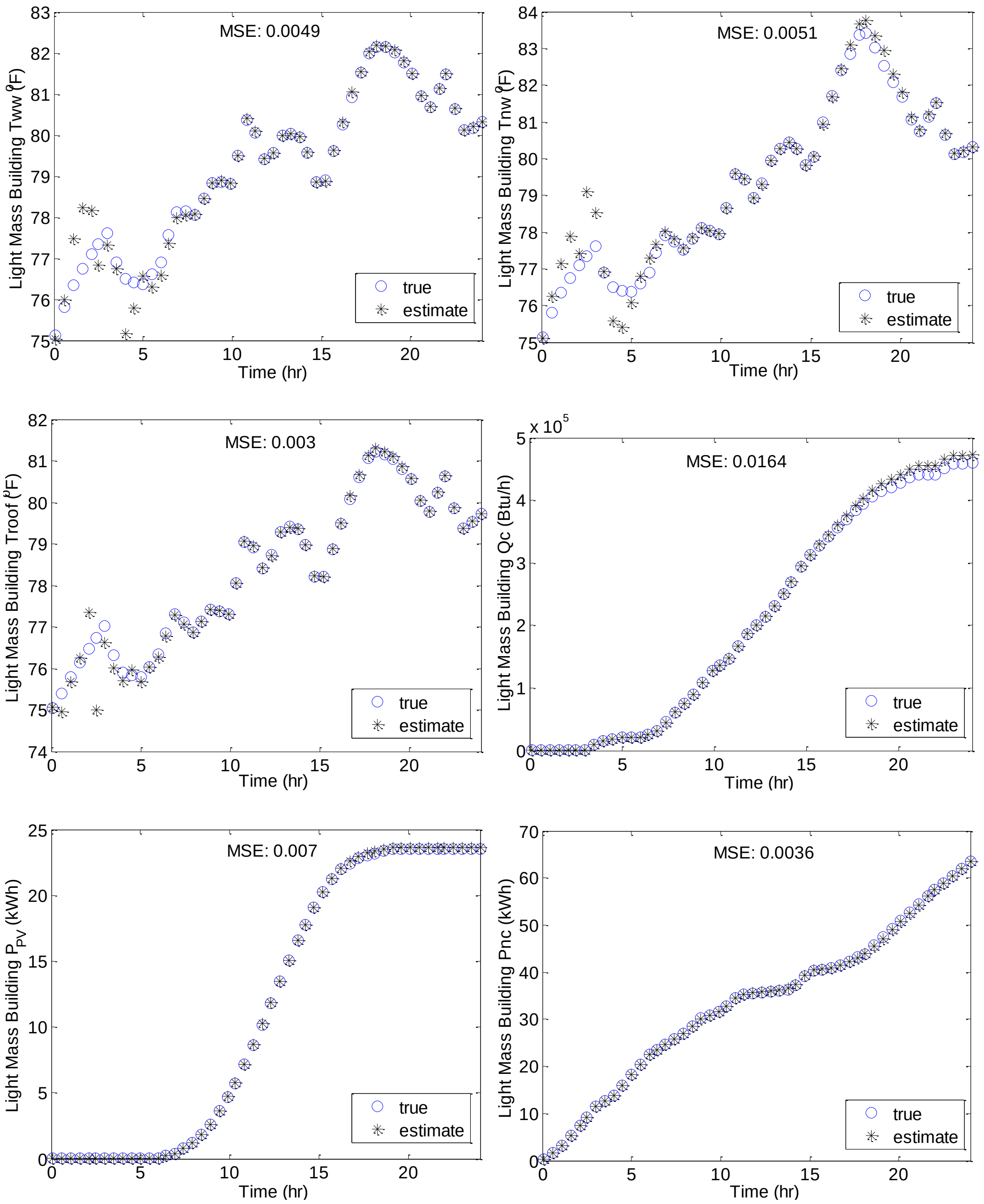


Figure 23 Calibration results for the two buildings

7.5.2 Operation Decisions using the Adaptive Decision Framework

In section 7.5.1, it is demonstrated that GMSPPF is able to accurately calibrate the building energy system. In this experiment, the impact of the model calibration for the building energy system operation decisions is studied.

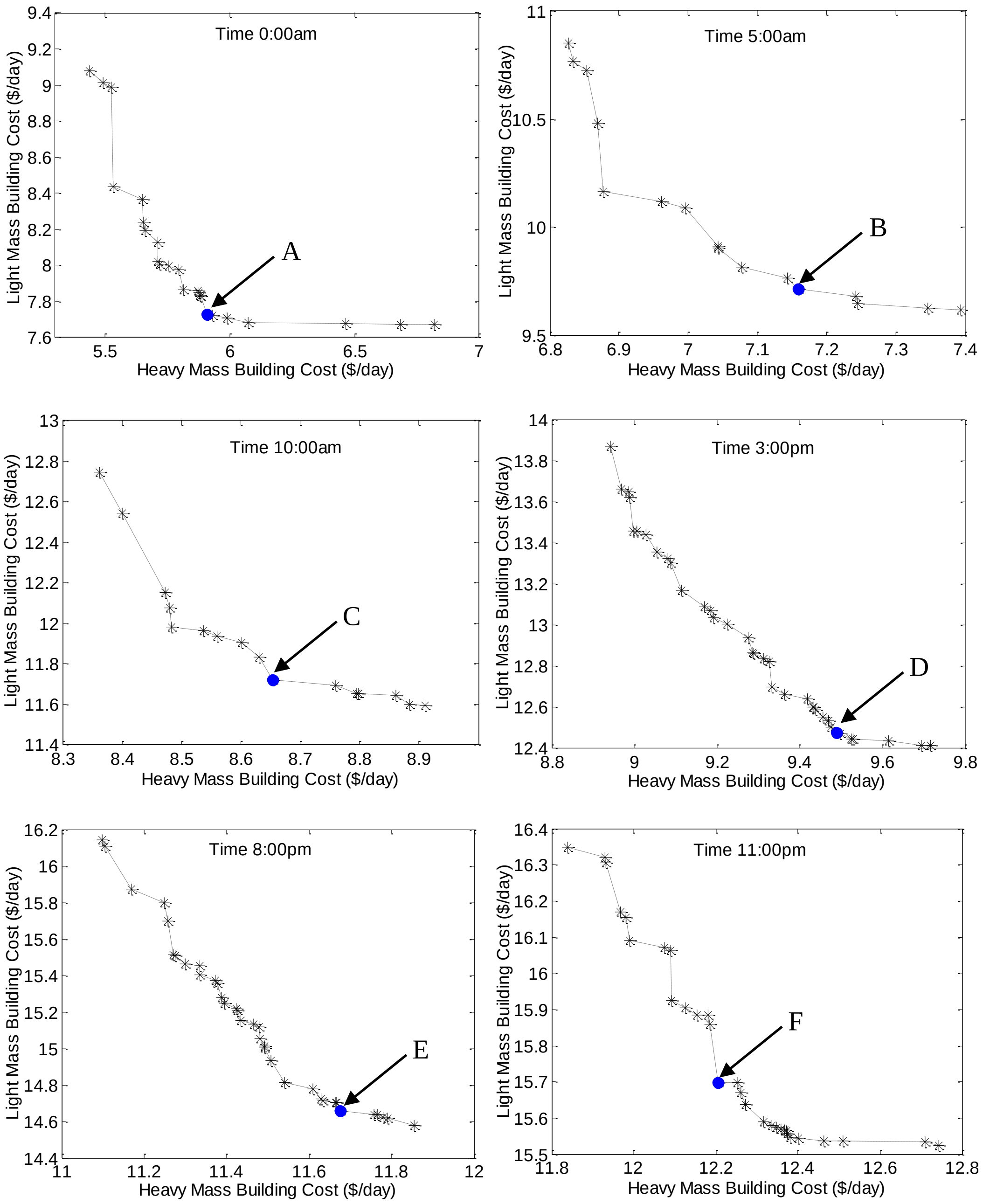


Figure 24 Adaptive operation decisions in the energy cost space

The Pareto frontiers in the single building energy cost performance space obtained by the AMOPSO based decision framework at each hour (six hours are selected for demonstration) are demonstrated in Figure 24. The circle points (solutions A~F) represent the Pareto decisions selected for implementation at the execution stage. The adaptive operation decisions for the heavy mass (HM) building and light mass (LM) building set-point temperature for the day of July 20, 2012 are demonstrated in Figure 25. Solutions A~F in Figure 24 are also shown in Figure 25.

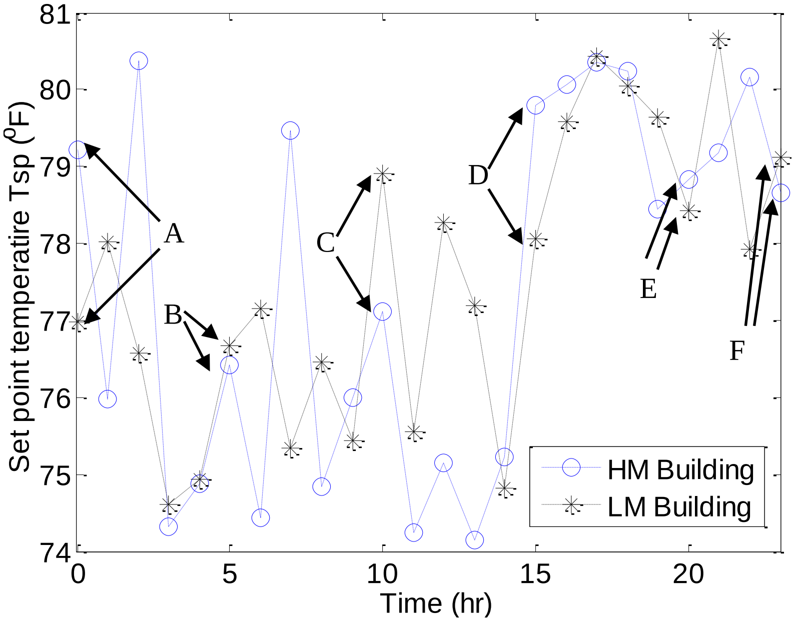


Figure 25 Building set-point temperature adaptive decisions

Table 35 Operation decisions comparison analysis

Solutions	HM building cost ($/day)	LM building cost ($/day)	Total cost ($/day)	HM building PGDR (%)	LM building PGDR (%)
Adaptive framework	**5.75**	**7.82**	**13.57**	**19.42**	**24.08**
Case I	5.98	8.04	14.02	19.50	24.62
Case II	6.27	8.44	14.71	19.81	24.82

In order to demonstrate the effectiveness of the adaptive decision framework, two additional sets of operation decisions are obtained: 1) case I: obtaining the operation decisions using the adaptive decision framework without calibration (equivalent to section 7.3.1 and section 7.3.2); 2) case II: obtaining the operation decisions using the adaptive decision framework without calibration, but considering uncertainties studied in section 7.4.2. In case II, the AMOPSO based decision framework will study a stochastic decision problem instead of a deterministic decision problem. The single building daily energy cost, total daily energy cost, power grid dependency rate (PGDR) (Hu et al., 2012) corresponding to these three sets of operations decisions under real case are recorded in Table 35. It is observed that the adaptive framework outperforms other two cases in these three performance metrics.

7.6 Conclusions

Due to the dynamics and complexity of the building energy system, a good operation strategy first requires an accurate model for building system energy usage which is currently lacking. In this chapter, an adaptive decision framework is developed to derive operation decisions for the building energy system considering uncertainties and noises existing in the building system, environment, sensors and meters. The adaptive decision framework has three stages: 1) the decisions generation stage is capable of deriving hourly Pareto operation decision; 2) the execution stage implements one of the decisions obtained in the decisions generation stage; 3) the GMSPPF is employed in the calibration stage to calibrate the building model. The experimental results demonstrate that the GMSPPF is capable of accurately calibrating the building model and the adaptive framework will be more cost effective for the building cluster.

In this chapter, the uncertainties exist in weather (e.g., temperature, solar radiation), and noises in measurement (e.g., sensor, meter) are considered. In the future, uncertainties in the model itself (e.g., thermal storage, battery) will be incorporated to the calibration model. Different time scales for each stage in the adaptive decision framework will be investigated. Some uncertainty quantification techniques will be employed to accurately derive the parameters for the random distributions considered in the calibration stage.

Chapter 8

CONCLUSIONS AND FUTURE RESEARCH

Two national initiatives (smart building and smart grid) urge the building industry to improve their energy efficiency and to have better capabilities to interact with the power grid. These initiatives have also driven research moving from centralized operation decisions on a single building to decentralized decisions on a group of buildings, termed building cluster which shares energy resources locally and globally. This dissertation envisions the next generation buildings could be interconnected physically or virtually, and they could share energy resources and freely exchange information. Several research issues are identified: 1) *what is the appropriate building model used to derive decision model for operation strategy identification?* 2) *What is a computationally efficient algorithm for building operation decision?* 3) *How to derive decisions for multiple building operation decisions?* 4) *How to handle dynamics, uncertainty and noise exist in the buildings and environment to guarantee the building could respond to the dynamic environment?*

In order to address these issues, an adaptive decision framework is developed to derive adaptive operation decisions for the building cluster. In chapter 2, an agent based building model is developed to obtain decision formulation, and Pareto operation decisions for the building cluster are derived using a multi-objective Memetic algorithm. The high computational cost for the multi-objective Memetic algorithm drives me to develop a computationally efficient algorithm. In chapter 3, a computationally efficient particle swarm optimization algorithm is developed which performs well on a diverse set of problems (e.g., uni-modal, multi-modal, shifted, rotated, noisy, mis-scaled). The augmented PSO with multiple adaptive methods (PSO-MAM) significantly outperforms the 10 existing published PSO algorithms on 36 out of 43 test functions. In chapter 4, an adaptive parameter tuning mechanism is developed to adaptively change the parameter settings for the PSO algorithm. The newly developed PSO algorithm which is termed as bi-local searches and mutation based adaptive particle swarm optimization (BLOSSM-APSO) is demonstrated to be robust to its parameter settings. In order to apply the PSO algorithm for building cluster operation decisions, the BLOSSM-APSO is extended to a multi-objective optimization (MOO) algorithm in chapter 5. The augmented PSO algorithm for MOO, termed AMOPSO is demonstrated to significantly outperform the existing 4 multi-objective PSOs and 3 Mulit-objective evolutionary algorithms and moderately outperform MOCell (cellular genetic algorithm). Based on the AMOPSO developed in chapter 5, a decision framework in chapter 6 is developed to obtain Pareto operation decisions for the building cluster. The AMOPSO based decision framework is demonstrated to be able to obtain hourly operation decisions and could achieve more cost savings for the building cluster. By considering uncertainties exist in the building systems and environment, noises exist in sensors and meters, chapter 7 integrates the Gaussian mixture sigma point particle filter (GMSPPF) algorithm with the multi-objective PSO decision framework to calibrate the building model with online measurement data. The calibrated high-fidelity model is then used for the AMOPSO based decision framework. The adaptive decision framework is able to accurately calibrate the building model and reduce energy cost. The operation strategies derived from the adaptive decision framework enable the buildings to respond to their dynamic environment.

In the future, more complex dynamic pricing model based on the demand will be developed and operation decisions by incorporating this pricing model into the decision model will be studied. Other than the uncertainties in weather (e.g., temperature, solar radiation), the uncertainties exist in the building model itself (e.g., thermal storage model, battery model) will be considered in the calibration stage. In addition, some uncertainty quantification techniques will be employed to accurately derive the parameters for the random distributions considered in the calibration stage. Furthermore, after the operation strategies for each decision stage are derived, it is necessary to develop an effective and smoothness transition trajectory from one time to the followed time since the trajectory for changing has direct implications on the system performance and the amount of time for each change to be completed.

From algorithm development perspective, several future efforts are needed. First, instead of fusing other techniques with PSO, the advantages of other search techniques (e.g., gradient method, non-uniform mutation-based method) will be incorporated into the velocity update equation of PSO to improve PSO's performance. Secondly, an intelligent strategy should be explored to determine when the adaptive parameter tuning will be needed. Last but not least, same as the mutation operator is employed to prevent premature convergence in the PSO, some equivalent "mutation" techniques could be investigated to prevent the rapid degeneracy problem in the particle filter.

REFERENCES

Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research, 36*(5), 1693-1702.

Al-Homoud, M. S. (2001). Computer-aided building energy analysis techniques. *Building and Environment, 36*(4), 421-433.

Allahverdi, A., & Al-Anzi, F. S. (2006). A PSO and a Tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application. *Computers & Operations Research, 33*(4), 1056-1080.

Alvarez-Benitez, J., Everson, R., & Fieldsend, J. (2005). A MOPSO Algorithm Based Exclusively on Pareto Dominance Concepts. In C. Coello Coello, A. Hernández Aguirre & E. Zitzler (Eds.), *Evolutionary Multi-Criterion Optimization* (Vol. 3410, pp. 459-473): Springer Berlin / Heidelberg.

Andrews, P. S. (2006). *An Investigation into Mutation Operators for Particle Swarm Optimization*. Paper presented at the Evolutionary Computation, 2006. CEC 2006. IEEE Congress on.

Angeline, P. J. (1998). *Using selection to improve particle swarm optimization*. Paper presented at the Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on.

Anghinolfi, D., Montemanni, R., Paolucci, M., & Maria Gambardella, L. (2011). A hybrid particle swarm optimization approach for the sequential ordering problem. *Computers & Operations Research, 38*(7), 1076-1085.

Arroyo, J. E. C., & Armentano, V. A. (2005). Genetic local search for multi-objective flowshop scheduling problems. *European Journal of Operational Research, 167*(3), 717-738.

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing, 50*(2), 174-188.

Arun, P., Banerjee, R., & Bandyopadhyay, S. (2009). Optimum design of battery-integrated diesel generator systems incorporating demand uncertainty. *Industrial & Engineering Chemistry Research, 48*(10), 4908-4916.

Auger, A., & Hansen, N. (2005). *Performance evaluation of an advanced local search evolutionary algorithm*. Paper presented at the Evolutionary Computation, 2005. The 2005 IEEE Congress on.

Aydinalp, M., Ismet Ugursal, V., & Fung, A. S. (2004). Modeling of the space and domestic hot-water heating energy-consumption in the residential sector using neural networks. *Applied Energy, 79*(2), 159-178.

Banks, A., Vincent, J., & Anyakoha, C. (2007). A review of particle swarm optimization. Part I: background and development. *Natural Computing, 6*(4), 467-484.

Bartz-Beielstein, T., Limbourg, P., Mehnen, J., Schmitt, K., Parsopoulos, K. E., & Vrahatis, M. N. (2003). *Particle swarm optimizers for Pareto optimization with enhanced archiving techniques*. Paper presented at the Evolutionary Computation, 2003. CEC '03. The 2003 Congress on.

Boyd, S. (2010). EE364b Course Notes: Sub-Gradient Methods, June 17, 2011

Braun, J. E. (1990). Reducing energy costs and peak electrical demand through optimal control of building thermal mass. *ASHRAE Transactions, 96*(2), 876-888.

Braun, J. E. (2003). Load control using building thermal mass. *Journal of Solar Energy Engineering, 125*(3), 292-301.

Braun, J. E. (2007). A near-optimal control strategy for cool storage systems with dynamic electric rates. *HVAC&R Research, 13*(4), 557-580.

Braun, J. E., & Chaturvedi, N. (2002). An Inverse Grey-box Model for Transient Building Load Prediction. *HVAC&R Research, 8*(1), 73-99.

Braun, J. E., Montgomery, K. W., & Chaturvedi, N. (2001). Evaluating the Performance of Building Thermal Mass Control Strategies. *HVAC&R Research, 7*(4), 403-428.

Cai, Y. P., Huang, G. H., Yang, Z. F., Lin, Q. G., & Tan, Q. (2009). Community-scale renewable energy systems planning under uncertainty—An interval chance-constrained programming approach. *Renewable and Sustainable Energy Reviews, 13*(4), 721-735.

Carpenter, J., Clifford, P., & Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proceedings -, 146*(1), 2-7.

Chatterjee, A., & Siarry, P. (2006). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers & Operations Research, 33*(3), 859-871.

Chen, H., Chou, P., Duri, S., Lei, H., & Reason, J. (2009). *The Design and Implementation of a Smart Building Control System*. Paper presented at the e-Business Engineering, 2009. ICEBE '09. IEEE International Conference on.

Chen, T. Y. (2001). Real-time predictive supervisory operation of building thermal systems with thermal mass. *Energy and Buildings, 33*(2), 141-150.

Chen, Y.-P., Peng, W.-C., & Jian, M.-C. (2007). Particle Swarm Optimization With Recombination and Dynamic Linkage Discovery. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 37*(6), 1460-1470.

Chen, Z. (2003). Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond: Technique Report, Adaptive System Lab., McMaster University

Clerc, M., & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58-73.

Coello Coello, C. A., & Lechuga, M. S. (2002). *MOPSO: a proposal for multiple objective particle swarm optimization*. Paper presented at the Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on.

Coello Coello, C. A., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 256-279.

Corrado, V., Fabrizio, E., & Filippi, M. (2007). *Modelling and optimization of multi-energy source building systems in the design concept phase*. Paper presented at the Proceedings of Clima 2007 WellBeing Indoors International Conference, Helsinki.

Daneshyari, M., & Yen, G. G. (2011). Cultural-Based Multiobjective Particle Swarm Optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 41*(2), 553-567.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182-197.

Dong, B., Cao, C., & Lee, S. E. (2005). Applying support vector machines to predict building energy consumption in tropical region. *Energy and Buildings, 37*(5), 545-553.

Drees, K. H., & Braun, J. E. (1996). Development and evaluation of a rule-based control strategy for ice storage systems. *HVAC&R Research, 2*(4), 312-336.

Durillo, J. J., & Nebro, A. J. (2011). jMetal: A Framework for Multi-Objective Optimization. Retrieved from http://jmetal.sourceforge.net/problems.html on Sept. 1, 2011

Eberhart, R., & Kennedy, J. (1995). *A new optimizer using particle swarm theory*. Paper presented at the Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on.

Eberhart, R., & Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. In V. Porto, N. Saravanan, D. Waagen & A. Eiben (Eds.), *Evolutionary Programming VII* (Vol. 1447, pp. 611-616): Springer Berlin / Heidelberg.

Eberhart, R. C., & Shi, Y. (2001). *Tracking and optimizing dynamic systems with particle swarms*. Paper presented at the Evolutionary Computation, 2001. Proceedings of the 2001 Congress on.

Ekici, B. B., & Aksoy, U. T. (2009). Prediction of building energy consumption by using artificial neural networks. *Advances in Engineering Software, 40*(5), 356-362.

El-shatter, T. F., Eskander, M. N., & El-Hagry, M. T. (2006). Energy flow and management of a hybrid wind/PV/fuel cell generation system. *Energy Conversion and Management, 47*(9-10), 1264-1280.

Elbeltagi, E., Hegazy, T., & Grierson, D. (2005). Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics, 19*(1), 43-53.

Engelbrecht, A. P. (2006). *Fundamentals of Computational Swarm Intelligence*: John Wiley & Sons.

Evensen, G. (1992). Using the Extended Kalman Filter with a Multilayer Quasi-Geostrophic Ocean Model. *Journal of Geophysical Research, 97*(C11), 17905-17924.

Evensen, G. (2003). The Ensemble Kalman Filter: theoretical formulation and practical implementation. *Ocean Dynamics, 53*(4), 343-367.

Fan, S., & Zahara, E. (2007). A hybrid simplex search and particle swarm optimization for unconstrained optimization. *European Journal of Operational Research, 181*(2), 527-548.

Fernandez Martınez, J. L., & Garcıa Gonzalo, E. (2008). The generalized PSO: a new door to PSO evolution. *J. Artif. Evol. App., 2008*, 1-15.

Fieldsend, J. E. (2004). Multi-objective particle swarm optimisation methods: Department of Computer Science, University of Exeter.

Fieldsend, J. E., & Singh, S. (2002). *A multiobjective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence*. Paper presented at the Proceedings of the 2002 U.K. Workshop on Computational Intelligence, Birmingham, UK.

Fleten, S.-E., & Kristoffersen, T. K. (2008). Short-term hydropower production planning by stochastic programming. *Computers & Operations Research, 35*(8), 2656-2671.

Fong, K., Hanby, V., & Chow, T. (2006). HVAC system optimization for energy management by evolutionary programming. *Energy and Buildings, 38*(3), 220-231.

Friedman, H. (2009). Wiring the smart grid for energy savings: integrating buildings to maximize investment. Portland, Oregon: Portland Energy Conservation Inc. (PECI.).

Garc ía-Gonz ález, J., Parrilla, E., & Mateo, A. (2007). Risk-averse profit-based optimal scheduling of a hydro-chain in the day-ahead electricity market. *European Journal of Operational Research, 181*(3), 1354-1369.

Garcia-Villoria, A., & Pastor, R. (2009). Introducing dynamic diversity into a discrete particle swarm optimization. *Computers & Operations Research, 36*(3), 951-966.

Goh, C. K., & Tan, K. C. (2007). An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation, 11*(3), 354-381.

Goh, C. K., Tan, K. C., Liu, D. S., & Chiam, S. C. (2010). A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design. *European Journal of Operational Research, 202*(1), 42-54.

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F, 140*(2), 107-113.

Hämäläinen, R. P., & Mäntysaari, J. (2002). Dynamic multi-objective heating optimization. *European Journal of Operational Research, 142*(1), 1-15.

Hassan, R., Cohanim, B., de-Weck, O., & Venter, G. (2005). *A comparison of particle swarm optimization and the genetic algorithm*. Paper presented at the 1st AIAA multidisciplinary design optimization specialist conference. No. AIAA-2005-1897, Austin, TX.

Henze, G. P. (2003a). An overview of optimal control for central cooling plants with ice thermal energy storage. *Journal of Solar Energy Engineering, 125*(3), 302.

Henze, G. P. (2003b). Impact of real-time pricing rate uncertainty on the annual performance of cool storage systems. *Energy and Buildings, 35*(3), 313-325.

Henze, G. P. (2004). Evaluation of optimal control for active and passive building thermal storage. *International Journal of Thermal Sciences, 43*(2), 173-183.

Henze, G. P., & Dodier, R. H. (2003). Adaptive optimal control of a grid-independent photovoltaic system. *Journal of Solar Energy Engineering, 125*(1), 34-42.

Henze, G. P., Dodier, R. H., & Krarti, M. (1997). Development of a Predictive Optimal Controller for Thermal Energy Storage Systems. *HVAC&R Research, 3*(3), 233-264.

Henze, G. P., Felsmann, C., Kalz, D. E., & Knabe, G. (2004). Impact of Forecasting Accuracy on Predictive Optimal Control of Active and Passive Building Thermal Storage Inventory. [Article]. *HVAC&R Research, 10*(2), 153-178.

Henze, G. P., Florita, A. R., Brandemuehl, M. J., Felsmann, C., & Cheng, H. (2010). Advances in Near-Optimal Control of Passive Building Thermal Storage. *Journal of Solar Energy Engineering, 132*(2), 021009-021009.

Henze, G. P., Kalz, D. E., Liu, S., & Felsmann, C. (2005). Experimental analysis of model-based predictive optimal control for active and passive building thermal storage inventory. *HVAC&R Research, 11*(2), 189-213.

Henze, G. P., Krarti, M., & Brandemuehl, M. J. (2003). Guidelines for improved performance of ice storage systems. *Energy and Buildings, 35*(2), 111-127.

Henze, G. P., & Schoenmann, J. (2003). Evaluation of reinforcement learning control for thermal energy storage systems. *HVAC&R Research, 9*, 259-275.

Hicks, T. W., & von Neida, B. (2000). *An Evaluation of America's First Energy Star Buildings: The Class of 1999*. Paper presented at the ACEEE 2000 Summer Study on Energy Efficiency in Buildings Proceedings, Washington, D.C.

Higashi, N., & Iba, H. (2003). *Particle swarm optimization with Gaussian mutation*. Paper presented at the Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE.

Ho, S. L., Yang, S., Ni, G., Lo, E. W. C., & Wong, H. C. (2005). A particle swarm optimization-based method for multiobjective design optimizations. *IEEE Transactions on Magnetics, 41*(5), 1756-1759.

Hoffmann, T. (2009). White paper: Smart Buildings: Johnson Controls, Inc.

Hu, M., Weir, J. D., & Wu, T. (2012). Decentralized operation strategies for an integrated building energy system using a memetic algorithm. *European Journal of Operational Research, 217*, 185-197.

Hu, M., Wen, J., Li, F., Haghnevis, M., Khodadadegan, Y., Sanchez, L. M., . . . Wu, T. (2010). *An agent based simulation for building energy system modeling*. Paper presented at the Proceeding of 2010 ASME Dynamic Systems and Control Conference, Cambridge, Massachusetts.

Hu, X., Eberhart, R. C., & Shi, Y. (2003). *Particle swarm with extended memory for multiobjective optimization*. Paper presented at the Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE.

Huang, H., Qin, H., Hao, Z., & Lim, A. (2010). Example-based learning particle swarm optimization for continuous optimization. *Information Sciences*.

Huang, V. L., Suganthan, P. N., & Liang, J. J. (2006). Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems. *International Journal of Intelligent Systems, 21*(2), 209-226.

Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation, 10*(5), 477-506.

Ito, K., & Xiong, K. (2000). Gaussian filters for nonlinear filtering problems. *Automatic Control, IEEE Transactions on, 45*(5), 910-927.

Iwasaki, N., Yasuda, K., & Ueno, G. (2006). Dynamic parameter tuning of particle swarm optimization. *IEEJ Transactions on Electrical and Electronic Engineering, 1*(4), 353-363.

Janson, S., & Merkle, D. (2005). A New Multi-objective Particle Swarm Optimization Algorithm Using Clustering Applied to Automated Docking Hybrid

Metaheuristics. In M. Blesa, C. Blum, A. Roli & M. Sampels (Eds.), (Vol. 3636, pp. 902-902): Springer Berlin / Heidelberg.

Janson, S., & Middendorf, M. (2005). A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 35*(6), 1272-1282.

Jiang, M., Luo, Y. P., & Yang, S. Y. (2007). Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters, 102*(1), 8-16.

Jiao, B., Lian, Z., & Gu, X. (2008). A dynamic inertia weight particle swarm optimization algorithm. *Chaos, Solitons & Fractals, 37*(3), 698-705.

Juang, Y.-T., Tung, S.-L., & Chiu, H.-C. (2010). Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions. *Information Sciences, In Press, Corrected Proof*.

Julier, S. J., Uhlmann, J. K., & Durrant-Whyte, H. F. (1995). *A new approach for filtering nonlinear systems*. Paper presented at the American Control Conference, 1995. Proceedings of the.

Julier, S. J., & Uhlmman, J. (1997). *A New Extension of the Kalman Filter to Nonlinear Systems*. Paper presented at the Int. Symp. Aerospace/Defense Sensing, Simul. and Controls.

Kadirkamanathan, V., Selvarajah, K., & Fleming, P. J. (2006). Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation, 10*(3), 245-255.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering, 82*, 35-45.

Kao, Y.-T., & Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing, 8*(2), 849-857.

Katipamula, S., & Lu, N. (2006). Evaluation of residential HVAC control strategies for demand response programs. *ASHRAE Transactions, 112*(1), 535-546.

Keeney, K., & Braun, J. E. (1996). A simplified method for determining optimal cooling control strategies for thermal storage in building mass. *HVAC&R Research, 2*(1), 59-78.

Kennedy, J. (1999). *Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance*. Paper presented at the Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.

Kennedy, J., & Eberhart, R. (1995). *Particle swarm optimization*. Paper presented at the Neural Networks, 1995. Proceedings., IEEE International Conference on.

Kennedy, J., & Eberhart, R. (2001). *Swarm Intelligence*. San Francisco, California: Morgan Kaufmann Publishers.

Kennedy, J., & Mendes, R. (2002). *Population structure and particle swarm performance*. Paper presented at the Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on.

Kennedy, J., & Spears, W. M. (1998). *Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator*. Paper presented at the Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on.

Kiesling, L. L. (2009). Markets, technology and institutions: increasing energy efficiency through decentralized coordination: EcoAlign Project Energy Code, series #2.

Kjeldsen, N. H., & Chiarandini, M. (2012). Heuristic solutions to the long-term unit commitment problem with cogeneration plants. *Computers & Operations Research, 39*(2), 269-282.

Kleissl, J., & Agarwal, Y. (2010). *Cyber-physical energy systems: focus on smart buildings*. Paper presented at the Proceedings of the 47th Design Automation Conference, Anaheim, California.

Knowles, J., & Corne, D. (1999). *The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation*. Paper presented at the Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.

Kosny, J., Petrie, T., Gawin, D., Childs, P., Desjarlais, A., & Christian, J. (2001). Thermal mass - energy savings potential in residential buildings Retrieved from http://www.ornl.gov/sci/roofs+walls/research/detailed_papers/thermal/index.html on April 14, 2011

Lee, W.-S., Chen, Y. T., & Wu, T.-H. (2009). Optimization for ice-storage air-conditioning system using particle swarm algorithm. *Applied Energy, 86*(9), 1589-1595.

Leong, W.-F., & Yen, G. G. (2008). PSO-Based Multiobjective Optimization With Dynamic Population Size and Adaptive Local Archives. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 38*(5), 1270-1293.

Lewis, K., & Mistree, F. (1998). Collaborative, sequential, and isolated decisions in design. *Journal of Mechanical Design, 120*(4), 643-652.

Li, X. (2003). A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization Genetic and Evolutionary Computation — GECCO 2003. In E. Cantú-Paz, J. Foster, K. Deb, L. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. Potter, A. Schultz, K. Dowsland, N. Jonoska & J. Miller (Eds.), (Vol. 2723, pp. 198-198): Springer Berlin / Heidelberg.

Lian, Z. (2010). A united search particle swarm optimization algorithm for multiobjective scheduling problem. *Applied Mathematical Modelling, 34*(11), 3518-3526.

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation, 10*(3), 281-295.

Liang, J. J., & Suganthan, P. N. (2005). *Dynamic multi-swarm particle swarm optimizer*. Paper presented at the Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE.

Liao, C.-J., Tseng, C.-T., & Luarn, P. (2007). A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers & Operations Research, 34*(10), 3099-3111.

Liu, D. S., Tan, K. C., Goh, C. K., & Ho, W. K. (2007). A Multiobjective Memetic Algorithm Based on Particle Swarm Optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 37*(1), 42-50.

Liu, D. S., Tan, K. C., Huang, S. Y., Goh, C. K., & Ho, W. K. (2008). On solving multiobjective bin packing problems using evolutionary particle swarm optimization. *European Journal of Operational Research, 190*(2), 357-382.

Liu, S., & Henze, G. P. (2004). Impact of Modeling Accuracy on Predictive Optimal Control of Active and Passive Building Thermal Storage Inventory. [Article]. *ASHRAE Transactions, 110*(1), 151-163.

Liu, S., & Henze, G. P. (2006a). Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory part 1: theoretical foundation. *Energy and Buildings, 38*(2), 142-147.

Liu, S., & Henze, G. P. (2006b). Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory part 2: results and analysis. *Energy and Buildings, 38*(2), 148-161.

Liu, S., & Henze, G. P. (2007). Evaluation of reinforcement learning for optimal control of building active and passive thermal storage inventory. *Journal of Solar Energy Engineering, 129*(2), 215-225.

Livengood, D., & Larson, R. (2009). The Energy Box: Locally Automated Optimal Control of Residential Electricity Usage. *Service Science, 1*(1), 1-16.

Lorenzo, E. (2003). Energy collected and delivered by PV modules *Handbook of Photovoltaic Science and Engineering* (pp. 905-970). New York, NY: John Wiley & Sons Ltd.

Loukil, T., Teghem, J., & Fortemps, P. (2007). A multi-objective production scheduling case study solved by simulated annealing. *European Journal of Operational Research, 179*(3), 709-722.

Lozano, M., Herrera, F., Krasnogor, N., & Molina, D. (2004). Real-coded memetic algorithms with crossover hill-climbing. *Evolutionary Computation, 12*(3), 273-302.

Lu, L. (2004). *Investigation on characteristics and application of hybrid solar-wind power generation systems.* Hong Kong Polytechnic University (Hong Kong) Ph.D., Hong Kong Polytechnic University (Hong Kong), Hong Kong.

Lu, L., Cai, W., Xie, L., Li, S., & Soh, Y. C. (2005). HVAC system optimization - in-building section. *Energy and Buildings, 37*(1), 11-22.

Lu, L., Yang, H., & Burnett, J. (2002). Investigation on wind power potential on Hong Kong islands--an analysis of wind power and wind turbine characteristics. *Renewable Energy, 27*(1), 1-12.

Lu, L., & Yang, H. X. (2004). A study on simulations of the power output and practical models for building integrated photovoltaic systems. *Journal of Solar Energy Engineering, 126*(3), 929-935.

Manolakos, D., Papadakis, G., Papantonis, D., & Kyritsis, S. (2001). A simulation-optimisation programme for designing hybrid energy systems for supplying electricity and fresh water through desalination to remote areas: case study: the Merssini village, Donoussa island, Aegean Sea, Greece. *Energy, 26*(7), 679-704.

Marinakis, Y., & Marinaki, M. (2010). A Hybrid Multi-Swarm Particle Swarm Optimization algorithm for the Probabilistic Traveling Salesman Problem. *Computers & Operations Research, 37*(3), 432-442.

Martin, N., & Ruiz, J. M. (2001). Calculation of the PV modules angular losses under field conditions by means of an analytical model. *Solar Energy Materials and Solar Cells, 70*(1), 25-38.

Mavrotas, G., Diakoulaki, D., Florios, K., & Georgiou, P. (2008). A mathematical programming framework for energy planning in services' sector buildings under uncertainty in load demand: The case of a hospital in Athens. *Energy Policy, 36*(7), 2415-2429.

Mefti, A. (2003). Generation of hourly solar radiation for inclined surfaces using monthly mean sunshine duration in Algeria. *Energy Conversion and Management, 44*(19), 3125-3141.

Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation, 8*(3), 204-210.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs (3rd ed.).* London, UK: Springer-Verlag.

Mihalakakou, G., Santamouris, M., & Tsangrassoulis, A. (2002). On the energy consumption in residential buildings. *Energy and Buildings, 34*(7), 727-736.

Mohammad Nezhad, A., & Mahlooji, H. (2011). A revised particle swarm optimization based discrete Lagrange multipliers method for nonlinear programming problems. *Computers & Operations Research, 38*(8), 1164-1174.

Mostaghim, S., & Teich, J. (2003a). *Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)*. Paper presented at the Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE.

Mostaghim, S., & Teich, J. (2003b). *The Role of e-dominance in Multi Objective Particle Swarm Optimization Methods*. Paper presented at the Congress on Evolutionary Computation, Canberra, Australia.

Mostaghim, S., & Teich, J. (2004). *Covering Pareto-optimal fronts by subswarms in multi-objective particle swarm optimization*. Paper presented at the Evolutionary Computation, 2004. CEC2004. Congress on.

Nassif, N., Kajl, S., & Sabourin, R. (2005). Optimization of HVAC control system strategy using two-objective genetic algorithm. [Article]. *HVAC&R Research, 11*(3), 459-486.

NCDC. (2010). Quality controlled local climatological data Retrieved Sept. 15, 2010, from http://cdo.ncdc.noaa.gov/qclcd/QCLCD

Nebro, A. J., Durillo, J. J., Garcia-Nieto, J., Coello Coello, C. A., Luna, F., & Alba, E. (2009a). *SMPSO: A new PSO-based metaheuristic for multi-objective optimization*. Paper presented at the Computational intelligence in miulti-criteria decision-making, 2009. mcdm '09. ieee symposium on.

Nebro, A. J., Durillo, J. J., Luna, F., Dorronsoro, B., & Alba, E. (2009b). MOCell: A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems, 24*(7), 726-746.

Nørgaard, M., Poulsen, N. K., & Ravn, O. (2000). New developments in state estimation for nonlinear systems. *Automatica, 36*(11), 1627-1638.

NREL. (2010). National solar radiation data base Retrieved Sept. 15, 2010, from http://www.nrel.gov/midc/pfci/

Omkar, S., Mudigere, D., Naik, G., & Gopalakrishnan, S. (2008). Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures. *Computers & Structures, 86*(1-2), 1-14.

Ozcan, E., & Mohan, C. K. (1998). Analysis of a simple particle swarm optimization system. *Intelligent Engineering Systems through Artificial Neural Networks*, 253-258.

Ozcan, E., & Mohan, C. K. (1999). *Particle swarm optimization: surfing the waves*. Paper presented at the Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.

Ozturk, H. K., Canyurt, O. E., Hepbasli, A., & Utlu, Z. (2004). Residential-commercial energy input estimation based on genetic algorithm (GA) approaches: an application of Turkey. *Energy and Buildings, 36*(2), 175-183.

Pan, Y., Huang, Z., & Wu, G. (2007). Calibrated building energy simulation and its application in a high-rise commercial building in Shanghai. *Energy and Buildings, 39*(6), 651-657.

Parks, N. (2009). Energy efficiency and the smart grid. *Environmental Science & Technology, 43*(9), 2999-3000.

Parsopoulos, K. E., & Vrahatis, M. N. (2004) UPSO - A unified particle swarm optimization scheme. (pp. 868-873): Lecture Series on Computational Sciences.

Peram, T., Veeramachaneni, K., & Mohan, C. K. (2003). *Fitness-distance-ratio based particle swarm optimization*. Paper presented at the Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE.

Perez, S., & Farnham, L. (2010). Building Efficiency Technology Overview: Johnson Control, Inc.

Petalas, Y. G., Parsopoulos, K. E., & Vrahatis, M. N. (2007). Memetic particle swarm optimization. *Annals of Operations Research, 156*(1), 99-127.

Plevris, V., & Papadrakakis, M. (2010). A Hybrid Particle Swarm-Gradient Algorithm for Global Structural Optimization. *Computer-Aided Civil and Infrastructure Engineering*.

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization An overview. *Swarm Intelligence, 1*(1), 33-57.

Pulido, G., & Coello Coello, C. (2004). Using Clustering Techniques to Improve the Performance of a Multi-objective Particle Swarm Optimizer Genetic and Evolutionary Computation – GECCO 2004. In K. Deb (Ed.), (Vol. 3102, pp. 225-237): Springer Berlin / Heidelberg.

Raquel, C. R., & Naval Jr., P. C. (2005). *An effective use of crowding distance in multiobjective particle swarm optimization*. Paper presented at the Proceedings of the 2005 conference on Genetic and evolutionary computation, Washington DC, USA.

Ratnaweera, A., Halgamuge, S. K., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation, 8*(3), 240-255.

Ray, T., & Liew, K. M. (2002). A swarm metaphor for multiobjective design optimization. *Engineering Optimization, 34*(2), 141-153.

Reyes-Sierra, M., & Coello Coello, C. A. (2005). Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ∈-Dominance. In C. Coello Coello, A. Hernández Aguirre & E. Zitzler (Eds.), *Evolutionary Multi-Criterion Optimization* (Vol. 3410, pp. 505-519): Springer Berlin / Heidelberg.

Reyes-Sierra, M., & Coello Coello, C. A. (2006). Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *International Journal of Computational Intelligence Research, 2*(3), 287-308.

Rong, A., Hakonen, H., & Lahdelma, R. (2008a). A variant of the dynamic programming algorithm for unit commitment of combined heat and power systems. *European Journal of Operational Research, 190*(3), 741-755.

Rong, A., & Lahdelma, R. (2007). An efficient envelope-based branch and bound algorithm for non-convex combined heat and power production planning. *European Journal of Operational Research, 183*(1), 412-431.

Rong, A., Lahdelma, R., & Luh, P. B. (2008b). Lagrangian relaxation based algorithm for trigeneration planning with storages. *European Journal of Operational Research, 188*(1), 240-257.

Salazar-Lechuga, M., & Rowe, J. E. (2005). *Particle swarm optimization and fitness sharing to solve multi-objective optimization problems*. Paper presented at the Evolutionary Computation, 2005. The 2005 IEEE Congress on.

Salomon, R. (1996). Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms. *Biosystems, 39*(3), 263-278.

Shi, Y., & Eberhart, R. (1998). *A modified particle swarm optimizer*. Paper presented at the Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on.

Shi, Y., & Eberhart, R. C. (1999). *Empirical study of particle swarm optimization*. Paper presented at the Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.

Shi, Y., & Eberhart, R. C. (2001). *Fuzzy adaptive particle swarm optimization*. Paper presented at the Evolutionary Computation, 2001. Proceedings of the 2001 Congress on.

Shi, Y., Liu, H., Gao, L., & Zhang, G. (2010). Cellular particle swarm optimization. *Information Sciences*.

Shu, J., & Li, J. (2009). *An Improved Self-Adaptive Particle Swarm Optimization Algorithm with Simulated Annealing*. Paper presented at the 2009 Third International Symposium on Intelligent Information Technology Application.

SIEMENS. (2009). Smart grids make it possible for building operators to actively participate in the electricity market. Barcelona, Spain: Siemens Industry Sector.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *Automatic Control, IEEE Transactions on, 37*(3), 332-341.

Srinivasan, D., & Seow, T. H. (2003). *Particle swarm inspired evolutionary algorithm (PS-EA) for multiobjective optimization problems*. Paper presented at the Evolutionary Computation, 2003. CEC '03. The 2003 Congress on.

SRP. (2010). SRP price plan Retrieved on Sept. 15, 2010, from http://www.srpnet.com/menu/PayBillPrice.aspx?res

Sun, C., Temple, K., Rossi, T., & Braun, J. E. (2006). Interaction between dynamic electric rates and thermal energy storage control: final report for RP-1252. Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

Thangaraj, R., Pant, M., & Abraham, A. (2009). *A new diversity guided particle swarm optimization with mutation*. Paper presented at the Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on.

Torcellini, P., Pless, S., Deru, M., & Crawley, D. (2006). Zero energy buildings: a critical look at the definition. Pacific Grove, California: NREL Report No. CP-550-39833.

Tripathi, P., Bandyopadhyay, S., & Pal, S. (2007). Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients. *Information Sciences, 177*(22), 5033-5049.

Valenzuela, J., Mazumdar, M., & Kapoor, A. (2000). Influence of temperature and load forecast uncertainty on estimates of power generation production costs. *IEEE Transactions on Power System, 15*(2), 668-674.

van den Bergh, F. (2002). *An analysis of particle swarm optimizers.* Doctoral, University of Pretoria, Pretoria, South Africa.

van den Bergh, F., & Engelbrecht, A. P. (2004). A Cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 225-239.

van der Merwe, R. (2004). *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models.* Oregon Health & Science University Ph.D., Oregon Health & Science University, United States -- Oregon.

Villalobos-Arias, M. A., Pulido, G. T., & Coello Coello, C. A. (2005). *A proposal to use stripes to maintain diversity in a multi-objective particle swarm optimizer*. Paper presented at the Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE.

Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., & Ventresca, M. (2011). Enhancing particle swarm optimization using generalized opposition-based learning. *Information Sciences*.

Wang, Y., Li, B., Weise, T., Wang, J., Yuan, B., & Tian, Q. (2010). Self-adaptive learning based particle swarm optimization. *Information Sciences*.

Wang, Y., & Yang, Y. (2009). Particle swarm optimization with preference order ranking for multi-objective optimization. *Information Sciences, 179*(12), 1944-1959.

Wang, Y., & Yang, Y. (2010). Particle swarm with equilibrium strategy of selection for multi-objective optimization. *European Journal of Operational Research, 200*(1), 187-197.

Welch, G., & Bishop, G. (1995). An Introduction to the Kalman Filter: Department of Computer Science, University of North Carolina at Chapel Hill.

Wen, J. (2003). *Development and validation of adaptive optimal operation methodology for building HVAC systems.* The University of Iowa Ph.D., The University of Iowa, United States -- Iowa.

West, J. D., & Braun, J. E. (1999). Modeling partial charging and discharging of area-constrained ice storage tanks. *HVAC&R Research, 5*(3), 209-228.

Wright, J. A., Loosemore, H. A., & Farmani, R. (2002). Optimization of building thermal design and control by multi-criterion genetic algorithm. *Energy and Buildings, 34*(9), 959-972.

Yamaguchi, T., & Yasuda, K. (2006). *Adaptive Particle Swarm Optimization; Self-coordinating Mechanism with Updating Information*. Paper presented at the Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on.

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*(2), 82-102.

Yapicioglu, H., Smith, A. E., & Dozier, G. (2007). Solving the semi-desirable facility location problem using bi-objective particle swarm. *European Journal of Operational Research, 177*(2), 733-749.

Yen, G. G., & Leong, W. F. (2009). Dynamic Multiple Swarms in Multiobjective Particle Swarm Optimization. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 39*(4), 890-911.

Youcef Ettoumi, F., Mefti, A., Adane, A., & Bouroubi, M. Y. (2002). Statistical analysis of solar measurements in Algeria using beta distributions. *Renewable Energy, 26*(1), 47-67.

Zhan, Z.-H., Zhang, J., Li, Y., & Chung, H. S. H. (2009). Adaptive Particle Swarm Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39*(6), 1362-1381.

Zhan, Z. H., Zhang, J., Li, Y., & Shi, Y. H. (2010). Orthogonal Learning Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation, PP*(99), 1-1.

Zhao, S.-Z., & Suganthan, P. N. (2011). Two-lbests based multi-objective particle swarm optimizer. *Engineering Optimization, 43*, 1-17.

Zhao, X. (2011). Simulated annealing algorithm with adaptive neighborhood. *Applied Soft Computing, 11*(2), 1827-1836.

Zhao, X., Gao, X.-S., & Hu, Z.-C. (2007). Evolutionary programming based on non-uniform mutation. *Applied Mathematics and Computation, 192*(1), 1-11.

Zhou, G., Krarti, M., & Henze, G. P. (2005). Parametric analysis of active and passive building thermal storage utilization. *Journal of Solar Energy Engineering, 127*(1), 37-46.

Zhou, Q., Wang, S., Xu, X., & Xiao, F. (2008). A grey-box model of next-day building thermal load prediction for energy-efficient control. *International Journal of Energy Research, 32*(15), 1418-1431.

Zielinski, K., & Laur, R. (2007). *Adaptive parameter setting for a multi-objective particle swarm optimization algorithm*. Paper presented at the Evolutionary Computation, 2007. CEC 2007. IEEE Congress on.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). *SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization*. Paper presented at the Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, Athens, Greece.

APPENDIX A

TEST FUNCTIONS FOR PARTICLE SWARM OPTIMIZATION

1) Sphere function

$$F_{Sphere}\left(\mathbf{z}\right)=\sum\nolimits_{i=1}^{D} z_i^2$$

2) Schwefel P2.22 function

$$F_{Schwefel\,2.22}\left(\mathbf{z}\right)=\sum\nolimits_{i=1}^{D}\left|z_i\right|+\prod\nolimits_{i=1}^{D}\left|z_i\right|$$

3) Schwefel P1.2 function

$$F_{Schwefel1.2}\left(\mathbf{z}\right)=\sum\nolimits_{i=1}^{D}\left(\sum\nolimits_{j=1}^{i} z_j\right)^2$$

4) Schwefel P2.21 function

$$F_{Schwefel\,2.21}\left(\mathbf{z}\right)=\max_{i=1,\ldots,D}\left|z_i\right|$$

5) Rosenbrock function

$$F_{Rosenbrock}\left(\mathbf{z}\right)=\sum\nolimits_{i=1}^{D-1}\left(100\left(z_i^2-z_{i+1}\right)^2+\left(z_i-1\right)^2\right)$$

6) Step function

$$F_{Step}\left(\mathbf{z}\right)=\sum\nolimits_{i=1}^{D}\left(\left\lfloor z_i+0.5\right\rfloor\right)^2$$

7) Tablet function

$$F_{Tablet}\left(\mathbf{z}\right)=\left(1000z_1\right)^2+\sum\nolimits_{i=2}^{D} z_i^2$$

8) Ellipse function

$$F_{Ellipse}\left(\mathbf{z}\right)=\sum\nolimits_{i=1}^{D}\left(a_i z_i\right)^2 \text{ where } a_i=20^{\frac{i-1}{D-1}},\ i=1,\ldots,D$$

9) Diff power function

$$F_{Diffpower}\left(\mathbf{z}\right)=\sum\nolimits_{i=1}^{D}\left|z_i\right|^{2+a_i} \text{ where } a_i=10^{\frac{i-1}{D-1}},\ i=1,\ldots,D$$

10) Schwefel function

$$F_{Schwefel}\left(\mathbf{z}\right)=418.982887273\times D-\sum\nolimits_{i=1}^{D} z_i\sin\left(\sqrt{\left|z_i\right|}\right)$$

11) 2^D minima function

$$F_{2Dminima}(\mathbf{z}) = 78.332331408 + \sum_{i=1}^{D}\left(z_i^4 - 16z_i^2 + 5z_i\right)\Big/D$$

12) Rastrigin function

$$F_{Rastrigin}(\mathbf{z}) = \sum_{i=1}^{D}\left(z_i^2 - 10\cos\left(2\pi z_i\right) + 10\right)$$

13) Noncontinuous Rastrigin function

$$F_{NoncRastrigin}(\mathbf{z}) = \sum_{i=1}^{D}\left(y_i^2 - 10\cos\left(2\pi y_i\right) + 10\right)$$

$$\text{where } y_i = \begin{cases} z_i & \left|z_i\right| < 0.5 \\ round\left(2z_i\right)/2 & \left|z_i\right| \geq 0.5 \end{cases}, \ i = 1,\ldots,D$$

14) Ackley function

$$F_{Ackley}(\mathbf{z}) = -20\exp\left(-0.2\sqrt{\sum_{i=1}^{D} z_i^2 \Big/ D}\right) - \exp\left(\sum_{i=1}^{D}\cos\left(2\pi z_i\right)\Big/D\right) + 20 + \mathrm{e}$$

15) Griewank function

$$F_{Griewank}(\mathbf{z}) = \sum_{i=1}^{D} z_i^2 / 4000 - \prod_{i=1}^{D}\cos\left(z_i/\sqrt{i}\right) + 1$$

16) Weierstrass function

$$F_{Weierstrass}(\mathbf{z}) = \sum_{i=1}^{D}\left(\sum_{k=0}^{kmax}\left[a^k\cos\left(2\pi b^k\left(z_i + 0.5\right)\right)\right]\right) - D\sum_{k=0}^{kmax}\left[a^k\cos\left(2\pi b^k 0.5\right)\right]$$

where $a = 0.5,\ b = 3,\ kmax = 20$

17) Salomon function

$$F_{Salomon}(\mathbf{z}) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} z_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} z_i^2}$$

18) Schwefel P2.13 function

$$F_{Schwefel\,2.13}(\mathbf{z}) = \sum_{i=1}^{D}\left(A_i - B_i(\mathbf{z})\right)^2$$

where $A_i = \sum_{j=1}^{D}\left(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j\right),\ B_i(\mathbf{z}) = \sum_{j=1}^{D}\left(a_{ij}\sin z_j + b_{ij}\cos z_j\right)$

$a_{ij},\ b_{ij}$ are integer random numbers in the range $\left[-100,\ 100\right],\ i = 1,\ldots,D,\ j = 1,\ldots,D$

α_j are random numbers in the range $\left[-\pi,\ \pi\right], j = 1,\ldots,D$

19) Penalized 1 function

$$F_{Penalized1}(\mathbf{z}) = \frac{\pi}{D}\left\{10\left(\sin(\pi y_1)\right)^2 + \sum_{i=1}^{D-1}(y_i - 1)^2\left[1 + 10\left(\sin(\pi y_{i+1})\right)^2\right] + (y_D - 1)^2\right\}$$
$$+ \sum_{i=1}^{D} u(z_i, 10, 100, 4)$$

$$\text{where } y_i = 1 + (z_i + 1)/4,\ u(z_i, a, k, m) = \begin{cases} k(z_i - a)^m & z_i > a \\ 0 & -a \le z_i \le a,\ i = 1, \ldots, D \\ k(-z_i - a)^m & z_i < -a \end{cases}$$

20) Penalized 2 function

$$F_{Penalized2}(\mathbf{z}) = 0.1\left\{\begin{matrix}\left(\sin(3\pi z_1)\right)^2 + \sum_{i=1}^{D-1}(z_i - 1)^2\left[1 + \left(\sin(3\pi z_{i+1})\right)^2\right] \\ +(z_D - 1)^2\left[1 + \left(\sin(2\pi z_D)\right)^2\right]\end{matrix}\right\}$$
$$+ \sum_{i=1}^{D} u(z_i, 5, 100, 4)$$

21) Noise Schwefel P1.2 function

$$F_{NoiseSch1.2}(\mathbf{z}) = \left(\sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2\right)\left(1 + 0.4\left|N(0,1)\right|\right)$$

22) Noise Quadric function

$$F_{NoiseQuadric}(\mathbf{z}) = \sum_{i=1}^{D} i z_i^4 + random[0,1)$$

23) Rastrigin10 function

$$F_{Rastrigin10}(\mathbf{z}) = \sum_{i=1}^{D}\left((a_i z_i)^2 - 10\cos(2\pi a_i z_i) + 10\right) \text{ where } a_i = 10^{\frac{i-1}{D-1}},\ i = 1, \ldots, D$$

24) Rastrigin100 function

$$F_{Rastrigin100}(\mathbf{z}) = \sum_{i=1}^{D}\left((a_i z_i)^2 - 10\cos(2\pi a_i z_i) + 10\right) \text{ where } a_i = 100^{\frac{i-1}{D-1}},\ i = 1, \ldots, D$$

APPENDIX B

TEST PROBLEMS FOR MULTI-OBJECTIVE OPTIMIZATION

1) ZDT1

$$f_1\left(\mathbf{x}\right)=x_1;\ f_2\left(\mathbf{x}\right)=g\left(\mathbf{x}\right)\left[1-\sqrt{x_1/g\left(\mathbf{x}\right)}\right];\ g\left(\mathbf{x}\right)=1+9\left(\sum\nolimits_{d=2}^{D}x_d\right)\Big/\left(D-1\right)$$

Optimal solutions **x***: $x_1\in[0,1],\ x_{2,\dots,D}=0$

2) ZDT2

$$f_1\left(\mathbf{x}\right)=x_1;\ f_2\left(\mathbf{x}\right)=g\left(\mathbf{x}\right)\left[1-\left(x_1/g\left(\mathbf{x}\right)\right)^2\right];\ g\left(\mathbf{x}\right)=1+9\left(\sum\nolimits_{d=2}^{D}x_d\right)\Big/\left(D-1\right)$$

Optimal solutions **x***: $x_1\in[0,1],\ x_{2,\dots,D}=0$

3) ZDT3

$$f_1\left(\mathbf{x}\right)=x_1;\ f_2\left(\mathbf{x}\right)=g\left(\mathbf{x}\right)\left[1-\sqrt{x_1/g\left(\mathbf{x}\right)}-\left(x_1/g\left(\mathbf{x}\right)\right)\sin\left(10\pi x_1\right)\right];$$

$$g\left(\mathbf{x}\right)=1+9\left(\sum\nolimits_{d=2}^{D}x_d\right)\Big/\left(D-1\right)$$

Optimal solutions **x***: $x_1\in[0,1],\ x_{2,\dots,D}=0$

4) ZDT4

$$f_1\left(\mathbf{x}\right)=x_1;\ f_2\left(\mathbf{x}\right)=g\left(\mathbf{x}\right)\left[1-\sqrt{x_1/g\left(\mathbf{x}\right)}\right];$$

$$g\left(\mathbf{x}\right)=1+10\left(D-1\right)+\sum\nolimits_{d=2}^{D}\left[x_d^2-10\cos\left(4\pi x_d\right)\right]$$

Optimal solutions **x***: $x_1\in[0,1],\ x_{2,\dots,D}=0$

5) ZDT6

$$f_1\left(\mathbf{x}\right)=1-\exp\left(-4x_1\right)\left(\sin\left(6\pi x_1\right)\right)^6;\ f_2\left(\mathbf{x}\right)=g\left(\mathbf{x}\right)\left[1-\left(f_1\left(\mathbf{x}\right)/g\left(\mathbf{x}\right)\right)^2\right];$$

$g\left(\mathbf{x}\right)=1+9\left[\left(\sum\nolimits_{d=2}^{D}x_d\right)\Big/\left(D-1\right)\right]^{0.25}$; Optimal solutions **x***: $x_1\in[0,1],\ x_{2,\dots,D}=0$

6) DTLZ1

$$f_1\left(\mathbf{x}\right)=0.5\left(1+g\left(\mathbf{x}\right)\right)\prod\nolimits_{m=1}^{M-1}x_m;\ f_{2,\dots,M-1}\left(\mathbf{x}\right)=0.5\left(1+g\left(\mathbf{x}\right)\right)\left(1-x_{M-m+1}\right)\prod\nolimits_{m=1}^{M-m}x_m;$$

$$f_M\left(\mathbf{x}\right)=0.5\left(1+g\left(\mathbf{x}\right)\right)\left(1-x_1\right);$$

$$g\left(\mathbf{x}\right)=100\left\{D-M+1+\sum\nolimits_{d=M}^{D}\left[\left(x_d-0.5\right)^2-\cos\left(20\pi\left(x_d-0.5\right)\right)\right]\right\}$$

Optimal solutions **x***: $x_{1,\dots,M}\in[0,1],\ x_{M,\dots,D}=0.5$

7) DTLZ2

$$f_1(\mathbf{x}) = (1+g(\mathbf{x}))\prod_{m=1}^{M-1}\cos(x_m\pi/2);$$
$$f_{2,\ldots,M-1}(\mathbf{x}) = (1+g(\mathbf{x}))\sin(x_{M-m+1}\pi/2)\prod_{m=1}^{M-m}\cos(x_m\pi/2);$$
$$f_M(\mathbf{x}) = (1+g(\mathbf{x}))\sin(x_1\pi/2);\ g(\mathbf{x}) = \sum_{d=M}^{D}(x_d-0.5)^2$$

Optimal solutions **x***: $x_{1,\ldots,M} \in [0,1],\ x_{M,\ldots,D} = 0.5$

8) DTLZ3

$f_1(\mathbf{x}), f_2(\mathbf{x}),\ldots, f_M(\mathbf{x})$ are the same as DTLZ2 and $g(\mathbf{x})$ is the same as DTLZ1.
Optimal solutions are the same DTLZ2.

9) DTLZ4

$f_1(\mathbf{x}), f_2(\mathbf{x}),\ldots, f_M(\mathbf{x})$, and $g(\mathbf{x})$ are the same as DTLZ2, and x_m $(m = 1,\ldots, M-1)$ is replaced by x_m^{α}, α>0 and $\alpha = 100$ in this research.
Optimal solutions are the same DTLZ2.

10) DTLZ5

$f_1(\mathbf{x}), f_2(\mathbf{x}),\ldots, f_M(\mathbf{x})$, and $g(\mathbf{x})$ are the same as DTLZ2, and x_m $(m = 2,\ldots, M-1)$ is replaced by $(1+2g(\mathbf{x})x_m)\big/\big(2(1+g(\mathbf{x}))\big)$.
Optimal solutions are the same DTLZ2.

11) DTLZ6

$f_1(\mathbf{x}), f_2(\mathbf{x}),\ldots, f_M(\mathbf{x})$ are the same as DTLZ5, and $g(\mathbf{x}) = \sum_{d=M}^{D} x_d^{0.1}$.
Optimal solutions **x***: $x_{1,\ldots,M} \in [0,1],\ x_{M,\ldots,D} = 0$

12) DTLZ7

$$f_{1,\ldots,M-1}(\mathbf{x}) = x_m;\ f_M(\mathbf{x}) = (1+g(\mathbf{x}))\left\{M - \sum_{m=1}^{M-1}\left[\frac{f_m(\mathbf{x})}{1+g(\mathbf{x})}\big(1+\sin(3\pi f_m(\mathbf{x}))\big)\right]\right\};$$
$$g(\mathbf{x}) = 1+9\sum_{d=M}^{D} x_d \Big/ (D-M+1)$$

Optimal solutions **x***: $x_{1,\ldots,M} \in [0,1],\ x_{M,\ldots,D} = 0$

APPENDIX C

COMPARISON STUDY FOR PSO-MAM

Table 36 *t*-test comparison results between PSO-MAM with: 1) PSO with sub-gradient and Cauchy mutation only; 2) PSO with non-uniform mutation-based method and Cauchy mutation only; 3) PSO-MAM without Cauchy mutation on 31 30-dimensional functions

Functions	PSO with Sub-gradient & Cauchy mutation	PSO with non-uniform mutation & Cauchy mutation	PSO-MAM without Cauchy mutation
1	=	+	=
2	=	+	=
3	-	+	-
4	-	+	-
5	=	+	=
6	=	+	=
7	=	+	=
8	-	+	-
9	=	+	=
10	=	+	=
11	-	+	-
12	-	+	=
13	+	+	=
14	+	+	=
15	+	+	+
16	+	+	+
17	+	+	+
18	=	+	=
19	+	=	+
20	=	+	+
21	+	+	=
22	=	+	+
23	-	+	+
24	+	=	=
25	=	+	=
26	=	=	=
27	=	+	+
28	-	+	+
29	+	+	=
30	+	+	+
31	+	+	+
Better (+)	11	28	11
Same (=)	13	3	16
Worse (-)	7	0	4