Resource Optimization for Connected Internet of Things

その他（別言語等）のタイトル	IoTにおけるリソースの最適化
著者	KUMRAI Teerawat
学位名	博士（工学）
学位の種別	課程博士
報告番号	甲第418号
研究科・専攻	工学専攻・先端情報電子工学コース
学位授与年月日	2018-03-23
URL	http://hdl.handle.net/10258/00009639

Resource Optimization for Connected Internet of Things

Teerawat KUMRAI

Department of Information and Electronic Engineering
Muroran Institute of Technology

This dissertation is submitted for the degree of
Doctor of Philosophy of Engineering

March 2018

Declaration

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

Teerawat KUMRAI
March 2018

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Mianxiong Dong and Prof. Kaoru Ota for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof.Jay Kishigami, Prof. Yasuo Kudo, and Prof. Jie Li, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

My sincere thanks also goes to Dr. Paskorn Champresert, and Dr. Kyoungsook Kim, who provided me an opportunity to join their team as intern, and who gave access to the laboratory and research facilities. Without they precious support it would not be possible to conduct this research.

I would like to acknowledge the financial, academic and technical support of the Muroran Institute of Technology. The Department of Information and Electronic Engineering has provided the support and equipment I have needed to produce and complete my thesis.

Last but not the least, I would like to thank my family: my parents and to my brother for supporting me spiritually throughout writing this thesis and my life in general.

Teerawat KUMRAI
March 2018

Abstract

Recently, there are more than 9 billion things that connected in the Internet of Things (IoT) and the number is exceed more than 24 billion in 2020. It means that numerous data will be generated because of increasing quickly of the number of things. An infrastructure should be developed to manage the connected things in IoT. Moreover, cloud computing will play important role in terms of data storage and analysis for IoT. Therefore, a cloud broker is considered as an intermediary in the infrastructure for managing the connected things. The cloud broker will find the best deal between clients and service providers. However, there are three problems among cloud broker, clients and service providers that are the response time of the request from clients, the energy consumption of the system and the profit of the cloud broker. The three problems are considered as multi-objective optimization problem to maximize the profit of the broker while minimizing the response time of the request and the energy consumption. A multi-objective particle swarm optimization (MOPSO) is proposed to solve the problem. MOPSO is compared with a non-dominated sorting genetic algorithm-II (NSGA-II) and a random search algorithm.

Since, there are a lot of data including social media and geographic location, generated in IoT. Coupling social media with geographic location has boosted the worth of understanding the real-world situations. Event detection aims to find more specific topic which represents real-world event. However, identification of unusual and seemingly inconsistent patterns in data, called outliers, is necessary. The problem is how to partition a spatio-temporal domain to find a meaningful local outlier pattern. A k-dimensional (KD) tree partitioning is applied to divide a spatio-temporal domain into sub-cells. The optimal partitioning problem in a spatio-temporal domain has been proven as an NP-complete problem. Therefore, a genetic algorithm is proposed to solve the problem.

Moreover, in the future, there will be a steady increase in energy consumption as well as emission levels from using energy in IoT. It needs some system or application to manage the energy consumption and emission pollutant in IoT. Moreover, the smart grid is strongly related to IoT technologies. It is enabled by IoT to handle big data and reduce the number of communication protocols. The micro-grid is studied because of micro-grids are part of a larger system that makes the smart grid to become reality. The micro-grid also is able

to manage energy consumption for reducing the emission pollutant. Thus, the operation management problem and pollutant emission problem are important problems for the micro-grid system. Thus, reducing the total energy expenses and pollutant emission of micro-grid and improving the renewable energy sources (battery energy storage) are considered together with the operation management of the micro-grid system. A fitness-based modified game particle swarm optimization (FMGPSO) algorithm is proposed to minimize the total costs of operation and pollutant emissions in the micro-grid and multi-microgrid system. FMGPSO is compared with A non-dominated sorting genetic algorithm-III (NSGA-III), a multi-objective covariance matrix adaptation evolution strategy (MO-CMAES), and a speed-constrained multi-objective particle swarm optimization (SMPSO).

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

AT	the arrival time of a request from client i at service provider j
CT	the current time
E	the total energy consumption of all the service providers
IR	the interest rate of the installed BES
k	the dimension of the particles
l	the number of particles
LT	the lifetime of the installed BES
M	the number of cloud service providers
N	the number of clients
n	the number of neighbors including the target cell
OR_t	an operating reserve requirements at time t
P	the profit of the cloud broker
RT	the response time
S	the set of cloud service providers
U	the set of clients
w	an inertia weight

Greek Symbols

α the significance level

$\mathscr{C} - metric$ a performance metric

Δt a time interval duration

$\eta_{discharge}$ a discharge efficiency of BES

μ_0 the mean value of the normal distribution

$\omega_1, \omega_2, \omega_3$ the weighting factors

σ the standard deviation of the local importance scores

Subscripts

i subscript index

j subscript index

Other Symbols

$\overline{P}_{BES,t}$ a maximum discharge rates of BES at time t

$\underline{P}_{BES,t}$ a maximum BES charge rates of BES at time t

$B_{BES,t}$ a bid of BES at time t

$B_{FC,t}$ a bid of FC at time t

$B_{Grid,t}$ a bid of utility at time t

b_{ij} a binary variable

$B_{MT,t}$ a bid of MT at time t

$B_{PV,t}$ a bid of PV at time t

$B_{WT,t}$ a bid of WT at time t

C_j the cost of service provider j

$C_{BES,max}$ a maximum size of BES

$C_{BES,min}$ a minimum size of BES

$C_{BES,t}$ the cost of BES at time t

$C_{DG,t}$ the cost of operating power and fuel of DGs at time t

$C_{Grid,t}$ the cost of grid at time t

$CO_{2,BES,t}$ the amounts of carbon dioxide from BES at time t

$CO_{2,DG_{i,t}}$ the amounts of carbon dioxide from MT and FC at time t

$CO_{2,Grid,t}$ the amounts of carbon dioxide from utility at time t

E_j the energy consumption that service provider j used to execute the job

$E^s_{BES,t}$ the emission of BES

$E^s_{DG,t}$ the emission of DG

$E^s_{Grid,t}$ the emission of utility

f_t the total costs of DG

FC_{BES} a fixed cost for BES

L_{ij} the latency between client i and service provider j

MC_{BES} a maintenance cost for BES

MO_{DG} the fixed maintenance and operation cost of DG

MO_{FC} the fixed maintenance and operation cost of FC

MO_{MT} the fixed maintenance and operation cost of MT

MO_{PV} the fixed maintenance and operation cost of PV

MO_{WT} the fixed maintenance and operation cost of WT

$NO_{x,BES,t}$ the amounts of nitrogen oxides from BES at time t

$NO_{x,DG_{i,t}}$ the amounts of nitrogen oxides from MT and FC at time t

$NO_{x,Grid,t}$ the amounts of nitrogen oxides from utility at time t

OTH an operation time

$\bar{x}$ the average of local importance scores of queen contiguity neighbors

$P(X)$ a Pareto optimal objective

P_i the price from client i

$P_{BES,t}$ a power of BES at time t

$P_{Demand,t}$ a load demand of electrical at time t

$P_{FC,t}$ a power of FC at time t

$P_{Grid,t}$ a power of utility at time t

$P_{j,max}$ the maximum power of MT, PV, FC and WT

$P_{j,min}$ the minimum power of MT, PV, FC and WT

$P_{MT,t}$ a power of MT at time t

$P_{PV,t}$ a power of PV at time t

$P_{WT,t}$ a power of WT at time t

$PM_{10,BES,t}$ the amounts of particulate matter 10 micrometers from BES at time t

$PM_{10,DG_{i,t}}$ the amounts of particulate matter 10 micrometers from MT and FC at time t

$PM_{10,Grid,t}$ the amounts of particulate matter 10 micrometers from utility at time t

R_i the number of requests from client i

$S(X)$ a supercriterion

$SD_{FC,t}$ the shut-down cost for FC at time t

$SD_{MT,t}$ the shut-down cost for MT at time t

Sh_i a shutdown cost coefficient for FC and MT at time t

$SO_{2,BES,t}$ the amounts of sulphur dioxide from BES at time t

$SO_{2,DG_{i,t}}$ the amounts of sulphur dioxide from MT and FC at time t

$SO_{2,Grid,t}$ the amounts of sulphur dioxide from utility at time t

St_i a start-up cost coefficient for FC and MT at time t

$SU_{FC,t}$ the start-up cost for FC at time t

$SU_{MT,t}$ the start-up cost for MT at time t

T_j a spend time to execute the request

tax a tax rate of grid

$TCPD_{BES}$ the Total Cost Per Day of BES

$u_{BES,t}$ a status (off or on) of BES at time t

$u_{FC,t}$ a status (off or on) of FC at time t

$u_{MT,t}$ a status (off or on) of MT at time t

c_1 a coefficient of the self-recognition component

c_2 a coefficient of the social component

$\vec{g}_l$ the best position of the swarm

$\vec{p}_l$ the best position of the particle

r_1, r_2 random numbers that are uniformly distributed in the interval 0 to 1

$\vec{v}_l$ a velocity vector

$\vec{x}_l$ a position vector

Acronyms / Abbreviations

ACO Ant Colony Optimization

BES Battery Energy Storage

BLE Bluetooth Low Energy

CC Cloud Computing

CO_2 Carbon Dioxide

D2D Device-to-Device

DGs Distributed Generation Sources

DVFS Dynamic Voltage Frequency Scaling

EA Evolutionary Algorithm

FC Fuel Cell

FiWi Fiber-Wireless

FMGPSO Fitness-Based Modified Game Particle Swarm Optimization

GA Genetic Algorithm

GGA Grouping Genetic Algorithm

HMM Hidden Markov Model

IaaS Infrastructure as a Service

IoT Internet of Things

lat latitude

lon longitude

MANETs Mobile Ad-hoc NETworks

MGCC Micro-Grid Central Controller

MG Micro-Grid

MLIP Mixed Linear Integer Problem

MOGA Multi-Objective Genetic Algorithm

MOPSO Multi-Objective Particle Swarm Optimization

MT Micro-Turbine

NO_x Nitrogen Oxides

NSGA Non-dominated Sorting Genetic Algorithm

OMMG Operation Management of Micro-Grid

OPC One-Point Crossover

PaaS Platform as a Service

PM_{10} Particulate Matter 10 micrometers

PSO Particle Swarm Optimization

PV Photo-Voltaic

QoS Quality of Service

RBA Rank-Based Allocation

RESs Renewable Energy Sources

RFID Radio-Frequency Identification

SaaS Software as a Service

SAPSO Self-Adaptive Particle Swarm Optimization

SO_2 Sulphur Dioxide

tf-idf Term Frequency–Inverse Document Frequency

VMs Virtual Machines

WSNs Wireless Sensor Networks

WT Wind Turbine

Z-test The Normal Distribution Test

Chapter 1

Introduction

1.1 Background and motivation

Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over an Internet technology. In the other word, it is an Internet technology that connects humans, machines, and tools with each other via the Internet or wireless technologies as shown in figure 1.1. On the other hand, IoT is complex environments including many heterogeneous devices, application, and components. In the real world, the large number of things connected and data generated by those things in IoT will define a great require in term of processing and storage resources to be changed into useful services, systems and information. Some services, applications, or systems will require complex processing which include an analysis in time series and history of data. Therefore, it is very difficult when to consider IoT's typical resource constraints to imagine ultra-scale real-world of IoT without including a cloud computing platforms or a smart devices/systems such as Smart Gateways [1], edge/fog nodes [31] and so on. Moreover, figure 1.2 shows the future of the Internet of things. The figure shows that there are 8 billion devices in use by 2014. In the future, there will be 10 connected objects for every man, woman, and child on the planet. It means there will be as many as 40 to 80 billion connected objects by 2020. Thus, a lot of data will be generated in the future because the number of things is rapidly increasing. Therefore, a resource management or resource optimization is important to make a powerful IoT. The resource management or optimization is a set of processes or methods which is used to adjust the available resources (devices, applications, systems and so on) and the planned resources to achieve established goals. Optimization is desired to achieve objectives and constraints within expected time and minimum resources usage. Therefore, the resource optimization is considered to manage connected things and analyze generated data in IoT.

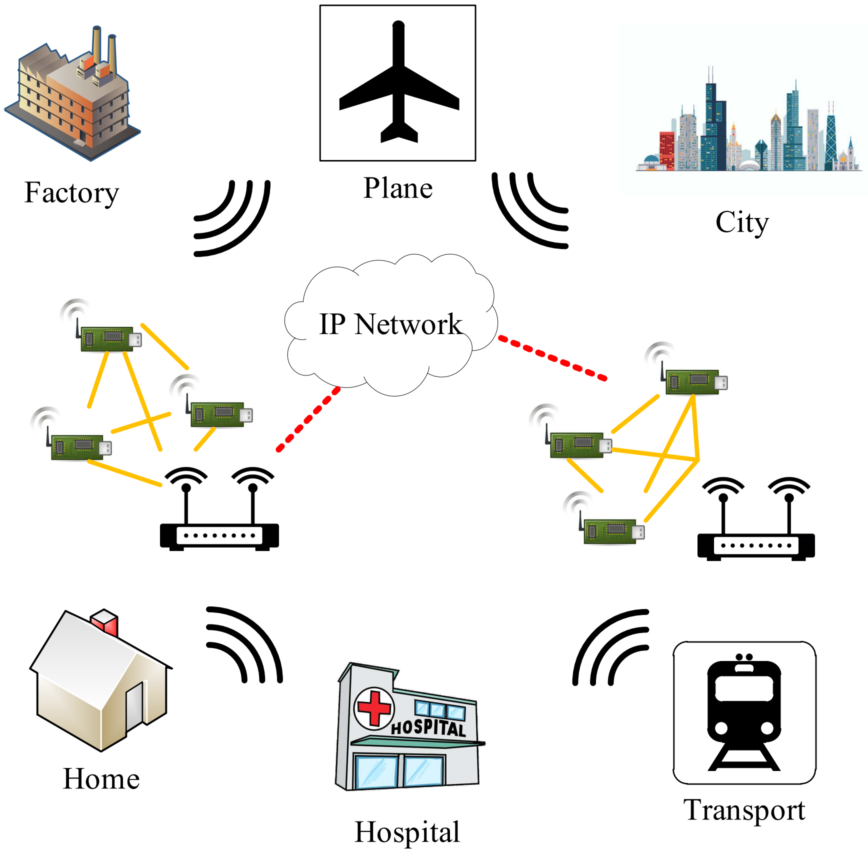


Fig. 1.1 An example of the Internet of things

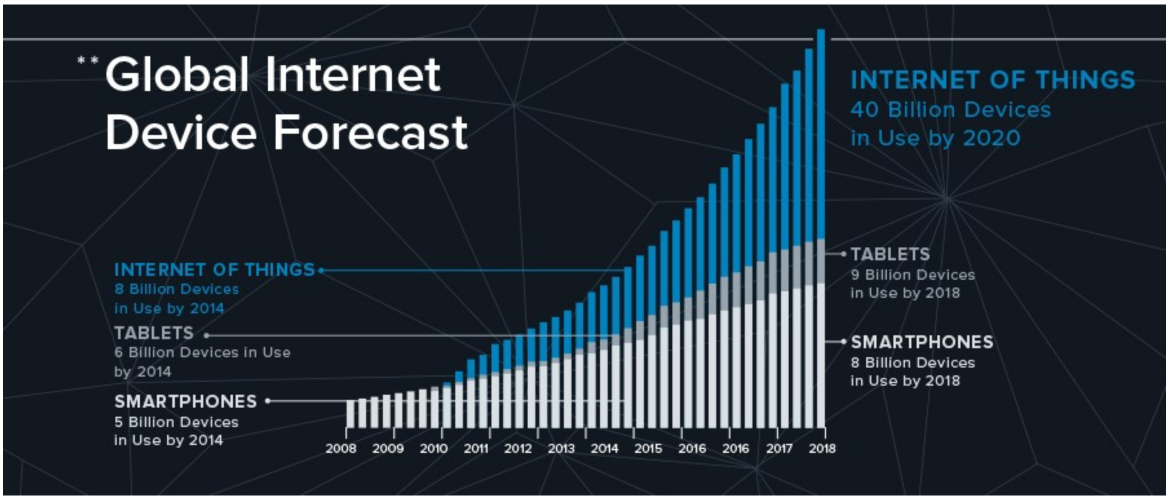


Fig. 1.2 The future of the Internet of things
(Source: http://visual.ly/future-internet-things)

The architecture of IoT could be classified into three-layer architecture as shown in figure 1.3. The three-layer consist of application as a top layer, network layer as a middle layer and sensing layer as a bottom layer. The application layer represents many applications such as environment prediction & protection, intelligent transport, smart grid & micro-grid and so on. Next, the sensing layer consists of various types of sensing devices or sensors, such as, RFID, ZigBee, GPS and so on. Each sensing devices or sensor collects sensing data around its location. Then, the sensing device transmits the sensing data to the applications through the cloud computing platforms at the network layer. From the architecture of IoT, It can see that the cloud computing is an important part of IoT. Therefore, cloud computing is considered as an important roles in terms of analysis and data storage in IoT.

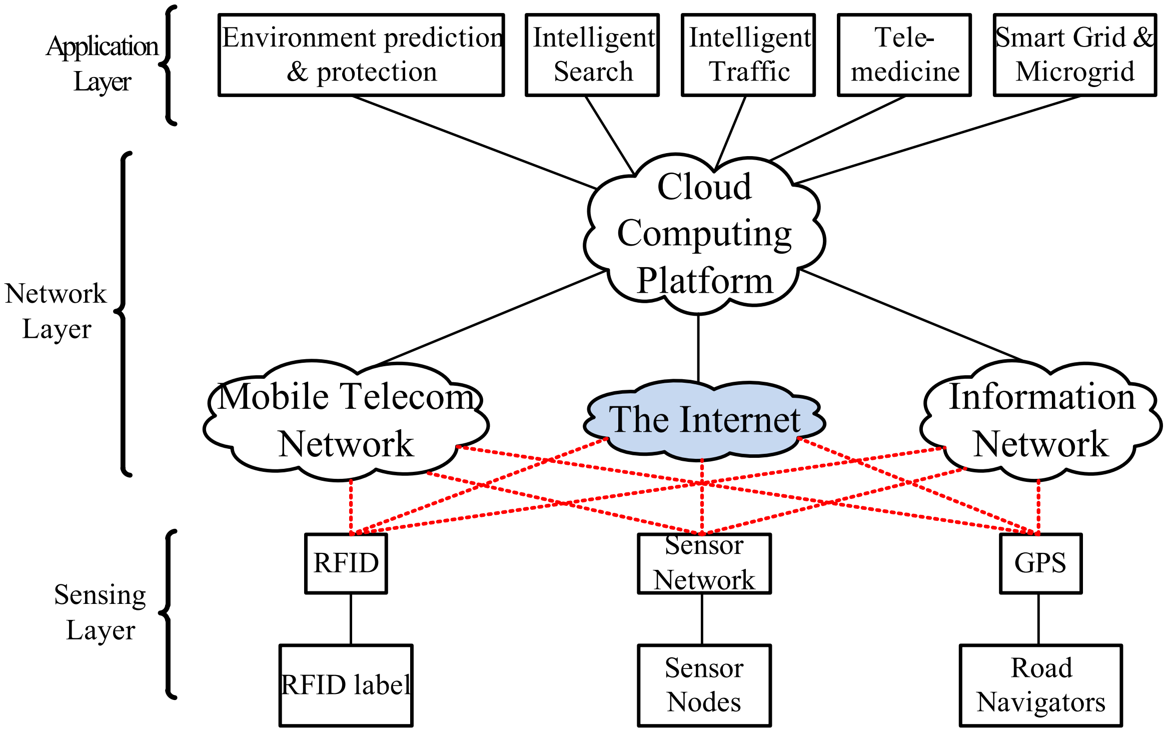


Fig. 1.3 The architecture of an IoT [37]

In general, cloud computing is a delivery of a computing services to clients on demand. Moreover, the service of cloud computing can be classified into three categories as follows: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). To handle connected things, Infrastructure as a Service (IaaS) is considered. In IaaS, organizations can outsource support operations, including hardware, storage, servers, and networks. Cloud service providers will own equipment, work responsibilities, and maintenance by clients, who have the option to pay for actual usage. An infrastructure should be developed for supporting and managing the connected things. Therefore, an intermediary of IaaS in cloud computing is utilized to manage a huge number of things and data that are connected in IoT. A cloud broker is considered as an intermediary in the infrastructure for managing the connected things. The cloud broker will find the best deal between clients and service providers. However, the response time of the request from clients, the energy consumption of the system and the profit of the cloud broker are important problems among cloud broker, clients and service providers. Because cloud broker is expected the profit to manage the process between clients and service providers, clients also are expected the speed to execute their request, and service providers are expected to minimize energy consumption in the system.

Moreover, as mentioned about a lot of data will be generated in the future because the number of things is rapidly increasing, the data include social media and geographic location such as Tweet, Facebook, Instagram and so on. Figure 1.4 shows the information of increasing the number of data in the future. Coupling social media with geographic location has boosted the worth of understanding the real-world situations. It means that we could be analyzed those data to capture and predict real-world situations in emergency management. Thus, social media and geographic location data raise the need for the enhanced spatio-temporal data analysis to automatically discover potentially useful patterns and knowledge. Moreover, sometimes identification of unusual and seemingly inconsistent patterns in data is necessary. These pattern called outliers. It may represent some useful information about abnormal situations in many applications such as credit card fraud detection, clinical trials, voting irregularity analysis, data cleansing, network intrusion, severe weather prediction, geographic information systems, athlete performance analysis, and other data-mining tasks. However, there are many outliers' definitions. The easy way is that an outlier of a keyword in geo-social media has a different pattern of the local importance measures with respect to the surrounding neighbors. It means geo-social media data are divided into sub-space and find a pattern outlier from sub-space. However, a different result could get depending on the size and way of sub-space partitioning. Thus, the problem is how to partition geo-social media data into sub-space while maximizing the number of outlier patterns.

Finally, since there will be a steady increase in energy consumption as well as emission levels from using energy in IoT as shown in figure 1.5. It needs some system or application to manage the energy consumption and emission pollutant in IoT. Therefore, a micro-grid is considered to manage the energy consumption and emission pollutant, because the micro-grids are a part of a larger system that could be made reality smart grid that is strongly related to IoT technologies. Moreover, the smart grid is enabled by IoT to reduce the number of communication protocols and handle big data. The micro-grid also is able to manage

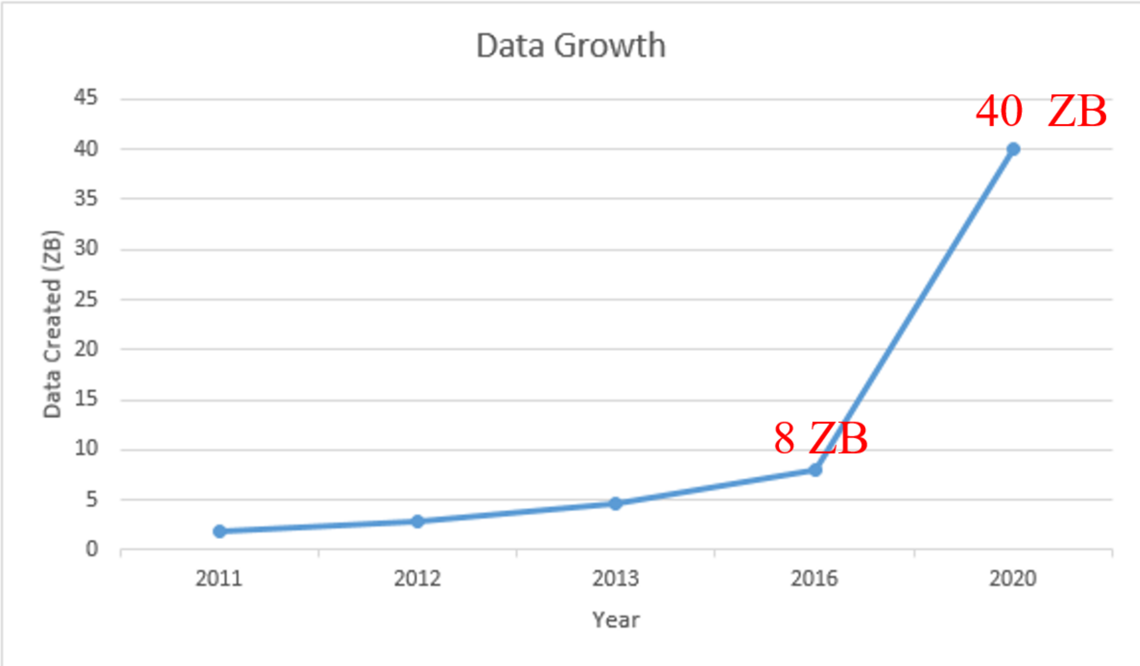


Fig. 1.4 Increasing the number of data in the future [54]

energy consumption in IoT to reduce the emission pollutant in the system. The micro-grid is an electricity distribution systems containing loads and distributed generation sources (DGs), such as fuel cell (FC), micro-turbine (MT), photovoltaic (PV), and wind turbine (WT), including the energy storage options operating system (battery energy storage (BES)). Figure 1.6 shows an example of the micro-grid system. Since there are many units in the micro-grid system, it is difficult to manage operations in the system with minimum cost and energy. Moreover, some pollutant emissions such as CO_2, NO_x, SO_2 and $PM-10$, are produced by some units in the micro-grid system. Therefore, the problem in the micro-grid system is how to manage operations while minimizing the pollutant emissions and the total operating cost.

1.2 Statement and significance of resource optimization for connected Internet of things

Recently, there are more than 9 billion things that connected in the Internet of Things (IoT) and the number is exceed more than 24 billion in 2020. It means that numerous data will be generated because of increasing quickly of the number of things. An infrastructure should be

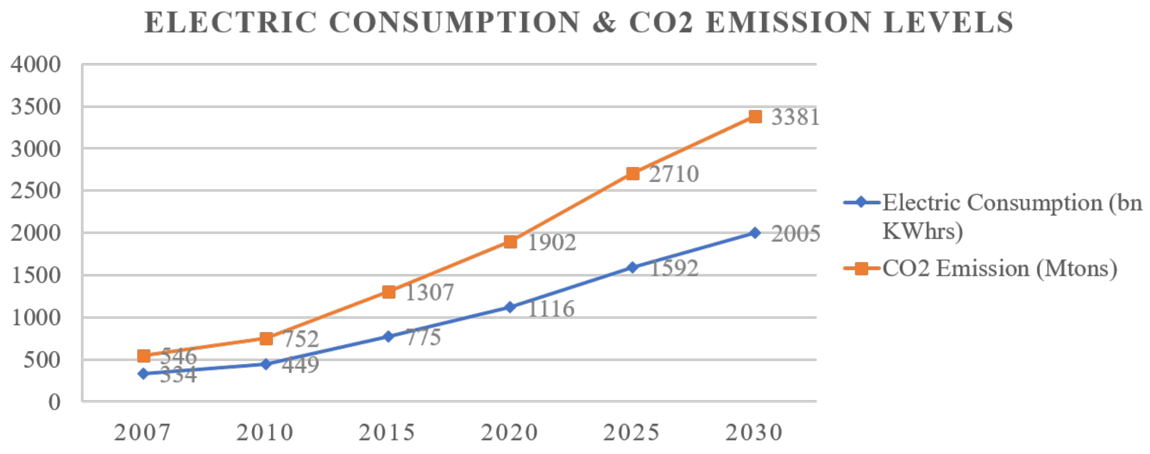


Fig. 1.5 Increasing electric consumption and emission levels in the future
(Source: U.S. Energy Information Administration, International Energy Outlook 2016)

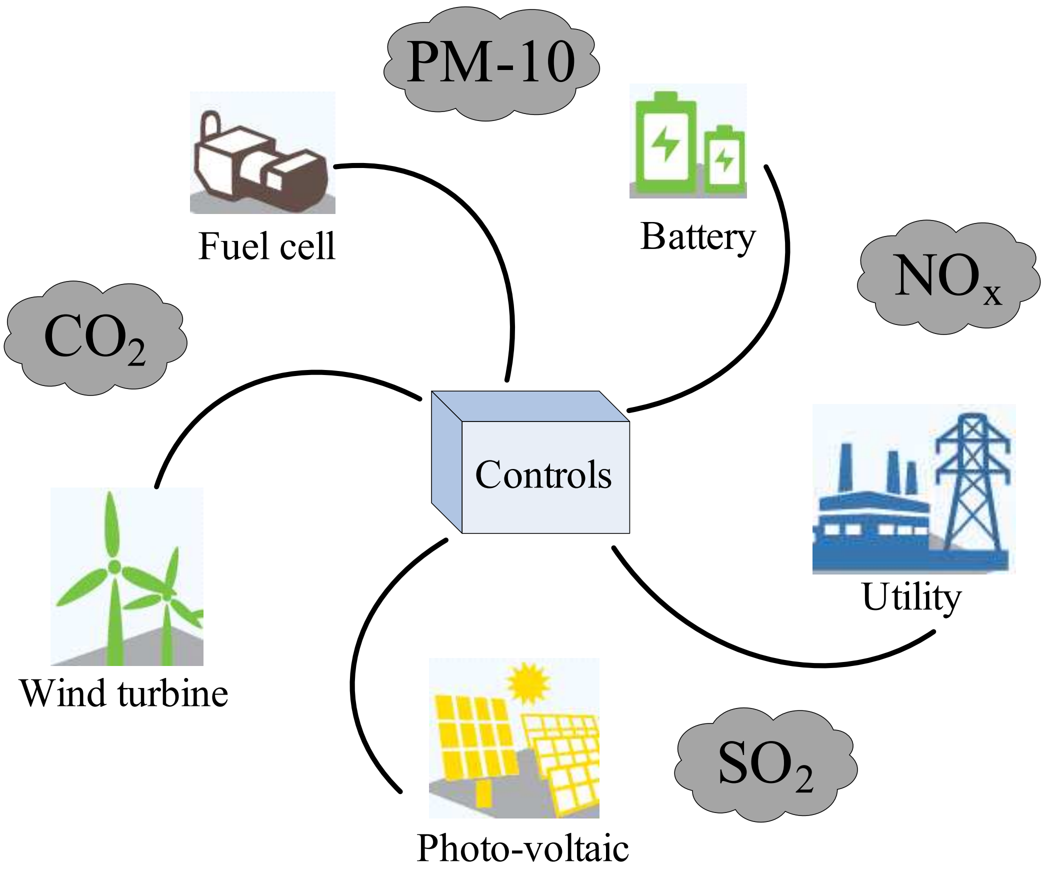


Fig. 1.6 Micro-grid

developed to manage the connected things in IoT. A cloud broker is used as the intermediary to optimize resource selection in cloud computing. The requests from clients are matched by the cloud broker with offers provided by the service providers. In a sense, when the cloud broker find the best deal between the request from client and the cloud service provider, the cloud broker is expected the maximum profit. On the other hand, the cloud service providers are expected to execute the request from client with the minimum energy consumption. Moreover, when the clients submit their requests to cloud broker, they are expected with the minimum response time. Since the scheduling problem has been proven as an NP-hard problem [32], it can conclude that the cloud brokering problem is NP-hard problem.

Since, there are a lot of data including social media and geographic location, generated in IoT. Coupling social media with geographic location has boosted the worth of understanding the real-world situations. Many Twitter messages were concentrated in the path of Hurricane Sandy as presented in [70]. The tweets and flooding had a similar pattern of the path. Therefore, those messages reflected human experiences of the storm and the local information can be obtained by analysis those messages. There are many tools and techniques of spatio-temporal data mining, such as outlier detection, spatio-temporal clustering, co-occurrence pattern discovery and hotspot detection [69]. In particular, unusual and seemingly inconsistent patterns in data, called outliers. Sometimes the outliers represent useful information about abnormal situations in many applications.

One way to detection outlier is that a spatio-temporal domain is divided into several small areas and measure the keyword importance of each area. A different pattern of the local importance measures with respect to the surrounding neighbors is assumed as a spatio-temporal outlier of a keyword in geo-social media. However, different results are found depending on the way and size of sub-space partitioning. Moreover, the optimal partitioning problem in a spatio-temporal domain has been proven as an NP-complete problem [48].

Moreover, since there will be a steady increase in energy consumption as well as emission levels from using energy in IoT. It needs some system or application to manage the energy consumption and emission pollutant in IoT. The micro-grid is studied to manage the energy consumption as well as emission pollutant in IoT because of micro-grids are part of a larger system that makes the smart grid to become reality and the smart grid which is enabled by IoT to handle big data, is strongly related to IoT technologies. The micro-grid also is able to manage energy consumption in IoT to reduce the emission pollutant in the system. In the micro-grid system, renewable energy sources poses a challenge in term of the fluctuation and irregularly of the units. This is the reason why the battery energy storage (BES) should be developed to increase the high availability time. The BES can be developed by the micro-grid central controller (MGCC) in the micro-grid system. However, the capacity and/or size of

BES is an important issue in the operation management problem in the micro-grid system. Therefore, the operation management problem is important problem in the micro-grid system. Moreover, some units in the micro-grid system produced some pollutant emissions such as CO_2, NO_x, SO_2 and $PM-10$. Therefore, it is important to consider reducing the pollutant emissions and the total operating cost in the operation management problem in the micro-grid system. Moreover, the scheduling problem has been proven as an NP-hard problem [32], it can conclude that the operation management problem in the micro-grid is NP-hard problem.

1.3 Purpose of the study

1. To investigate a cloud brokering system for connected IoT system with considering the difficulty issues.
2. To investigate an operation management for an micro-grid system with considering operation cost and pollutant emissions.
3. To investigate an optimal solution of partitioning for finding patterns in geo-social points.

1.4 Research rationale

A novel optimization problem is formulated for solving a problem among of clients, service providers, and cloud broker. This aims to help manage connections in a cloud IoT system. Next, a novel algorithm for operation management in a multi-micro-grids system is proposed. This aims to help control cost and pollutant emissions in the system. Finally, a k-dimensional tree is applied to divide a spatio-temporal domain of geo-social media for finding patterns. This aims to help us to know a situation in the interested area.

1.5 Educational advantages

1. To obtain a cloud brokering system for connected IoT system and an algorithm for solving an optimization problem in the cloud brokering system.
2. To obtain a partitioning for finding patterns in geo-social points and an algorithm for finding the optimal solution of partitioning.
3. To obtain an operation management for a multi-micro-grids system and a novel algorithm for solving an optimization problem of the multi-micro-grids system.

1.6 Main contribution of the studies

1.6.1 Cloud brokering systems for connected Internet of things

1. A cloud brokering problem in a cloud IoT system is investigated by considering the energy consumption, broker profit, and response time.

2. A novel optimization problem is formulated to maximize the broker profit and to minimize the energy consumption of the system and response time of clients.

3. Particle swarm optimization (PSO) is proposed to solve the formulated problem as a single-objective optimization problem. Moreover, multi-objective particle swarm optimization (MOPSO) is designed to solve the formulated problem as a multi-objective optimization problem.

4. MOPSO is proposed to solve the multi-objective optimization problem by using extensive simulations. Moreover, there are three algorithms to compare the performance: MOPSO, a well-known genetic algorithm and a random search algorithm.

1.6.2 KD-partitioning for the local outlier detection

1. A k-dimensional (KD) tree is applied to divide a spatio-temporal domain into sub-cells.

2. Two outlier patterns of a keyword are formulated by the local importance measures among the surrounding sub-cells.

3. The optimal solution of partitioning are investigated to maximize the number of two outlier patterns.

4. Genetic algorithm (GA) is proposed to seek the optimal solution of KD-tree partitioning while maximizing the number of outlier patterns by using extensive simulations.

1.6.3 Operation management for multi-microgrids control

1. An optimization operation management problem in an micro-grid system is investigated by considering the total cost of distributed generation sources, the maintenance and operation cost of fuel cell, micro-turbine, photovoltaic, and wind turbine, and the total battery energy storage cost per day and the pollutant emissions.

2. A cost function, an emission function, and constraints are formulated to minimize total cost and pollutant emissions in operation management problem as a single-objective and multi-objective optimization problem, respectively.

3. Fitness-based modified game particle swarm optimization (FMGPSO) is designed and proposed to solve the formulated cost, emission function, and constraints by using extensive simulations. Moreover, there are four algorithms to compare the performance: FMGPSO, A non-dominated sorting genetic algorithm-III (NSGA-III), multi-objective covariance matrix adaptation evolution strategy (MO-CMAES), and speed-constrained multi-objective particle swarm optimization (SMPSO).

1.7 Scope of the studies

1.7.1 Cloud brokering systems for connected Internet of things

1. The Cloud brokering system will be implemented and tested in a simulator.

2. The simulation configuration

   (a) The simulated system is used to find the best deals between 7 service providers and 5 clients.

   (b) The simulated system is used to find the best deals with the maximum profit for the cloud broker and minimum response time and minimum energy consumption of the cloud.

   (c) There are 3 types of instance: small, large and extra-large.

   (d) The average execution time and instance prices depend on the instance type.

1.7.2 KD-partitioning for the local outlier detection

1. The KD-partitioning for the local outlier detection in geo-social media will be implemented and tested in a simulator.

2. The simulation configuration

   (a) The number of the sample data is 33,030. Each data consists of geographical coordinates and timestamps.

   (b) A k-dimensional (KD) tree is applied to divide a spatio-temporal domain in to sub-cells.

(c) A term frequency-inverse document frequency ($tf - idf$) values of keyword 'landslide' in each tweet is calculated.

(d) A normal distribution test is applied to find the optimal number of patterns.

(e) Two-tailed normal distribution with α is considered by 0.05.

1.7.3 Operation management for multi-microgrids control

1. The multi-microgrids control system will be implemented and tested in a simulator.

2. The simulation configuration

   (a) The micro-grid test system consists of the PV, WT, MT, FC and BES. There are 2 cases: the micro-grid system with BES and without the BES.

   (b) The multi-micro-grid test system consists of 4 micro-grids. Each micro-grid consists of different DGs.

   (c) There are limitation of productions and coefficients of each DG.

1.8 Research methodology

1. Define the formal definition of the systems, and formulate optimization problems
2. Study and review related theories and literature
3. Design an efficient optimization algorithms for the research problems
4. Test and discuss the performance of algorithms
5. Conclude the experiment result and error correction
6. Conclude the research results and submit research papers for publication

Chapter 2

Principles and Theories of the Study

This chapter describes some principles that will be used in this study. The chapter starts with introducing the fundamental concepts of cloud brokering systems for connected Internet of Things, operation management for micro-grids control and outlier detection in geo-social media, respectively. Then, the multi-objective optimization techniques are presented. Finally, the literature review is presented.

2.1 Cloud Brokering Systems for Connected Internet of Things

This section starts with introducing the fundamental concepts of an Internet of things and cloud brokering systems for connected Internet of things, respectively. Recently, there are many humans, machines and tools that can connect with each other via the wireless and/or Internet technologies such as Bluetooth Low Energy (BLE), Wi-Fi, ZigBee and Radio-Frequency Identification (RFID) as shown in figure 2.1. The Internet technology is called IoT. In 2011, the number of data and things that connected in IoT outnumber the number of population in the world. There are more than 9 billion thins that connected in currently and it could be exceed more then 24 billion by 2020 [33]. It means there are a lot of data that will be generated in the future. Therefore, it is very difficult to handle analysis and data storage problem. Cloud computing and big data will play important roles to those problems. However, only cloud computing is considered in this study.

.

Cloud computing or CC is a service delivered based on visualization technologies through data centers. Cloud computing delivers a service to clients on demand [14]. It means software, systems, and computing resources of the cloud service provider can be used by the clients

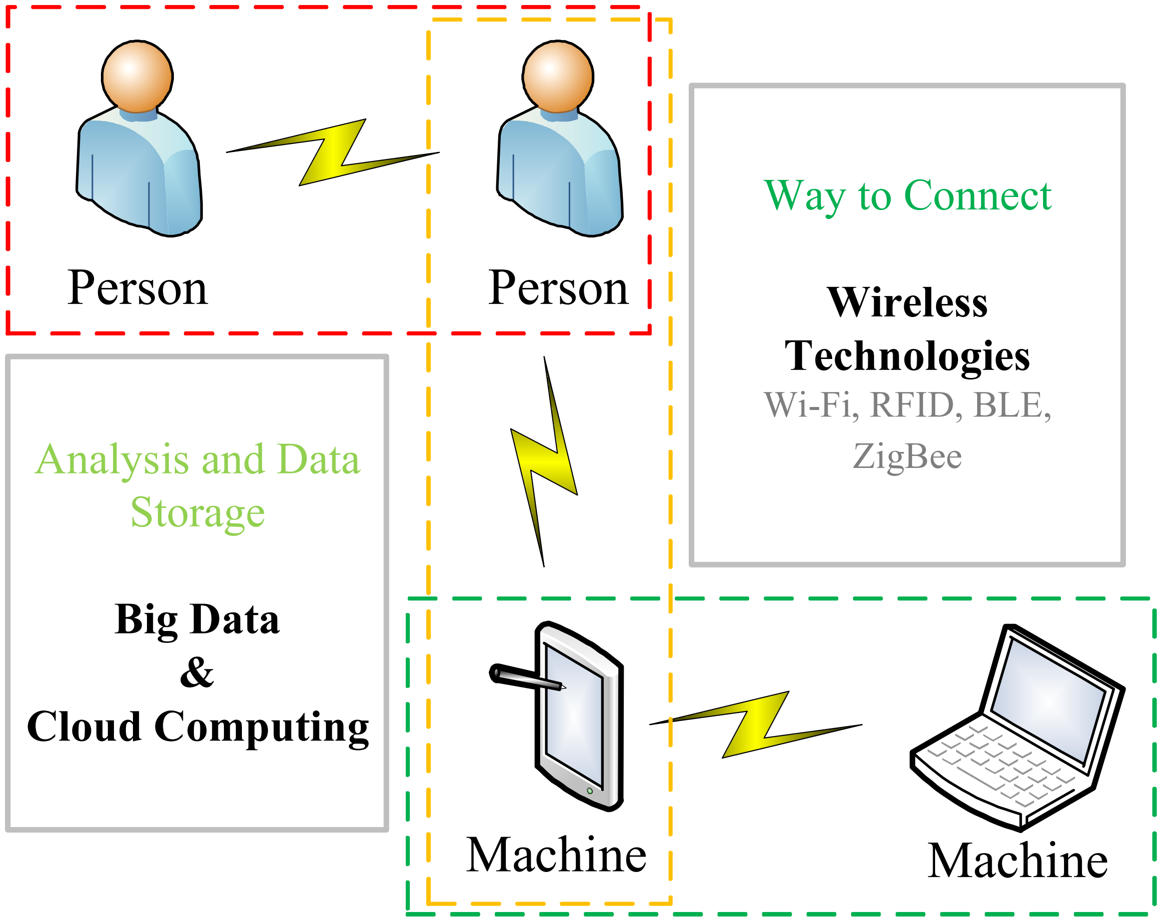


Fig. 2.1 Internet of Things: Things can be connected and communicated with each other via wireless technologies. Big data and cloud computing will play important roles in terms of analysis and data storage.

through the Internet anytime and anywhere. Additional, a mobile device (e.g., smart-phone and tablet) of some clients can be detected and used the computing resources on other mobile devices through D2D (device-to-device) communication [53]. Figure 2.2 shows the cloud computing architecture. In general, the service of cloud computing can be classified into three categories as follows:

1. The first service is Software as a Service (SaaS). In this service, clients can run software through the Internet without installing and maintaining the software.

2. The second service is Platform as a Service (PaaS). This service is to provide a platform allowing clients to develop, run, and manage applications visualization servers.

3. The third service is Infrastructure as a Service (IaaS). In this service, organizations can outsource support operations, including hardware, storage, servers, and networks. Cloud service providers will own equipment, work responsibilities, and maintenance by clients, who have the option to pay by actual usage.

Since, in the future, IoT will be a challenge by huge connected things and data on the Internet. An infrastructure is needed to develop for supporting and managing the connected things' quantity. Therefore, an intermediary of IaaS in cloud computing is utilized to manage a huge number of things and data that are connected in IoT. Because sometimes clients can be submitted their request to the cloud service providers through the intermediary. The intermediary resides between the cloud service providers and clients. Thus, a cloud broker is used as the intermediary to optimize resource selection in cloud computing. The requests from clients are matched by the cloud broker with offers provided by the service providers. In a sense, when the cloud broker find the best deal between the request from client and the cloud service provider, the cloud broker is expected the maximum profit. On the other hand, the cloud service providers are expected to execute the request from client with the minimum energy consumption. Moreover, when the clients submit their requests to cloud broker, they are expected with the minimum response time. Therefore, how to maximize the profit of the cloud broker and minimize the total energy consumption in cloud computing and the response time of requests from clients is considered in this study.

Since the scheduling problem has been proven as an NP-hard problem [32], it can conclude that the cloud brokering problem is NP-hard problem. The NP problem can be solved by using a meta-heuristic algorithm (evolutionary algorithm). Moreover, Several studies focused on meta-heuristic algorithm and cloud brokering in cloud computing [43, 71, 39]. The optimization problem of the energy consumption, CO2 emissions, and deployment cost in cloud computing is studied in [43]. A greedy heuristic algorithm and a multi-objective genetic algorithm were proposed to solve the problem. A parallel hybrid evolutionary algorithm (EA) was proposed in [39] to maximize the broker profit of virtual machine subletting in cloud computing. The veritable data from the cloud providers used for comparison between the proposed EA and a greedy heuristic algorithm. Moreover, a novel framework was proposed in [71], called CLOUD Resource Broker. A particle swarm optimization-based resource allocation scheme and a deadline-based job scheduling were integrated into the proposed framework. However, neither method solved the profit of the cloud broker, the energy consumption in cloud computing and/or the response time of requests from clients simultaneously. Moreover, almost the proposed algorithm are based on a genetic algorithm. The genetic algorithm converges slowly and has a high complexity than a particle swarm optimization (PSO) scheme. Thus, the formulating, designing and developing the PSO for

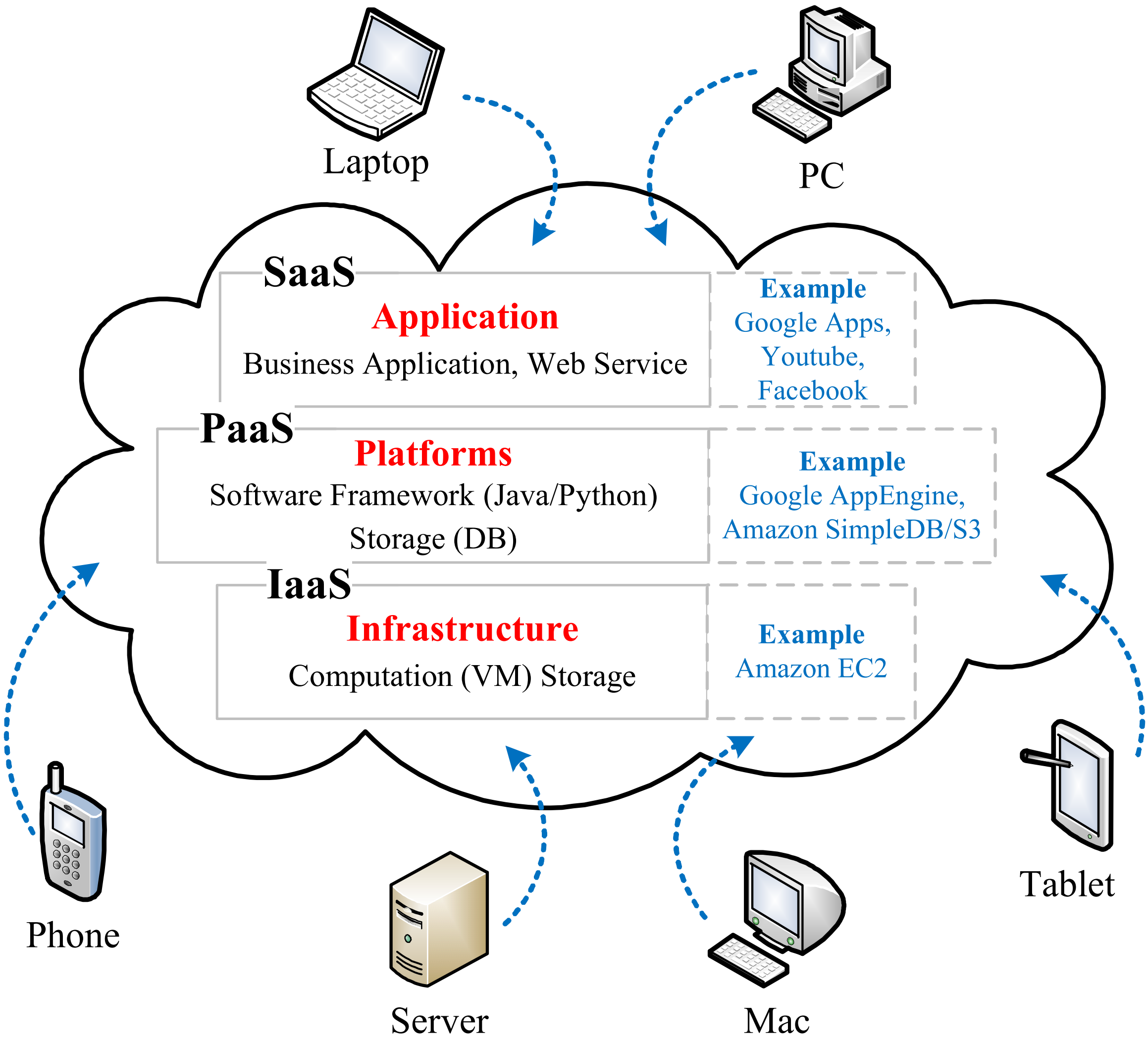


Fig. 2.2 Cloud Computing Architecture

the cloud brokering system are the main motivations. The PSO is used to increase the profit of the cloud broker and reduce the total energy consumption in cloud computing and the response time of requests from clients.

Therefore, a cloud brokering problem is considered as a multi-objective optimization problem with three objectives: (1) maximizing the profit of the broker, (2) minimizing the energy consumption in cloud computing, and (3) minimizing the response time of requests from clients. Moreover, a multi-objective particle swarm optimization (MOPSO) scheme is used to seek the optimal solutions set (Pareto-set) for the cloud brokering problem.

2.2 Operation Management for Micro-grids Control

This section starts with introducing the fundamental concepts of micro-grid and operation management for micro-grids control, respectively. Micro-grid is an electricity distribution systems containing loads and distributed generation sources (DGs) including the energy storage options operating system (a single system provides both of heat and power) that can be operated in a controlled, coordinated way either while connected to the main power network. Distributed generation sources are renewable resources such as the solar energy system and wind energy system. To raise energy crisis and centralize modern power grids, small-scale distributed generation sources and renewable energy sources (RESs) are combined with micro-grid [75]. Renewable energy sources in the micro-grid system in recently poses a challenge in whereby the fluctuation and irregularly of the units, i.e., wind turbine and photo-voltaic units. Thus, the battery energy storage (BES) should be developed to increase the high availability time. The BES can be developed by the micro-grid central controller (MGCC) in the micro-grid system. However, the capacity and/or size of BES is an important issue in the operation management problem in the micro-grid system. Moreover, some units in the micro-grid system produced some pollutant emissions such as CO_2, NO_x, SO_2 and $PM-10$. Therefore, it is important to consider reducing the pollutant emissions and the total operating cost in the operation management problem in the micro-grid system.

Several studies focused on the minimization operation management problem of the micro-grid system. There are some studies that consider the problem with the size and/or capacity of BES. For example, the authors in [18] proposed a mixed linear integer problem (MLIP) which is a solver in a modeling language for mathematical programming (AMPL) that considers the cost-benefit analysis, to optimize the size and/or capacity of an energy storage system in the micro-grid system. The optimizing the size and/or capacity of BES and the minimizing total operation cost of the micro-grid system problem were considered in [8]. An improved bat algorithm was proposed to solve the problems. Moreover, a novel software was proposed in [5], called a PSCAD/EMTDC software. The size of BES in the micro-grid system is optimized by using the proposed software.

Otherwise, there are some studies that not consider the problem with the size and/or capacity of BES. For example, in [15], the authors proposed linear programming for minimizing the operation management in the micro-grid system with optimizing the state of a charger of BES. A mathematical model based on linear programming and multi-agent system for the micro-grid operation was proposed in [44]. In [35], the authors proposed particle swarm optimization scheme to optimize the operation of a typical micro-grid interconnected with the main grid. The system included a local load, wind power, hydro-power and storage devices.

Therefore, the operation management problem is considered as an optimization problem. The objectives of the problem are to minimize the total cost of DGs, operation cost of a fuel cell (FC), micro-turbine (MT), photovoltaic (PV) and wind turbine (WT) and the total BES cost per day as well as the pollutant emissions. Then, a fitness-based modified game particle swarm optimization (FMGPSO) scheme is proposed to find the optimal solutions set for the operation management problem. FMGPSO is an integration of game theory and PSO, because of both game theory and PSO can be used to solve both single-objective optimization problem and multi-objective optimization problem.

2.3 The partitioning for Local Outlier Detection in Geo-Social Media

This section starts with introducing the fundamental concepts of geo-social media and the local outlier detection in geo-social media, respectively. Geo-tagged social media (in short, geo-social media) are pieces of information that can be attached to a photo, status or tweet with the physical location on a social networking site. Recently, the analysis of geo-social media is being emphasized to capture real-world situations in emergency management as well as to predict real-world situation. For example, the Hurricanes in the USA (Sandy and Katrina) and Haiti (Fay, Gustav, Hannah, and Ike), and the Tsunami and earthquake in Japan. These examples can be used as evidence. Moreover, many Twitter messages were concentrated in the path of Hurricane Sandy as presented in [70]. The tweets and flooding had a similar pattern of the path. Therefore, those messages reflected human experiences of the storm and the local information can be obtained by analysis those messages. Working with geo-location data in social media raises the need for the enhanced spatio-temporal data analysis to automatically discover potentially useful patterns and knowledge.

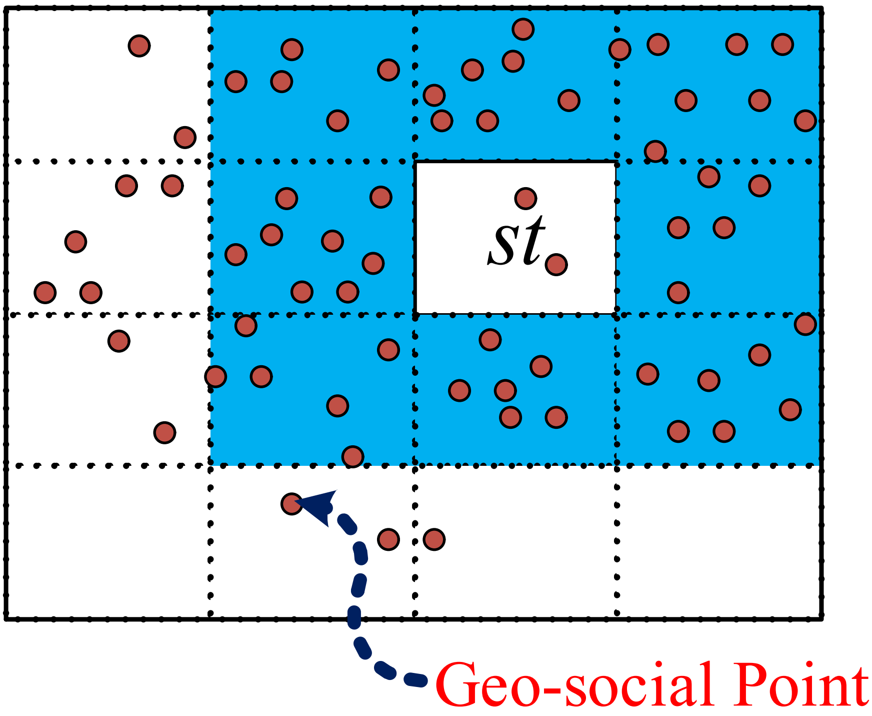
There are many tools and techniques of spatio-temporal data mining, such as outlier detection, spatio-temporal clustering, co-occurrence pattern discovery and hotspot detection [69]. Clustering techniques have been used to extract localized events within a small geographic area in [2] and [73]. More specific topics that represent real-world events, are extracted by using spatio-temporal clustering methods such as tweets related to breaking news from noise are captured by implementing online clustering, namely TwitterStand[66]. Moreover, in [19], the authors introduced a space-time scan statistic approach. The approach is used to detect hotspots as topic events within a dataset across both space and time. Outliers is an unusual and seemingly inconsistent patterns in data. Sometimes it represents useful information about abnormal situations in many applications, for example, credit card fraud detection, network traffic monitoring and outbreak of disease. Figure 2.3a and figure 2.3b show and example of abnormal situations. Some area likes *st* in the figure 2.3a sometimes has small tweets but surrounding neighbors have many tweet. It means there is something happen in st area. On the other hand, if *st* area in figure 2.3b has many tweets but surrounding neighbors have small tweets. Thus, it also means something may happen in the area.

Therefore, this study focus on the outlier detection. A genetic algorithm based approach that can detect spatio-temporal outliers with a multi-objective optimization problem is proposed. In the beginning, a spatio-temporal domain is divided into several small areas and measure the keyword importance of each area. A different pattern of the local importance measures with respect to the surrounding neighbors is assumed as a spatio-temporal outlier of a keyword in geo-social media. The outlier patterns can be divided into two categories: H-pattern which is outbound and L-pattern which is lower bound. However, different results are found depending on the way and size of sub-space partitioning. Thus, an optimal partitioning problem is considered in this study. The optimal partitioning problem in a spatio-temporal domain has been proven as an NP-complete problem [48]. Therefore, a genetic algorithm is used to find the optimal way to partition a space. Because GA is capable of making a global search and requires a shorter processing time than other meta-heuristics algorithms.

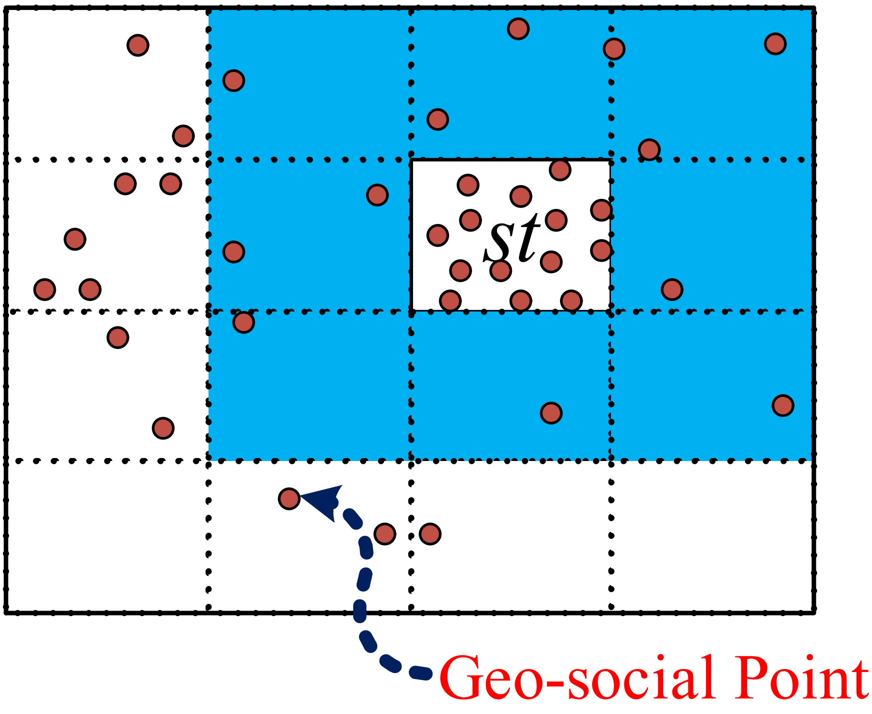
2.4 Particle Swarm Optimization

The particle swarm optimization (PSO) is described in this section. In 1995, Kennedy, Eberhart, and Shi designed PSO by imitating the motion of a flock of birds. Figure 2.4 shows a basis idea of a flock of birds. Each bird randomly out to search food and gains knowledge about the presence of food and his/her other members. The best strategy would be to search of food in the surrounding area and/or follow the distance the best food birds covered as shown in the figure. PSO is a population-based optimization tool and can be used to solve optimization problems. The set of potential solutions is represented by the swarm in PSO. A solution in the search space is represented by a particle in the swarm. Each particle moves repeatedly in the search space to find a new position as well as the best fitness value.

A position vector $\vec{x}_l = (x_{l1}, x_{l2}, ..., x_{lk})$ with the l-th particles in k-dimensions is used to represent the particle in the swarm. $\vec{v}_l = (v_{l1}, v_{l2}, ..., v_{lk})$ represents the velocity vector. $\vec{p}_l = (p_{l1}, p_{l2}, ..., p_{lk})$ and $\vec{g}_l = (g_{l1}, g_{l2}, ..., g_{lk})$ represent the best position of the particle and the best position of the swarm, respectively. In the beginning, PSO randomly generated particles to form an initial swarm. The updated velocity vector is computed at each iteration t in PSO. It can be calculated as follows:



(a) Many points in surrounding neighbors of *st* area



(b) Many points in *st* area

Fig. 2.3 An example of situation in geo-social points

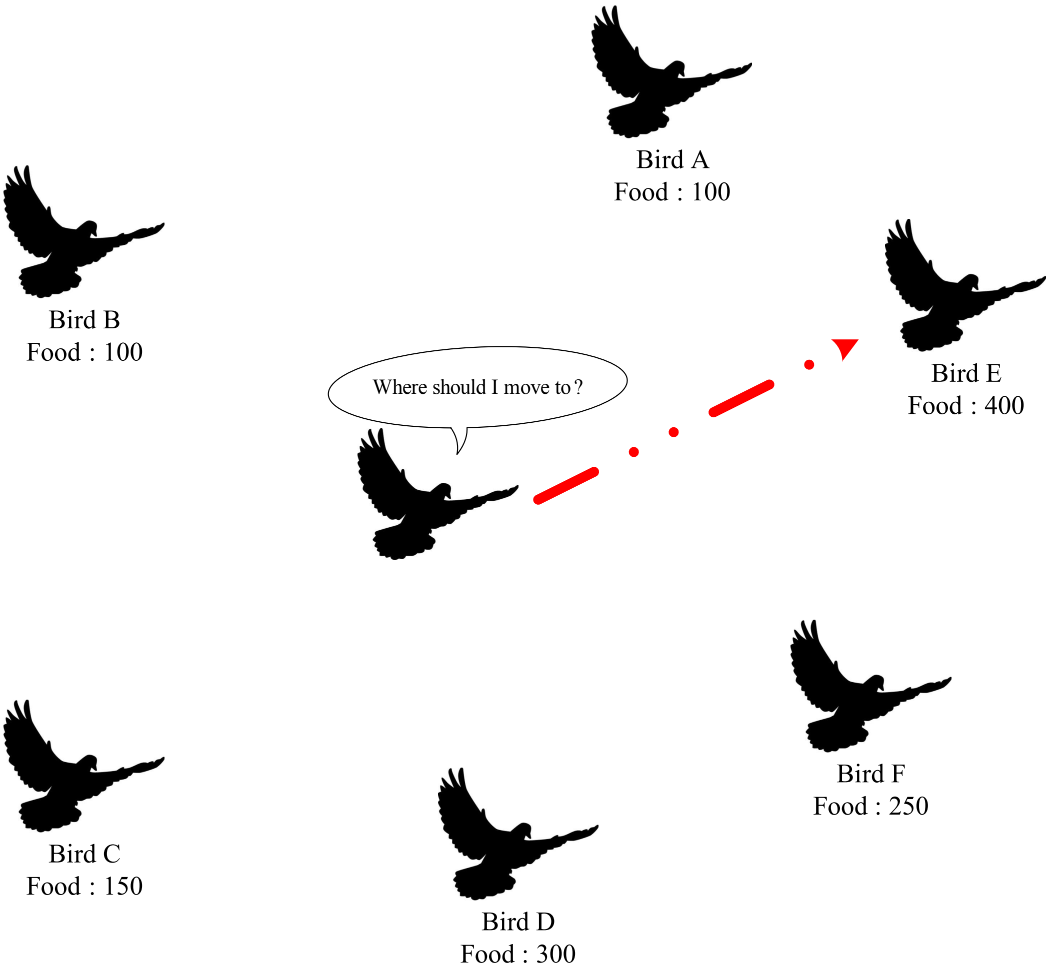


Fig. 2.4 The basic idea of a flock of birds

$$v_{lk}(t+1) = wv_{lk}(t) + c_1r_1[p_{lk} - x_{lk}(t)] + c_2r_2[g_{lk} - x_{lk}(t)] \tag{2.1}$$

where c_1 denotes a learning factors called the coefficient of the self-recognition component and c_2 denotes a learning factors called the coefficient of the social component. w denotes an inertia weight and r_1, r_2 denote random numbers that are uniformly distributed in the interval 0 to 1.

Then the positions of the particle are updated after updating the velocity. The updated positions can be calculated as follows:

$$x_{lk}(t+1) = x_{lk}(t) + v_{lk}(t+1) \tag{2.2}$$

where l and k denote the number of particles and the dimension of the particles, respectively. Next, after updating the position, the turbulence operator is used. Then, after all processes have finished, the set of gBest is updated. Finally, when the minimum/maximum objective function error is satisfied or the maximum limit of iteration is found, MOPSO terminates its processes.

An example of position updates and velocity in particle swarm optimization is shown in figure 2.5. The procedures of PSO can be implemented as follows:

Step 1 (Initialize): Set the parameters of the particle swarm.

Step 2: Generated randomly the particles (position and velocity vector) and calculated the fitness value .

Step 3: Set each particle's *pBest* to the particle position.

Step 4: Collect the set of *gBest* by choosing the particle position with the best fitness value.

Step 5: Improve the position of particle by calculating the particle's updated velocity, the particle's position and the fitness value, respectively.

Step 6: Check the fitness value by comparing between the new position with *pBest*. If the new position is better than *pBest*, then set a new position as *pBest*.

Step 7: Collect the *gBest* by choosing the particle position with the best fitness value.

Step 8: Check the fitness value error or the maximum limit of the number of iterations. If the fitness value error is not satisfied or the maximum limit of the number of iterations is not found go to *Step 3*, else next step.

Step 9: Report the *gBest* as the result.

2.5 Genetic Algorithm (GA)

The genetic algorithm (GA) is described in this section. GA is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. Figure 2.6 shows the genetic optimization process. In the beginning, the initial population (P^0) randomly generated μ individuals. In each generation (g), a pair of individuals who have high fitness value, are chosen from the population P^g by using a selection operator, for example, a binary tournament randomly takes two individuals from the population P^g, then compare them based on fitness values, finally chooses the one with the best fitness as a parent. Next, two offspring (q_c^1 and q_c^2) are reproduced by two parents (p_1 and p_2) by using a crossover operator with crossover rate. Then, each offspring is mutated by using a mutation operator with mutation rate. The three operators (selection, crossover, and mutation operator) repeatedly until the maximum number of offspring size is found. Next, the offspring ($|Q^g|$) is combined with

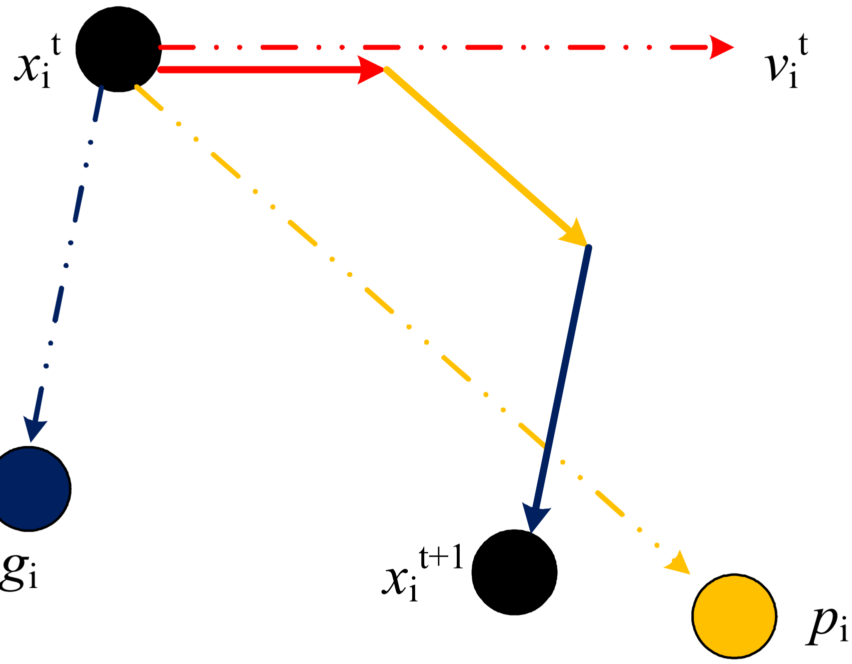


Fig. 2.5 Example of position updates and velocity in particle swarm optimization

the population P^g. Then, the top μ individuals are selected by using a selection operator as the population of next generation (P^{g+1}). When the number of the generations (g) of GA reaches its maximum limit (g_{max}), GA terminates its genetic optimization process.

However, recently, the optimization problem is not considered only single-objective optimization problem. There are many optimization problem that considered as multi-objective optimization problem. Therefor, a multi-objective genetic algorithm are presented for solving the multi-objective problem. Figure 2.7 shows procedure of a multi-objective genetic algorithm (MOGA). The MOGA is separated into four processes as follows:

2.5.1 An initialize process

In this process, the genetic algorithm randomly generated current individual values in population. The individual values will regenerate until the number of individuals reaches the predefined value M.

2.5.2 Selection process

In this process, two individuals in population are selected become parents. The domination ranking technique with objective or constraint domain is applied as a selection operator. In

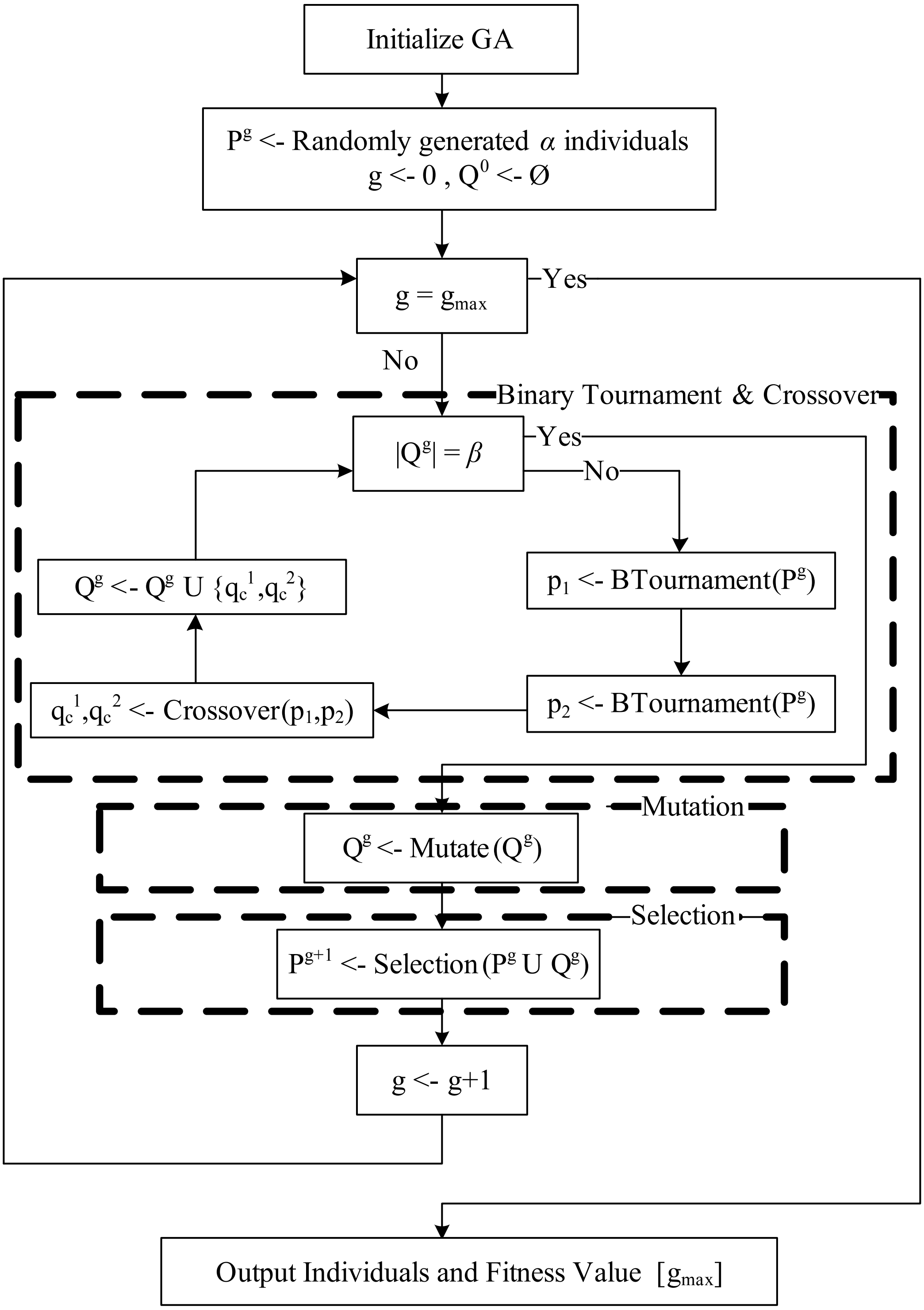


Fig. 2.6 Genetic Optimization Process

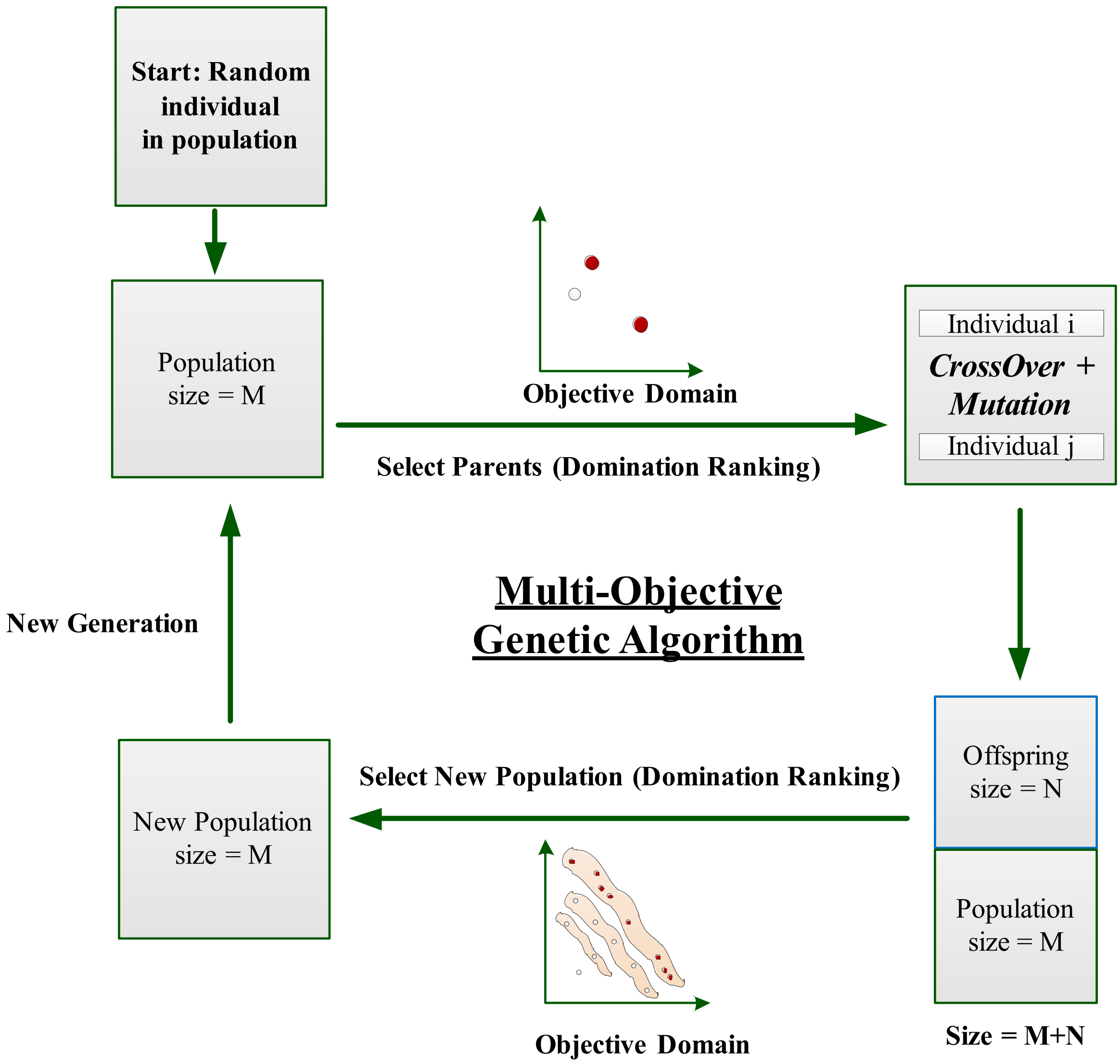


Fig. 2.7 Multi-objective Genetic Algorithm

the beginning, domination ranking technique will randomly takes two individuals (I_a, I_b) and plot their fitness value in objective domain. For example, if I_b dominate I_a, I_b is chosen as parent and drive this step again to find remaining parent. Moreover, the constraint-dominance operator is presented as follows.

Constraint-dominance operator

The fitness is used in selection operator in MOGA. It shows that an individual is inferior or superior to the others. It is determined by relationships among individuals. The rank of

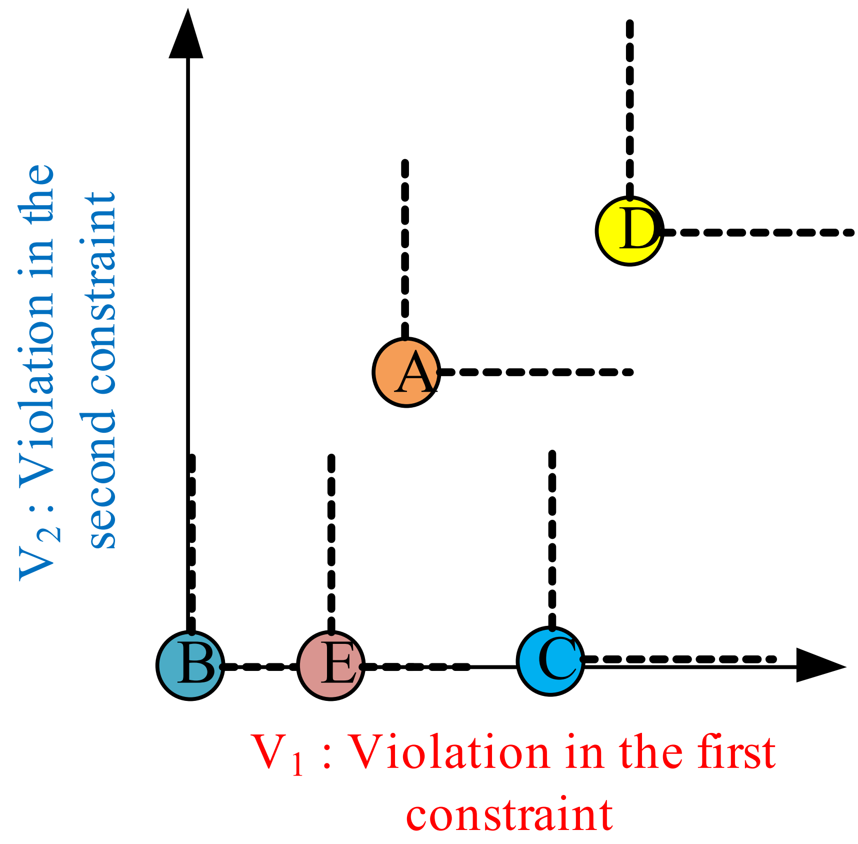


Fig. 2.8 An example of calculating constraint-dominance fitness

individuals based on the constraint violation and objective values. Individual X_i is said to dominate X_j with respect to constraints violation if:

$V_k(X_i) \leq V_k(X_j)$ for all $k \in 1,2,3,...,m$ and $V_k(X_i) < V_k(X_j)$ at least one $k \in 1,2,3,...,m$

where $V_k(X_i)$ denotes the violation that X_i yields in the $k-th$ constraint. m is number of constraints. Individual X_i is said to dominate X_j with respect to objective values if:

$F_l(X_i) \leq F_l(X_j)$ for all $l \in 1,2,3,...,n$ and $F_l(X_i) < F_l(X_j)$ at least one $l \in 1,2,3,...,n$

$F_l(X_i)$ denotes the objective value that X_i yields in the $l-th$ objective. n is number of objectives.

Individual X_i is said to constraint-dominate X_j if:

1. X_j does violate at least one constraint but X_i does not,
2. Both X_i and X_j violate at least one constraint, but X_i dominate X_j with respect to constraint violation, or

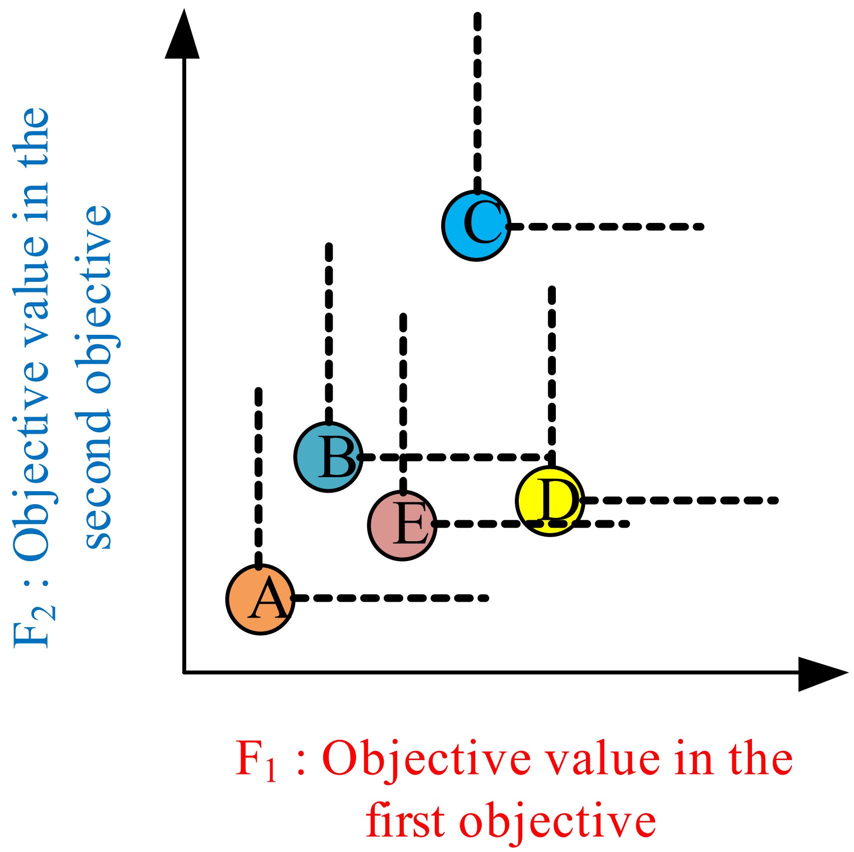


Fig. 2.9 An example of calculating constraint-dominance fitness

3. Both X_i and X_j do not violate any constraints, but X_i dominate X_j with respect to objective values.

Fitness is calculated for each individual (X_i) as follows.

$$Fitness(X_i) = \mu - d' \tag{2.3}$$

Where μ denotes the population size and d' denotes the number of individuals that constraint-dominate X_i. Thus, non-dominated individuals have the highest fitness.

Figure 2.8 and 2.9 show an example of calculating constraint-dominance fitness. Five individuals are considered in constraint-dominance including A, B, C, D, and E. The fitness of D is one because there are four individuals that constraint-dominate D. The fitness of B is five because no individual constraint-dominate B. The fitness of C and A is three because A and C have the same ranking. However, if it is necessary to select between individual A and C, it can be calculated new fitness value for A and C by using an objective domain. Thus, the new fitness of A is five because there is no individual that objective-dominate A and the new fitness of C is one because there are four individuals that objective-dominate C.

2.5.3 Genetic operations process

In this process, The two parents from parent selection process, reproduce two children by using a crossover operator. There are many crossover operators such as one-point crossover, two-point crossover, and simulated binary crossover (SBX). A one-point crossover is presented as an example. In the one-point crossover, the parents individuals (l_a, l_b) are separated into two pieces, then these pieces again. After that, there are two new children, called offspring. However, if there is only crossover operator, the solution may toward to local optimal solution.

Therefore, mutation operator is applied to scatter the solution from local optimal solution. There are many mutation operators such as uniform random mutation, non-uniform mutation, and polynomial mutation. In the mutation operation, a new value randomly and instead of the an old value in gene of offspring.

2.5.4 Population selection process

In this process, the offspring set is combined with the current population set. Then, the selection operator is applied to choose the best set of individuals as a new population for next generation. The GA process will repeat until termination criteria. The termination criteria are a predefined number of iterations or until a predefined fitness value is not reached. The finalize result of GA operation is expected toward global optimal solution.

2.6 Literature Review

This section is divided into three sub-sections that are (1) literature review of optimization in cloud Computing, (2) literature review of optimization in micro-grids, and (3) literature review of the local outlier detection in geo-social points.

2.6.1 Optimization in Cloud Computing

Cloud computing resource management problems have been proven as NP-hard, such as scheduling, load balancing, and mapping. These problems could be solved by using a meta-heuristic algorithm, such as evolutionary algorithm, particle swarm optimization, genetic algorithm, ant colony optimization, and neural networks. The related work on the resource management optimization problem in cloud computing is presented in this section.

Evolutionary algorithms have been proposed in several studies to solve the resource management optimization problems [43, 39, 71]. A multi-objective genetic algorithm and a greedy heuristic algorithm are proposed in [43] to reduce energy consumption which consists of the energy consumption, CO_2 emissions and deployment cost. The MOGA outperforms the greedy heuristic in terms of energy consumption and CO_2 emissions.

Moreover, evolutionary algorithms have been used in several studies to optimize an optimization problems in a wireless sensor network (WSN) [80, 47, 16]. In [80], the authors proposed a GA to optimize the connectivity of sensor nodes and the coverage in a k-covered hotspot area. GA also was proposed to solve a node placement problem in WSN in [47]. The objectives consist of maximizing the coverage and the network lifetime. Moreover, evolutionary algorithm with new genetic operators was proposed in [16]. The proposed EA used to solve the problem using high-dimensional quality of service (QoS) optimization for power grid communication networks and new genetic operators consist of *aging* operator, *age-based crossover* operator, *aged-based mutation* operator, and *fitness-based crossover* operator.

Otherwise, novel algorithms/frameworks have been proposed in several studies to solve optimization problems in wireless networks [74, 52]. An optimal forwarding problem in mobile ad-hoc networks (MANETs) was considered in [52]. The problem solved by using a novel game-theoretic framework to examine the optimal forwarding problem of a two-hop f-cast relay algorithm in MANETs. The relationship between the forwarding behaviors and the final throughput capacity were indicated by the proposed framework. In [74], the authors proposed a complex alliance strategy with multi-objective optimization of coverage. The proposed strategy could improve the effectiveness of sensor node coverage and the network lifetime in WSNs. A novel scheme also was proposed to calculate coverage expectation and proportionality.

Moreover, novel models/frameworks have been proposed in several studies to solve problems in cloud computing [77, 51, 13, 24, 50]. The authors in [24] proposed a multi-cloud architecture, called MCES, which can tolerate pressure more than single cloud service under emergency environment. Smart evacuation services are deployed in multi-cloud providers. The authors in [50] proposed an online-deduplication mechanism based on an energy efficient storage system to manage virtual machine (VM) storage. A deduplication selection algorithm also was designed for minimizing the storage energy consumption. The results show that the proposed mechanism is able to reduce both of the energy consumption and the redundant data blocks without service interruption. In [13], a cloud-centric multi-level authentication as a service was proposed to secure public safety networks in the cloud and IoT devices. To show the effectiveness of the proposed approach, scalability and time constraints are considered.

The authors in [77] proposed CSAM-IISG which is an imperfect information Stackelberg game with hidden Markov model (HMM) to increase the profit of both the service providers and the applicant. The novel model was proposed for a cloud resource allocation model in a cloud computing environment. A service framework and a pricing strategy for a multi-cloud environment were proposed in [51]. The proposed framework is able to provide streaming big data computing service and maximize the profits of the multi-cloud intermediary. The results show that the profits from the proposed strategy are higher than the other.

Moreover, novel models/frameworks have been proposed in several studies to solve an optimization problem in cloud brokering system as shown in table 2.1. In [71], the authors proposed a framework called CLOUD Resource Broker with particle swarm optimization-based resource allocation and deadline-based job scheduling to optimize the multi-objective optimization problem. The multi-objective consists of minimize the execution time and cost and maximize the number of jobs that are completed within a deadline. A genetic algorithm (GA), ant colony optimization (ACO) and rank-based allocation (RBA) mechanism were used to compare with the proposed framework to show the performance.In [39], the author proposed a novel evolutionary algorithm, which is a parallel hybrid evolutionary algorithm, to optimize the broker profit in the cloud system. The proposed algorithm was compared with a greedy heuristic algorithm to show the performance. The results show that, in terms of the profit values, the proposed algorithm outperforms the greedy heuristic algorithm. Moreover, the authors in [79] proposed A grouping genetic algorithm (GGA), which works with fuzzy multi-objective evaluation, to minimize the wasted resources, power consumption and the temperature of a cloud system. A control system also was proposed to manages the mapping of workloads to virtual machines (VMs) and VMs to physical resources. Four bin-packing algorithms and two single-objective approaches were used to compare with the proposed algorithm to show the performance.

Several studies proposed particle swarm optimization for solving various problems in cloud computing[40, 60]. A PSO-based heuristic scheme was proposed in [60] to minimize communication and computation costs for work-flow scheduling in cloud environments. A greedy best resource selection algorithm was used to compare with the proposed algorithm for showing the performance. In [40], an adaptive power-aware virtual machine provisioner was proposed as a novel meta-scheduler. A self-adaptive particle swarm optimization (SAPSO) was used to optimize a virtual machine placement problem. Five experiments were used for comparing among standard PSO, multi-ensemble particle swarm optimization and SAPSO.

Table 2.1 Optimization in cloud brokering

Author or Method	System	Resources	Objectives	Multi-Objective	Technique
Legillon et al. [49]	Multi-cloud	Virtual Machine Type	Cost	No	Genetic Algorithm
CDOXplorer [29]	Multi-cloud	Virtual Machine Type	Response Times, Costs, and Service Level Agreement Violations	Yes	Genetic Algorithm
MOGA-CB [42]	Cloud	Virtual Machine Type	Response Time, Cost	Yes	Genetic Algorithm
Kessaci et al. [43]	Cloud Federation	Clouds (Data Centers)	Energy, CO_2, Cost	Yes	Genetic Algorithm
Iturriaga et al. [39]	Multi-cloud	Reserved Virtual Machine Instances	Broker Revenue	No	Genetic Algorithm + Simulated Annealing
CLOUDRB [71]	Multi-cloud	Cloud Resources	Makespan, Cost, Number of Rejected Requests, Successful Jobs, User Satisfaction	Aggregation	Particle Swarm Optimization

The five experiments consist of a performance comparison of detecting and tracking an optimal target server, the number of failures in VM provisioning, the rate of failure in VM provisioning with fixed and variable evaporation factors, the impact of exploiting power-saving states along with dynamic voltage frequency scaling (DVFS) in VM provisioning, and a performance comparison of the power trade-offs

Moreover, ant colony optimizations have been proposed to solve various problems in cloud computing [30, 55, 28]. A workload placement problem in cloud computing was investigated in [28]. The authors proposed a multi-dimensional bin-packing problem. Then, ant colony optimization was proposed to compute the placement dynamically. A greedy algorithm, such as First-Fit Decreasing, was used to compare with the proposed algorithm for showing the performance. In [55], the authors proposed a novel cloud scheduler based on ant colony optimization. The proposed algorithm was used to execute parameter sweep experiments in clouds. A minimizing the weighted flowtime and makespan also was formulated. A virtual machine placement problem in a cloud computing environment was investigated in [30]. The authors proposed a multi-objective ant colony algorithm to improve the power efficiency and resource utilization. A multi-objective genetic algorithm and two single-objective algorithms were used to compare with the proposed algorithm.

2.6.2 The Local Outlier Detection in Geo-Social Points

Many outlier detection algorithms have studied in knowledge discovery research communities for recognizing unusual characteristics of systems. In general, the outlier detection approach can be classified into five categories (As shown in Table 2.2): *distribution*, *depth*, *clustering*, *distance*, and *density*. In [46], a new outlier detection algorithm based on the DB-SCAN (Density Based Spatial Clustering of Applications with Noise) clustering algorithm is proposed for spatio-temporal outlier detection in large databases. The algorithm consists of three steps: clustering, checking spatial neighbors and checking temporal neighbors. The algorithm has discovered outliers according to the spatial, temporal and non-spatial values of data objects. Outstretch in [78] discovers the top-k outlier for each time period by extending the Exact-Grid and Approx-Grid algorithm. Comparing with the usage of the distance metric, the space-partitioning and density approaches are suitable to apply a large dataset. However, they have a critical issue about dependency of the partitioning. Therefore, we take account of an optimization problem to divide the spatio-temporal space for the outlier detection.

The well-known meta-heuristic algorithm is proposed to solve the optimization problem that is genetic algorithm (GA). It has a good ability to making a global search and also preferable to get a good optimal solution in short time. The outlier detection using genetic algorithm is proposed in several researches. [4] proposed two algorithms to detecting out-

Table 2.2 Approaches to outlier detection

Approaches	Descriptions
Distribution-based [78, 9]	•Statistical test is applied to find the data that belong to a particular or not. •The data with low probability are considered as outliers.
Clustering-based [46, 3]	•The clusters of small size are considered as outliers. •The capability of handling exceptions are in some clustering algorithm e.g., CURE [34], DBSCAN [27]. •Discovering clusters is the main objective of the clustering algorithm. It is not considered optimizing outlier detection.
Depth-based [41, 65]	•The different layers of k-d convex hulls are computed in this approach. •The data with smaller depths are considered as outliers.
Distance-based [45, 6]	•The distance measure on the feature space is used to detect the outliers. •K-nearest neighbor method is used in some research to identify the outliers. •The difficulty of finding a particular value in a data set is a problem of this approach.
Density-based [12]	•A local outlier Factor (LOF) is assigned to each data based on local neighborhood density. •The data with high LOF (low density regions) are considered as outliers.

liers in high dimensional problem. The algorithms consist a brute force algorithm and an evolutionary algorithm. The technique observes the density distributions of projections from the data set to find outliers. The new algorithm to detect outliers using genetic algorithm is proposed in [62]. The outliers are considered as the fitness function in the genetic algorithm. The results show that the proposed algorithm is able identifying the accurate outliers of the datasets. However, the kd-tree partitioning is not considered to use with GA to detect outliers.

2.6.3 Optimization in Micro-grids

Recently, the suitable capacity or sizing of BES for optimizing an operation management of micro-grid (OMMG) has been considered in several studied as shown in table 2.3. Micro-grid central controller in the micro-grid system implemented the operation management of micro-grid and has the liability to optimize the MG operation. The one of the optimization problems is an operation management of micro-grid problem for the smart energy manager or micro-grid central controller. In general, the objective of this optimization problem is to minimize the total operation costs problem. There are two groups of several studies as follow:

1. The impact of BES optimal sizing on the operation management of micro-grid problem did not consider.

   In [17], the authors proposed a new smart energy management system to optimize the operation management of micro-grid. The system based on the matrix real-coded genetic algorithm.

2. The BESs optimal sizing and its presence performance on the operation management of micro-grid problem are considered.

   The authors in [56] proposed an appropriate technique of selecting the BES sizing to satisfy a reliability index. In [26], the authors proposed a simulated annealing algorithm to optimize the PV/WT sizing of hybrid energy conversion system with BES. In [18], A modeling language for mathematical programming was proposed to determine the BES sizing for the micro-grid system. Recently, a new method was proposed in [5] to determine the BES optimal sizing. The authors considered the primary frequency control of the micro-grid system. The system consists of FC, MT, diesel generator and PV system. It can conclude that the BES sizing and its role in MG system are a topic of interest in many researches.

Moreover, the real-time optimal control of a large number of DGs in smart distribution grids was considered in [76]. A consensus-based dimension-distributed computational intelligence technique also was proposed to optimize the problem. In [81], a resource allocation problem in Device-to-Device communications was considered as a non-cooperative game. A distributed interference-aware energy-efficient resource allocation algorithm also was proposed for maximizing each user equipment's energy efficient in an interference-limited environment. The authors in [72] proposed a game theoretic resource allocation scheme to allocate the resource of mobile social users through brokers for media cloud. A four-stage Stackelberg game was used to formulate the interactions among mobile social users, brokers and media cloud.

Table 2.3 Optimization in Micro-grids

Author or Method	System	Objectives	Multi-Objective	Technique
Baziar et al. [10]	Micro-grid	Operation Cost	No	Self Adaptive Modified Bat Algorithm
Sharma et al. [67]	Micro-grid	Operation Cost	No	Quasi-Oppositional Swine Influenza Model Based Optimization with Quarantine
Sharma et al. [68]	Micro-grid	Operation Cost	No	Grey Wolf Optimization
Hosseini [36]	Micro-grid	Operation Cost, Emission Pollutant	Yes	Quantum-Inspired Evolutionary Algorithm

Chapter 3

Cloud Brokering Systems for Connected Internet of Things

This chapter describes problem statements and methodologies that will be used in this study. The chapter starts with a problem statement of cloud brokering systems for connected Internet of Things and methodology for solving the optimization problem of cloud brokering systems, respectively. Next, simulation configuration and results are described. Finally, a conclusion of the study in cloud brokering systems for connected Internet of things is presented.

3.1 Cloud Brokering System

The problem statement of cloud brokering system in cloud computing is described in this section. Infrastructure as a Service is considered as the cloud model. The model is shown in figure 3.1. There are one cloud broker, M cloud service providers and N clients in the model. The goal of a cloud broker is that to find the best deal among of cloud service providers and clients. A set $S = s_1, ..., s_M$ denotes M service providers in the model and a set $U = u_1, ..., u_N$ denotes N clients in the model. Assumed that there is a limitation of a capacity of each cloud service provider to handle requests from clients. Moreover, the total number of handling requests of each cloud service provider must be bigger than the number of requests from a client. A binary variable b_{ij} with $i = 1, ..., N$ and $j = 1, ..., M$ is used to describe a process of cloud service providers as follows:

$$b_{ij} = \begin{cases} 1 & , if\ s_j \text{ handles the request from } u_i \\ 0 & , otherwise \end{cases} \tag{3.1}$$

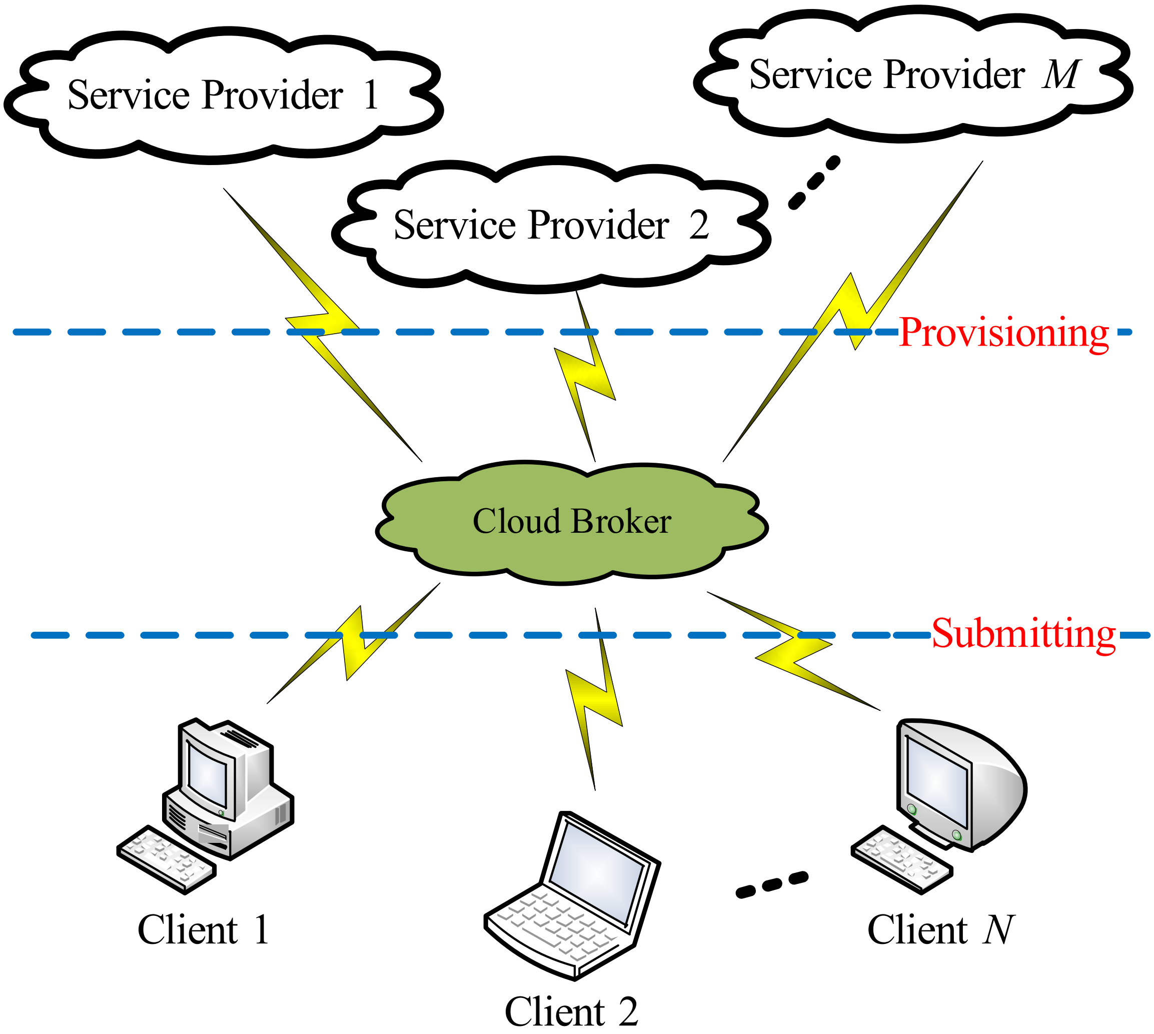


Fig. 3.1 Cloud Brokering Model. The model consists of N clients, M cloud service providers and one cloud broker.

Generally, when a client submits requests to cloud broker and cloud service provider, he/she is expected to complete his/her job in a minimal time. Therefore, a response time of requests from clients is considered as an objective. Assumed that L_{ij} denotes a latency between client i and service provider j. The latency can be calculated as follows:

$$L_{ij} = CT - AT \tag{3.2}$$

, where CT and AT denote the current time and the arrival time of a request from client i at service provider j, respectively. Next, the cloud service providers have to spend time T_j to execute requests after they receive requests from clients. Therefore, the minimization of the response time (RT) of requests is considered as the first objective and can be formulated as follows:

$$RT = \sum_{i=1}^{N}\sum_{j=1}^{M} b_{ij}(L_{ij} + T_j) \tag{3.3}$$

Next, clients sometimes can submit their request to the cloud service provider through a cloud broker. Then, the cloud broker will find the best cloud service provider to the satisfaction of the client. However, the cloud broker has expected a profit from the process. Thus, the cloud broker's profit is considered as an objective. Assumed that P_i denotes a price from client i and C_j denotes a cost of cloud service provider j. Therefore, the maximization of the profit (P) of the cloud broker is considered as the second objective and can be formulated as follows:

$$P = \sum_{i=1}^{N}\sum_{j=1}^{M} b_{ij}(P_i - C_i) \tag{3.4}$$

Finally, the cloud service provider is expected to finish a job with minimal energy consumption when executing the request from the client. Thus, the energy consumption in cloud computing is considered as an objective. Assumed that E_j denotes an energy consumption that the cloud service provider j used to execute the job. Therefore, the minimization of the total energy consumption of all cloud service providers is considered as the third objective and can be formulated as follows:

$$E = \sum_{i=1}^{N}\sum_{j=1}^{M} b_{ij} \cdot E_j \tag{3.5}$$

There are three objective for cloud brokering system: minimize the response time and the total energy consumption and maximize the profit of the cloud broker. Thus, the optimization problem of the cloud broker is considered as a single-objective optimization problem and a multi-objective optimization problem. The single-objective optimization problem is described in Section 3.2 and the multi-objective optimization problem is described in Section 3.3.

3.2 Single-Objective Optimization Problem

At first, the three objectives are considered as a single-objective optimization problem by using a weighting factor. Thus, the utility function of the cloud brokering system can be formulated as follows:

$$U = \omega_1 RT + \omega_2 E - \omega_3 P \tag{3.6}$$

, where ω_1 denotes a weighting factor of the response time of requests, ω_2 denotes a weighting factor of the profit of the cloud broker, and ω_3 denotes a weighting factor of the total energy consumption of the system. Note that the summation of weights is equal to 1. It means that $\omega_1 + \omega_2 + \omega_3 = 1$. RT denotes the response time of requests of the system which can be calculated by using Eq. (3.3). E denotes the total energy consumption of the system which can be calculated using Eq. (3.5). P denotes the profit of the cloud broker which can be calculated using Eq. (3.4). Therefore, the single-objective optimization problem of the cloud brokering system is to minimize the utility function as follows:

$$\underset{U,S}{\text{Minimize}} \; U = \omega_1 RT + \omega_2 E - \omega_3 P \tag{3.7}$$

$$\text{Subject to} \; \sum_{i=1}^{N} x_{ij} \leq A_j \tag{3.8}$$

$$\sum_{j=1}^{M} b_{ij} \geq R_i \tag{3.9}$$

$$A_j, R_i \geq 0 \quad ; i = 1, ..., N; j = 1, ..., M \tag{3.10}$$

where R_i and A_j denote the number of requests from client i and the capacity of the service provider j, respectively. Constraint (3.8), constraint (3.9) and constraint (3.10) represent that the total number of requests from clients to service provider j is less than or equal to the capacity of service provider j, the total number of requests from client i to all service providers is greater than or equal to the number of requests from client i, and the number of requests and the capacity are positive values, respectively.

However, the best value for the weights ω_1, ω_2 and ω_3 are very difficult to find. Therefore, the three objectives are considered as a multi-objective optimization problem and the multi-objective optimization problem is described in the next section.

3.3 Multi-objective Optimization Problem

The three objective of the cloud brokering problem is considered as a multi-objective optimization problem to fix the weight values assignment problem. The multi-objective optimization problem of the cloud brokering can be stated as follows:

$$\underset{U,S}{\text{Minimize}}\, RT = \sum_{i=1}^{N}\sum_{j=1}^{M} b_{ij}(L_{ij}+T_j) \tag{3.11}$$

$$\underset{U,S}{\text{Minimize}}\, E = \sum_{i=1}^{N}\sum_{j=1}^{M} b_{ij}\cdot E_j \tag{3.12}$$

$$\underset{U,S}{\text{Maximize}} P = \sum_{i=1}^{N}\sum_{j=1}^{M} b_{ij}(P_i - C_i) \tag{3.13}$$

$$\text{Subject to } \sum_{i=1}^{N} x_{ij} \leq A_j \tag{3.14}$$

$$\sum_{j=1}^{M} b_{ij} \geq R_i \tag{3.15}$$

$$A_j, R_i \geq 0 \quad ; i = 1,...,N; j = 1,...,M \tag{3.16}$$

where R_i and A_j denote the number of requests from client i and the capacity of the service provider j, respectively. Constraints (3.14), (3.15) and (3.16) have the same meaning as the constraints (3.8), (3.9) and (3.10) of the single-objective optimization problem, respectively.

Since the scheduling problem has been proven as an NP-hard problem [32], it can conclude that the cloud brokering problem is NP-hard problem. The NP problem can be solved by using a meta-heuristic algorithm (evolutionary algorithm). Thus, for solving the cloud brokering problem, a multi-objective particle swarm optimization (MOPSO) is chosen. The details of MOPSO is described in the next section.

3.4 Multi-Objective Particle Swarm Optimization for Cloud Brokering Problem

Since a multi-objective optimization problem is considered for cloud brokering problem, the standard PSO needs to be modified for solving the problem. The authors in [64] presented a multi-objective particle swarm optimization (MOPSO). The goal of MOPSO is a seeking a Pareto-set (a set of optimal solutions).

In the beginning, the initial swarm is randomly generated. Then, non-dominated particles from the swarm are chosen to initialize a set of gBest. Next, an external archive store the set of gBest. Then, a gBest is selected and the positions of the particles are updated at each iteration. The updated velocity vector can be calculated as follows:

$$v_{lk}(t+1) = wv_{lk}(t) + c_1r_1[p_{lk} - x_{lk}(t)] + c_2r_2[g_{lk} - x_{lk}(t)] \tag{3.17}$$

where c_1 denotes a learning factors called the coefficient of the self-recognition component and c_2 denotes a learning factors called the coefficient of the social component. w denotes an inertia weight and r_1, r_2 denote random numbers that are uniformly distributed in the interval 0 to 1.

Then the positions of the particle are updated after updating the velocity. The updated positions can be calculated as follows:

$$x_{lk}(t+1) = x_{lk}(t) + v_{lk}(t+1) \tag{3.18}$$

where l and k denote the number of particles and the dimension of the particles, respectively. Next, after updating the position, the turbulence operator is used. Then, after all processes have finished, the set of gBest is updated. Finally, when the minimum/maximum objective function error is satisfied or the maximum limit of iteration is found, MOPSO terminates its processes.

The procedures of MOPSO can be implemented as follows:

Step 1 (Initialize): Set the parameters of the particle swarm.

Step 2: Generated randomly the particles (position and velocity vector) and calculated the fitness value .

Step 3: Set each particle's *pBest* to the particle position.

Step 4: Collect the set of *gBest* by choosing the particle position with using non-dominated sorting based on fitness value.

Step 5: Initialize an external archive by adding the set of *gBest*.

Step 6: Improve the position of particle by calculating the particle's updated velocity, the particle's position and the fitness value, respectively.

Step 7: Apply the turbulence (which is a mutation operator that operates on the velocity value) operators.

Step 8: Check the fitness value by comparing between the new position with *pBest*. If the new position is better than *pBest*, then set a new position as *pBest*.

Step 9: Collect the *gBest* by choosing the particle position with using non-dominated sorting based on fitness value.

Step 10: Update an external archive by adding the set of *gBest*.

Step 11: Check the fitness value error or the maximum limit of the number of iterations. If the fitness value error is not satisfied or the maximum limit of the number of iterations is not found go to *Step 6*, else next step.

Step 12: Report the external archive as the results.

3.5 Computational Complexity of MOPSO

The computational complexity of MOPSO is described in this section. Let a swarm size be N, an external archive size be L and the number of objectives be M. The complexity of MOPSO is mainly influenced by a variety computation operation such as calculating the updated velocity of particle, calculating the updated position of particle and calculating the fitness function as well as a non-dominated sorting process. At first, $M(N+L)$ comparisons are needed to check the non-dominance of particles based on fitness value within $N+L$ particles and M objectives. Thus, the complexity of this process will be $O(M(N+L)^2)$ in the worst case. Next, the sorting based on fitness value for only the external archive requires $O(MLlog(L))$ computations. Therefore, the overall worst-case complexity of MOPSO is $O(M(N+L)^2)$ in the worst-case scenario with $N+L$ elements in the archive.

3.6 Simulation Configuration and Results of MOPSO

At first, the setup of MOPSO simulation for cloud brokering system is presented. Then, an analysis of the performance of MOPSO and a comparison with other two algorithms are presented. MOPSO is compared with well-known GA and random search algorithm.

3.6.1 Simulation Configuration

The simulated cloud brokering system is assumed that there are seven service providers, five clients, and one cloud broker. The minimum energy consumption of the cloud and response time and maximum profit for the cloud broker are considered. The instances consist of three types: extra-large, large and small. The parameters for cloud brokering system is shown in Table 3.1. The prices and average execution time of instance depend on the type of the instance.

Table 3.1 Parameters for Cloud Brokering System

Parameter	Value
Instance type	small, large, extra-large
Hourly price	0.1, 0.125, 0.143
Instance prices	Amazon EC2 pricing history
Average execution time(s)	980 ± 71, 616 ± 61, 697 ± 13

jMetal is used to execute three algorithms in this simulations. The three algorithms are MOPSO, a well-known genetic algorithm, and random search algorithm. The jMetal is a framework based on an object-oriented JAVA for simulating meta-heuristics with multi-objective optimization problems. The MOPSO simulation results are compared with NSGA-II [23] which is a well-known genetic algorithm and random search algorithm. NSGA-II is used as the well-known genetic algorithm for solving the multi-objective optimization problem because of elitist approach and parameter-less sharing of the algorithm as well as low computational requirements. The complexity of NSGA-II is $\mathscr{O}(mN^2)$, m and N denote the number of objectives and the number of population size, respectively. The simulated binary crossover (SBX) [20] is applied as a crossover operator in NSGA-II. The simulation configurations for MOPSO and NSGA-II is presented in table 3.2. The number of swarms and archives of MOPSO are 100. The number of population of NSGA-II is 100 populations. The number of iterations and a mutation rate of both algorithms are 250 iterations and $1/n$, respectively. A crossover rate of NSGA-II is 0.9. The average results of 10 runs of each scenario are compared.

Table 3.2 The Simulation Configurations

Configuration	MOPSO	NSGA-II
Number of iterations	250	250
Population Size	-	100
Swarm Size	100	-
Archive Size	100	-
crossover rate	-	0.9
degree of SBX crossover	-	15
mutation rate	$1/n$	$1/n$
degree of polynomial mutation	20	20

3.6.2 Simulation Results

The simulation results is divided into three parts: (1) the simulation results of single-objective particle swarm optimization for solving single-objective optimization problems, (2) the simulation results of MOPSO, NSGA-II and random search algorithm for solving multi-objective optimization problems, and (3) $\mathscr{C}$-metric which represents the performance of the solution of the three algorithms. At first, a comparison of the weight values for the single-objective optimization problem is presented. Next, the best solutions of the three algorithms are presented in each iteration. Finally, the performance of the solutions of the three algorithms is shown in term of the $\mathscr{C}$-metric.

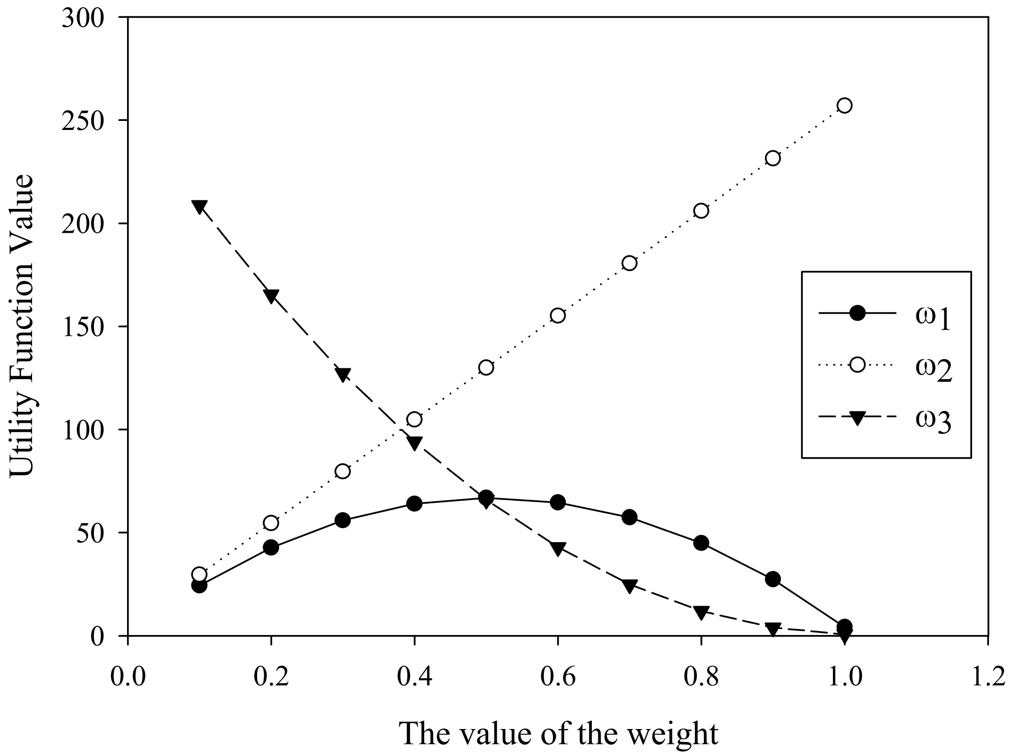


Fig. 3.2 The utility function value of each weight value

Single-Objective Optimization Problem

In this section, the three objectives (to maximize the broker profit and to minimize the energy consumption of the system and response time of users) are considered as a single-objective optimization problem by using the weight values. This process is used to show how difficult to find the best weight value for the single-objective optimization problem. There are three different weight values (ω_1, ω_2 and ω_3) for the single-objective PSO are used to compare the results. In the simulations, the weight values can be calculated as follows:

$$\omega_1 = random(\lambda)/\lambda \tag{3.19}$$

$$\omega_2 = (1-\omega_1)(\omega_1) \tag{3.20}$$

$$\omega_3 = 1-\omega_1-\omega_2 \tag{3.21}$$

where λ denotes a random number which is a positive number by using uniformly distributed.

The best utility function value of each weight value is shown in figure 3.2. The range of weight values is from 0.1 to 1.0 which is used to compare the best utility function values. The results from the figure show that the utility function value in the single-objective optimization depends on the weight value. Therefore, the weight value can be chosen according to an interested objective. However, it is difficult to find the best weight value for the problem.

Multi-Objective Optimization Problem

In this section, the three objectives in the cloud brokering system are considered as a multi-objective optimization problem. MOPSO is used to solve the multi-objective optimization problem. The best solutions of three algorithms (MOPSO, GA, and random search algorithm) are presented to compare the performance of each algorithm.

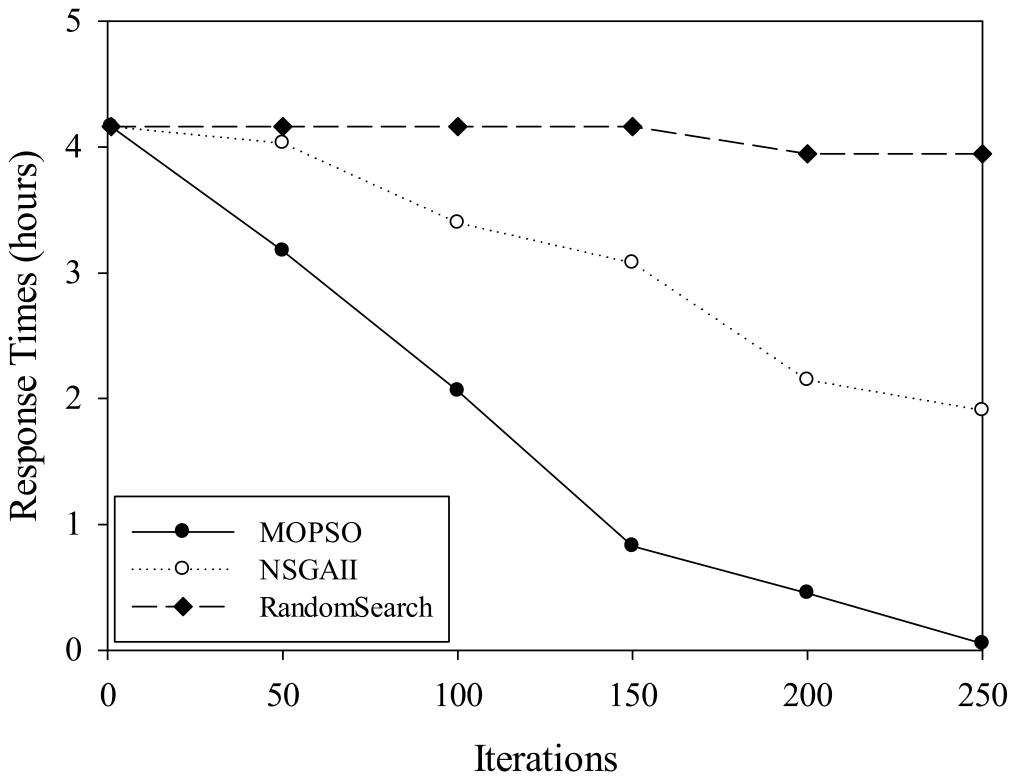


Fig. 3.3 The minimum response time at the end of each iteration

The best solutions of each algorithm are shown in figures 3.3, 3.4 and 3.5 based on 10 independent runs. The minimum response time at the end of each iteration is shown in figures 3.3. The minimum energy consumption of the systems at the end of each iteration is

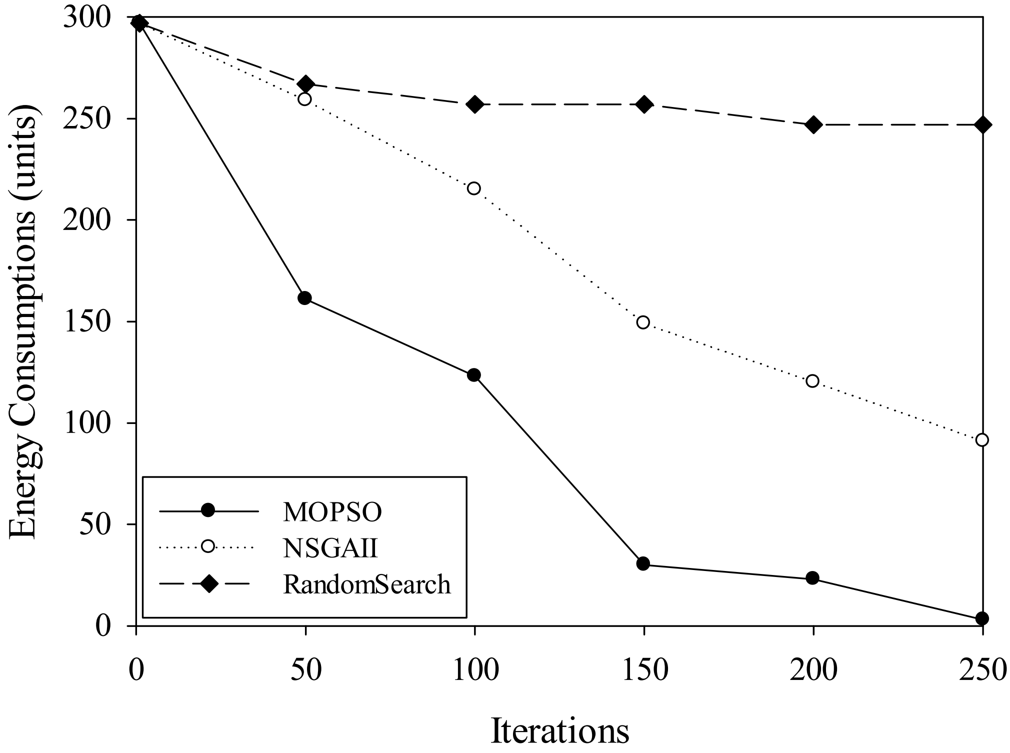


Fig. 3.4 The minimum energy consumption at the end of each iteration

shown in 3.4. The maximum profit of the cloud broker at the end of each iteration is shown in 3.5. The simulation results show that MOPSO success to fine the optimality of three objectives better than NSGA-II and random search algorithm. Since the random process of the random search algorithm has a possibility of leading high values for three objectives, the solutions in term of the profit of cloud broker from random search algorithm are similar to the results from MOPSO.

Moreover, figure 3.6, 3.7 and 3.8 show the minimum total response time and energy consumption, and the maximum profit of the cloud broker in the cloud brokering system at the end of each iteration with 500 max iterations, respectively. The figures show the results from three algorithms: MOPSO, NSGA-II, and random search algorithm. The results in figure 3.6 shows that the minimum total response time of MOPSO and NSGA-II less change after the 300 iterations. Next, the results in figure 3.7 shows that the minimum energy consumption of those three algorithm less change after the 300 iterations. Last, the results in figure 3.8 shows that the maximum profit of the cloud broker of those three algorithm less change after the 250 iterations.

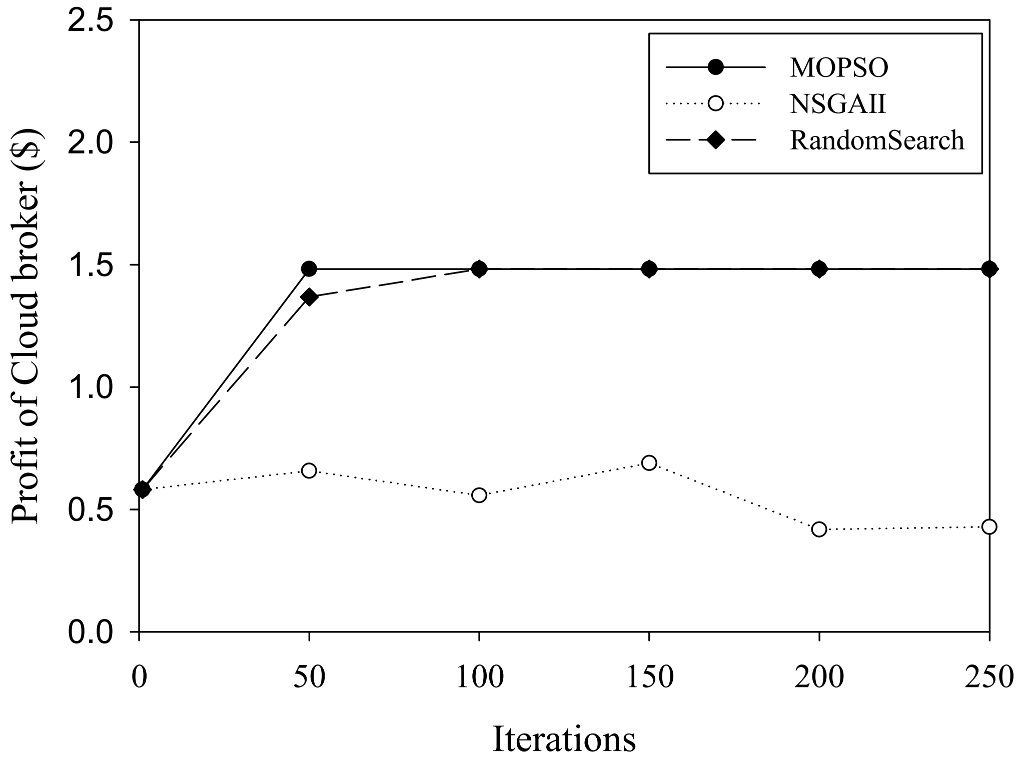


Fig. 3.5 The maximum profit of the cloud broker at the end of each iteration

Moreover, there is a similarity in terms of population-based search approaches and depend on information sharing among members (particle or individual) in the population (or swarm), between MOPSO and NSGA-II to enhance their processes. MOPSO and NSGA-II have a high ability for global searching as well. However, MOPSO could be implemented simply with not many operators and/or parameters. On the other hand, there are many operator such as selection, crossover and mutation operator, and many parameters such as the number of population size, crossover and mutation rate, in NSGA-II. Therefore, NSGA-II has a complexity and a slower convergence higher than MOPSO. Although, MOPSO has a faster convergence, it has a possibility the solutions of MOPSO fall into local optima higher than NSGA-II.

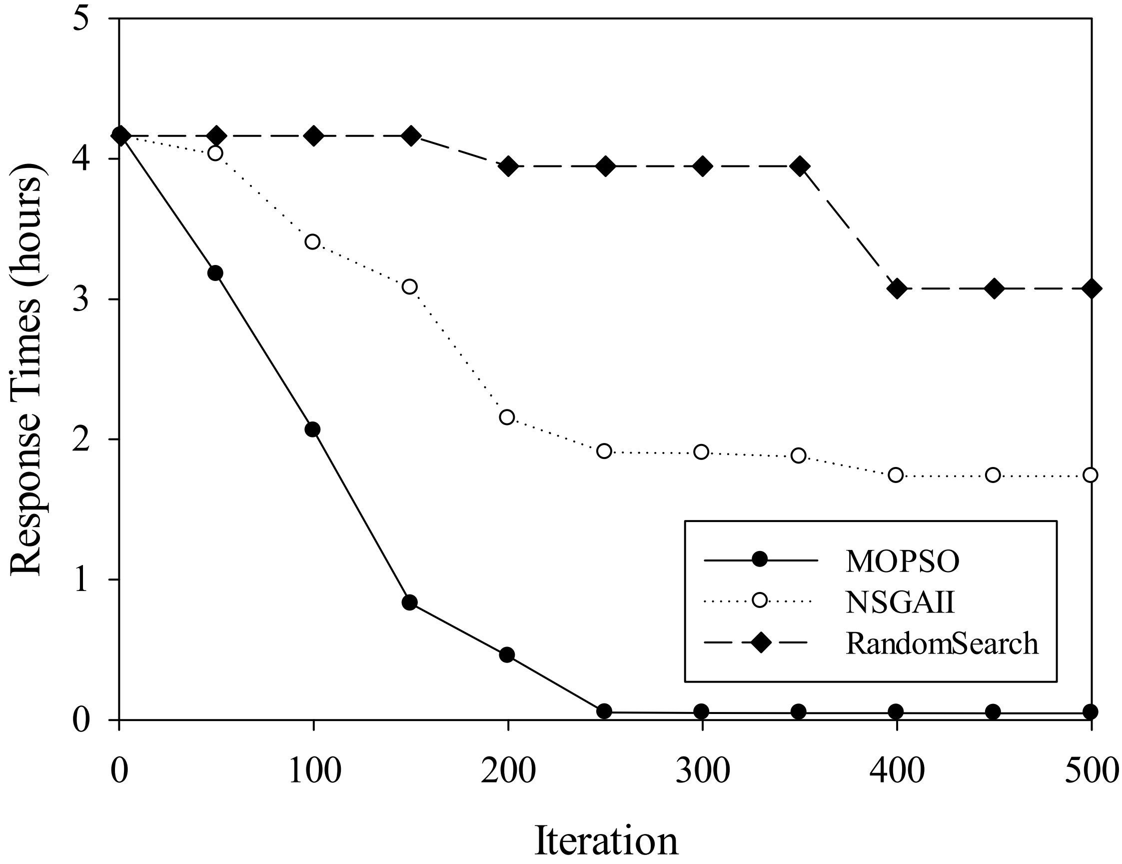


Fig. 3.6 The minimum response time in case of 500 max iterations

$\mathscr{C}$-metric

$\mathscr{C}$-metric [82] is used as a performance metric to show that how the solutions (individuals) of algorithm outperform the solutions from another algorithm. The $\mathscr{C}$-metric between algorithm A and B is represented by $\mathscr{C}(A,B)$. Moreover, $\mathscr{C}(A,B)$ can be calculated as follows:

$$\mathscr{C}(A,B) = |\{b \in B \mid \exists a \in A : a \succ b\}|/|B|, \tag{3.22}$$

where an operator $\succ$ is the dominating algorithm. For example, $b \succ a$, this means that individual (solution) b dominate individual a. A range of $\mathscr{C}(A,B)$ is between 0 to 1. Therefore, if $\mathscr{C}(A,B)$ is equal to 1, this means that there is at least one individual in A which dominates all individuals in B. On the other hand, if there is no individual in B which is dominated by any individual in A, $\mathscr{C}(A,B)$ is equal to 0.

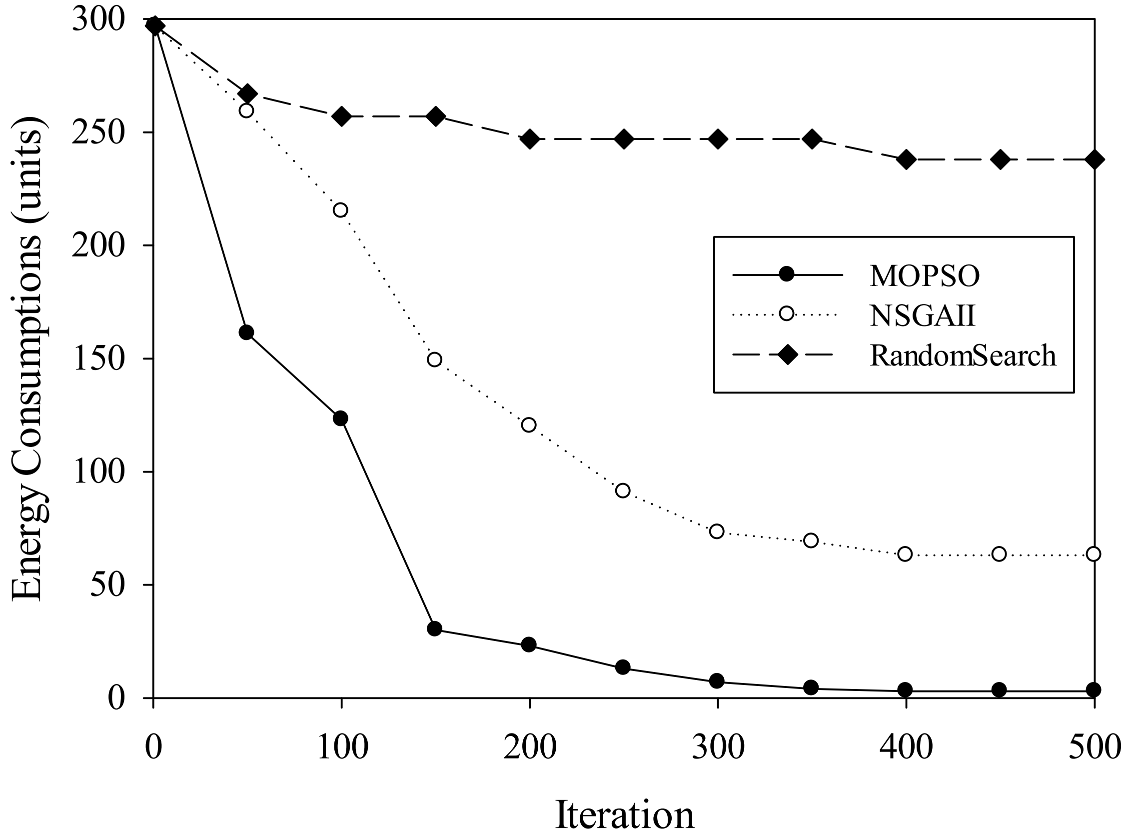


Fig. 3.7 The minimum energy consumption in case of 500 max iterations

$\mathscr{C}$-metric at iteration 250 is shown in table 3.3. The average, minimum and maximum of 10 independent runs are shown in the table. The results show that NSGA-II and random search algorithm contribute less to the non-dominated frontier than MOPSO. From the table, the maximum of $\mathscr{C}(MOPSO, NSGA-II)$ and $\mathscr{C}(MOPSO, RandomSearch)$ are 1, which indicate that all individuals in NSGA-II and random search algorithm that were dominated by at least one individual from MOPSO, respectively. Moreover, the average of $\mathscr{C}(MOPSO, NSGA-II)$ and $\mathscr{C}(MOPSO, RandomSearch)$ of 10 independent runs are more than the average of $\mathscr{C}(NSGA-II, MOPSO)$ and $\mathscr{C}(RandomSearch, MOPSO)$.

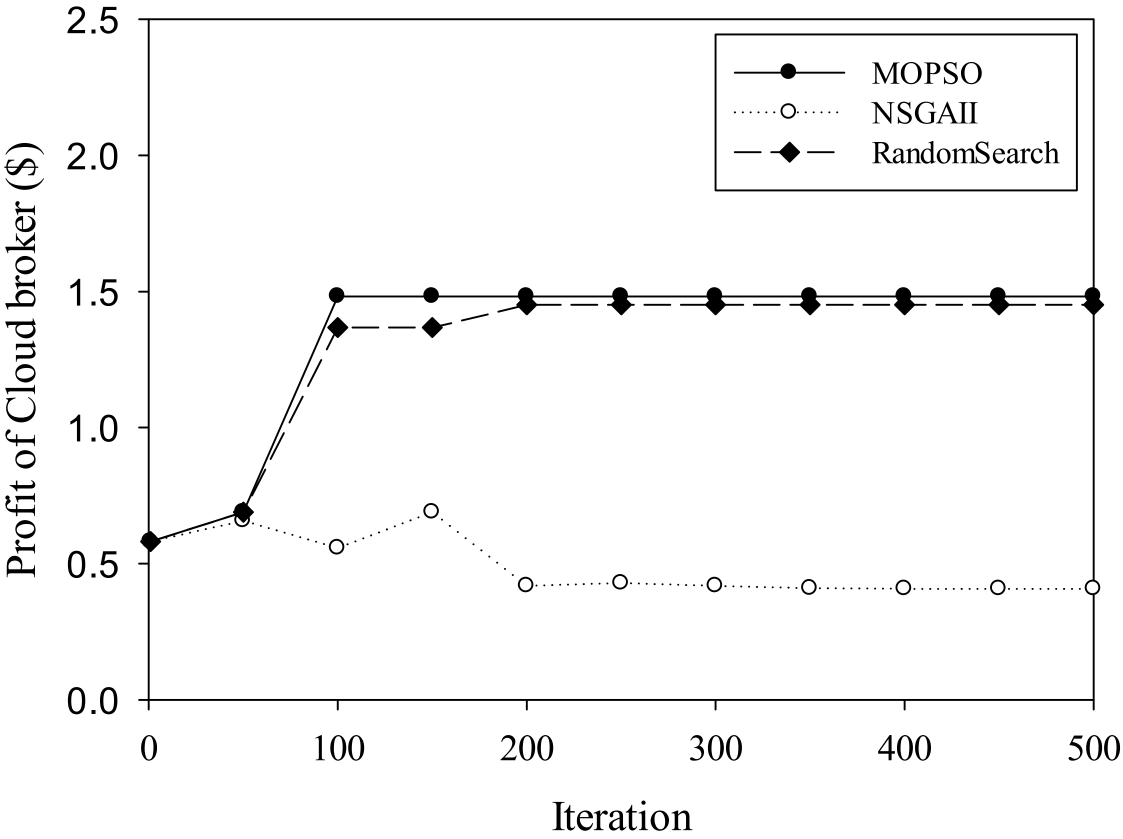


Fig. 3.8 The maximum profit of the cloud broker in case of 500 max iterations

3.7 Conclusion

This study considered a cloud brokering system to manage connected things in IoT. The cloud broker is to find the best deal among clients and service providers. The cloud brokering system was considered as an optimization problem. The objectives are to minimize the response time of requests from clients, the energy consumption of service providers and to maximize the profit of the cloud broker. There are three optimization algorithms which is used to solve the optimization problem: (1) Multi-objective Particle Swarm Optimization (MOPSO), (2) Non-dominated Sorting Genetic Algorithm II (NSGA-II) and (3) Random Search Algorithm. The cloud brokering system and those three algorithm have been conducted by using extensive simulations. The simulation results show that MOPSO succeed to find the minimum response time of requests from clients, the energy consumption of service providers and the maximum profit of the cloud broker better than other algorithms.

Table 3.3 $\mathscr{C}$-metric

$\mathscr{C}(A,B)$	min	avg	max
$\mathscr{C}(MOPSO, NSGA-II)$	0.04	0.82	1
$\mathscr{C}(MOPSO, RandomSearch)$	0.75	0.87	1
$\mathscr{C}(NSGA-II, MOPSO)$	0.16	0.73	0.83
$\mathscr{C}(NSGA-II, RandomSearch)$	0.75	0.78	0.82
$\mathscr{C}(RandomSearch, MOPSO)$	0.26	0.50	0.75
$\mathscr{C}(RandomSearch, NSGA-II)$	0.34	0.54	0.75

Chapter 4

The Local Outlier Detection in Geo-Social Points

This chapter describes problem statements and methodologies that will be used in this study. The chapter starts with a problem statement of the local outlier detection in geo-social media and methodology for solving the optimization problem of the outlier detection are presented, respectively. Next, simulation configuration and results are described. Finally, a conclusion of the study in the local outlier detection in geo-social points is presented.

4.1 Problem Statement

The problem statement of a partitioning a spatio-temporal domain and finding outlier patterns is presented in this section. Assumed that gp denotes a geo-social point as a tuple of (s,t,B), where $s = (lat, lon)$, t and B are a geographical coordinate of latitude (lat) and longitude (lon), a time-stamp and a bag of words, respectively. $GP = \{gp_1, gp_2, ..., gp_n\}$ indicate a set of geo-social points in a spatio-temporal bounding box ($ST = ([lon_{min}, lon_{max}]$, $[lat_{min}, lat_{max}], [time_{min}, time_{max}])$) which consists of three bounding intervals of longitudes, latitudes, and timestamps to cover all elements in GP. Term frequency–inverse document frequency (*tf-idf*) weighting scheme is applied to examine the local importance measurement of a keyword. Figure 4.2 shows that ST is divided into non-overlapping small cells, i.e., $ST = \{st_1, st_2, ..., st_N\}$, and each cell is regarded as a document to calculate *tf-idf*. The local importance score (x) of each cell for each keyword. Then, the outlier pattern of each cell is identified with respect to the surrounding neighbors. The difficult of partitioning is shown in figure 4.1. if the ST is divided as left-hand side figure in figure 4.1, we cannot find any pattern. On the other hand, if the ST is divided as right-hand side figure in figure 4.1, we can get the pattern.

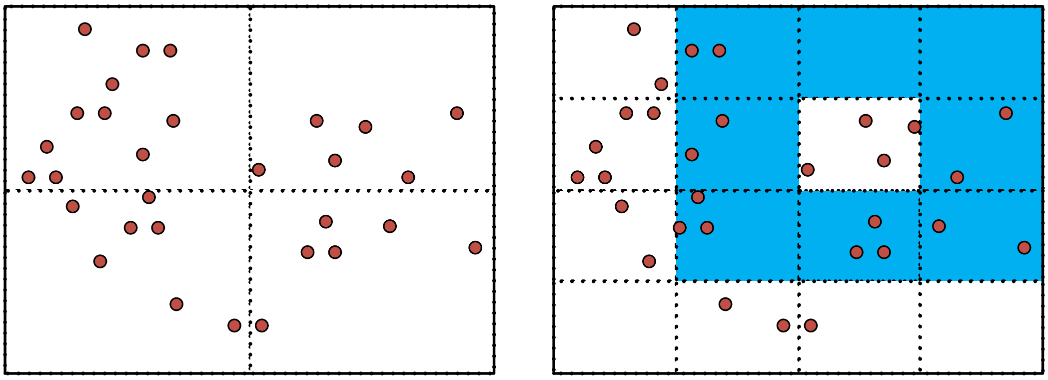


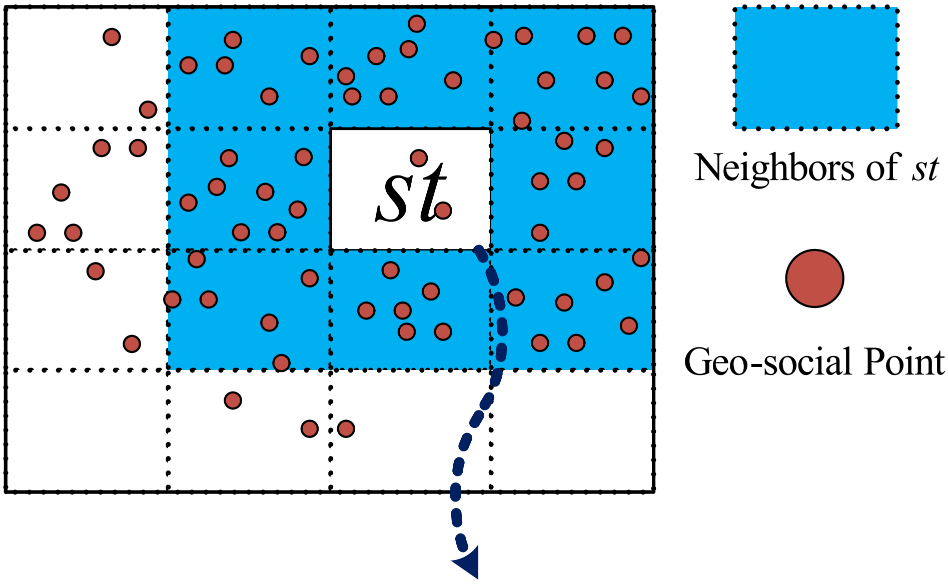
Fig. 4.1 An example of partitioning in geo-social points

Since the goal of this study is a way how to partition ST, a normal distribution test (Z-test) [9] is used to detect outlier patterns. Given the local importance scores of cells, the Z value of a target cell can be calculated as follows:

$$Z = (\bar{x} - \mu_0)/(\sigma/\sqrt{n}) \tag{4.1}$$

where μ_0 and σ are the mean value of the normal distribution and the standard deviation of the local importance scores, respectively. n denotes the number of neighbors including the target cell. $\bar{x}$ denotes the average of local importance scores of queen contiguity neighbors (blue rectangles in Figure 4.2) of the target cell. There are two patterns which are considered in this study: pattern of outbound outlier, called H-pattern and pattern of lower bound, called L-pattern. The target cell is counted as one pattern of H-pattern, if Z values is more than $Z_{\alpha/2}$. On the other hand, the target cell is counted as one pattern of L-pattern, Z values of the target cell is less than $-Z_{\alpha/2}$.

Next, the optimization problem of the space partitioning is considered. Since, partitioning is suitable to apply a large dataset such as geo-social media, the space-partitioning approach is used instead of the distance metric. However, the space-partitioning approach leads to to different results of the outlier detection depending on the size and shape of cells. Thus, k-dimensional tree or KD-tree [11] is used to divide ST as a 3D-space (three-dimensional space) into small disjoint cells. The KD-tree is a binary search tree and hybrid spatial grid. It supports an efficient processing of nearest neighbor searches for high-dimensional point data comparing to other space partitioning methods. Figure 4.3 shows an example of 2D-space (two-dimensional space) partitioning by using kd-tree. m-number of sampling points is



$x_{w,st} = tf_{w,st} \times \log(N/df_w)$: local importance of w in cell st

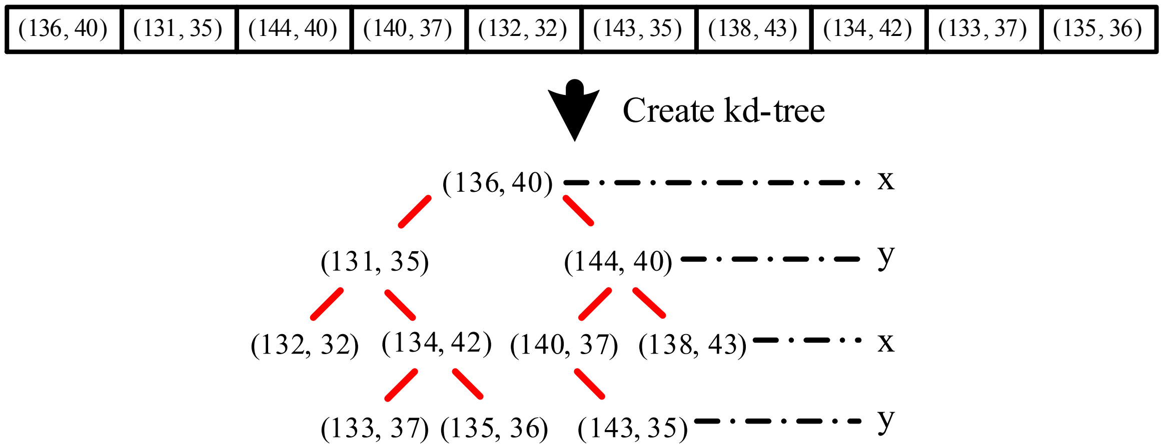
$tf_{w,st}$: the number of occurrences of w in cell st

df_w : the number of cells containing w

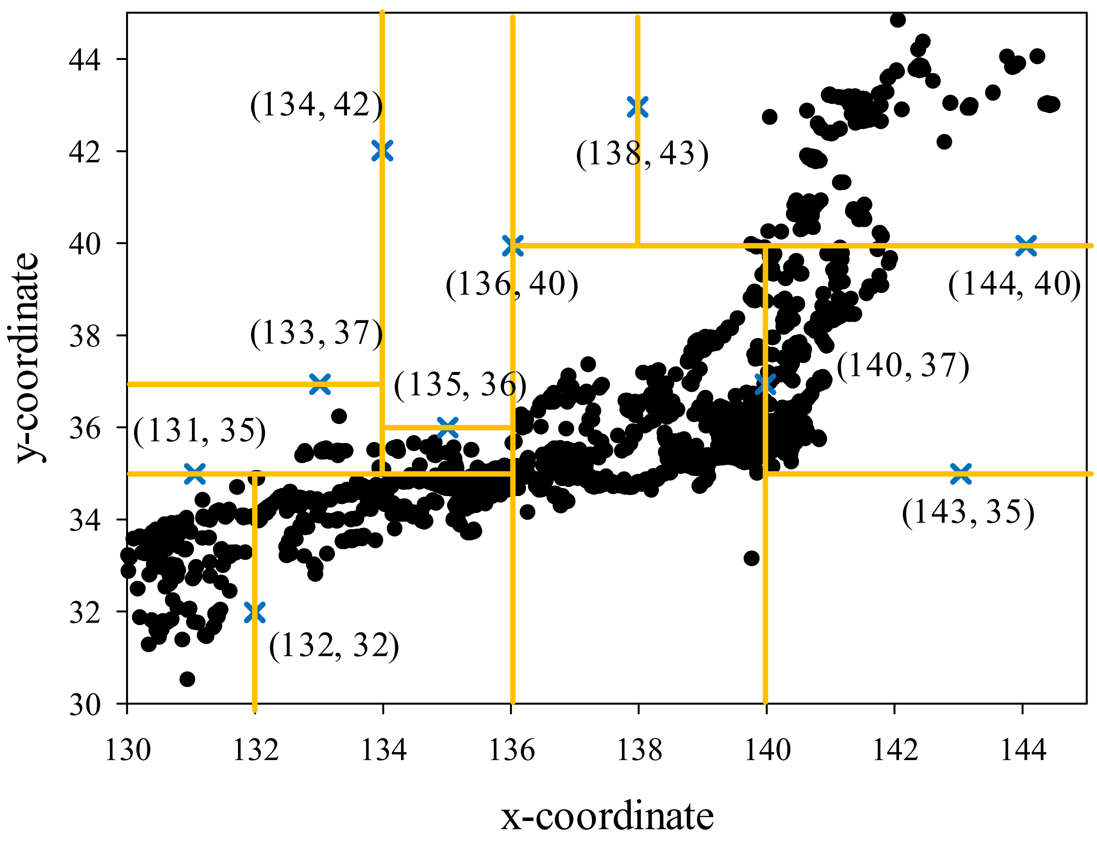
N : the total number of cells

Fig. 4.2 TF-IDF based on the two-dimensional grid partitioning

randomly selected from a subset of geo-social points containing keyword w, $GP_w \in GP$ and a kd-tree is created with them. One subdivision of ST is represented by each node in the kd-tree. Then, Z-test is calculated to each cell and an outlier pattern is determined such as H-pattern and L-pattern. However, a dependency of the samples is a critical issue of the results. Therefore, a genetic algorithm (GA) is considered to find the optimal partition of space. GA is a well-known meta-heuristic algorithm for solving an optimization problem. Several studied applied GA for the outlier detection, for example, two detection algorithms were proposed in [4] for high dimensional data based on the data distribution, and the authors in [62] proposed a fitness function based GA approach to detect outliers. However, a spatio-temporal outlier pattern with respect to the neighbors' point pattern is considered, unlike the existing methods that focus on the detection of outlier objects.



(a) Create kd-tree



(b) KD-tree partitioning

Fig. 4.3 2-dimensional pattern example

4.2 Optimization Problem for Micro-grid

The goal of this work is to seek the optimal solutions set of KD-tree partitioning for the local outlier pattern detection in geo-social points. Thus, the objective function is to maximize the number of H-pattern and L-pattern. It can be formulated as follows:

$$U = \omega_1 HP + \omega_2 LP = \omega_1 \sum_{i=1}^{N} p_i^H + \omega_2 \sum_{i=1}^{N} p_i^L, \tag{4.2}$$

where HP denotes the sums of H-pattern and LP denotes the sums of L-pattern, ω_1 and ω_2 denote the weighting factors of the number of two patterns, respectively, and N denotes the total number of sub-cells. The p_i^H is each outlier indicator of H-pattern of i-th sub-cell in ST and p_i^L is each outlier indicator of L-pattern of i-th sub-cell in ST. They can be formulated as follows:

$$p_i^H = \begin{cases} 1 & , if\ Z_i \geq Z_{\alpha/2} \\ 0 & , otherwise \end{cases}, \qquad p_i^L = \begin{cases} 1 & , if\ Z_i \leq -Z_{\alpha/2} \\ 0 & , otherwise \end{cases}$$

In the other hand, the goal is to partition ST to get the maximum number of both H-patterns and L-patterns. It means if more patterns are found, the area would be smaller. Moreover, more cells are needed to obtain candidates where a certain unusual event happens. In Eq. (4.2), there are two weight factors: ω_1 and ω_2. The function depends on these weight factors. However, it is very difficult to find the best values for these weight factors. Therefore, the two objectives are considered as a multi-objective optimization problem as follows:

$$\text{Maximize } HP = \sum_{i=1}^{n} p_i^H \tag{4.3}$$

$$\text{Maximize } \ LP = \sum_{i=1}^{n} p_i^L \tag{4.4}$$

4.3 Genetic Optimization Process

The genetic optimization process is described in this section. The precess is to seek the optimal solutions set (Pareto-set) of KD-tree partitioning for the local outlier pattern detection in geo-social points.

Since the local outlier detection problem is considered as a multi-objective optimization problem, a multi-objective genetic algorithm (MOGA) is presented to solve the problem. MOGA executes its optimization process to adjust the 3D kd-tree partitioning until the maximum number of generation found. The solutions set and objective values are provided after MOGA is finished. One of the solution can be selected from the solutions set for using in 3D kd-tree partitioning for detection of spatio-temporal outlier patterns. At first, the population of MOGA consists of M individuals. Each individual i is represented by multiple segments. Each multiple segments is a set of the 3D-coordinates (three-dimensional coordinates) of (x, y, t) in ST for KD-tree. the number of multiple segments in each individual represented the number of nodes in the kd-tree. An example of the individual structure is shown in figure 4.4.

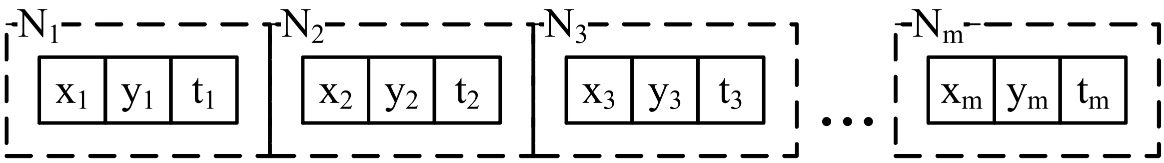


Fig. 4.4 The structure of an individual

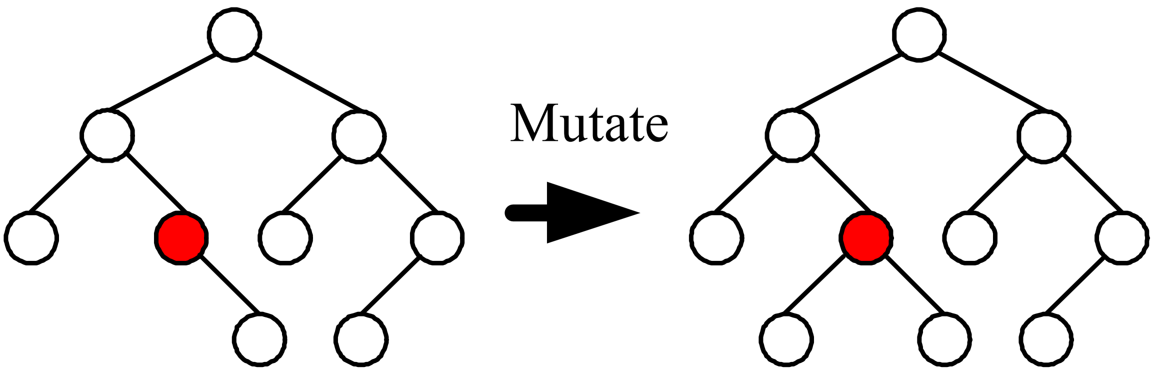


Fig. 4.5 Mutation operator example

In the beginning, the population randomly generated by coordinates in ST. Then, eq. (3.11) and eq. (3.12) are used to calculate fitness values of each individual. Next, a selection operator (e.g., binary tournament and random selection) is applied to select a pair of individuals with the highest fitness values as parents. After that, a crossover operator (e.g., one-point crossover, two-point crossover, and simulated binary crossover) with crossover rate is applied to reproduce two offspring from selected parents. In this study, one-point crossover [61] and two-point crossover are applied to implement the MOGA. Figures 4.6a and figure 4.6b show an example of one-point and two-point crossover operator, respectively.

After the crossover operator is finished, a mutation operator (e.g., bit-flip mutation, uniform mutation, and polynomial mutation) with a mutation rate is applied to randomly selects a node in KD-tree. Moreover, the mutation operator also decide to delete or add nodes. Figure 4.5 shows an example of mutation operator. Those operators (selection, crossover and mutation operator) are repeated until the maximum number of the offspring size N is found. Next, the offspring set will be combined with the population set. Then, the best M individuals from $M+N$ individuals are selected by using a selection operator as a new population for the next generation. Finally, the MOGA processes are terminated, if the maximum limit of the number of generations is found or the fitness value error is satisfied.

4.4 Computational Complexity of MOGA

Let the population size be N, and the number of objectives is M. The complexity of MOGA is mainly influenced by a genetic computation operation (e.g., selection operator, crossover operator, mutation operator and the fitness value) and the non-dominated sorting process. MN comparisons are needed for checking a particle for its non-dominance based on fitness value within N individuals and M objectives. Thus, the worst case complexity of this process will be $O(MN^2)$. Therefore, the overall worst-case complexity of MOGA is $O(MN^2)$ in the worst-case scenario with N individuals in the population.

4.5 Simulation Configuration and Results of MOGA

In this section, the setup of simulation of MOGA-based partitioning for maximizing the number of H and L-patterns is presented. Next, the performance of MOGA-based partitioning is shown. A dataset of samples of Twitter messages is used in the simulation. The dataset contains keyword 'landslide' that related to landslide situation in Hiroshima in 2014.

4.5.1 Simulation Configuration

The number of the dataset of samples of Twitter messages is 33,030. Each the sample data consists of timestamps (t) and geographical coordinates (x and y). *tf-idf* values of keyword 'landslide' are calculated in each tweet. A normal distribution test (Z-test) is applied in the simulation to find the optimal number of H-patterns and L-patterns. Two-tailed Z-test is used with α and α is set to 0.05. This means that $Z \leq -1.960$ or $Z \geq 1.960$ are considered as the decision rule for finding the optimal number of H-patterns or L-patterns, respectively. The population size, a crossover rate, a mutation rate and the maximum generation of MOGA are assumed as 100, 0.9, $1/n$ and 250, respectively. Moreover, there are three cases of the size of an individual which are considered in the simulation as shown in table 4.1. The size of an individual in MOGA represents the number of KD-tree partitioning for finding the optimal number of H-patterns and L-patterns. MOGA is evaluated to find the optimal

KD-tree partitioning with the maximum number of H and L-patterns. One-point crossover and two-point crossover are applied as a crossover operator in MOGA for comparison in term of convergence.

Table 4.1 The size of an individual

Individual Size	1% of Hiroshima	5% of Hiroshima	10% of Hiroshima
The number of coordinates	330	1652	3303

4.5.2 Simulation Results

The simulation results is divided into three parts: (1) $\mathscr{C}$-metric which represents the performance of the solution of MOGA, and (2) generation distance which represents the speed of an algorithm that can find the optimal solution, and (3) the optimal solutions of MOGA.

$\mathscr{C}$-metric

$\mathscr{C}$-metric [82] is used as a performance metric to show that how the solutions (individuals) of algorithm outperform the solutions from another algorithm. The $\mathscr{C}$-metric between algorithm A and B is represented by $\mathscr{C}(A,B)$. Moreover, $\mathscr{C}(A,B)$ can be calculated by $\mathscr{C}(A,B) = |\{b \in B | \exists a \in A : a \succ b\}|/|B|$, where an operator $\succ$ is the dominating algorithm. For example, $b \succ a$, this means that individual (solution) b dominate individual a. A range of $\mathscr{C}(A,B)$ is between 0 to 1. Therefore, $\mathscr{C}(A,B)$ is equal to 1, if there is at least one individual in A which dominates all individuals in B. On the other hand, there is no individual in B which is dominated by any individual in A, if $\mathscr{C}(A,B)$ is equal to 0.

Table 4.2 $\mathscr{C}$-metric

Individual Size	1% of Hiroshima	5% of Hiroshima	10% of Hiroshima
$\mathscr{C}$(OPX,TPX)	0.23	1.00	0.00
$\mathscr{C}$(TPX,OPX)	0.02	0.41	1.00

The $\mathscr{C}$-metric at generation 200 of MOGA is shown in table 4.2. In the case of 1% and 5% of the sample data are considered as the number of KD-tree partitioning, the two-point crossover (TPX) operator contributes in MOGA to less non-dominated frontier than the one-point crossover (OPX), because value of $\mathscr{C}$(TPX,OPX) are less than value of $\mathscr{C}$(OPX,TPX) in both 1% and 5% of the sample data. In the case of 10% of the sample data is considered as the number of KD-tree partitioning, the two-point crossover operator contributes in MOGA to better non-dominated frontier than the one-point crossover, because $\mathscr{C}$(TPX,OPX) is equal to 1.00 and $\mathscr{C}$(OPX,TPX) is equal to 0.00. It means there is no individual from two-point crossover operator that dominated by any individual from one-point crossover operator, and all individuals from one-point crossover operator are dominated by at least one individual from two-point crossover operator.

Generation Distance

Generation distance (GD) is used to show a speed of an algorithm that can fine an optimal solution. GD calculates by using the minimum Euclidean distance from the Utopian point in objective space (U) to the non-dominated individuals at the end of each generation of MOGA. It can be calculated as follows:

$$GD = \min_{i \in \mu} \sqrt{\sum_{k=1}^{n} (x_i[k])^2} \tag{4.5}$$

The generation distance in scenario 1%, 5%, and 10% of the individual size are shown in figure 4.7, figure 4.8, and figure 4.9, respectively. In the case of 1% of the sample data are considered as the number of KD-tree partitioning, two-point crossover operator contributes to increasing the generation distance slower than one-point crossover operator at 100 and 150 generation. At 50 and 200 generation, two-point crossover operator contributes to increasing the generation distance faster than the one-point crossover. Next, in the case of 5% of the sample data are considered as the number of KD-tree partitioning, two-point crossover operator contributes to increasing the generation distance slower than one-point crossover operator at 50, 100, 150 and 200 generation. Finally, in the case of 10% of the sample data are considered as the number of KD-tree partitioning, two-point crossover operator contributes to increasing the generation distance slower than one-point crossover operator at 50 and 150 generation. At 100 and 200 generation, two-point crossover operator contributes to increasing the generation distance faster than the one-point crossover.

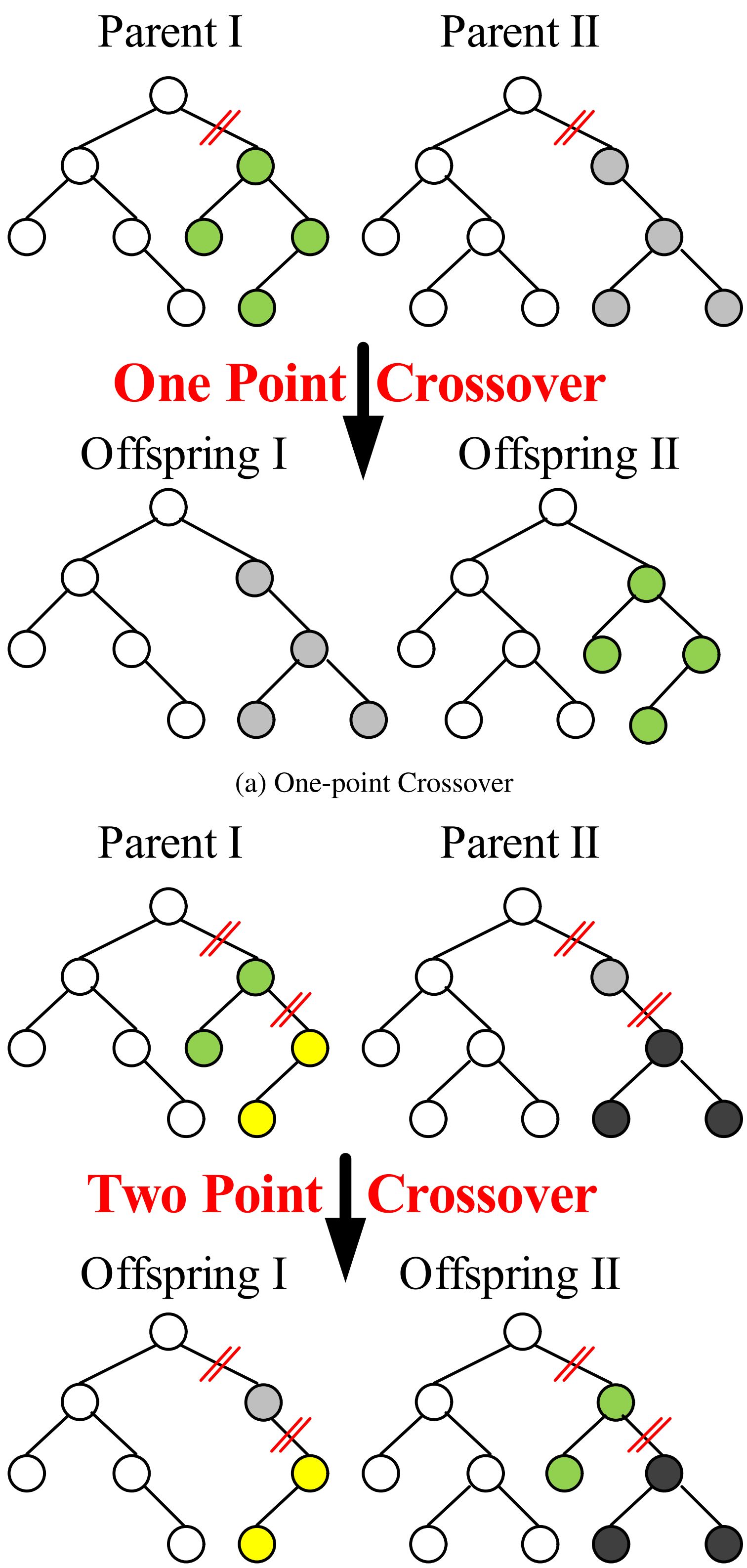
Optimal Solution of MOGA

The maximum number of the H-patterns and L-patterns are shown in figure 4.10a and figure 4.11a by considering one-point crossover (OnePointXover) and two-point crossover (TwoPointXover) as a crossover operator in MOGA. In the case of 1% of the sample data are considered as the number of KD-tree partitioning, one-point crossover operator is able to find the maximum number of L-patterns better than two-point crossover operator, but the two-point crossover is able to find the maximum number of H-patterns better than one-point crossover operator. Next, in the case of 5% of the sample data are considered as the number of KD-tree partitioning, two-point crossover operator is able to find the maximum number of L-patterns better than one-point crossover operator, but the one-point crossover is able to find the maximum number of H-patterns better than two-point crossover operator. Finally, in the case of 10% of the sample data are considered as the number of KD-tree partitioning, two-point crossover operator is able to find the maximum number of L-patterns better than one-point crossover operator as well as the maximum number of H-patterns.

Moreover, figure 4.12 and 4.13 show the maximum number of the H-patterns and L-patterns at the end of each generation with 500 max generations, respectively. The figures show in case of 5% and 10% of the sample data. The results show that GA is able to find more H-patterns and L-patterns in more generations. The two-point crossover operator is still able to find the maximum number of H-patterns and L-patterns in the case of a large number of partition better than the one-point crossover operator.

4.6 Conclusion

This study applied KD-tree partitioning for the outlier detection in a spatio-temporal domain. However, it is very difficult to find the best kd-tree partitioning for the outlier detection. Therefore, the multi-objective genetic algorithm (MOGA) was investigated to optimize KD-tree partitioning. The surrounding neighbors with a normal distribution test (Z-test) of the values of keyword was used to detect the outlier pattern. The pattern was classified into 2 categories: H-pattern which is outbound outlier, and L-pattern which is lower bound. The kd-tree partitioning for the outlier detection and MOGA have been conducted by using extensive simulations. There are two crossover operators for comparison: one-point crossover and two-point crossover operator. The simulation results show that MOGA is able to find solution sets for the KD-tree partitioning for the outlier detection in a spatio-temporal domain. Moreover, the two-point crossover operator in MOGA succeeded to find the maximum number of H-patterns and L-patterns better than the one-point crossover in the case of a large number of the partition.



(a) One-point Crossover

(b) Two-point Crossover

Fig. 4.6 Crossover operator example

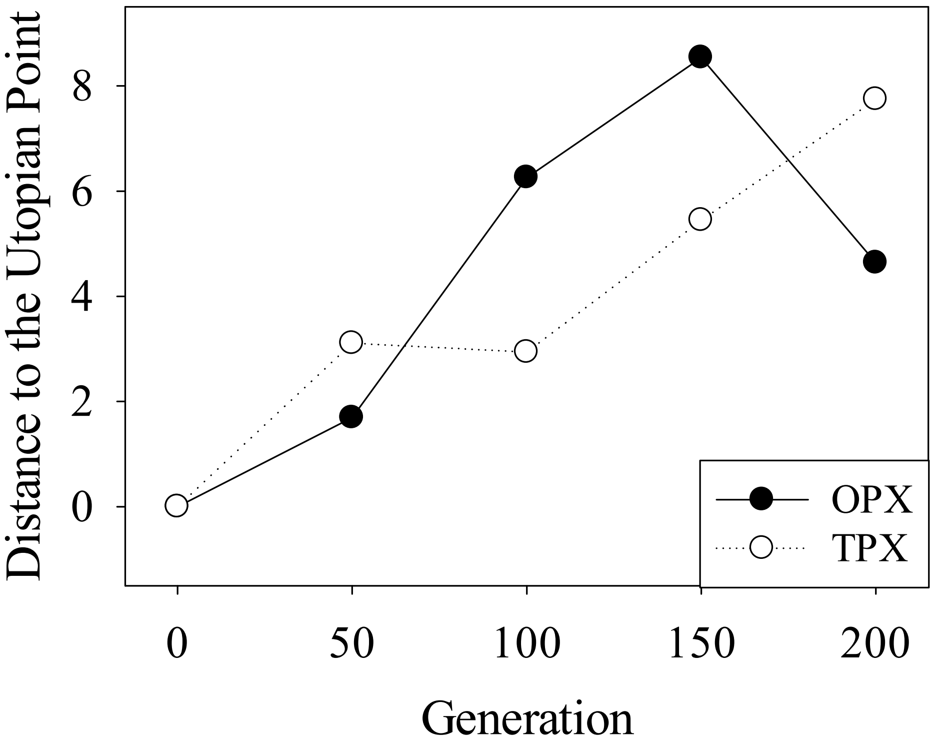


Fig. 4.7 1% of Hiroshima

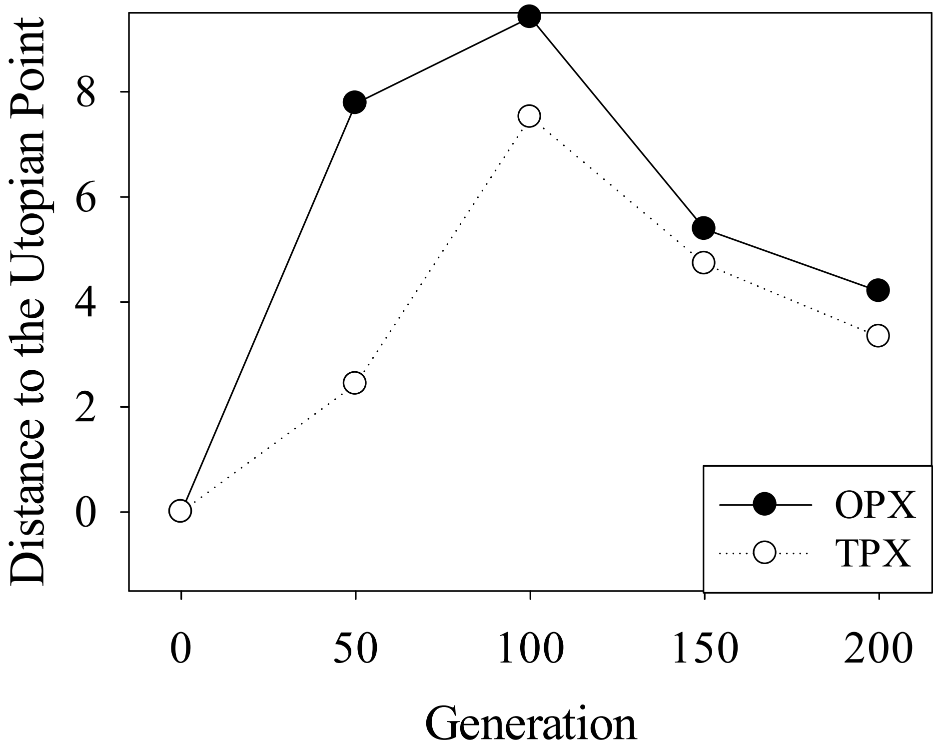


Fig. 4.8 5% of Hiroshima

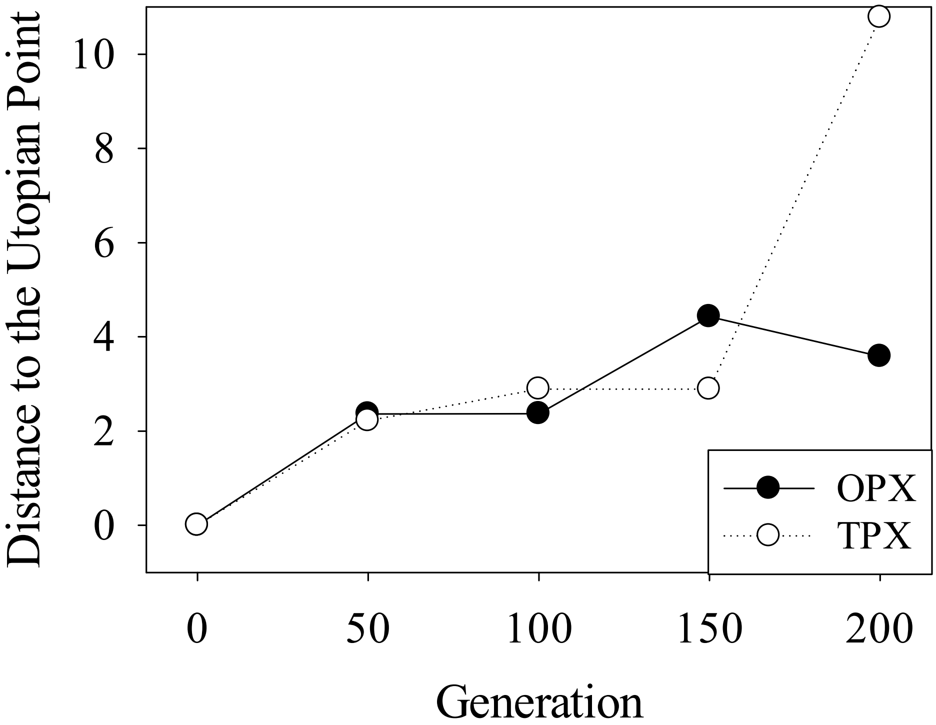
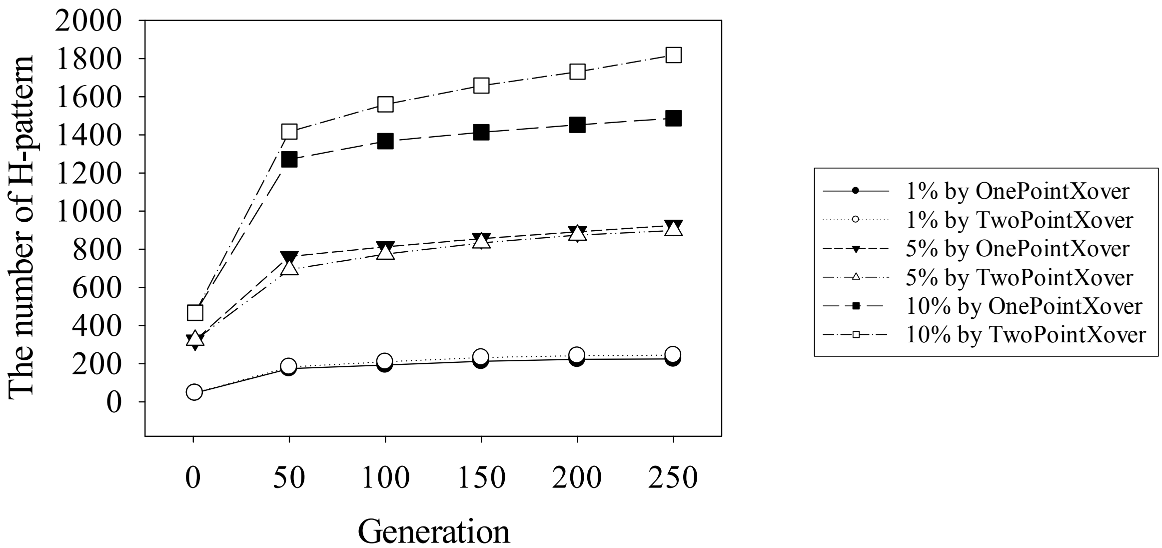
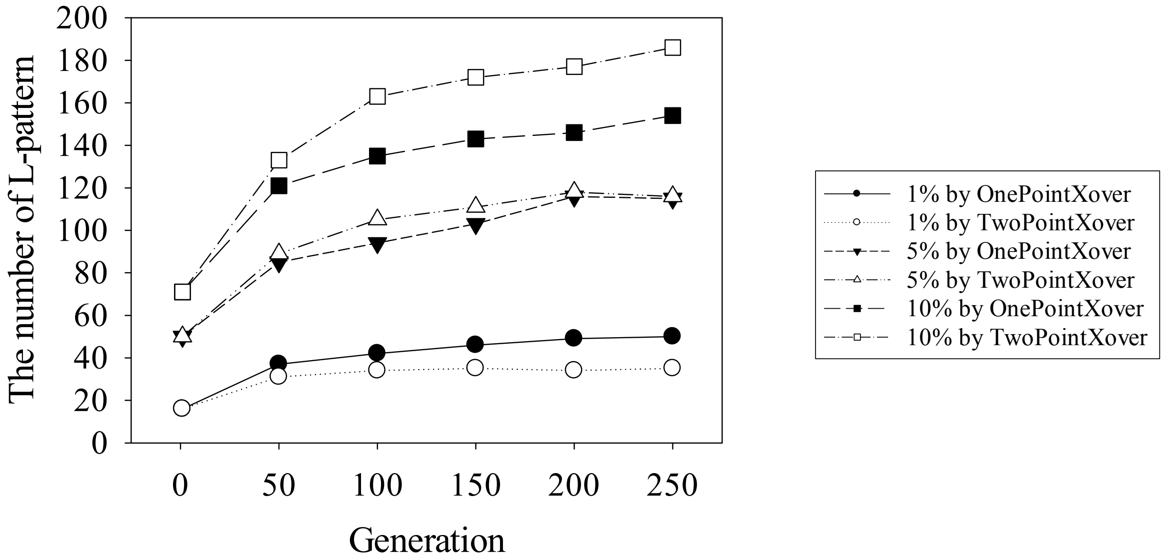


Fig. 4.9 10% of Hiroshima



(a) H-pattern

Fig. 4.10 The number of H-patterns



(a) L-pattern

Fig. 4.11 The number of L-patterns

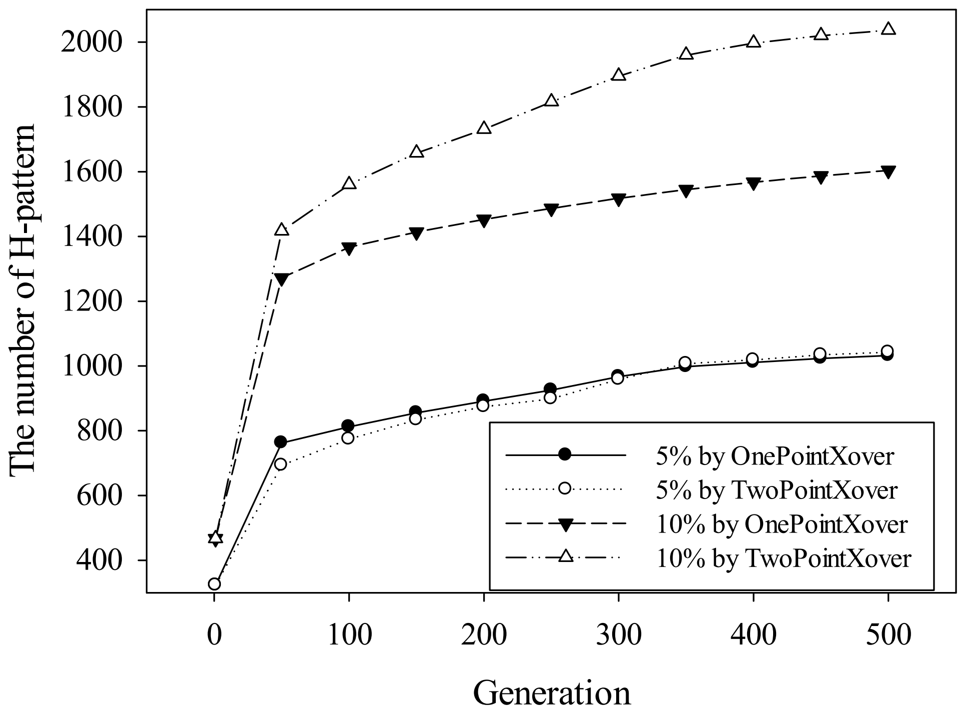


Fig. 4.12 The number of H-patterns in case of 500 max generations

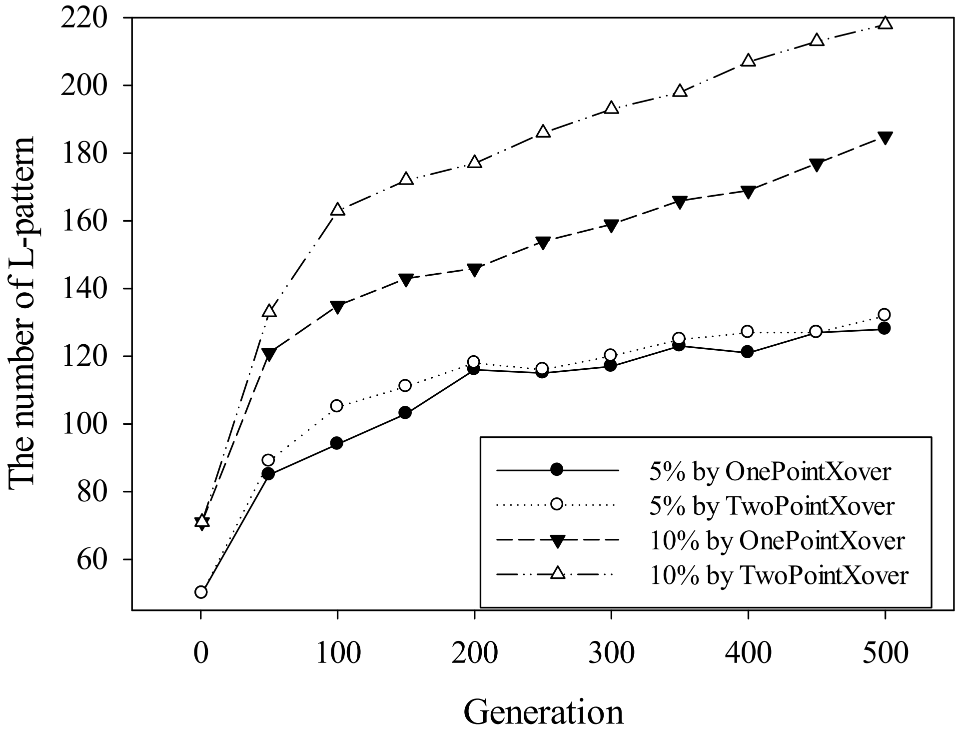


Fig. 4.13 The number of L-patterns in case of 500 max generations

Chapter 5

Operation Management for Multi-Micro-Grids Control

This chapter describes problem statements and methodologies that will be used in this study. The chapter starts with a problem statement and methodology of operation management for micro-grids control. Next, simulation configuration and results are described. Finally, a conclusion of the study in operation management of micro-grid is presented.

5.1 Operation Management for Micro-grids System

A micro-grid system is described in this section. The system consists of DGs such as WT, FC, PV, MT, and BES. Figure 5.1 [8] show a typical low voltage micro-grid system. A power exchange between DGs and the utility is managed by the micro-grid central controller (MGCC) in the system. However, there is some problem in term of the incompatibility between renewable energy generators and the energy consumption. Thus, a MT/FC/BES backup power sources are located in a various location in the system and also used to store the power from renewable energy generators. When non-generation and/or low-power in time periods, the system can use the stored power in the sources. Moreover, some DGs such as MT, FC and BES produced some pollutants such as carbon dioxide, nitrogen oxides, sulphur dioxide and particulate matter 10 micrometers. Therefore, an operating management problem is considered for the micro-grid system. In this work, the total cost of the system and the pollutant emissions from the system are considered as two objectives of the problem.

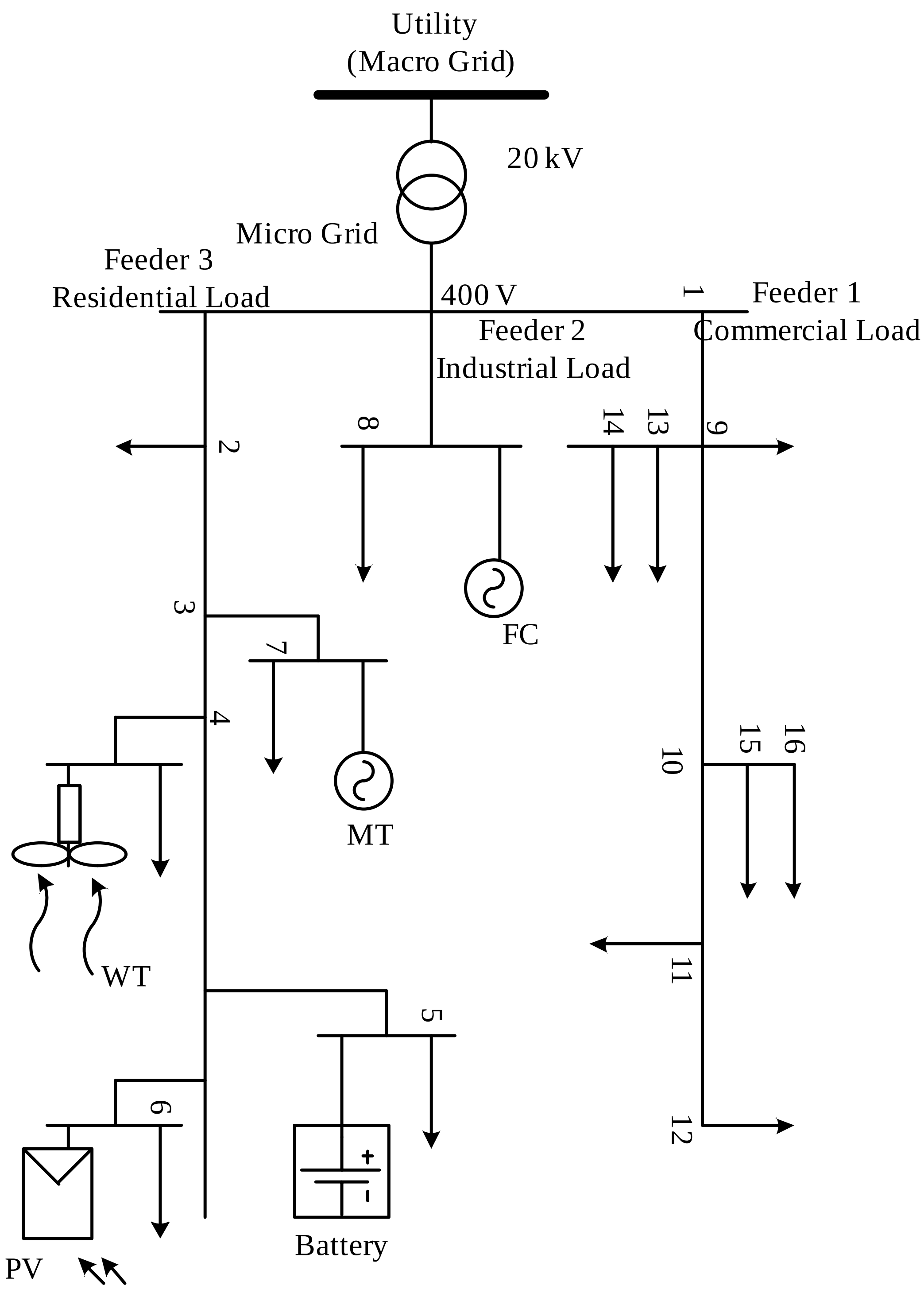


Fig. 5.1 A typical micro grid test system

5.2 Problem Statement

Next, the problem statement of the operation management problem in the micro-grid system is described. Since there are two objectives: to minimize the pollutant emissions and to minimize the total cost of the system, the problem is considered as multi-objective optimization problem. First, the pollutant emissions function is described. Then, describe the cost function. Finally, the multi-objective optimization problem is described.

5.2.1 Objective Functions

This section describes the the pollutant emissions function and the total cost function for the optimization operation management problem in the micro-grid system.

Pollutant Emission Function

The most important pollutant emissions from the micro-grid system are carbon dioxide (CO_2), nitrogen oxides (NO_x), sulphur dioxide (SO_2) and particulate matter 10 micrometers (PM_{10}). These important pollutant emissions are considered as an objective in the optimization operation management problem. The pollutant emission function can be divided into three functions, which are the pollutant emission function of fuel cell (FC), photo-voltaic (PV), micro-turbine (MT) and wind turbine (WT), the pollutant emission function of BES, and the pollutant emission function of utility.

The emission of DG: It can be formulated as follows.

$$E^s_{DG,t} = \sum_{i=1}^{N} E_{DG_{i,t}} P_{DG_{i,t}} u_{DG_{i,t}} \tag{5.1}$$

$$E_{DG_{i,t}} = CO_{2,DG_{i,t}} + NO_{x,DG_{i,t}} + SO_{2,DG_{i,t}} + PM_{10,DG_{i,t}} \tag{5.2}$$

where $CO_{2,DG_{i,t}}$, $NO_{x,DG_{i,t}}$, $SO_{2,DG_{i,t}}$ and $PM_{10,DG_{i,t}}$ are the amounts of carbon dioxide, nitrogen oxides, sulphur dioxide and particulate matter 10 micrometers from MT and FC at time t, respectively.

The emission of BES: It can be formulated as follows.

$$E^s_{BES,t} = \sum_{i=1}^{N} E_{BES,t} P_{BES,t} u_{BES,t} \tag{5.3}$$

$$E_{BES,t} = CO_{2,BES,t} + NO_{x,BES,t} + SO_{2,BES,t} + PM_{10,BES,t} \tag{5.4}$$

where $CO_{2,BES,t}$, $NO_{x,BES,t}$, $SO_{2,BES,t}$ and $PM_{10,BES,t}$ are the amounts of carbon dioxide, nitrogen oxides, sulphur dioxide and particulate matter 10 micrometers from BES at time t, respectively.

The emission of utility: It can be formulated as follows.

$$E^{s}_{Grid,t} = E_{Grid,t}P_{Grid,t} \tag{5.5}$$

$$E_{Grid,t} = CO_{2,Grid,t} + NO_{x,Grid,t} + SO_{2,Grid,t} + PM_{10,Grid,t} \tag{5.6}$$

where $CO_{2,Grid,t}$, $NO_{x,Grid,t}$, $SO_{2,Grid,t}$ and $PM_{10,Grid,t}$ are the amounts of carbon dioxide, nitrogen oxides, sulphur dioxide and particulate matter 10 micrometers from utility at time t, respectively.

Cost Function

The cost function can be divided into three functions, which are the maintenance and operation cost of a fuel cell (FC), photo-voltaic (PV), micro-turbine (MT) and wind turbine (WT), the total BES cost per day, and the total cost of DG.

The fixed maintenance and operation cost of DG (MO_{DG}) can be calculated as follows:

$$MO_{DG} = (MO_{FC} + MO_{MT} + MO_{WT} + MO_{PV}) \times OTH \tag{5.7}$$

where MO_{MT}, MO_{FC}, MO_{PV}, and MO_{WT} denote the fixed maintenance and operation cost of MT, FC, PV and WT.

Total Cost Per Day of BES ($TCPD_{BES}$) can be calculated as follows [18, 8]:

$$TCPD_{BES} = \frac{C_{BES,max}}{365}\left(\frac{IR(1+IR)^{LT}}{(1+IR)^{LT}-1}FC_{BES} + MC_{BES}\right) \tag{5.8}$$

where $C_{BES,max}$ is a maximum size of BES (kWh). FC_{BES} and MC_{BES} are a fixed cost for BES ($/kwh) and a maintenance cost for BES ($/kwh), respectively. IR denotes the interest rate of the installed BES and LT denotes the lifetime of the installed BES.

The total costs of DG (f_t) can be formulated as follows:

$$f_t = C_{Grid,t} + C_{DG,t} + C_{BES,t} + SU_{FC,t} + SU_{MT,t} + SD_{FC,t} + SD_{MT,t} \tag{5.9}$$

where $C_{Grid,t}$ is the cost of grid at time t ($). $C_{DG,t}$ and $C_{BES,t}$ denote the cost of operating power and fuel of DGs at time t ($), respectively. $SU_{MT,t}$, $SD_{MT,t}$ and $SU_{FC,t}$, $SD_{FC,t}$ denote the start-up and shut-down cost for MT and FC at time t ($), respectively.

The cost of grid ($C_{Grid,t}$) at time t can be calculated as follows:

$$C_{Grid,t} = \begin{cases} (1-tax)B_{Grid,t}P_{Grid,t} & ,if\ P_{Grid,t} < 0 \\ B_{Grid,t}P_{Grid,t} & ,if\ P_{Grid,t} > 0 \\ 0 & ,if\ P_{Grid,t} = 0 \end{cases} \tag{5.10}$$

where $B_{Grid,t}$ and $P_{Grid,t}$ denote a utility's bid at time t (\$/kWh) and a utility's power at time t (kwh), respectively. tax is a grid's tax rate. Next, the cost of operating power and fuel of DGs ($C_{DG,t}$) and BES ($C_{BES,t}$) at time t (\$) can be calculated as follow:

$$C_{DG,t} = B_{MT,t}P_{MT,t}u_{MT,t} + B_{PV,t}P_{PV,t} + B_{FC,t}P_{FC,t}u_{FC,t} + B_{WT,t}P_{WT,t} \tag{5.11}$$

where $u_{MT,t}$, $u_{FC,t}$ and $u_{BES,t}$ are a status (off or on) of MT, FC and BES at time t, respectively. $B_{MT,t}$, $B_{FC,t}$, $B_{PV,t}$ and $B_{WT,t}$ are a bid cost of the MT, FC, PV, and WT at time t (\$/kwh). $P_{MT,t}$, $P_{FC,t}$, $P_{PV,t}$ and $P_{WT,t}$ are the power of MT, FC, PV, and WT at time t (kwh), respectively.

$$C_{BES,t} = B_{BES,t}P_{BES,t}u_{BES,t} \tag{5.12}$$

where $B_{BES,t}$ and $P_{BES,t}$ are a bid of BES (\$/kwh) and a power of BES (kwh) at time t.

The start-up and shut-down cost for MT ($SU_{MT,t}$), ($SD_{MT,t}$) and FC ($SU_{FC,t}$), ($SD_{FC,t}$) at time t (\$) can be calculated as follows.

$$\begin{aligned} SU_{i,t} &= St_i \times max(0, u_{i,t} - u_{i,t-1}) \\ SD_{i,t} &= Sh_i \times max(0, u_{i,t} - u_{i,t-1}) \end{aligned} \tag{5.13}$$

where $St_{i,t}$ and $Sh_{i,t}$ denote a start-up and shutdown cost coefficient for FC and MT, respectively. $u_{i,t}$ is a binary variable which denotes the status (off or on) of FC and MT at time t.

5.2.2 Constraints

The constraints of the operation management problem in the micro-grid system consist of three requirements, which are BES constraint, operating reserve constraint, and a dispatchable DGs and grid constraint.

Battery energy storage constraints [2]: there are two modes as follows:

- Charging mode:

$$C_{BES,t+1} = min\{(C_{BES,max}, C_{BES,t} - \Delta t P_{BES,t} \eta_{charge})\};\ t = 1,...,OTH \qquad (5.14)$$

where $C_{BES,max}$ denotes a maximum BES size (kwh). Δt and η_{charge} denote a time interval duration and a charge efficiency of BES, respectively.

$$\underline{P}_{BES,t} \leq P_{BES,t} \leq \overline{P}_{BES,t};\ t = 1,...,OTH \qquad (5.15)$$

where $\underline{P}_{BES,t}$ and $\overline{P}_{BES,t}$ denote a maximum BES charge rates and a maximum discharge rates of BES at time t (kw), respectively. They can be formulated as follows:

$$\overline{P}_{BES,t} = min\{P_{BES,max}, (C_{BES,t} - C_{BES,max} \eta_{discharge}/\Delta t)\};\ t = 1,...,OTH \qquad (5.16)$$

$$\underline{P}_{BES,t} = max\{P_{BES,max}, (C_{BES,t} - C_{BES,max}/\eta_{charge} \Delta t)\};\ t = 1,...,OTH \qquad (5.17)$$

- Discharging mode:

$$C_{BES,t+1} = max\{(C_{BES,min}, C_{BES,t} - \Delta t P_{BES,t}/\eta_{discharge})\};\ t = 1,...,OTH \qquad (5.18)$$

where $C_{BES,min}$ denotes a minimum size of BES (kwh). Δt and $\eta_{discharge}$ denote a time interval duration and a discharge efficiency of BES, respectively.

Eq. (5.18), Eq. (5.14), and Eq. (5.15) indicated that the BES released energy limits and BES power discharged, a limitation of BES on the stored energy and power charged of BES, respectively. Next, Eq. (5.16) indicates the minimum charging/discharging rates and Eq. (5.17) indicates the maximum charging/discharging rates. Note that if several conditions of the battery are in the same range, the battery strings will be sorted up or down in the priority list for charge/discharge according to the current state of charge. If a new range is reached, the condition of the battery is charged/discharged. Finally, the battery will be charged/discharged with the next lower/higher state of charge with the same range.

Operating reserve constraint: An an operating reserve (OR) [8] is a summation of reserved electrical power generation capacity when utility, BES, FC, and MT are turned on in each time periods. It can be formulated as follow:

$$OR_t + P_{Demand,t} \leq P_{grid,max} + P_{MT,max} u_{MT,t} \\ + P_{FC,max} u_{FC,t} + \overline{P}_{BES,t} u_{BES,t};\ t = 1, ..., OTH \quad (5.19)$$

where OR_t denotes an operating reserve requirements (kW) at time t. $P_{FC,max}$, $P_{MT,max}$ and $P_{grid,max}$ denote a maximum power of FC, MT and the utility, respectively. $u_{FC,t}$, $u_{MT,t}$ and $u_{BES,t}$ denote the status of FC, MT and BES (off or on) at time t. $P_{Demand,t}$ denotes a load demand of electrical at time t (kW).

Generating and grid capacity constraints can be formulated as follows:

$$P_{j,min} \leq P_{j,t} \leq P_{j,max}; t = 1, ..., OTH \quad (5.20)$$

where $P_{j,min}$ and $P_{j,max}$ denote minimum and maximum power of MT, PV, FC and WT (kw) as well as the utility, respectively.

5.3 Single-Objective Optimization Problem for Micro-grid

At first, the two objectives are considered as a single-objective optimization problem by using a weighting factor. Thus, the utility function of the micro-grid system can be formulated as follows:

$$U = \omega_1 C(X) + \omega_2 E(X) \quad (5.21)$$

, where ω_1 denotes a weighting factor of cost function, and ω_2 denotes a weighting factor of pollutant emission function. Note that the summation of weights is equal to 1. It means that $\omega_1 + \omega_2 = 1$. $C(X)$ denotes the cost function of the micro-grid system which can be calculated as follows:

$$C(X) = \sum_{t=1}^{OTH} f_t + MO_{DG} + TCPD_{BES} \quad (5.22)$$

, and $E(X)$ denotes the emission function of the micro-grid system which can be calculated as follows:

$$E(X) = \sum_{t=1}^{OTH} \{E^s_{Grid}(t) + E^s_{DG}(t) + E^s_{BES}(t)\} \quad (5.23)$$

Therefore, the single-objective optimization problem of the micro-grid system is to minimize the utility function as follows:

$$\underset{U,S}{\text{Minimize}}\ U = \omega_1 C + \omega_2 E \tag{5.24}$$

$$\text{Subject to} \tag{5.25}$$

$$C_{BES,t+1} = max\{(C_{BES,min}, C_{BES,t} - \Delta t P_{BES,t}/\eta_{discharge})\} \tag{5.26}$$

$$C_{BES,t+1} = min\{(C_{BES,max}, C_{BES,t} - \Delta t P_{BES,t}\eta_{charge})\} \tag{5.27}$$

$$OR_t + P_{Demand,t} \leq P_{grid,max} + P_{MT,max}u_{MT,t} + P_{FC,max}u_{FC,t} + \overline{P}_{BES,t}u_{BES,t} \tag{5.28}$$

$$\underline{P}_{BES,t} \leq P_{BES,t} \leq \overline{P}_{BES,t};\ t = 1,...,OTH \tag{5.29}$$

where constraint (5.26) and (5.27) indicate battery energy storage constraint in charging mode and in discharging mode, respectively. Next, constraint (5.28) indicates an operating reserve constraint. Finally, constraint (5.29) indicates generating and grid capacity constraint.

However, the best value for the weights ω_1, and ω_2 are very difficult to find. Therefore, the two objectives are considered as a multi-objective optimization problem and the multi-objective optimization problem is described in the next section.

5.4 Multi-Objective Optimization Problem for Micro-grid

In this section, the pollutant emission function and the cost function are considered as a multi-objective optimization problem for micro-grid system.

This section first considers a multi-objective optimization problem for micro-grid system by using the three cost functions and three emission functions which are described in Section 5.2.1. Thus, the cost function of the operation management problem is formulated as follows:

$$C(X) = \sum_{t=1}^{OTH} f_t + MO_{DG} + TCPD_{BES} \tag{5.30}$$

where f_t is the total cost of DG ($). t and OTH are i-th time stamp (h) and operation time (h), respectively. MO_{DG} is the fixed maintenance and operation cost of DG ($/kwh). $TCPD_{BES}$ denotes the total cost per day of battery energy storage ($).

Then, the emission function of the operation management problem is formulated as follows:

$$E(X) = \sum_{t=1}^{OTH} \{E_{Grid}^{s}(t) + E_{DG}^{s}(t) + E_{BES}^{s}(t)\} \tag{5.31}$$

Hence, the optimization operation management problem of the MG system is to minimize the cost and emission function as follows:

$$\text{Minimize } C(X) = \sum_{t=1}^{OTH} f_t + MO_{DG} + TCPD_{BES} \tag{5.32}$$

$$\text{Minimize } E(X) = \sum_{t=1}^{OTH} \{E_{Grid}^{s}(t) + E_{DG}^{s}(t) + E_{BES}^{s}(t)\} \tag{5.33}$$

Subject to three constraints which are described in Section 5.2.2.

Next, the processes of fitness-based modified game particle swarm optimization (FMG-PSO) are described. The presented algorithm tries to seek the optimal solution set (Pareto-set) for the optimization operation management problem of the micro-grid system. At first, the multi-objective PSO is introduced. Then, a modified game is presented. Finally, FMGPSO is proposed to solve the optimization operation management problem in the micro-grid system.

5.5 Multi-objective Particle Swarm Optimization

In this section, the processes of multi-objective particle swarm optimization are described. In the beginning, the initial swarm is randomly generated. Then, non-dominated particles from the swarm are chosen to initialize a set of gBest. Next, an external archive store the set of gBest. Then, a gBest is selected and the positions of the particles are updated at each iteration. The updated velocity vector can be calculated as follows:

$$v_{lk}(t+1) = wv_{lk}(t) + c_1 r_1 [p_{lk} - x_{lk}(t)] + c_2 r_2 [g_{lk} - x_{lk}(t)] \tag{5.34}$$

where c_1 denotes a learning factors called the coefficient of the self-recognition component and c_2 denotes a learning factors called the coefficient of the social component. w denotes an inertia weight and r_1, r_2 denote random numbers that are uniformly distributed in the interval 0 to 1.

Then the positions of the particle are updated after updating the velocity. The updated positions can be calculated as follows:

$$x_{lk}(t+1) = x_{lk}(t) + v_{lk}(t+1) \tag{5.35}$$

where l and k denote the number of particles and the dimension of the particles, respectively. Next, after updating the position, the turbulence operator is used. Then, after all processes have finished, the set of gBest is updated. Finally, when the minimum/maximum objective function error is satisfied or the maximum limit of iteration is found, MOPSO terminates its processes.

5.6 Modified Game

A game theory is a mathematical model that is studied to both of the situations of cooperation and conflicts. However, it does not cover in cases of the decision makers do not have effect on the results [63]. The phenomenon of negotiation between in a very general setting and rational agents in conflict situations is studies in game theory. Therefore, game theory can be called as a rational behavior general theory. The situation in the theory is that rational players (decision makers) have available with their limit number of action. In terms of the number of payoffs participated with each combination of action are shown by a well-defined outcome or end with wins and losses for each player (decision maker) [25].

The authors in [7] proposed the modified game theory or the cooperative game theory for solving an optimization problem. Since the operation management problem in the micro-grid system is considered as optimization problem, the modified game theory is proposed to solve the optimization problem. Thus, two objectives can be assumed as three players. The two objectives consist of the pollutant emission and the total cost of the micro-grid system. Players in the modified game theory try to improve their own situations. In the other word, players try to decrease their objective value.

Since the optimization pollutant emission and total operation costs of the micro-grid system have to be minimized, an objective function for modified game can be created as follows:

$$\text{Minimize: } Obj(X) = P(X) - S(X) \tag{5.36}$$

where $P(X)$ denotes a Pareto optimal objective and $S(X)$ denotes a supercriterion.

The Pareto optimal objective $P(X)$ can be calculated as follows:

$$P(X) = \sum_{i=1}^{n} c_i f_{ni}(X) \tag{5.37}$$

where n represents the number of objectives and $\sum_{i=1}^{n} c_i = 1$.

The supercriterion $S(X)$ can be calculated as follows: The Pareto optimal objective $P(X)$ can be calculated as follows:

$$S(X) = \prod_{i=1}^{n} [1 - f_{ni}(X)] \tag{5.38}$$

where $f_{ni}(X)$ denotes a normalization of the i-th objective function which can be calculated as follows:

$$f_{ni}(X) = [f_i(X) - f_i(X_i^*)]/[F_{iu} - f_i(X_i^*)] \tag{5.39}$$

where F_{iu} denotes the worst value, $f_i(X)$ and $f_i(X_i^*)$ denote the i-th objective value and the i-th optimum objective value, respectively.

5.7 Fitness-based Modified Game Particle Swarm Optimization

The fitness-based modified game particle swarm optimization (FMGPSO) is described in this section. The modified game theory is applied to calculate fitness value of MOPSO for solving the multi-objective operation management problem in the micro-grid system. It means that MOPSO is needed to modify. Therefore, FMGPSO is proposed to seek a Pareto-set (the optimal solutions set).

The procedures of FMGPSO can be implemented as follows:

Step 1 (Initialize): Set the parameters of the particle swarm.

Step 2: Generated randomly the particles (position and velocity vector) and calculated the fitness value by Equation (5.36).

Step 3: Set each particle's *pBest* to the particle position.

Step 4: Collect the set of *gBest* by choosing the particle position with using non-dominated sorting based on fitness value.

Step 5: Initialize an external archive by adding the set of *gBest*.

Step 6: Improve the position of particle by calculating the particle's updated velocity, the particle's position and the fitness value.

Step 7: Apply the turbulence (which is a mutation operator that operates on the velocity value) operators.

Step 8: Check the fitness value by comparing between the new position with *pBest*. If the new position is better than *pBest*, then set a new position as *pBest*.

Step 9: Collect the *gBest* by choosing the particle position with using non-dominated sorting based on fitness value.

Step 10: Update an external archive by adding the set of *gBest*.

Step 11: Check the fitness value error or the maximum limit of the number of iterations. If the fitness value error is not satisfied or the maximum limit of the number of iterations is not found go to *Step 6*, else next step.

Step 12: Report the external archive as the results.

5.8 Computational Complexity of FMGPSO

The computational complexity of FMGPSO is described in this section. Let a swarm size be N, an external archive size be L and the number of objectives be M. The complexity of MOPSO is mainly influenced by a variety computation operation such as calculating the updated velocity of particle, calculating the updated position of particle and calculating the fitness function as well as a non-dominated sorting process. At first, $M(N+L)$ comparisons are needed to check the non-dominance of particles based on fitness value within $N+L$ particles and M objectives. Thus, the complexity of this process will be $O(M(N+L)^2)$ in the worst case. Next, the sorting based on fitness value for only the external archive requires $O(MLlog(L))$ computations. Therefore, the overall worst-case complexity of MOPSO is $O(M(N+L)^2)$ in the worst-case scenario with $N+L$ elements in the archive.

5.9 Simulation Configuration and Results of FMGPSO

In this section, the setup of simulation of FMGPSO for operation management in micro-grids is presented. Next, an analysis performance of FMGPSO for micro-grid and multi-micro-grids are presented. The performance is shown by comparison among FMGPSO, non-dominated sorting genetic algorithm III (NSGA-III), multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) and Speed-constrained Multi-objective PSO (SMPSO).

5.9.1 Simulation Configuration

A typical micro-grid test system consists of the MT, WT, FC, PV and BES as shown in 5.1 [8]. More information of the test system can be found in [18, 57, 59]. Moreover, a maintenance and fixed cost is 15.97 ($/kwh) and an installation and operation of BES is 495.09 ($/kwh). 500 kwh is assumed as the BES's full size. The BES's minimum size is set to 10% of the BES's full size. The BES's discharge and charge rate is set to be 90%. Assumed that the IR and LT are 0.06 and a financing the installed BES is 3. 5% of the load demand is assumed for the OR requirement in each time step. 10% is assumed as the tax. There are two test systems for operation management in micro-grids. Both of test systems are executed in one day. Two test systems are considered in this section as follows.

Micro-grid System

Micro-grid test system consists of one MT, one PV, one FC, one PV and one BES. Minimum and maximum power and bid cost of units in the micro-grid test system is shown in table 5.1. The fixed maintenance and operation, and start-up/shut-down cost of units in the micro-grid system is shown in table 5.2. The pollutant emission of each DG in the system is shown in table 5.3. Moreover, there are two cases of the test system: micro-grid test system with BES and without BES.

Table 5.1 Minimum and maximum power and bid cost of units in micro-grid system

Type	Min Power (kw)	Max Power (kw)	Bid cost ($/kwh)
Micro-turbine (MT)	6	30	0.49
Fuel-cell (FC)	3	30	0.31
Photo-voltaic (PV)	0	25	2.75
Wind-turbine (WT)	0	15	1.14
Battery	-30	30	0.4
Utility	-30	30	-

Table 5.2 MO and start-up/shut-down cost of units in micro-grid system

Type	MO cost ($/kwh)	Start-up/shut-down cost ($)
Micro-turbine (MT)	0.0475	1.02
Fuel-cell (FC)	0.0918	1.76
Photo-voltaic (PV)	0.22	0
Wind-turbine (WT)	0.56	0
Battery	-	0
Utility	-	-

Multi-microgrid System

The multi-micro-grids test system is also considered for the operation management optimization problem. Fours micro-grids are assumed as the multi-micro-grids test system. Each MG consists of a different number of PV, WT, MT, FC and BES. Figure 5.2 show the multi-micro-grids test system. Moreover, minimum and maximum power and bid cost of

Table 5.3 Emissions of the DG sources

Pollitants	CO_2	NO_x	SO_2	$PM-10$
Micro-turbine (MT)	724.6	0.2	0.004	0.041
Fuel-cell (FC)	489.4	0.014	0.003	0.001
Photo-voltaic (PV)	0	0	0	0
Wind-turbine (WT)	0	0	0	0
Battery	10	0.001	0.0002	0

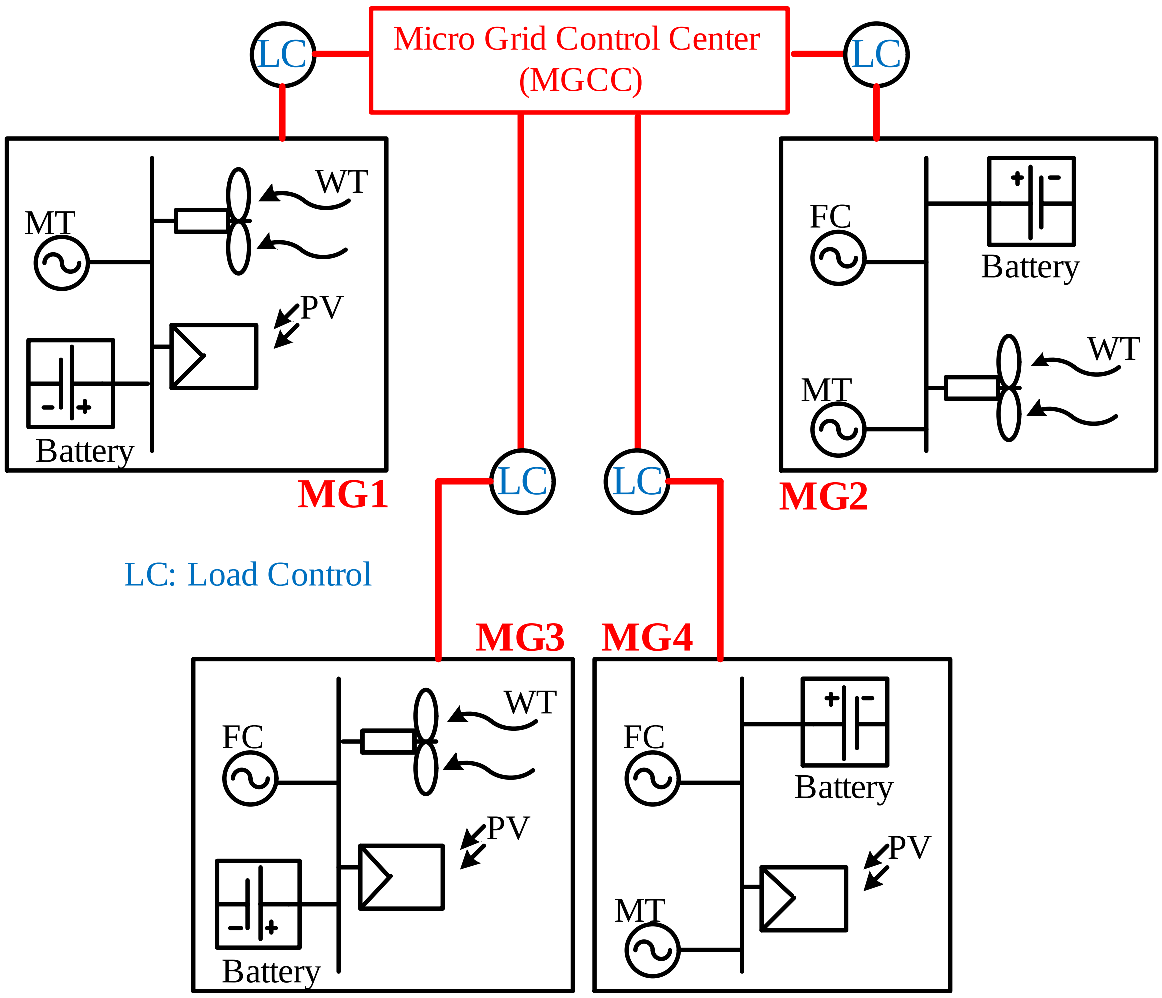


Fig. 5.2 A multi micro grid test system (four MGs)

units in the multi-micro-grids test system is shown in table 5.1. The fixed maintenance and operation, and start-up/shut-down cost of units in the multi-micro-grids test system is shown in table 5.2. The pollutant emission of each DG in the system is shown in table 5.3.

A non-dominated sorting genetic algorithm-III (NSGA-III) [22], multi-objective - covariance matrix adaptation evolution strategy (MO-CMAES) [38] and speed-constrained multi-objective particle swarm optimization (SMPSO) [58] are used to compare with FMGPSO. Each algorithm is repeatedly 30 independent trial runs. The swarm size, the archives size, a mutation rate, and the maximum iteration of FMGPSO are assumed as 100, 100, $1/n$ and 300. The population size, a crossover rate, a mutation rate and the maximum generation of NSGA-III are assumed as 100, 0.9, $1/n$ and 300, respectively. The simulated binary crossover (SBX) [21] is applied as a crossover operator in NSGA-III. The population size, neighborhood size, recombination weights, and the maximum generation of MO-CMAES are assumed as 100, 10, 0.5 and 300, respectively. The swarm size, the archives size, a mutation rate, and the maximum iteration of SMPSO are assumed as 100, 100, $1/n$ and 300.

Table 5.4 Minimum and maximum power and bid cost of units in multi-micro-grids system

MG	Type	Minimum Power (kw)	Maximum Power (kw)	Bid cost ($/kwh)
1	MT	6	30	0.49
	PV	0	25	2.75
	WT	0	15	1.14
	Battery	-30	30	0.4
	Utility	-30	30	-
2	MT	6	30	0.49
	FC	3	30	0.31
	WT	0	15	1.14
	Battery	-30	30	0.4
	Utility	-30	30	-
3	FC	3	30	0.31
	PV	0	25	2.75
	WT	0	15	1.14
	Battery	-30	30	0.4
	Utility	-30	30	-
4	MT	6	30	0.49
	FC	3	30	0.31
	PV	0	25	2.75
	Battery	-30	30	0.4
	Utility	-30	30	-

5.9.2 Simulation Results

There are two parts of the simulation results in this section. (1) the optimal solutions of the operation costs and pollutant emission optimization problem in the micro-grids test

Table 5.5 MO and start-up/shut-down cost of units in multi-micro-grids system

MG	Type	MO cost ($/kwh)	Start-up/shut-down cost ($)
1	MT	0.0475	1.02
	PV	0.22	0
	WT	0.56	0
2	MT	0.0475	1.02
	FC	0.0918	1.76
	WT	0.56	0
3	FC	0.0918	1.76
	PV	0.22	0
	WT	0.56	0
4	MT	0.0475	1.02
	FC	0.0918	1.76
	PV	0.22	0

system which solved by using FMGPSO, NSGA-III, MO-CMAES and SMPSO and (2) the optimal solutions of the operation costs and pollutant emission optimization problem in the multi-micro-grids test system which solved by using FMGPSO, NSGA-III, MO-CMAES and SMPSO. The comparison of the solutions are shown by the best solutions from 30 independent trail runs in each iteration of four algorithms.

Optimal Solution of MG System in Single-Objective Optimization Problem

In this section, the two objectives (to minimize the total cost and the total emission pollutant of the MG system) are considered as a single-objective optimization problem by using the weight values. This process is used to show how difficult to find the best weight value for the single-objective optimization problem. There are two different weight values (ω_1, and ω_2) for the single-objective PSO are used to compare the results. In the simulations, the weight values can be calculated as follows:

$$\omega_1 = random(\lambda)/\lambda \tag{5.40}$$

$$\omega_2 = 1 - \omega_1 \tag{5.41}$$

where λ denotes a random number which is a positive number by using uniformly distributed.

Therefore, the operation management problem is considered as a single-objective optimization problem in the micro-grid test system. FMGPSO is proposed to solve the problem. The best solutions of 30 independent runs of four algorithms: FMGPSO, NSGA-III, MO-

CMAES, and SMPSO, are compared and presented. The setup of simulation of each algorithm is described in section 5.9.1. There are two cases that are presented: the micro-grid test system with BES and without BES.

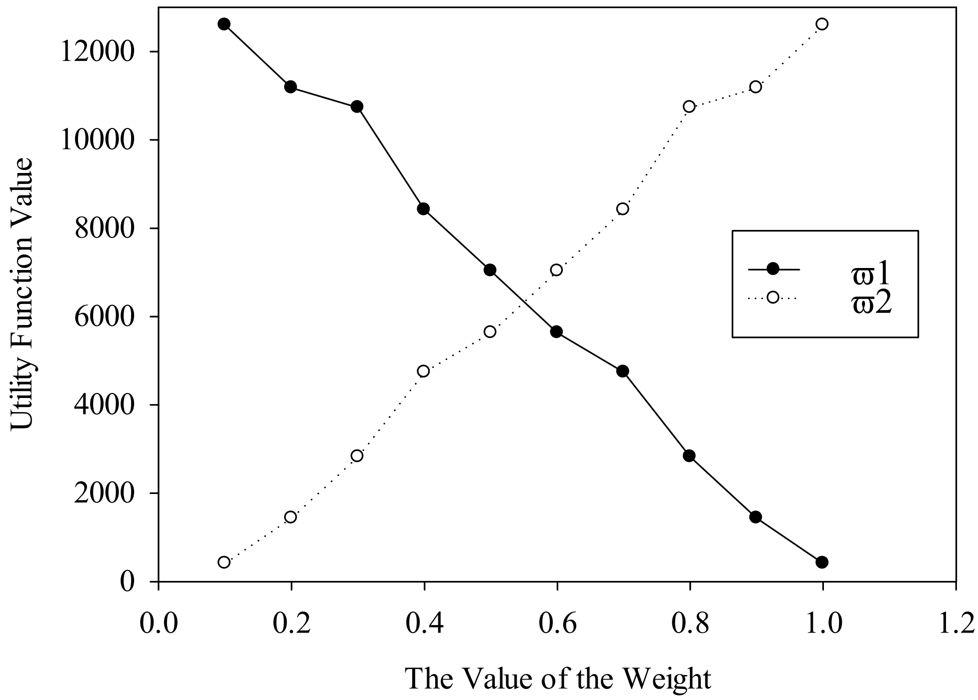


Fig. 5.3 Utility function value of each weight value of the MG system with BES

The best utility function value of each weight value of the MG system with and without BES are shown in figure 5.3 and figure 5.4, respectively. The range of weight values is from 0.1 to 1.0 which is used to compare the best utility function values. The results from the two figures show that the utility function value in the single-objective optimization depends on the weight value. Therefore, the weight value can be chosen according to an interested objective. However, it is difficult to find the best weight value for the problem.

Optimal Solution of MG System in Multi-Objective Optimization Problem

In this section, the operation management problem is considered as a multi-objective optimization problem in the micro-grid test system. FMGPSO is proposed to solve the problem. The best solutions of 30 independent runs of four algorithms: FMGPSO, NSGA-III, MO-CMAES, and SMPSO, are compared and presented. The setup of simulation of each algorithm is described in section 5.9.1.

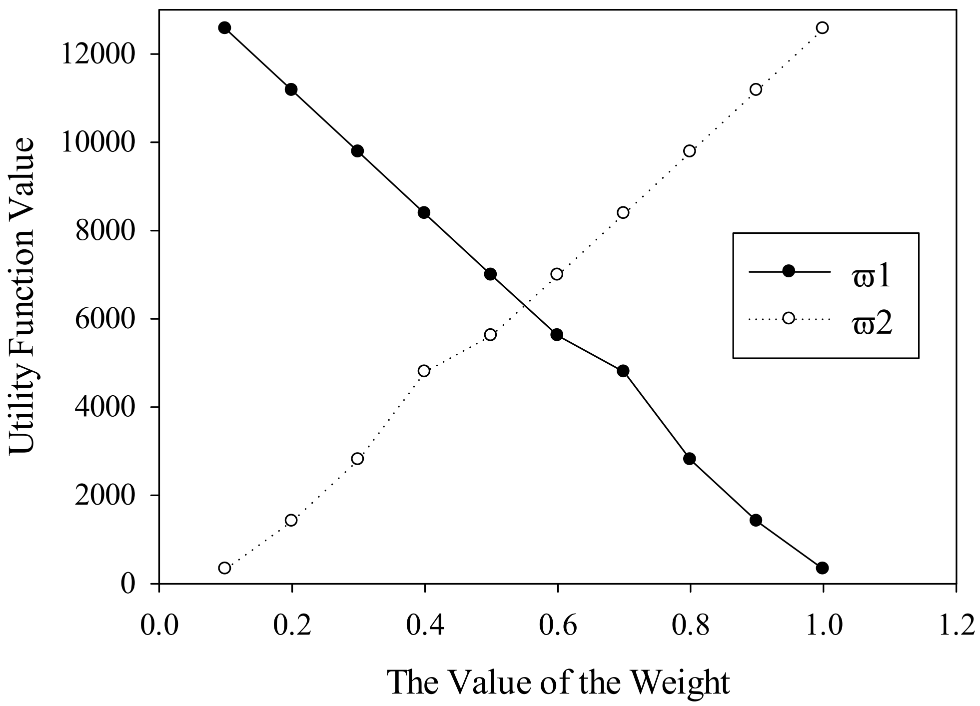


Fig. 5.4 Utility function value of each weight value of the MG system without BES

There are two cases that are presented in this section: the micro-grid test system with BES and without BES. First, the simulation results in the case of the micro-grid test system with BES are presented. The comparison of optimal operation costs and pollutant emission in the micro-grid test system with BES of 30 independent runs are shown in table 5.6 and table 5.7, respectively. The total operating cost and pollutant emission in the micro-grid system with BES at the end of each iteration are shown in figure 5.5 and figure 5.6, respectively.

Table 5.6 shows that the minimum value of the minimal operation cost of FMGPSO is \$396.10 and the maximum value of the minimal operation cost of FMGPSO is \$472.06. Moreover, table 5.7 shows that the minimum value of the minimal pollutant emissions of FMGPSO is 1396.15 kg/MWh and the maximum value of the minimal pollutant emissions of FMGPSO is 1661.86 kg/MWh. The results show that the minimum and maximum value of the minimal operation cost of FMGPSO are less than the results of other algorithms as well as the minimum and maximum value of the minimal pollutant emissions of FMGPSO. Therefore, FMGPSO is able to contribute minimal operation cost and pollutant emissions better than NSGA-III, MO-CMAES and SMPSO do. Moreover, the simulation time is also shown in table 5.6. It can see that the simulation time of the proposed algorithm FMGPSO is not the best because of fitness function based on modified game theory is used in PSO. In the other word, FMGPSO consists of two algorithms that are PSO and modified game theory. Therefore, FMGPSO has a possibility that the simulation time is higher than NSGA-III, MO-CMAES, and SMPSO. Finally, the status of the units in the micro-grid test system with BES by FMGPSO are shown in table 5.8 and the optimal output power of the units in the micro-grid test system with BES by FMGPSO are shown in table 5.9.

Moreover, figure 5.7 and 5.8 show the minimum total operating cost and pollutant emission in the micro-grid system with BES at the end of each iteration with 600 max iterations, respectively. The figures show the results from four algorithms: the proposed algorithm (FMGPSO), NSGA-III, MO-CMAES, and SMPSO. The results show that the minimum total operating cost and pollutant emission of those four algorithm less change after the 300 iterations. It is difficult to see the difference. However, the proposed algorithm of FMGPSO is still able to find the minimum total operating cost and pollutant emission better than other algorithms.

Table 5.6 Comparison of operation cost ($) and simulation time of 30 runs in case of the MG system with BES

Algorithm	Min	Avg	Max	Mean time (ms)
FMGPSO	396.10	447.70	472.06	455.8
NSGA-III	502.72	563.33	605.34	336.2
MO-CMAES	451.79	518.48	522.13	259.4
SMPSO	520.80	591.19	621.32	359.6

Table 5.7 Comparison of emissions (kg/MWh) in case of the MG system with BES

Algorithm	Min	Avg	Max
FMGPSO	1396.15	1578.03	1661.86
NSGA-III	1463.49	1639.91	2015.28
MO-CMAES	1407.64	1569.85	1952.19
SMPSO	1429.38	1708.78	2537.27

Second, the simulation results in the case of the micro-grid test system without BES are presented. The comparison of optimal operation costs and pollutants emission in the micro-grid test system without BES of 30 independent runs are shown in table 5.10 and table 5.11, respectively. The total operation cost and pollutant emission in the MG system at the end of each iteration are shown in figure 5.9 and figure 5.10, respectively.

Table 5.10 shows that the minimum value of the minimal operation cost of FMGPSO is $320.50 and the maximum value of the minimal operation cost of FMGPSO is $533.16.

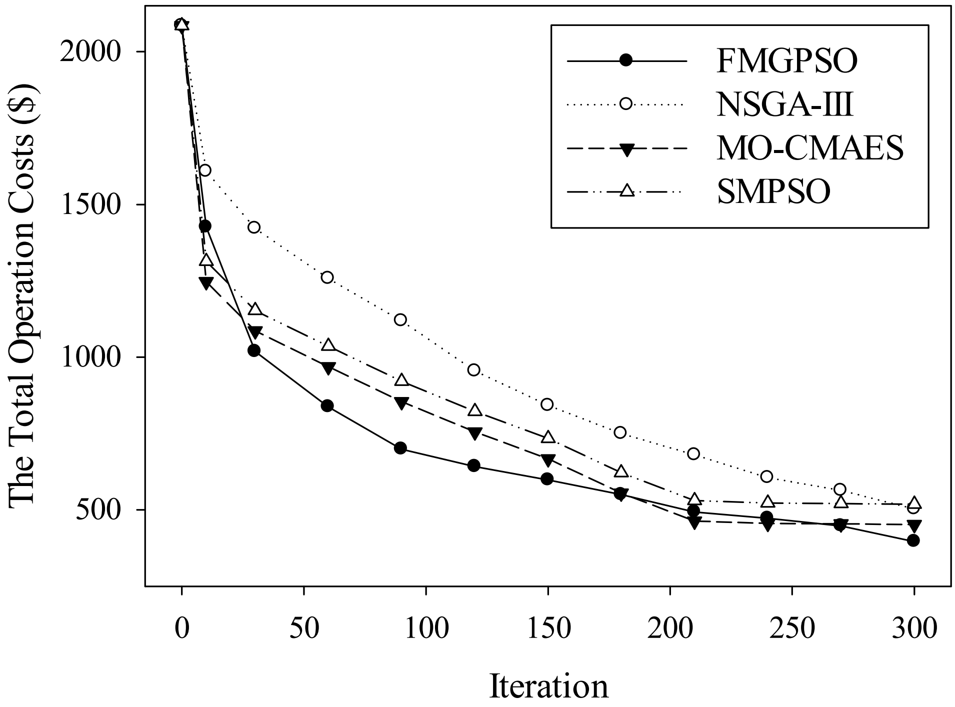


Fig. 5.5 The total operation cost of the MG system with BES at the end of each iteration

Moreover, table 5.11 shows that the minimum value of the minimal pollutant emissions of FMGPSO is 1396.15 kg/MWh and the maximum value of the minimal pollutant emissions of FMGPSO is 1569.85 kg/MWh. The results show that the minimum and maximum value of the minimal operation cost of FMGPSO are less than the results of other algorithms as well as the minimum and maximum value of the minimal pollutant emissions of FMGPSO. Therefore, FMGPSO is able to contribute minimal operation cost and pollutant emissions better than NSGA-III, MO-CMAES and SMPSO do. Moreover, the simulation time is also shown in table 5.10. It can see that the simulation time of the proposed algorithm FMGPSO is still not the best because of the same reason as FMGPSO in the case of the micro-grid test system with BES.

Moreover, figure 5.11 and 5.12 show the minimum total operating cost and pollutant emission in the micro-grid system without BES at the end of each iteration with 600 max iterations, respectively. The figures show the results from four algorithms: the proposed algorithm (FMGPSO), NSGA-III, MO-CMAES, and SMPSO. The results show that the minimum total operating cost and pollutant emission of those four algorithm less change after the 300 iterations. It is difficult to see the difference. However, the proposed algorithm

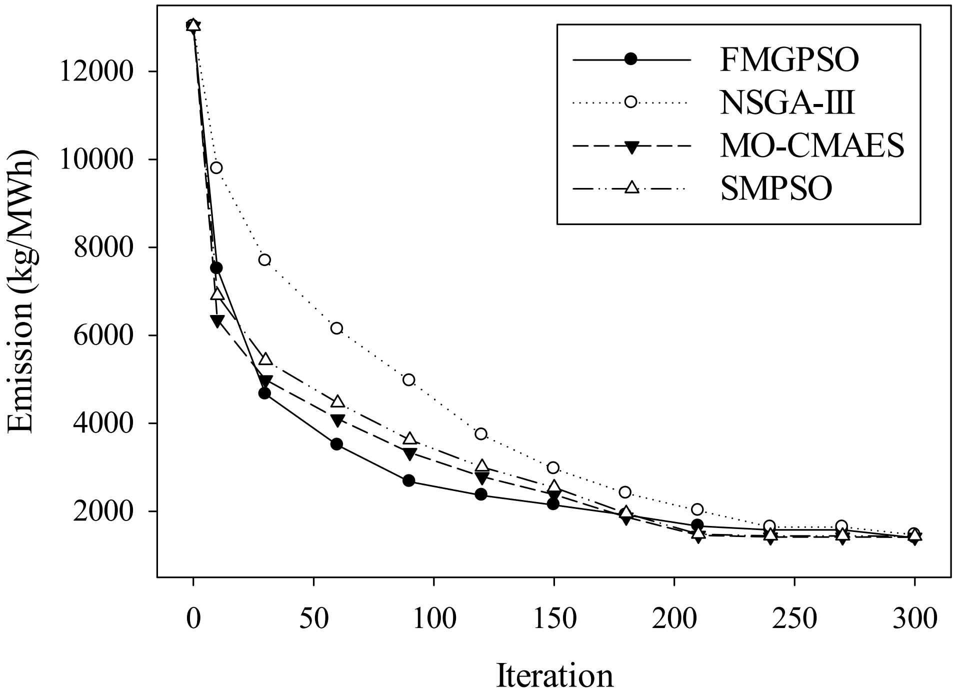


Fig. 5.6 The total emission of the MG system with BES at the end of each iteration

of FMGPSO is still able to find the minimum total operating cost and pollutant emission better than other algorithms.

Optimal Solution of Multi-micro-grids System in Multi-objective Optimization Problem

The objectives of the multi-micro-grids test system are the same with the micro-grid test system. Since there are different units in each micro-grid in the system, each term in the multi-objective optimization problem depends on the units of each micro-grid. The best solutions of 30 independent runs of four algorithms: FMGPSO, NSGA-III, MO-CMAES, and SMPSO, are compared and presented. The setup of simulation of each algorithm is described in section 5.9.1.

The comparison of optimal operation costs and pollutant emission in the multi-micro-grids test system of 30 independent runs are shown in table 5.12 and table 5.13, respectively. The total operating cost and pollutant emission in the multi-micro-grids test system at the end of each iteration are shown in figure 5.13 and figure 5.14, respectively. Table 5.12 shows the minimum value, average and maximum value of the operation cost of each micro-grid of

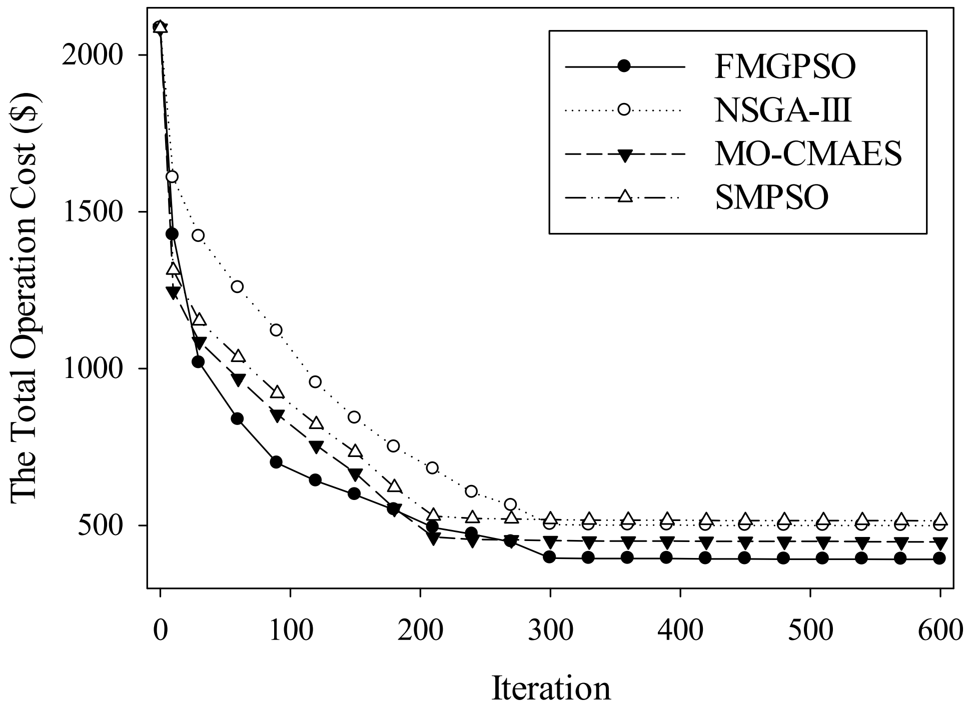


Fig. 5.7 The total operation cost of the MG system with BES in case of 600 max iterations

the four algorithms and table 5.13 shows the minimum value, average and maximum value of the pollutant emissions of each micro-grid of the four algorithms. The results show that the minimum and maximum value of the minimal operation cost and pollutant emissions of FMGPSO are better than NSGA-III, MO-CMAES, and SMPSO. Therefore, FMGPSO is able to contribute minimal total operation cost and pollutant emission better than the other algorithms do.

5.10 Conclusion

This study considered an operation management problem of micro-grid and multi-micro-grids system as an optimization problem. The objectives are to minimize the to total costs of operation and the pollutant emission in both of the micro-grid and multi-micro-grids system. There are two cases of operation management problem in micro-grid system: with BES and without BES. The micro-grid, multi-micro-grids and optimization algorithms have been conducted by using extensive simulations. There are four algorithms for comparison: (1)

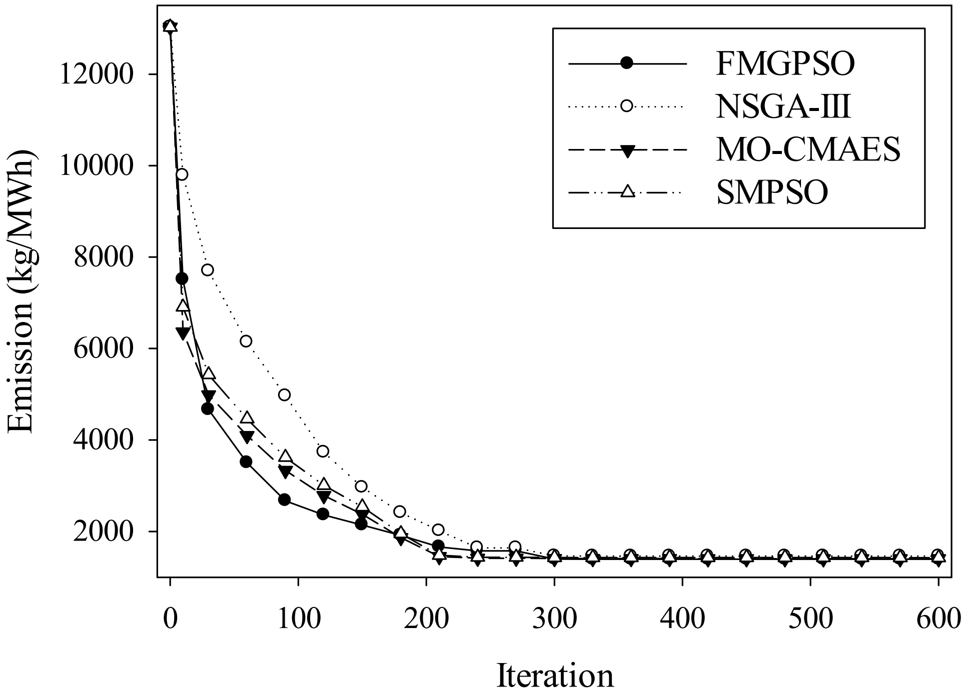


Fig. 5.8 The total emission of the MG system with BES in case of 600 max iterations

Fitness-based modified game particle swarm optimization (FMGPSO), (2) Non-dominated Sorting Genetic Algorithm III (NSGA-III), (3) multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) and (4) Speed-constrained Multi-objective PSO (SMPSO). The simulation results show that the proposed algorithm or FMGPSO succeed to find the minimum total operation costs and pollutant emission of the micro-grid and multi-micro-grids system better than other three algorithms. However, The other three algorithms are better than FMGPSO in term of the simulation time.

Table 5.8 Status of the units in the MG system with BES by FMGPSO (Total cost = 396.10 and Emission = 1396.15 kg/MWh)

Time (h)	Status (0 or 1)					
	MT	FC	PV	WT	BES	Utility
1	1	1	0	1	1	1
2	1	1	0	1	1	1
3	1	1	0	0	1	1
4	1	1	0	0	1	1
5	1	1	0	0	1	1
6	1	1	0	0	1	1
7	1	1	0	0	1	1
8	1	1	0	1	1	1
9	1	1	0	0	1	1
10	1	1	1	0	1	1
11	1	1	0	0	1	1
12	1	1	0	0	1	1
13	1	1	1	1	1	1
14	1	1	0	0	1	1
15	1	1	0	0	1	1
16	1	1	0	0	1	1
17	1	1	0	1	1	1
18	1	1	0	0	1	1
19	1	1	0	0	1	1
20	1	1	0	1	1	1
21	1	1	0	0	1	1
22	1	1	0	0	1	1
23	1	1	0	1	1	1
24	1	1	0	1	1	1

Table 5.9 Optimal output power of the units in the MG system with BES by FMGPSO (Total cost = 396.10 and Emission = 1396.15 kg/MWh)

Time (h)	Optimal Output Power (kw)					
	MT	FC	PV	WT	BES	Utility
1	6.0000	3.0000	0	0.0038	-30.0000	27.5091
2	6.0000	3.0049	0	0.0027	-26.7206	29.7407
3	6.0000	3.0055	0	0	-26.6405	22.6219
4	6.0000	3.0000	0	0	1.3605	22.3713
5	6.0000	3.0000	0	0	5.2510	28.8121
6	30.0000	3.0000	0	0	2.4112	21.8917
7	6.0000	3.0000	0	0	2.8716	23.5719
8	6.0000	3.0000	0	14.3648	0.0016	24.6808
9	6.0029	3.0006	0	0	2.5306	-12.4109
10	6.0000	3.0000	1.3305	0	11.2719	-11.1708
11	6.0000	30.0000	0	0	15.2714	-11.0020
12	6.0000	3.0000	0	0	11.2206	-17.9213
13	6.0000	3.0000	1.4705	15.0000	14.1909	-17.9781
14	6.0000	3.0000	0	0	16.7719	-18.6715
15	29.9999	3.0000	0	0	17.5309	-15.3322
16	6.0000	3.0000	0	0	14.1909	-11.7907
17	6.0622	3.0000	0	0.1552	18.1107	13.9107
18	6.0000	3.0000	0	0	22.8909	24.7015
19	6.0000	3.0000	0	0	17.8218	22.3613
20	6.0000	3.0000	0	0.0109	22.1511	11.1515
21	6.0000	3.00784	0	0	15.5838	-11.1518
22	6.0000	3.0000	0	0	13.2906	-2.5308
23	6.0000	30.0000	0	0.0043	9.1515	25.7207
24	6.0000	3.0000	0	0.0121	11.0717	21.0117

Table 5.10 Comparison of operation cost ($) and simulation time of 30 runs in case of the MG system without BES

Algorithm	Min	Avg	Max	Mean time (ms)
FMGPSO	320.50	396.10	533.16	455.8
NSGA-III	455.59	555.59	616.19	436.2
MO-CMAES	397.32	488.35	490.67	259.4
SMPSO	481.35	591.19	623.18	359.6

Table 5.11 Comparison of emissions (kg/MWh) in case of the MG system without BES

Algorithm	Min	Avg	Max
FMGPSO	1396.15	1405.09	1569.85
NSGA-III	1536.12	1543.41	1581.88
MO-CMAES	1409.64	1550.40	1649.35
SMPSO	1532.37	1620.93	1702.84

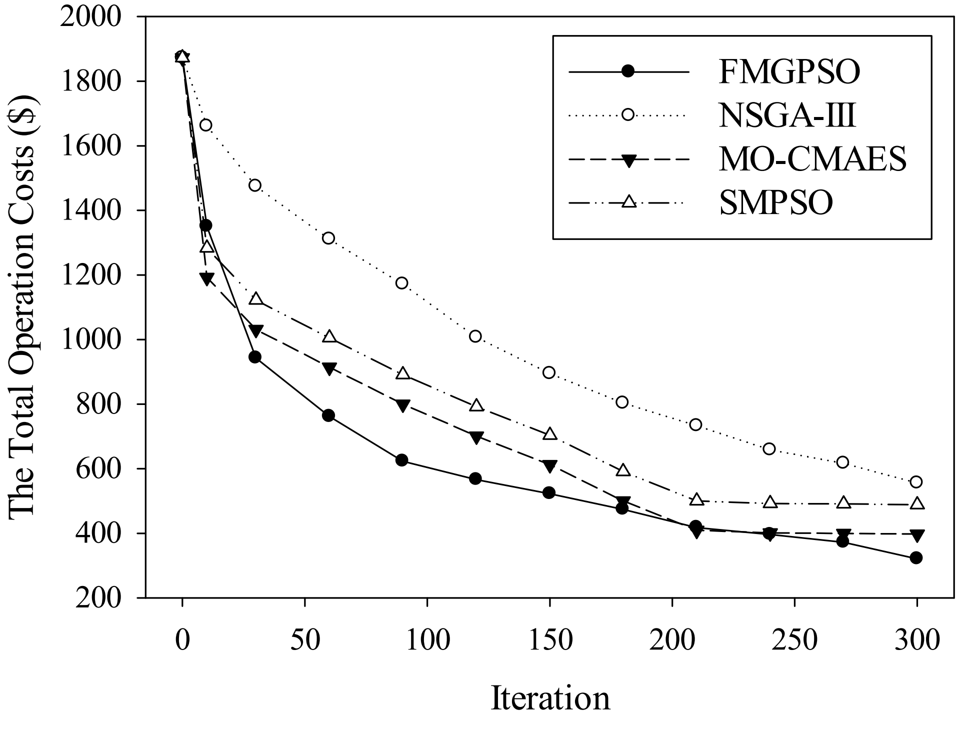


Fig. 5.9 The total operation cost of the MG system without BES at the end of each iteration

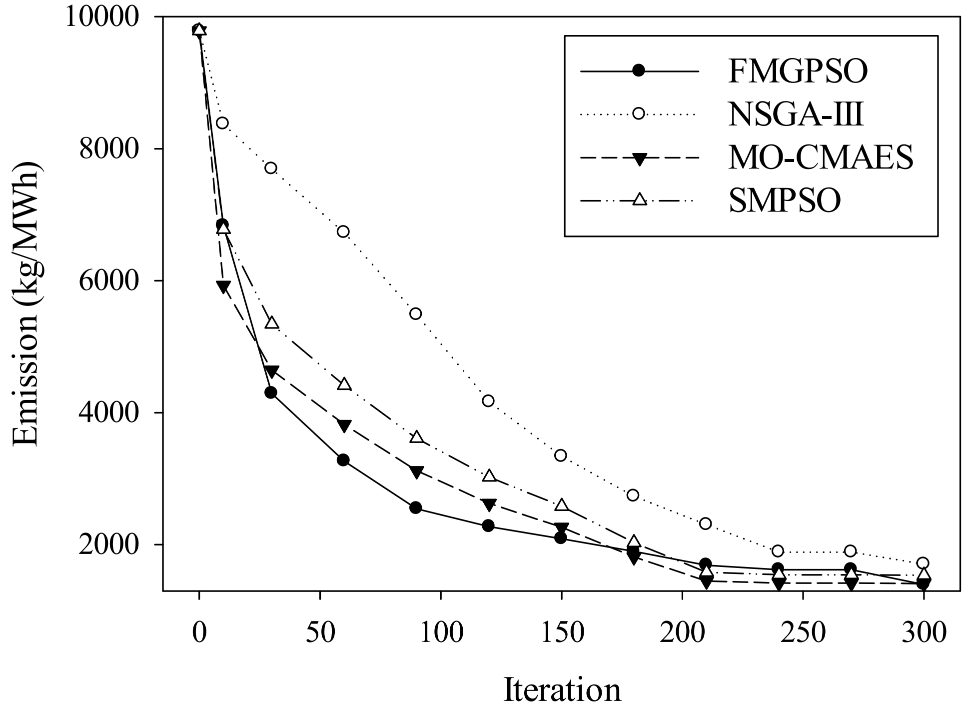


Fig. 5.10 The total emission of the MG system without BES at the end of each iteration

Table 5.12 Comparison of operation cost ($) of 30 simulation runs

Algorithm	MG	Min	Avg	Max
FMGPSO	1	378.26	378.26	378.26
	2	396.10	396.10	396.10
	3	365.32	365.32	365.32
	4	376.84	376.84	376.84
NSGA-III	1	506.31	603.87	676.39
	2	502.72	575.59	623.98
	3	509.73	598.74	645.10
	4	504.10	601.76	668.98
MO-CMAES	1	453.71	562.78	653.24
	2	451.79	569.52	631.97
	3	491.28	543.75	626.13
	4	478.98	521.32	632.78
SMPSO	1	580.27	612.58	657.14
	2	571.64	599.52	631.97
	3	593.05	623.75	656.10
	4	577.30	609.32	643.80

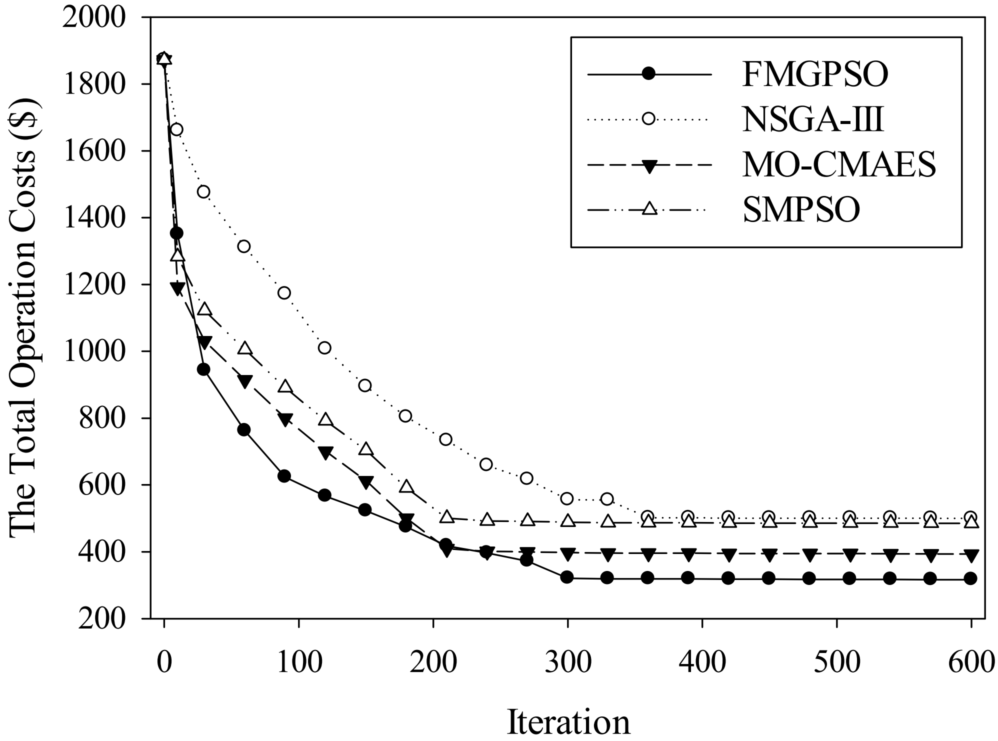


Fig. 5.11 The total operation cost of the MG system without BES in case of 600 max iterations

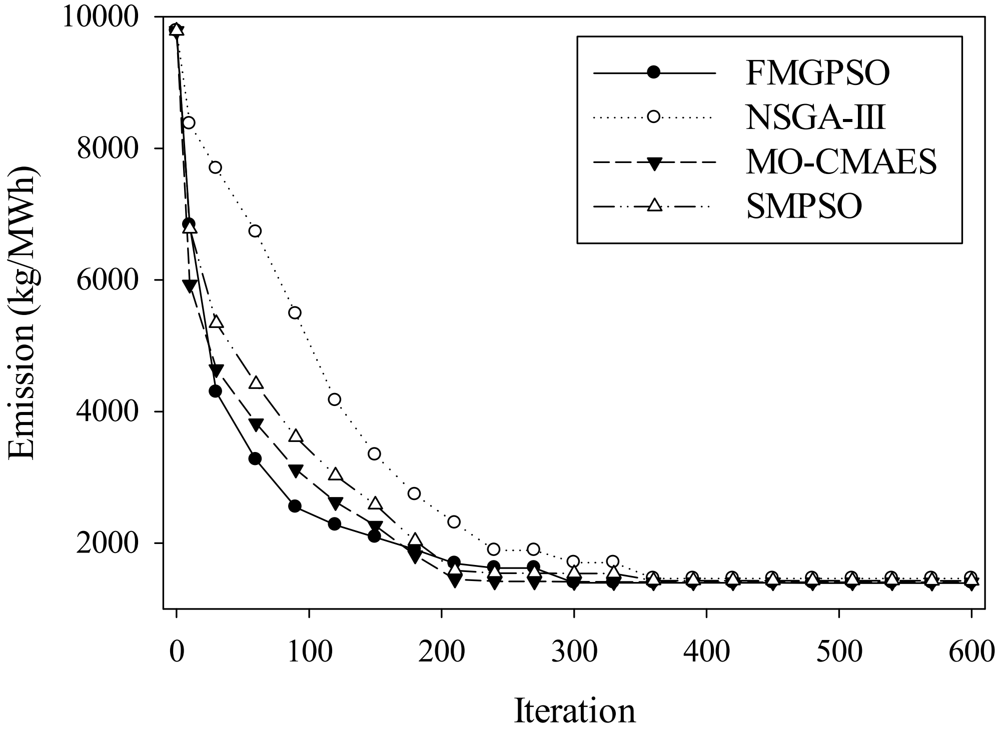


Fig. 5.12 The total emission of the MG system without BES in case of 600 max iterations

Table 5.13 Comparison of emission (kg/MWh) of 30 simulation runs

Algorithm	MG	Min	Avg	Max
FMGPSO	1	1396.15	1396.15	1396.15
	2	1362.78	1362.78	1362.78
	3	1379.13	1379.13	1379.13
	4	1366.92	1366.92	1366.92
NSGA-III	1	1459.07	1503.32	1533.97
	2	1463.49	1500.24	1530.76
	3	1456.74	1504.37	1535.61
	4	1444.04	1492.18	1520.71
MO-CMAES	1	1407.59	1452.58	1505.34
	2	1407.64	1479.53	1551.95
	3	1458.54	1503.65	1556.16
	4	1460.81	1511.37	1573.82
SMPSO	1	1417.35	1532.57	1595.34
	2	1417.41	1549.51	1632.97
	3	1429.38	1553.65	1603.52
	4	1417.32	1541.37	1598.43

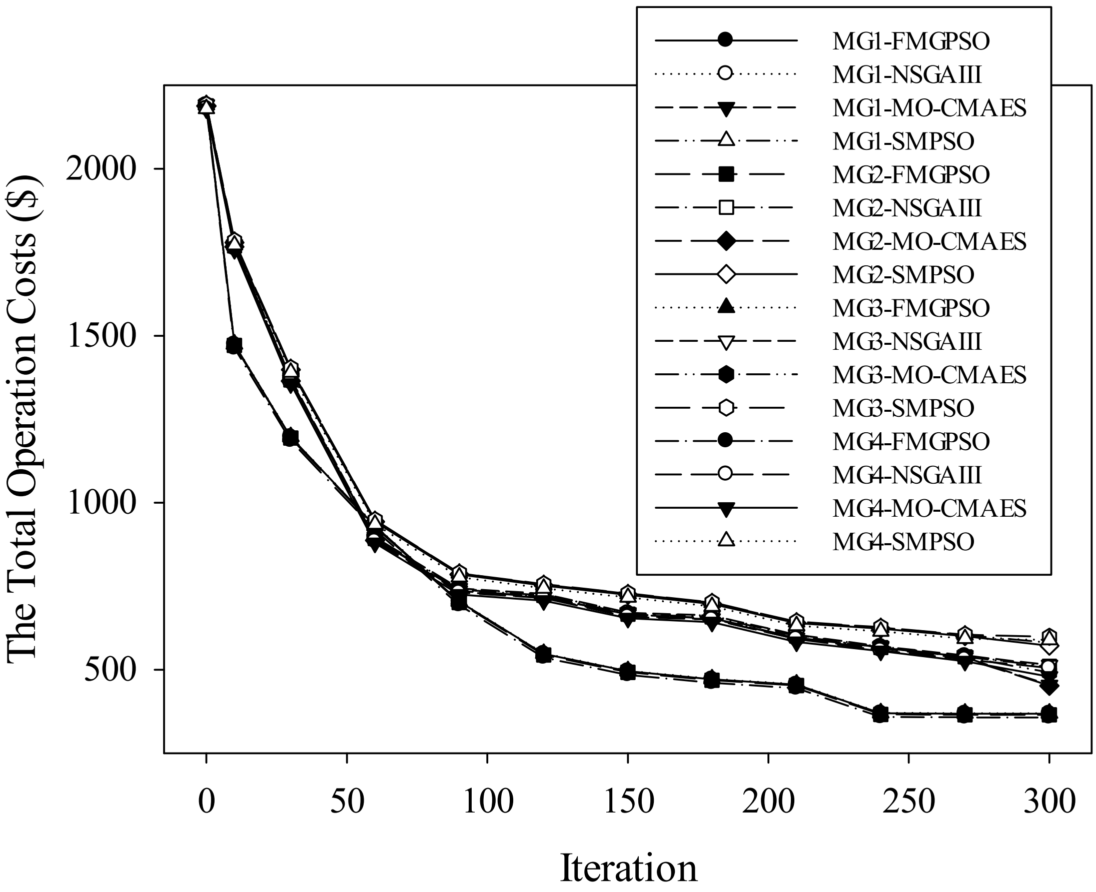


Fig. 5.13 The total operation cost of the multi-MG system at the end of each iteration

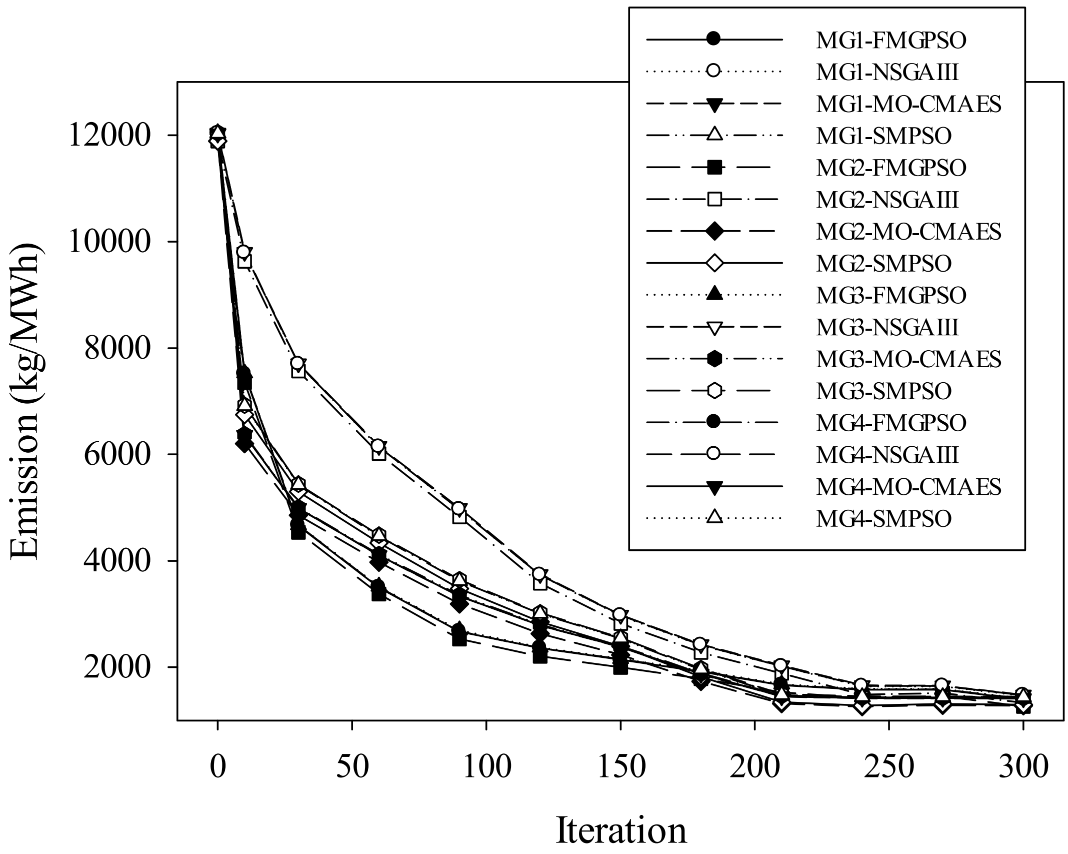


Fig. 5.14 The total emission of the multi-MG system at the end of each iteration

Chapter 6

Conclusions and Future Directions

This chapter describes conclusions and future directions of each study. Therefore, The chapter consists of three sections: (6.1) a conclusion and a future direction of the optimization of cloud brokering systems for connected IoT, (6.2) a conclusion and a future direction of the partitioning for the local outlier detection in geo-social media, and (6.3) a conclusion and a future direction of the operation management of multi-micro-grids system.

6.1 Cloud Brokering System

A multi-objective particle swarm optimization (MOPSO) was proposed for cloud brokering system to find the best deal among clients, service providers and cloud broker. The proposed algorithm is to optimize the response time of requests from clients, the energy consumption of service providers and the profit of the cloud broker. The results from extensive simulations show that the proposed algorithm is able to find suitable solutions sets for the system. Moreover, the proposed algorithm was compared with two algorithms: (1) Non-dominated Sorting Genetic Algorithm II (NSGA-II) which is a well-known genetic algorithm and (2) a random search algorithm. The comparison results show that MOPSO is able to increases the profit of the cloud broker and reduces the energy consumption and the response time better than the other two algorithms.

In future work, since there are many well-known cloud simulations such as GreenCloud, iCanCloud and CloudSim, the experiments under large-scale of the cloud brokering system on a well-known cloud simulation are able to conduct. Moreover, the complexity of MOPSO is high. Novel operators are able to propose to reduce the complexity of MOPSO.

6.2 Local Outlier Detection in Geo-social Media

The multi-objective genetic algorithm (MOGA) was investigated to optimize KD-tree partitioning for the outlier detection in a spatio-temporal domain. The surrounding neighbors with a normal distribution test (Z-test) of the values of keyword were considered in the outlier detection. There are two outlier patterns which are considered: H and L outlier patterns. Extensive simulations have been conducted, the results show that MOGA is able to find appropriate solutions sets for a KD-tree partitioning in a spatio-temporal domain. Moreover, two crossover operators were compared: (1) one-point crossover and (2) two-point crossover. The simulation results show that the two-point crossover successfully increases the number of H and L outlier patterns in the case of a large number of partition better than the one-point crossover.

In future work, a parallel processing is considered to support the processes. It is able to handle experiments with large volumes of geo-social media. Moreover, novel operators will be considered to reduce the complexity of MOPSO.

6.3 Multi-micro-grids System

The fitness-based modified game particle swarm optimization (FMGPSO) scheme was proposed for multi-micro-grids control to find the appropriate control. FMGPSO is to optimize the total costs of operation and the pollutant emission in both of the micro-grid and multi-micro-grids system. Extensive simulations have been conducted to show the performance of the proposed algorithm, the results show that the proposed algorithm is able to find appropriate solutions sets for both of the micro-grid and multi-micro-grids system. Moreover, FMGPSO was compared with three well-known algorithms: (1) Non-dominated Sorting Genetic Algorithm II (NSGA-II), (2) multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) and (3) Speed-constrained Multi-objective PSO (SMPSO). The simulation results show that the proposed algorithm successfully reduces both of the total operation costs and pollutant emission of the micro-grid and multi-micro-grids system better than the other algorithms. However, The other algorithms are better than the proposed algorithm in term of the average simulation time.

In future work, there is a cost model of purchased and sold powers among micro-grids. Therefore, it is able to consider and conduct to make realistic micro-girds system real-able. Moreover, finding novel operators for the proposed algorithm will be proposed to reduce the complexity.

References

[1] Aazam, M. and Huh, E.-N. (2014). Fog computing and smart gateway based communication for cloud of things. In *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*, pages 464–470. IEEE.

[2] Abdelhaq, H., Sengstock, C., and Gertz, M. (2013). Eventweet: Online localized event detection from twitter. *Proc. VLDB Endow.*, 6(12):1326–1329.

[3] Acuna, E. and Rodriguez, C. (2004). A meta analysis study of outlier detection methods in classification. *Technical paper, Department of Mathematics, University of Puerto Rico at Mayaguez.*

[4] Aggarwal, C. C. and Yu, P. S. (2001). Outlier detection for high dimensional data. In *ACM Sigmod Record*, volume 30, pages 37–46. ACM.

[5] Aghamohammadi, M. R. and Abdolahinia, H. (2014). A new approach for optimal sizing of battery energy storage system for primary frequency control of islanded microgrid. *International Journal of Electrical Power & Energy Systems*, 54:325–333.

[6] Angiulli, F., Basta, S., and Pizzuti, C. (2006). Distance-based detection and prediction of outliers. *Knowledge and Data Engineering, IEEE Transactions on*, 18(2):145–160.

[7] Annamdas, K. K. and Rao, S. S. (2009). Multi-objective optimization of engineering systems using game theory and particle swarm optimization. *Engineering optimization*, 41(8):737–752.

[8] Bahmani-Firouzi, B. and Azizipanah-Abarghooee, R. (2014). Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algorithm. *International Journal of Electrical Power & Energy Systems*, 56:42–54.

[9] Barnett, V. and Lewis, T. (1994). *Outliers in statistical data*, volume 3. Wiley New York.

[10] Baziar, A., Kavoosi-Fard, A., and Zare, J. (2013). A novel self adaptive modification approach based on bat algorithm for optimal management of renewable mg. *Journal of Intelligent Learning Systems and Applications*, 5(01):11.

[11] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.

[12] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM.

[13] Butun, I., Erol-Kantarci, M., Kantarci, B., and Song, H. (2016). Cloud-centric multi-level authentication as a service for secure public safety device networks. *IEEE Communications Magazine*, 54(4).

[14] Buyya, R., Broberg, J., and Goscinski, A. M. (2010). *Cloud computing: principles and paradigms*. John Wiley & Sons.

[15] Chakraborty, S., Weiss, M. D., and Simoes, M. G. (2007). Distributed intelligent energy management system for a single-phase high-frequency ac microgrid. *IEEE Transactions on Industrial electronics*, 54(1):97–109.

[16] Champrasert, P., Suzuki, J., and Otani, T. (2011). Evolutionary high-dimensional QoS optimization for safety-critical utility communication networks. *Natural Computing*, 10(4):1431–1458.

[17] Chen, C., Duan, S., Cai, T., Liu, B., and Hu, G. (2011). Smart energy management system for optimal microgrid economic operation. *IET renewable power generation*, 5(3):258–267.

[18] Chen, S., Gooi, H. B., and Wang, M. (2012). Sizing of energy storage for microgrids. *IEEE Transactions on Smart Grid*, 3(1):142–151.

[19] Cheng, T. and Wicks, T. (2014). Event detection using twitter: A spatio-temporal approach. *PLoS ONE*, 9(6):e97807.

[20] Deb, K. (2001a). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Son.

[21] Deb, K. (2001b). *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons.

[22] Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation*, 18(4):577–601.

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

[24] Dong, M., Li, H., Ota, K., Yang, L. T., and Zhu, H. (2014). Multicloud-based evacuation services for emergency management. *IEEE Cloud Computing*, 1(4):50–59.

[25] Dutta, D., Goel, A., and Heidemann, J. (2003). Oblivious aqm and nash equilibria. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 106–113. IEEE.

[26] Ekren, O. and Ekren, B. Y. (2010). Size optimization of a pv/wind hybrid energy conversion system with battery storage using simulated annealing. *Applied Energy*, 87(2):592–598.

[27] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1998). Clustering for mining in large spatial databases. *KI*, 12(1):18–24.

[28] Feller, E., Rilling, L., and Morin, C. (2011). Energy-aware ant colony based workload placement in clouds. In *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*. IEEE Computer Society.

[29] Frey, S., Fittkau, F., and Hasselbring, W. (2013). Search-based genetic optimization for deployment and reconfiguration of software in the cloud. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 512–521. IEEE Press.

[30] Gao, Y., Guan, H., Qi, Z., Hou, Y., and Liu, L. (2013). A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *Journal of Computer and System Sciences*, 79(8):1230–1242.

[31] Garcia Lopez, P., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., Barcellos, M., Felber, P., and Riviere, E. (2015). Edge-centric computing: Vision and challenges. *ACM SIGCOMM Computer Communication Review*, 45(5):37–42.

[32] Garey, M. R. and Johnson, D. S. (1979). Computers and intractability: a guide to the theory of np-completeness. 1979. *San Francisco, LA: Freeman*.

[33] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

[34] Guha, S., Rastogi, R., and Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, volume 27, pages 73–84. ACM.

[35] Hao, G., Cong, R., and Zhou, H. (2014). Pso applied to optimal operation of a micro-grid with wind power. In *Parallel Architectures, Algorithms and Programming (PAAP), 2014 Sixth International Symposium on*, pages 46–51. IEEE.

[36] Hosseini, S. M. et al. (2014). Optimization of microgrid using quantum inspired evolutionary algorithm. *International Journal of Intelligent Systems and Applications*, 6(9):47.

[37] Hwang, K., Dongarra, J., and Fox, G. C. (2013). *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann.

[38] Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evolutionary computation*, 15(1):1–28.

[39] Iturriaga, S., Nesmachnow, S., Dorronsoro, B., Talbi, E.-G., and Bouvry, P. (2013). A parallel hybrid evolutionary algorithm for the optimization of broker virtual machines subletting in cloud systems. *IEEE 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pages 594–599.

[40] Jeyarani, R., Nagaveni, N., and Ram, R. V. (2012). Design and implementation of adaptive power-aware virtual machine provisioner (APA-VMP) using swarm intelligence. *Future Generation Computer Systems*, 28(5):811–821.

[41] Johnson, T., Kwok, I., and Ng, R. T. (1998). Fast computation of 2-dimensional depth contours. In *KDD*, pages 224–228. Citeseer.

[42] Kessaci, Y., Melab, N., and Talbi, E.-G. (2013a). A pareto-based genetic algorithm for optimized assignment of vm requests on a cloud brokering environment. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2496–2503. IEEE.

[43] Kessaci, Y., Melab, N., and Talbi, E.-G. (2013b). A pareto-based metaheuristic for scheduling hpc applications on a geographically distributed cloud federation. *Cluster Computing*, 16(3):451–468.

[44] Kim, H.-M. and Kinoshita, T. (2010). A new challenge of microgrid operation. *Security-Enriched Urban Computing and Smart Grid*, pages 250–260.

[45] Knorr, E. M. and Ng, R. T. (1997). A unified notion of outliers: Properties and computation. In *KDD*, pages 219–222.

[46] Kut, A. and Birant, D. (2006). Spatio-temporal outlier detection in large databases. *CIT. Journal of computing and information technology*, 14(4):291–297.

[47] Le Berre, M., Hnaien, F., and Snoussi, H. (2011). Multi-objective optimization in wireless sensors networks. In *IEEE International Conference on Microelectronics (ICM)*.

[48] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R. (2006). Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 25–25. IEEE.

[49] Legillon, F., Melab, N., Renard, D., and Talbi, E.-G. (2013). Cost minimization of service deployment in a multi-cloud environment. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2580–2587. IEEE.

[50] Li, H., Dong, M., Liao, X., and Jin, H. (2015). Deduplication-based energy efficient storage system in cloud environment. *The Computer Journal*, 58(6):1373–1383.

[51] Li, H., Dong, M., Ota, K., and Guo, M. (2016). Pricing and repurchasing for big data processing in multi-clouds. In *IEEE Transactions on Emerging Topics in Computing*. 2016, 10.1109/TETC.2016.2517930.

[52] Liu, J., Jiang, X., Nishiyama, H., Miura, R., Kato, N., and Kadowaki, N. (2012). Optimal forwarding games in mobile ad hoc networks with two-hop f-cast relay. *IEEE Journal on Selected Areas in Communications*, 30(11):2169–2179.

[53] Liu, J., Kato, N., Ma, J., and Kadowaki, N. (2014). Device-to-device communication in LTE-advanced networks: a survey. *IEEE Communications Surveys & Tutorials*, 17(4):1923–1940.

[54] Man, M. (2015). Big data and the internet of things.

[55] Mateos, C., Pacini, E., and Garino, C. G. (2013). An ACO-inspired algorithm for minimizing weighted flowtime in cloud-based parameter sweep experiments. *Advances in Engineering Software*, 56:38–50.

[56] Mitra, J. (2010). Reliability-based sizing of backup storage. *IEEE Transactions on Power Systems*, 25(2):1198–1199.

[57] Moghaddam, A. A., Seifi, A., Niknam, T., and Pahlavani, M. R. A. (2011). Multi-objective operation management of a renewable mg (micro-grid) with back-up micro-turbine/fuel cell/battery hybrid power source. *Energy*, 36(11):6490–6507.

[58] Nebro, A. J., Durillo, J. J., Garcia-Nieto, J., Coello, C. C., Luna, F., and Alba, E. (2009). Smpso: A new pso-based metaheuristic for multi-objective optimization. In *Computational intelligence in miulti-criteria decision-making, 2009. mcdm'09. ieee symposium on*, pages 66–73. IEEE.

[59] Niknam, T., Golestaneh, F., and Malekpour, A. (2012). Probabilistic energy and operation management of a microgrid containing wind/photovoltaic/fuel cell generation and energy storage devices based on point estimate method and self-adaptive gravitational search algorithm. *Energy*, 43(1):427–437.

[60] Pandey, S., Wu, L., Guru, S. M., and Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*.

[61] Poli, R. and Langdon, W. B. (1998). Schema theory for genetic programming with one-point crossover and point mutation. *Evolutionary Computation*, 6(3):231–252.

[62] Raja, P. V. and Bhaskaran, V. M. (2012). An effective genetic algorithm for outlier detection. *International Journal of Computer Applications*, 38(6):30–33.

[63] Rextin, A. T., Irfan, Z., and Uzmi, Z. A. (2004). Games networks play a game theoretic approach to networks. In *Parallel Architectures, Algorithms and Networks, 2004. Proceedings. 7th International Symposium on*, pages 451–456. IEEE.

[64] Reyes-Sierra, M. and Coello, C. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3):287–308.

[65] Ruts, I. and Rousseeuw, P. J. (1996). Computing depth contours of bivariate point clouds. *Computational Statistics & Data Analysis*, 23(1):153–168.

[66] Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D., and Sperling, J. (2009). TwitterStand: News in Tweets. In *Proc. of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 42–51.

[67] Sharma, S., Bhattacharjee, S., and Bhattacharya, A. (2015). Operation cost minimization of a micro-grid using quasi-oppositional swine influenza model based optimization with quarantine. *Ain Shams Engineering Journal*.

[68] Sharma, S., Bhattacharjee, S., and Bhattacharya, A. (2016). Grey wolf optimisation for optimal sizing of battery energy storage device to minimise operation cost of microgrid. *IET Generation, Transmission & Distribution*, 10(3):625–637.

[69] Shekhar, S., Jiang, Z., Ali, R., Eftelioglu, E., Tang, X., Gunturi, V., and Zhou, X. (2015). Spatiotemporal Data Mining: A Computational Perspective. *ISPRS International Journal of Geo-Information*, 4:2306–2338.

[70] Shelton, T., Poorthuis, A., Graham, M., and Zook, M. (2014). Mapping the data shadows of hurricane sandy: Uncovering the sociospatial dimensions of 'big data'. *Geoforum*, 52(0):167–179.

[71] Somasundaram, T. S. and Govindarajan, K. (2014). Cloudrb: A framework for scheduling and managing high-performance computing (hpc) applications in science cloud. *Future Generation Computer Systems*, 34:47–65.

[72] Su, Z., Xu, Q., Fei, M., and Dong, M. (2016). Game theoretic resource allocation in media cloud with mobile social users. *IEEE Transactions on Multimedia*, 18(8):1650–1660.

[73] Sugitani, T., Shirakawa, M., Hara, T., and Nishio, S. (2013). Detecting local events by analyzing spatiotemporal locality of tweets. In *27th International Conference on Advanced Information Networking and Applications Workshops*, pages 191–196.

[74] Sun, Z., Zhang, Y., Nie, Y., Wei, W., Lloret, J., and Song, H. (2016). Casmoc: a novel complex alliance strategy with multi-objective optimization of coverage in wireless sensor networks. *Wireless Networks*, pages 1–22.

[75] Tan, X., Li, Q., and Wang, H. (2013). Advances and trends of energy storage technology in microgrid. *International Journal of Electrical Power & Energy Systems*, 44(1):179–191.

[76] Utkarsh, K., Trivedi, A., Srinivasan, D., and Reindl, T. (2017). A consensus-based distributed computational intelligence technique for real-time optimal control in smart distribution grids. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(1):51–60.

[77] Wei, W., Fan, X., Song, H., Fan, X., and Yang, J. (2016). Imperfect information dynamic stackelberg game based resource allocation using hidden markov for cloud computing. *IEEE Transactions on Services Computing*.

[78] Wu, E., Liu, W., and Chawla, S. (2010). Spatio-temporal outlier detection in precipitation data. In *Knowledge discovery from sensor data*, pages 115–133. Springer.

[79] Xu, J. and Fortes, J. A. (2010). Multi-objective virtual machine placement in virtualized data center environments. In *2010 IEEE/ACM Int'l Conference on Green Computing and Communications (GreenCom) & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*.

[80] Yildirim, K., Kalayci, T., and Ugur, A. (2008). Optimizing coverage in a k-covered and connected sensor network using genetic algorithms. In *Proceedings of the 9th WSEAS International Conference on Evolutionary Computing*. World Scientific and Engineering Academy and Society (WSEAS).

[81] Zhou, Z., Dong, M., Ota, K., Shi, R., Liu, Z., and Sato, T. (2015). Game-theoretic approach to energy-efficient resource allocation in device-to-device underlay communications. *IET Communications*, 9(3):375–385.

[82] Zitzler, E. and Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study And the Strength Pareto Approach. *IEEE Trans. on Evolutionary Computation*, 3(4):257–271.

Publications

Journals

1. Teerawat Kumrai, Kaoru Ota, Mianxiong Dong, Kazuhiko Sato, Jay Kishigami, "Optimizing Operation Management for Multi-Microgrids Control," IET Cyber-Physical Systems: Theory & Applications, Volume 3, Issue 1, pp. 24 – 33, March 2018.

2. Kaoru Ota, Teerawat Kumrai, Mianxiong Dong, Jay Kishigami, Minyi Guo, "Smart infrastructure design for Smart Cities," IEEE IT Professional, Vol. 19, Issue. 5, pp. 42-49, September/October 2017. (IF: 1.661)

3. Teerawat Kumrai, Kaoru Ota, Mianxiong Dong, Jay Kishigami, Dan Keun Sung, "Multi-objective Optimization in Cloud Brokering Systems for Connected Internet of Things," IEEE Internet of Things Journal, Vol. 4, Issue. 2, pp. 404-413, April 2017. (IF: 7.596) **(2017 IEEE Sapporo Section Best Paper Award)**

Proceeding of International Conference

1. Teerawat Kumrai, Kyoung-Sook Kim, Mianxiong Dong, Hirotaka Ogawa, "Optimal KD-Partitioning the Local Outlier Detection," The 14th International Symposium on Neural Networks (ISNN 2017), Sapporo, Hokkaido, Japan, June 21-23, 2017.