Optimal Coalition Structure Generation on Large-Scale Renewable Energy Smart Grids

SAPERE AUDE

UNIVERSITY
of
OTAGO
Te Whare Wānanga o Otāgo
NEW ZEALAND

Sean Hsin-Shyuan Lee

a thesis submitted for the degree of

Doctor of Philosophy

at the University of Otago, Dunedin
New Zealand

December 4, 2020

To my parents and my wife

Abstract

Most renewable energy sources are dependent on unpredictable weather conditions, which have considerable variation over space and time. The intermittent nature of this production means that any renewable energy prosumer may sometimes produce an amount of energy in excess of its local consumption needs and sometimes in deficiency. This thesis is concerned with developing methods that can improve the effectiveness and widespread adoption of renewable energy usage. In order for renewable energy to be more economically viable, there needs to be a scheme for sharing energy among the prosumers so that those with excess energy can give their excess amounts to those in energy deficiency. That is the task addressed in this thesis.

The way to deal with this problem is to setup an optimal arrangement of local coalitions of renewable energy prosumers such that energy is shared within the coalitions in an optimally efficient manner. As is formally explained early on in this work, finding such an optimal coalition arrangement is an example of a Coalition Structure Generation (CSG) problem. The most straightforward way to find an optimal solution for a given pool of prosumer agents in these circumstances is to examine every possible coalition partition (coalition structure) and evaluate its comparative utility. This is known as "exhaustive search" (ES) and can be computationally expensive. As has been shown earlier, the number of such evaluations in ES even for a pool of twenty agents can be in the tens of trillions.

The problem for us in the renewable energy domain is that, because of the constantly changing weather conditions among the scattered prosumers, the CSG optimization calculation must be carried out every hour of the day. This means that the ES approach in the CSG optimization calculation for a reasonable number of prosumer agents is computationally intractable. So a more computationally feasible stochastic optimization method must be used, which searches through the coalition structure search space in order to find a reasonably good solution even if it is not the global optimum.

To this end a number of stochastic optimization search methods have been investigated in this thesis, including some of our own novel extensions to existing approaches. These search methods have been examined with respect to two different connection arrangements with respect to the outside world – (1) when the local prosumer networks have a connection to a public utility power grid and can therefore buy needed energy (at a high price) from the grid and sell excess energy

(at a low price) to the grid and (2) when the local prosumer networks are isolated from any public utility, which is referred to as "island mode". The overall goal of these investigations has been to find an optimization approach that arrives at a near-optimal (near the global optimum of the given search space) that is computationally efficient (i.e. it does not require a vast amount of computer memory or running time).

Based on these empirical examinations, which have employed realistic parameters drawn from existing consumption and renewable energy data sets, the following conclusions concerning renewable energy can be drawn from this study:

- It is feasible to employ ordinary computer resources to obtain on an hourly basis near-optimal energy-sharing coalition structures that will lead to more effective and economical use of renewable energy.
- This energy-sharing approach will contribute to a more rapid adoption and proliferation of existing renewable energy equipment and infrastructure.

The principal contributions towards these end that this thesis work has made are as follows:

- A modelling framework has been setup that can be used for extensive empirical determinations of near-optimal energy-sharing coalition structures.
- A detailed empirical study has been carried out that has examined the relative capabilities in this context of various optimal coalition structure search methods, including genetic algorithms (GA), dynamic programming (DP), particle-swarm optimization (PSO), population-based incremental learning (PBIL), and several variants to PBIL.
- The novel extensions to basic PBIL optimization have included Top-k Merit Weighting PBIL (PBIL-MW), Set-ID Encoding Schemes, and Hierarchical PBIL-MW.

Acknowledgements

In the study and research leading up to this thesis, I would like to take the opportunity to thank those who provided support and assistance. First and foremost, I would like to express my deep appreciation to my three supervisors, Associate Professor Jeremiah D. Deng, Professor Martin K. Purvis, and Dr Maryam Purvis, for their unwavering support and consistent encouragement during my study. I have benefited excessively from their advice, enthusiasm and immense knowledge. I have learned the fundamentals of research as well as of life from them. I also thank Dr Lizhi Peng and Dr Xiao-Min Hu for their valuable feedback and suggestions. This thesis would not have been possible without their help.

As a student of the Information Science Department, University of Otago, I wish to thank all the staff who have given me their generous assistance during the journey. I am grateful to all the members of the department. They gave me much useful information on settling into university life and provided enormous help with preparing for my conference travelling. I also want to thank the technical support group team who gave me great assistance in setting up the research environment and found practical solutions whenever I had problems with the computer or network. I also thank all lecturers and administration staff in the department for their priceless advice.

I am also grateful to my talented and friendly postgraduate colleagues: Muhammad Yasir, Hanhe Lin, and Robert Hou for sharing their research skills and experiences. I earned valuable benefits from their generous feedback and assistance in preparing for the conference. My gratitude to the University of Otago and Otago Energy Research Centre for providing funding for attending the conference. Last but not least, I wish to thank all those who have also supported me during my PhD study course.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

This thesis describes the design and evaluation of a model that facilitates the efficient sharing of electrical energy among a distributed network of energy producers and consumers. This model is particularly suited for sharing energy from renewable resources, which by their nature have variable patterns of energy production. Thus, the sharing arrangement needs repeated rescheduling as the weather and consumption conditions frequently change. The model described here provides a highly efficient method of calculating collective energy-sharing arrangements that up untill now have been considered computationally intractable for a reasonable number of agents.

The rest of this chapter provides some contextual background information for the motivation for this work. Section 1.2 of this chapter briefly describes the issues of global warming and why fossil fuels need to be replaced by renewable energy sources. Section 1.3 gives a brief description of renewable energy characteristics, particularly those associated with the two primary sources of renewable power generation – solar energy and wind energy. Following on from what has been discussed in Section 1.3, Section 1.4 describes why it is advantageous, both economically and environmentally, to have renewable energy sources connected in a network. We call a locally connected network of renewable energy sources a coalition, and in Section 1.5, we describe why it is necessary to frequently recalculate and reassign memberships of coalitions. This optimisation is the major research issue addressed in this thesis: how to accomplish this optimal coalition membership calculation in an efficient and scalable manner. Section 1.6 briefly summarises how this issue has been addressed in this thesis and what our principal contributions in this area have been.

1.2 Emergence of Renewable Energy

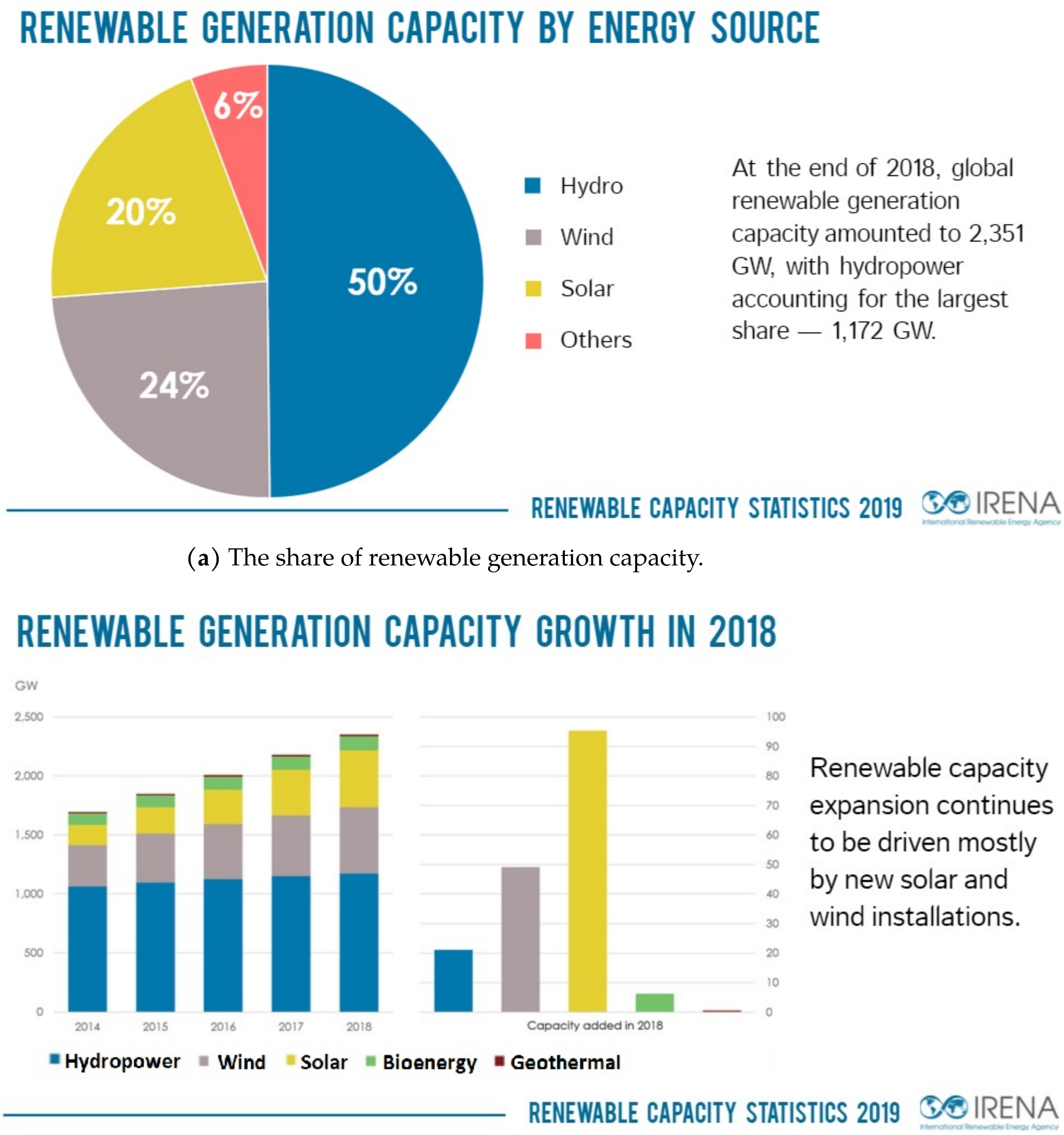
Throughout history, and still today, fossil fuels (such as petroleum, coal, and natural gas) have been the primary sources to support the world energy requirements (Hubbert, 1949). Fossil fuels are known as depletion resources, that are only available in limited locations. Therefore, due to their restrictive character, they have caused a lot of political issues and economic conflicts in the world. Also, to make matters worse, the combustion of fossil fuel to provide energy has brought substantial greenhouse gases into the Earth's atmosphere (Hoel & Kverndokk, 1996).

Since the early 1990s, remarkable scientific evidence and research results have revealed that the global warming phenomenon is a clear and inevitable trend (Höök & Tang, 2013; Zou et al., 2016). Additionally, the consequential disasters being caused by global climate change are looming (Lobell & Field, 2007; Weber, 2006). Concerning the origin of global warming, most scientists confirm that greenhouse gases, which are mainly produced by the use of fossil fuel, are the primary sources which accelerate global warming (Capellán-Pérez et al., 2014; Ellabban et al., 2014; Hoel & Kverndokk, 1996).

To eliminate fossil fuel production in order to mitigate the massive emission of greenhouse gasses and to avert unprecedented catastrophes of the Earth's ecosystem, the UN adopted the United Nations Framework Convention on Climate Change (UNFCCC) in 1992, aiming to "stabilise greenhouse gas concentrations in the atmosphere at a level that would prevent dangerous anthropogenic interference with the climate system" (Bodansky, 1993)(*p.455*). Besides, to face the ees of climate change and to realise energy sustainability and a cleaner air target, the requirement to speed up the development of low-carbon energy technologies is becoming a primary subject worldwide (IEA, 2011; Zou et al., 2016).

Renewable resources (such as firewood, waterwheels, windmills and hot spring heating) have long been used in human history (Sørensen, 1991). However, due to the limitation of scientific knowledge and technology, except for hydro-power generation, a large portion of these abundant resources have been ineffectively utilised, and some even were unavailable until a few decades ago (Manolopoulos et al., 2016; Kelly, 2011). Nevertheless, arising from the concerns of global climate change and with the substantial progress of science and technology in renewable energy (RE), the exploitation and application of RE have now become part of mainstream research in the goal of a sustainable future for the Earth and humanity (Hussain et al., 2017).

Nowadays, the utilisation of RE has been widely acknowledged as one of the best solutions for substituting the massive utilisation of fossil fuel (Hussain et al., 2017; Zou et al., 2016). The growth of RE generation, as shown in Figure 1.1, has continuously increased the penetration of RE and gradually taken over fossil fuel in energy supplies (Sequeira & Santos, 2018; Ellabban et al., 2014; IRENA, 2019). Furthermore, from the Figure, it is clear that solar and wind have driven more capacity among these resources in recent years.


RENEWABLE GENERATION CAPACITY BY ENERGY SOURCE

At the end of 2018, global renewable generation capacity amounted to 2,351 GW, with hydropower accounting for the largest share — 1,172 GW.

RENEWABLE CAPACITY STATISTICS 2019 IRENA

(a) The share of renewable generation capacity.

RENEWABLE GENERATION CAPACITY GROWTH IN 2018

Renewable capacity expansion continues to be driven mostly by new solar and wind installations.

RENEWABLE CAPACITY STATISTICS 2019 IRENA

(b) The growth of renewable generation capacity.

Figure 1.1: Statistics for global renewable generation capacity (IRENA, 2019)). (Source: https://www.irena.org/-/media/Images/IRENA/Infographics/2019/Mar/ Accessed: 31 Dec. 2019.)

1.3 The Nature of Renewable Energy Resources

From the previous section, we know that solar and wind energy have grown more rapidly than other resources. This success is attributed to their natural characteristics, economic scalability, and easy customisation for versatile consumers.

Sunshine and wind always surround us in our environment and are the most convenient and available RE sources for human consumption. For example, a household can directly install solar panels (notably photovoltaic (PV) panels) on top of the roof and integrate the solar power generator with the private power supply system. Besides this, customers can also connect a personal power line to the public grid via a smart meter (Kádár & Varga, 2012).

Nowadays, the traditional power distribution network is rapidly transforming into a smart grid. A smart grid, as defined by the European Union Commission (2010), "is an electricity network that can cost-efficiently integrate the behaviour and actions of all users connected to it – generators, consumers and those that do both – in order to ensure economically efficient, sustainable power systems with low losses and high levels of quality and security of supply and safety."(*p.3*)

Accordingly, when there is sufficient energy generated by solar radiation, then the household can fulfil the power consumption without the input from the public grid. What is more, the customer might sell excess energy to the utility as well. As a result, the solar power user could generate power to meet personal energy consumption during the bright sunshine hours. However, the user may demand power from the grid while there is less or no available solar source. Figure 1.2 shows a typical roof with installed solar panels in New Zealand.

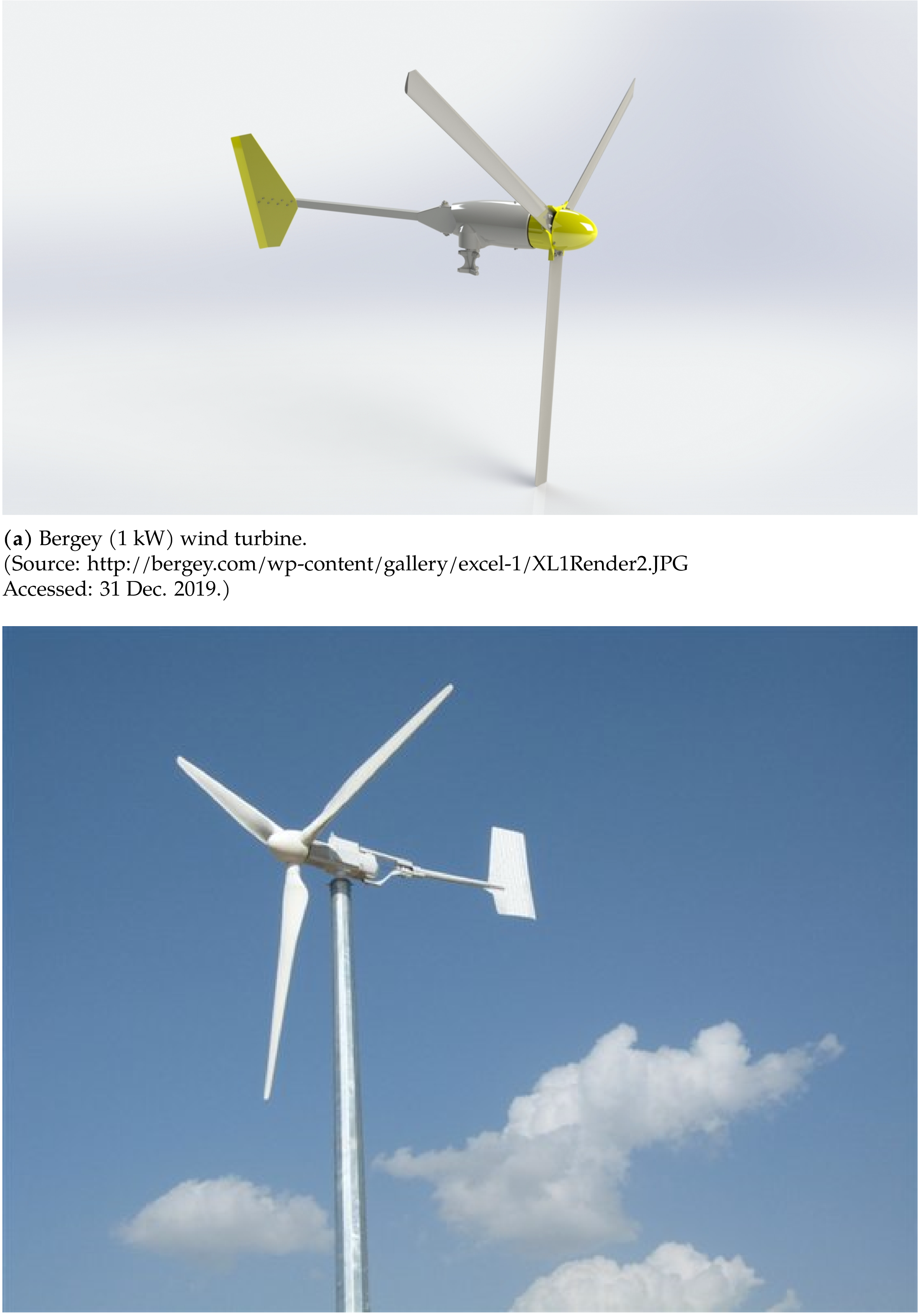
Similar to solar energy, there are many commercialised wind turbines available for a customer to select from. Consequently, the wind energy generated from turbines can provide sufficient power for the users, or even contribute to the grid, while the wind speed is fast and prolonged enough. However, like the solar user, wind turbine users will need additional energy from the grid when the wind is mild or still. Figure 1.3 (a) and (b) show pictures of the two selected wind turbine types used to generate the simulated data for our experiments in the thesis.

To summarise, the kinds of energy generated from the renewable devices (such as solar and wind) are convenient and produce no carbon pollution, but they have the limitation of a smaller scale and the intermittent nature of energy production in comparison with fossil fuels. Therefore, users will need to frequently buy from and sell to the utilities on the grids. The spontaneous solution for solving this issue is to install an energy backup system, notably batteries, to store the excess energy

Figure 1.2: A typical roof with solar panels installed.
(Source:https://solargroup.co.nz/wp-content/uploads/2015/01/solar-panel-roof-nz.jpg
Accessed: 31 Dec. 2019.)

for requirements in deficit hours. Hence, this additional cost of the backup system will make the RE less profitable.

However, the utilisation of RE can bring forward the contribution of reducing the pollution of greenhouse gases. As a consequence, some governments may provide subsidies to the users to encourage people to install RE facilities. Alternatively, some co-operative models have been proposed for RE users to share their energy in order to support the users to use the RE efficiently and economically.



(**a**) Bergey (1 kW) wind turbine.
(Source: http://bergey.com/wp-content/gallery/excel-1/XL1Render2.JPG
Accessed: 31 Dec. 2019.)

(**b**) Eco (3 kW) wind turbine.
(Source: https://image.ec21.com/image/windsystem/oimg_GC04832269_CA04452851/
3kw-Wind-Turbine-Generator.jpg Accessed: 31 Dec. 2019.)

Figure 1.3: Two commercialised wind turbines used in our experiment.

1.4 Prosumers and Networked Energy Sharing

A prosumer is a person who consumes and produces a product (Toffler, 1980). Thus, in the application of smart grids, a consumer who also generates energy (notably from renewable sources) is regarded as a prosumer. Note that the power line of a local prosumer is still connected to the public network, which allows the prosumer to trade with a provider (namely a utility) while it has an excess or shortage. Generally, a RE prosumer is said to be a modelling agent in this study. For example, a prosumer can be a household with a wind turbine or solar panels installed.

As discussed in Section 1.3, a prosumer will often have an excess or a deficit of power so that it needs to trade with the utility. However, the buy-back rates (paid by the utility) are much lower than the purchase prices (paid by the prosumer) in some countries, including New Zealand (Philpott et al., 2019). Therefore, a prosumer will need to pay extra to the utility, even when the prosumer has, on average, generated sufficient power.

For instance, let us assume a PV prosumer has generated PV power (e.g. 20 kWh) in the daytime on a sunny day and sells the extra power (e.g. 10 kWh) to the utility. Outside the daylight hours, the prosumer has consumed power (e.g. 10 kWh) which is purchased from the utility. In this case, this prosumer has generated sufficient power to meet the requirement in one day. However, due to the price difference, this prosumer still has to pay an additional fee to the utility.

As mentioned, to avoid the prosumer selling power to the utility, one can install an auxiliary storage (notably, a battery backup system) to store the extra energy for when a shortage of hours arises. However, the storage is expensive and makes the investment less profitable.

Alternatively, another preference is for the prosumer to share the energy with others by forming a networked energy-sharing coalition. Therefore, suppose that while some prosumers in the coalition have surplus energy, they will share the energy with others who require energy. Accordingly, while these prosumers are short of energy, they can then be supported by other prosumers who have an excess.

Note that buying from and selling to the electric utility is less efficient and much more costly than sharing energy among coalition partners. Thus, it is a better option for a prosumer to join the co-operation of a coalition. Consequently, the prosumer can be connected in a network with others in order to form a coalition for sharing energy with others. Besides, the networked prosumers can still be connected to the public grid or else operate in island mode. (Island mode refers to a distributed power network that is applied in an isolated location where its operation is independent of any national or regional power distribution network (IEEE, 2011; Saleh et al., 2016).)

Hence, the problems of how to organise and operate the distributed agents of small scale prosumers, such as households with solar panels or wind turbines, to form decentralised power systems will be the primary consideration in this thesis.

In summary, the advantages for agents to form a networked energy-sharing coalition are:

1. Sharing energy among a networked collection of prosumers to accommodate their variable amounts of energy production and consumption can reduce the need for expensive storage and can also facilitate community co-operation.

2. Because of their localised and distributed nature, a networked collection of energy prosumers does not have a single point of failure in the case of power failure. Thus, the coalition can not only share energy during the daily routine, but also serve as an emergency support system while there is a blackout caused by a catastrophic event.

3. Owing to the sharing mechanism, forming a coalition is economical and efficient. Therefore, prosumer proliferation can support affordable and incremental increases of renewable energy.

1.5 The Optimisation of Agents' Coalitions

From the previous discussion, we argue that forming coalitions is one of the best ways for RE agents to work together. Also, according to the variable weather and consumption conditions, the demand and supply situation of agents may frequently change. Hence, the electricity dispatch arrangement will need to be periodically rescheduled (Dulău et al., 2016). Therefore, our goal is to design a model that facilitates efficient energy sharing among a distributed network of energy agents with recurrent demand and supply.

In our problem of forming coalitions, a coalition can be any non-empty subset of an alliance with n agents, and a coalition structure is a partition of that alliance. However, the total number of possible coalition structures is exponential to the number of agents n, and, more importantly, it is large, even relative to the number of coalitions (which is equal to 2^n-1) (Chalkiadakis et al., 2011). To demonstrate the challenge, let us consider a simple example. For instance, 15 possible coalition structures can be formed for an alliance of four agents. When the number of agents in an alliance is up to 10, the number of possible coalition structures will reach 115,975. Thus, this coalitional model is usually not considered scalable for a working alliance of agents. Therefore, an exhaustive search for evaluating every possible coalition structure in order to find the optimal solution is impractical and unfeasible, even for a small alliance such as a community with an hourly dispatch schedule. Alternatively, other optimal algorithms may provide more prompt and appropriate solutions to meet the requirements of scalability and immediateness. This issue will be covered rigorously and in more detail in Chapter 2.

In order to make our proposed coalition model effective and practical, the research objectives are listed below:

- Given the intermittent and mutable characteristics of agents, can a co-operative model be designed with specific mechanisms for an alliance of agents that can internally exchange energy?

- To overcome the challenge of the rapidly growing number of coalitions, can we propose an optimisation method to solve the design model within a local area efficiently (e.g. a community or town), where the result can remain at or nearly at an optimal level? Will this method fit the periodic electricity dispatch arrangement without losing much accuracy?

- Can we improve the scalability while retaining the accuracy of the proposed optimisation approach to solve a large-scale multi-region application?

1.6 Contributions

Much of the work reported in this thesis is based on our three published papers as listed below:

- Lee, S. H.-S., Deng, J. D., Peng, L., Purvis, M. K., & Purvis, M. (2017). Top-k merit weighting pbil for optimal coalition structure generation of smart grids. In international conference on neural information processing (pp. 171–181).
- Lee, S. H.-S., Deng, J. D., Purvis, M. K., Peng, L., & Purvis, M. (2018a). An improved PBIL algorithm for optimal coalition structure generation of smart grids. In workshop on data ming for energy modelling and optimisation, damemo), the 22nd Pacific-Asia conference on knowledge discovery and data mining (pp. 1–8).
- Lee, S. H.-S., Deng, J. D., Purvis, M. K., & Purvis, M. (2018b). Hierarchical Population-Based Learning for Optimal Large-Scale Coalition Structure Generation in Smart Grids. In T. Mitrovic, B. Xue, & X. Li (Eds.), AI 2018: Advances in artificial intelligence (pp. 16–28). Cham: Springer International Publishing.

The components of Chapters 1 to 3 (introduction, methodology and models) of the thesis are extensions of these papers. Furthermore, the approaches and experiments in Chapters 4 to 6 are extensions of each paper, accordingly.

In summary, this study proposes a model for prosumers to share energy internally. The solution not only gives consumers more incentive to become new RE prosumers, but also encourages current prosumers to enlarge facilities to generate more RE. This approach utilises the formation of coalitions, which is essential in Co-operative Game Theory.

Furthermore, this thesis solves the coalitional formation problem by using the Coalition Structure Generation (CSG) in Co-operative Game Theory. However, the number of coalition structures (CS) is exponential to the number of agents. Traditional deterministic search methods, such as Dynamic Programming (DP), can only solve the CSG problems with a limited number of agents because they demand computer resources, such as memory and the number of iterations proportional to the number of CS. Therefore, even though these approaches can guarantee finding the exact (optimal) solution, they are still impractical in the applications of a large number of agents. Alternatively, the Stochastic Optimisation (SO) approaches may not guarantee reaching the exact solution, but they may provide relative optimisation solutions in applications which have a large agent number and repeatedly require a solution.

Based on the aforementioned concept and motivation, this thesis contributes to the following aspects:

- Based on Co-operative Game Theory, this thesis builds a coalition model for an alliance of RE agents to share energy internally. This model could increase the benefit for agents in the alliance through the mechanism of energy sharing. This mechanism can operate on both smart grids and off-grid. In both of the operations, the agents within the alliance can obtain more benefits by sharing energy with other members. Besides, if an alliance, such as a community, is disconnected from the public grid because of a disaster, the energy sharing model can still operate in island mode.

- Since the exhaustive search and its improved methods are inefficient and not scalable for the application of our coalition model, we have proposed an improved stochastic algorithm, named Top-k Merit Weighting Population-Based Incremental Learning (PBIL-MW), to generate an optimal coalition model which maximises the benefit of the alliance. When comparing existing state-of-the-art methods with the proposed PBIL-MW algorithm, the latter not only optimises power-sharing within the alliance in our experiments, but also outperforms the methods by delivering faster running speeds and consuming fewer computer resources.

- Furthermore, we have also proposed a new scalable approach, namely, Hierarchical PBIL-MW (H-PBIL-MW), to solve a coalition formation with a large number of agents in an extended region. Our results have revealed the promising potential of applying the H-PBIL-MW algorithm to deal with co-operative coalitions across extensive areas where there are constraints such as landform restrictions in multi-areas which power-share.

- The proposed method for sharing renewable energy among the members (prosumers) in an alliance can be directly applied to smart grids by repeatedly forming coalitions. Consequently, prosumers can share renewable energy with other members and may make more profit in comparison with trading energy with the utilities.

1.7 Organisation of Thesis

The rest of this thesis is organised as follows.

In Chapter 2, we introduce the research background, relevant concepts, and theories; and their relevant literature is reviewed accordingly. Firstly, we briefly present and explain the definition of a Coalition Formation and Coalition Structure Generation (CSG) in Cooperative Game Theory. The detailed construction of CSG is then illustrated, and the methods for solving CSG problems are correspondingly discussed.

The relative methodology and concepts in this study are introduced and explained in Chapter 3. Concerning the research objectives of forming a co-operative alliance, we analyse the incentives and models for forming the coalition structures. Based on the analysis, the design of the coalition model has been set up, which covers several aspects, including coalition constraints, coalition evaluation, and scenarios for experiments in terms of different scales. In Section 3.4, the process of obtaining and analysing the consumption profiles and the generating power of RE is presented and demonstrated separately. Finally, the integration of both consumption and generation information to obtain the prosumer's data is explained.

After we have designed the coalition structure model for solving the co-operative multi-agent problem in Chapter 4, we begin with the leading experiments to demonstrate the feasibility of the proposed model with a small number of agents. To solve these experiments, we have utilised four approaches for comparison. Apart from using the direct search method and two popular stochastic algorithms mentioned in Chapter 2, we propose an improved stochastic approach using Population-Based Incremental Learning algorithm with top-k Merit Weighting, namely, the PBIL-MW, and a customised strategy for choosing the initial probability to solve the problem. Empirical results show that our new proposed algorithm gives competitive performance compared with some stochastic optimisation algorithms.

In Chapter 5, we demonstrate the superiority of our proposed stochastic algorithm, i.e. the PBIL-MW for 20 agents, and compare the outcome with the results we obtained by using a dynamic programming algorithm. Furthermore, to improve our proposed algorithm presented in Chapter 4, we propose a customised genotype encoding scheme as an improved approach to finding the optimal coalition structure of agents. Our empirical results show that the proposed approach gives competitive performance compared with the exact solutions obtained by dynamic programming.

In Chapter 6, we advance the problem to large multi-region scales, e.g. 80 agents in four urban areas. We know from Chapter 5 that our proposed algorithm, namely, the PBIL-MW, could efficiently obtain an approximately optimal solution. Furthermore, we propose a new approach, Hierarchical PBIL-MW with a termination scheme, namely, the Hi-PBIL-MW. In comparison with the PBIL-MW, the new algorithm achieves significant efficiency with only a small loss of accuracy. The Hi-PBIL-MW algorithm has provided an alternative solution when the number of agents is large, and the time restriction is an essential factor in some applications.

Finally, we conclude by summarising the contributions of this study and point to some possible further directions in the last chapter.

Chapter 2

Preliminary

This chapter will firstly discuss the existing co-operative methods for prosumers, including the fundamentals of coalition structure generation and the solutions applied in this thesis. At the same time, a relevant literature review will be presented. Firstly, the co-operative methods for prosumers (i.e. agents) will be introduced and discussed. Then, the optimisation problem considered in this thesis will be defined. In Section 2.3, we will introduce the traditional methods for solving optimisation problems.

2.1 Co-operative Methods for Prosumers

As presented in Chapter 1, the integration of RE with smart grids involves emerging decentralised power management systems which harmonise the requirements and availabilities of all generators, customers, grid operators, electricity market stakeholders, grid regulators, and so on (Siano, 2014; IEA, 2011). This increasing trend of implementation has brought about enormous applications (Gungor et al., 2013) and research (Stoustrup et al., 2019; Fadlullah et al., 2011) from all aspects, including social welfare (Wolsink, 2012), global economics (Derksen & Weber, 2017), political co-operation, management technologies (Derksen & Weber, 2017), environment protection, and power grid security (Cuellar, 2013).

To summarise the existing smart grids' research and applications, Yasir (2018) has conducted a comprehensive literature review and classifies all research into five domains: energy management, communication technology, system protection, power storage and energy subsystem. Among those domains, one of the research topics in energy management is focused on the optimisation of cost and profits. As will be discussed later in this section, our research concern is one of the essential subjects about the optimisation of benefit. Therefore, as mentioned in Section 1.4, the problems of how to organise and operate the distributed agents of small scale prosumers, such as households with solar panels or wind turbines, to form decentralised power systems will be the primary consideration in this thesis.

2.1.1 Coalition Formation

In game theory, a coalition is defined as a group (namely an alliance) of agents. A coalition is formed by binding an agreement, committing to work together in order to achieve a specific goal, such as increasing payoff or decreasing cost (Chalkiadakis et al., 2011). Hence, under the definition, a grand coalition, i.e. the whole group of agents, may be divided into the form of separated smaller coalitions to pursue an optimal goal of the group (Shoham & Leyton-Brown, 2008).

Bogomolnaia et al. (2019) summarised the Coalition Formation (CF) theory, which is composed of two aspects. One is about how to partition the agents, and the other involves the group's actions. The researchers also provided a significant number of articles which are related to CF theories and applications.

CF applies to almost every aspect in the world, such as political parties, electricity markets, organisations for environmental protection, and the league of sports teams (Bogomolnaia et al., 2019; Ray & Vohra, 2015).

Since the efficiency of wind turbines and solar PV panels keeps increasing, and their installation cost keeps dropping, the penetration of them has experienced rapid growth in recent decades (Sawin et al., 2014). However, due to the intermittency of RE, prosumers may face a frequent shortage. As mentioned in Chapter 1, two conventional methods are either to trade with the power utility or to enlarge the capacity of their backup storage (e.g. battery). However, both approaches are costly. An alternative solution may be forming a coalition of prosumers to share energy regularly and internally.

To improve co-operation among RE prosumers, some researchers utilise CF to provide different strategies to share the RE among prosumers (Ellabban et al., 2014; Rennkamp et al., 2017). For example, Yasir et al. (2013) proposed agent-based community co-ordination to reduce the power deficit of communities by forming some co-operative groups, namely coalitions, based on the social aspect of sustainability by reducing the overall discomfort levels. Wolsink (2012) addressed the social construction of smart grids. The contribution of a smart grid among prosumers is optimised to collaborate and share RE in a co-operative micro-grid with mutual delivery. By using the Markov decision process, Mondal and Misra (2013) optimised energy distribution by dynamically changing the size of the coalition in which the number of customers in a single micro-grid varies. However, in Mondal and Misra (2013), the power suppliers and consumers are separated, whereas in our study, there are prosumers coalitions.

Apart from the above research, Saad et al. (2011) applied coalitional game theory to study the co-operative strategies in distributed smart grids in order to minimise the total power loss. Chakraborty et al. (2014) extended the study of Saad et al. and proposed a more scalable hierarchical coalition formation mechanism which considered both power loss and price mechanisms.

However, these two articles focused only on finding coalitions to minimise the power loss caused by power transportation in which the power loss is transferable. Here, 'transferable' means that the payoff or utility can be freely transferred to any agent in the coalition. Approaches to distribute the payoff equitably to agents are called solution concepts such as the core, the Shapley value, and the nucleolus (Peters, 2015). Thus, these methods belong to a coalitional game with transferable utility (Shoham & Leyton-Brown, 2008). Furthermore, as mentioned by Shoham and Leyton-Brown (2008), the coalitions can be addressed as a single grand coalition. To solve the grand coalition, the calculation of the Shapley value is efficient for dividing the payoff of the grand coalition members among the agents.

2.1.2 Coalitional Game in Partition Form

Different from the transferable coalitional game, this thesis focuses on a non-transferable problem, namely, the coalition structure generation (CSG), which is based on the co-operative game theory (Chalkiadakis et al., 2011), aka "Coalitional Game in Partition Form" (Shoham & Leyton-Brown, 2008). The CSG works as a multi-agent system, whose agents join together to form a specified partition, termed the coalition structure (CS), for achieving an optimal solution of the overall benefit for all the participating agents (Chalkiadakis et al., 2011).

In the multi-agent game, each agent is an autonomous, interactive and intelligent software entity. In this thesis, agents act on behalf of the prosumers. Thus, agents can bring information and make decisions, just like the prosumers themselves. Furthermore, the intelligence of the agents includes functional mechanisms, procedural strategies and other roles that work in the program (Wooldridge, 2009; Chalkiadakis et al., 2011; Shoham & Leyton-Brown, 2008).

The key to finding the optimal solution is to optimise the CS. The goal for the coalitions of a given alliance can be defined, such as maximising the overall profit or minimising the power loss. However, a grand coalition is generally inapplicable in a multi-agent system. For example, RE sources, such as wind and solar power, have intermittent energy outputs, so that they can be used to form temporary coalitions in order to share their energy. For agents that have excess energy (i.e. they are producing more energy than their local consumption) can provide the energy to other members in their coalition who are facing an energy deficit.

For a broad alliance of agents producing RE, it then becomes an optimisation problem as to what coalition arrangement will result in the optimal distribution of available power. Such a co-operative mechanism for sharing the RE among agents is also suitable in the case where agents behave unselfishly and volitionally bind to an agreement to accomplish coalitions as a single system (Chalkiadakis et al., 2011). In the following section, the CSG problem will be defined in detail.

2.2 Coalition Structure Generation

The CSG is a non-deterministic polynomial hard (NP-hard) problem (Sanjeev Arora, 2009), which means that there are no polynomial-time algorithms for solving the problem. The larger the number of agents, the more the potential search domain will increase exponentially. In the following subsections, the definition of CSG and its estimation of the problem scales are presented, including the number of exclusive subsets, partitions, and total coalitions.

2.2.1 Problem Definition

Suppose there are n agents, denoted by $a_i,\ i \in n$. The set of all agents is denoted by $\mathcal{S}$, such that $\mathcal{S} = \{a_1, ..., a_n\}$. The term "coalition", denoted by $C_{(k,i)}$, refers to any subset, indexed by i, of which $\mathcal{S}$ contains k agents, where $C_{(k,i)} \subseteq \mathcal{S},\ C_{(k,i)} \neq \emptyset$. In addition to $\mathcal{S}$ and $C_{(k,i)}$, a coalition structure is a partition of $\mathcal{S}$ and can be defined (Chalkiadakis et al., 2011; Rahwan et al., 2015) by:

Definition 1. For any set $\mathcal{S}$, a **coalition structure**, denoted by CS, is a partition of $\mathcal{S}$, and coalitions are subsets of CS, such that $\text{CS} = \{C_{(k,i)}, ..., C_{(k',j)}\}$, $\cup\text{CS} = \mathcal{S}$, and $C_{(k,i)} \cap C_{(k',j)} = \emptyset$ for any $C_{(k,i)}, C_{(k',j)} \subset \mathcal{S}, C_{(k,i)} \neq C_{(k',j)}$.

In this thesis, each coalition is evaluated by the total payoff that the coalition can get, denoted by $v(C)$. This game is a characteristic function game of CSG proposed by von Neumann (1944). Therefore, the value (or worth) of a CS is the sum of the values of the coalitions $C_{(k,i)}$ in CS and can be defined (Chalkiadakis et al., 2011; Rahwan et al., 2015) by:

Definition 2. For any coalition, $C_{(k,i)} \subseteq S$, the **value** of a CS is denoted by $v(\text{CS})$ and is given by:

$$v(\text{CS}) = \sum_{C_{(k,i)} \in \text{CS}} v(C_{(k,i)}). \tag{2.1}$$

Hence, the optimisation objective of the CSG problem (Chalkiadakis et al., 2011; Rahwan et al., 2015) can be defined by:

Definition 3. The **CSG problem** is the optimisation problem of searching for an optimal CS, denoted by CS*, over $\mathcal{S}$ whose value is optimal. Mathematically, a coalition structure CS* is said to be globally optimal if CS* gives the maximal overall characteristic value:

$$\text{CS}^* = \arg \max_{\text{CS} \in \mathcal{F}(\text{CS})} v(\text{CS}), \tag{2.2}$$

where $\mathcal{F}(\mathrm{CS})$ denotes the whole collection of all possible CSs into which the set $\mathcal{S}$ of n agents can be formed.

Thus, the size of $\mathcal{F}(\mathrm{CS})$ for $\mathcal{S}$ is denoted by $|\mathcal{F}(\mathrm{CS})|$.

In this thesis, we focus on designing algorithms for solving the above problem and endeavour to efficiently search for the optimal or near-optimal solutions.

2.2.2 Number of Exclusive Subsets

Given a set $\mathcal{S}$ of n agents, let $C_{(k)}$ denote any possible subset which contains k agent(s) exactly and $|C_{(k)}|$ denote the total number of exclusive $C_{(k)}$, respectively. From the binomial theorem, we know that:

$$|C_{(k)}| = \tbinom{n}{k} = \frac{n!}{(n-k)!\,k!}\,, \tag{2.3}$$

which is a binomial coefficient. Consequently, the total number of all possible subsets can be calculated by:

$$\sum_{k=1}^{n} |C_{(k)}| = 2^n - 1, \tag{2.4}$$

which is the sum of binomial coefficients, excluding the empty set.

Let us consider the set of 4 agents, $\mathcal{S} = \{a_1, a_2, a_3, a_4\}$. There are:

- 4 subsets, $|C_{(1)}| = \tbinom{4}{1} = 4$, containing 1 agent: $\{a_1\},\ \{a_2\},\ \{a_3\}$ and $\{a_4\}$.
- 6 subsets, $|C_{(2)}| = \tbinom{4}{2} = 6$, containing 2 agents: $\{a_1, a_2\}, \{a_1, a_3\}, \{a_1, a_4\}$, $\{a_2, a_3\}, \{a_2, a_4\}$ and $\{a_3, a_4\}$.
- 4 coalitions, $|C_{(3)}| = \tbinom{4}{3} = 4$, containing 3 agents: $\{a_1, a_2, a_3\},\ \{a_1, a_2, a_4\}$ and $\{a_2, a_3, a_4\}$.
- 1 grand coalition, $|C_{(4)}| = \tbinom{4}{4} = 1$, containing all 4 agents: $\{a_1, a_2, a_3, a_4\}$.

Therefore, the total number of possible coalitions equals $2^4 - 1 = 15$.

2.2.3 Number of Partitions

Given a set $\mathcal{S}$ of n agents, let $\mathrm{CS}_{(m)}$ and $|\mathrm{CS}_{(m)}|$ denote any possible partitions of $\mathcal{S}$ containing precisely m coalitions and their total number, accordingly. From the

Combinatorics Theorem, we know that $|\mathrm{CS}_{(m)}|$ is a Stirling number of the second kind (Graham et al., 1989), and is given by:

$$|\mathrm{CS}_{(m)}| = \left\{ {n \atop m} \right\} = \frac{1}{m!} \sum_{j=0}^{m} (-1)^{m-j} \binom{m}{j} j^n, 1 \leq m \leq n. \tag{2.5}$$

Therefore, a set of 4 agents has

- $\left\{ {4 \atop 1} \right\} = \frac{1}{1!} \sum_{j=0}^{1} (-1)^{1-j} \binom{1}{j} j^4 = 1$ $\mathrm{CS}_{(1)}$ for the grand coalition, i.e. $\{\{a_1, a_2, a_3, a_4\}\}$.
- $\left\{ {4 \atop 2} \right\} = \frac{1}{2!} \sum_{j=0}^{2} (-1)^{2-j} \binom{2}{j} j^4 = \frac{1}{2!}[(-1)^2 \binom{2}{0} 0^4 + (-1)^1 \binom{2}{1} 1^4 + (-1)^0 \binom{2}{2} 2^4] = \frac{0-2+2^4}{2} = 7$ $\mathrm{CS}_{(2)}$ containing 2 subsets, i.e. $\{\{a_1, a_2, a_3\}, \{a_4\}\}, \{\{a_1, a_2, a_4\}, \{a_3\}\}, \{\{a_1, a_3, a_4\}, \{a_2\}\}, \{\{a_2, a_3, a_4\}, \{a_1\}\}, \{\{a_1, a_2\}, \{a_3, a_4\}\}, \{\{a_1, a_3\}, \{a_2, a_4\}\}, \{\{a_1, a_4\}, \{a_2, a_3\}\}$.
- $\left\{ {4 \atop 3} \right\} = \frac{1}{3!} \sum_{j=0}^{3} (-1)^{3-j} \binom{3}{j} j^4 = 6$ $\mathrm{CS}_{(3)}$ containing 3 subsets, i.e. $\{\{a_1, a_2\}, \{a_3\}, \{a_4\}\}, \{\{a_1, a_3\}, \{a_2\}, \{a_4\}\}, \{\{a_1, a_4\}, \{a_2\}, \{a_3\}\}, \{\{a_2, a_3\}, \{a_1\}, \{a_4\}\}, \{\{a_2, a_4\}, \{a_1\}, \{a_3\}\}, \{\{a_3, a_4\}, \{a_1\}, \{a_2\}\}$.
- $\left\{ {4 \atop 4} \right\} = \frac{1}{4!} \sum_{j=0}^{4} (-1)^{4-j} \binom{4}{j} j^4 = 1$ $\mathrm{CS}_{(4)}$ containing 4 subsets, i.e. $\{\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}\}$.

Moreover, from Eq. 2.5 we can compute the number of the collection $\mathcal{F}(\mathrm{CS})$ of all possible CSs for a given set $\mathcal{S}$ of n agents, denoted by $|\mathcal{F}(\mathrm{CS})|$, which is $B(n)$ and can be calculated by:

$$|\mathcal{F}(\mathrm{CS})| = \sum_{m=1}^{n} |\mathrm{CS}_{(m)}| = B(n) = \sum_{m=1}^{n} \left\{ {n \atop m} \right\}. \tag{2.6}$$

The $B(n)$ in Eq. 2.6 is known as a Bell's number, which is NP-hard in computational complexity. Chalkiadakis et al. (2011) have proven that $B(n)$ satisfies:

$$(n/4)^{n/2} \leq B(n) < n^n. \tag{2.7}$$

Note that the bounds of Eq. 2.7 grow much faster than the single exponential number. For example, $B(3) = 5$, $B(12) = 4.213597 \times 10^6$, $B(20) = 5.172416 \times 10^{13}$ and $B(80) \approx 9.913 \times 10^{86}$.

2.2.4 Total Number of Coalitions

The total number $T_{(k)}$ of any k-agent subsets in the CS can be given by:

$$T_{(k)} = |C_{(k)}|B(n-k) = \binom{n}{k}B(n-k), \ where\ 1 \leq k \leq n. \tag{2.8}$$

Therefore, the total number of all subsets in the CS, denoted by $Ts_{(n)}$, is given by:

$$Ts(n) = \sum_{k=1}^{n} T_{(k)} = \sum_{k=1}^{n} \binom{n}{k}B(n-k). \tag{2.9}$$

In the 4 agents example, there are 6 $(=\binom{4}{2})$ subsets that contain 2 agents, i.e. $\{a_1,a_2\},\{a_1,a_3\},\{a_1,a_4\},\{a_2,a_3\},\{a_2,a_4\}$ and $\{a_3,a_4\}$. Moreover, there are two CSs, i.e. $\{\{a_1,a_2\},\{a_3,a_4\}\},\{\{a_1,a_2\},\{a_3\},\{a_4\}\}$, and both have the subset $\{a_1,a_2\}$ inside; and there are another two CSs, i.e. $\{\{a_1,a_3\},\{a_2,a_4\}\},\{\{a_1,a_3\},\{a_2\},\{a_4\}\}$ and both have the subset $\{a_1,a_3\}$ inside, and so forth. Therefore, from Eq. 2.8, we get $T_{(2)} = \binom{4}{2}B(4-2) = 12$ which is the total number of subsets with just 2 agents inside the 4 agents CS.

The calculation of $T_{(k)}$ and $Ts(n)$, for the 4-agent CS case is tabulated in Table 2.1. Furthermore, the numbers of the CS, i.e. $B(n)$ and $Ts(n)$, where n is from 1 to 80, are shown in Table 2.2, and a plot is shown in Figure 2.1.

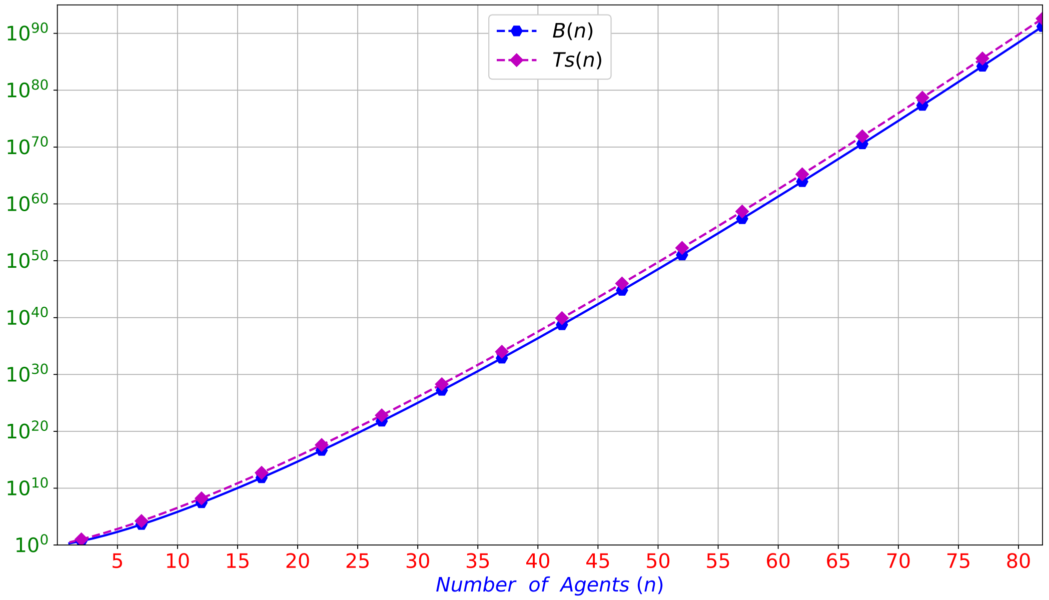


Figure 2.1: Number of CSs and Ts in comparison with the number of agents.

Table 2.1: Calculating $Ts_{(n)}$ by summing $T_{(k)}$ of a CS with $n = 4$ agents.

4-Agent set				
k	1	2	3	4
$\|C_{(k)}\|$	4	6	4	1
$B(4-k)$	5	2	1	1
$T_{(k)}$	20	12	4	1
$Ts(n) = T_{(1)} + T_{(2)} + T_{(3)} + T_{(4)} = 20 + 12 + 4 + 1 = 37$				

Table 2.2: No. of CSs and $Ts_{(n)}$ in terms of n-Agent.

Agent-size	1	2	3	4	10	20	40	80
$B(n)$	1	2	5	15	115,975	$\approx 5.172 \times 10^{13}$	$\approx 1.575 \times 10^{35}$	$\approx 9.913 \times 10^{86}$
$Ts(n)$	1	3	10	37	562,595	$\approx 4.231 \times 10^{14}$	$\approx 2.194 \times 10^{36}$	$\approx 2.377 \times 10^{88}$

Therefore, according to Eq. 2.9, the total number of inclusive subsets $C_{(k,i)}$ in $\mathcal{F}$(CS) with a set of 4 agents is 37. Figure 2.2 illustrates the classes and relations of the CSs with 4 agents. The number shown on the right of the Figure represents the partition size of the CS aligned with the same row. There are three characteristic numbers related to the CS of a set $\mathcal{S}$ of agents, namely, the "number of exclusive subsets $C_{(k)}$ in $\mathcal{S}$", the "number of partitions CS in $\mathcal{S}$" and the "number of total coalitions C(k,i) in $\mathcal{S}$".

To sum up, it can be observed that it is impossible to use traditional algorithms such as a direct search or dynamic programming when the problem size becomes large. However, there may be hundreds or even thousands of prosumers in an application, so new methods need to be researched to tackle the problem.

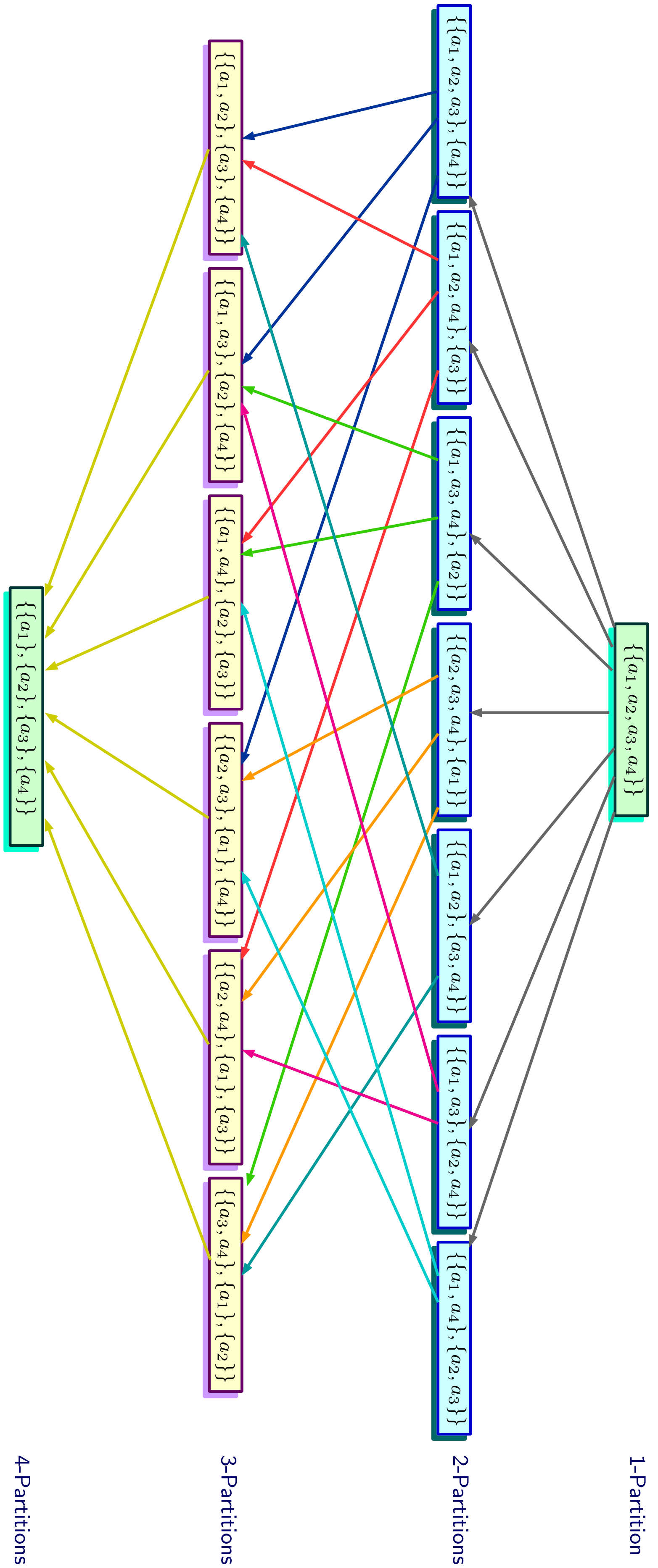

$\{\{a_1,a_2,a_3,a_4\}\}$
$\{\{a_1,a_2,a_3\},\{a_4\}\}$
$\{\{a_1,a_2,a_4\},\{a_3\}\}$
$\{\{a_1,a_3,a_4\},\{a_2\}\}$
$\{\{a_2,a_3,a_4\},\{a_1\}\}$
$\{\{a_1,a_2\},\{a_3,a_4\}\}$
$\{\{a_1,a_3\},\{a_2,a_4\}\}$
$\{\{a_1,a_4\},\{a_2,a_3\}\}$
$\{\{a_1,a_2\},\{a_3\},\{a_4\}\}$
$\{\{a_1,a_3\},\{a_2\},\{a_4\}\}$
$\{\{a_1,a_4\},\{a_2\},\{a_3\}\}$
$\{\{a_2,a_3\},\{a_1\},\{a_4\}\}$
$\{\{a_2,a_4\},\{a_1\},\{a_3\}\}$
$\{\{a_3,a_4\},\{a_1\},\{a_2\}\}$
$\{\{a_1\},\{a_2\},\{a_3\},\{a_4\}\}$
1-Partition
2-Partitions
3-Partitions
4-Partitions

Figure 2.2: The coalition structure graph for $S_4 = \{a_1, a_2, a_3, a_4\}$.

2.3 Related Work to Solve the CSG Problem

Since CSG can be considered as an optimisation problem, some state-of-the-art algorithms for solving such kind of problems will be introduced. Firstly, two classical methods, i.e. exhaustive search (ES) and dynamic programming (DP), are introduced. These two algorithms provide the optimal solution for the problem, but they become inapplicable when the size of the problem enlarges.

Next, the stochastic optimisation (SO) methods are described. SO methods are a kind of stochastic algorithm that are based on a probabilistic search or nature-inspired search behaviours. They make fewer or no assumptions about a problem domain and can search in the vast space of candidate solutions (Beheshti & Shamsuddin, 2013). SO algorithms are proposed for large-scale searches due to their global exploration and local exploitation ability. SO algorithms are recommended when enumerative techniques are difficult or even unable to find an exact solution.

Although SO algorithms do not guarantee finding an exact solution, they may provide promising solutions with high efficiency and scalability (Rahwan et al., 2015). They are alternative solutions to the CSG problem. The stochastic algorithms discussed in this thesis include the genetic algorithm (GA), particle swarm optimisation (PSO), and population-based incremental learning (PBIL).

2.3.1 Deterministic Method

Exhaustive Search (ES)

From Section 2.2, we know that the value of every CS is a discrete point in the searching space. Thus, the direct way to find the optimal solution is to check the value of every single point.

ES or the so-called brute force method is a direct search method, which searches the whole search space for finding the optimal solution. Although it can obtain the optimal solution, aka “exact solution” theoretically, it may take too long to find it when the solution space is too large (Chalkiadakis et al., 2011).

Dynamic Programming (DP)

Bellman and Dreyfus (1962) propose the original basic concept of using the DP method. The DP method refers to re-constructing a complex problem by separating it into some elementary sub-problems in a recursive form. Hence, by recursively searching for the optimal solutions to the sub-problems, a complex problem can be solved exhaustively and optimally. In CSG problems, the first DP method was proposed by Yeh (1986). It has an advantage over ES by employing re-use in problem-solving. The approach of DP can be explained by using the lemma (Rothkopf et al., 1998; Chalkiadakis et al., 2011) below.

Lemma 2.1 Let $v^*(C)$ denote the optimal value of a given coalition. For any $C \subseteq S_n$ we have

$$v^*(C) = \max\Big\{\max\{v^*(C') + v^*(C'')\}, v(C)\Big\},$$

where $C' \cup C'' = C,\ C' \cap C'' = \emptyset,\ C'$ and $C'' \neq \emptyset$.

When searching for the optimal coalition C^*, the partitions of C are considered as two disjoint coalitions, i.e. the value of a grand coalition $v(C)$ itself or the sum of its 2-partition sub-coalition values $v^*(C^{'})$ and $v^*(C^{''})$.

In return to Lemma 2.1, a flowchart is shown in Figure 2.3 to demonstrate the process of DP.

The standard operation steps and an example for explaining the steps of DP are given below.

- For all sets of singleton $\{a_i | i \in n\}$, we compute the values of $v(\{a_i\})$ directly.
- For all coalitions of size 2, $\{a_i, a_j | i \neq j, i, j \in n\}$, we compute the values of $v(\{a_i, a_j\})$ and compare them to $v(\{a_i\}, \{a_j\})$. Thus, the optimal $v^*(\{a_i, a_j\})$ could be obtained.
- For the coalitions of size $= k$, $3 \leq k \leq n$, the $v^*(C_{(k)})$ could be obtained by iterating over all partitions of size $k-1$ based on Lemma 2.1.

Besides, Chalkiadakis et al. (2011) have proven that "its running time of DP is bounded by $O(\sum\limits_{k=1}^{n} \binom{n}{k} 2^k) = O(3^n)$. For each $k = 1, ..., n$, there are $\binom{n}{k}$ coalitions of size k, and for each coalition of size k, there are $2^{k-1} - 1$ ways to split it into two non-empty subcoalitions. Observe that $3^n < (n/4)^{n/2}$ for large enough values of n—i.e., this algorithm avoids searching all coalition structures. Moreover, its running time is polynomial in the number of coalitions 2^n

By using the 4 agents example, the fitness values of each subset of $C = \{a1, a2, a3, a4\}$ are listed in Table 2.3, accordingly. Here, we adopt the fitness values gathered in Rahwan et al. (2015)(p.145). According to Lemma 2.1, we then recursively calculate the $v^*(c)$ and the optimal c^* for the subsets with 2 and 3 agents, which are tabulated in Table 2.4.

By using Lemma 2.1, we can recursively calculate the $v^*(c)$ and optimal c^* for the subsets of 2 and 3 agents, which are listed in Table 2.4.

Table 2.3: Subset of a 4 agents coalition and their fitness values

subset c	Fitness $v(c)$
$\{a_1\}$	$v(\{a_1\}) = 30$
$\{a_2\}$	$v(\{a_2\}) = 40$
$\{a_3\}$	$v(\{a_3\}) = 25$
$\{a_4\}$	$v(\{a_4\}) = 45$
$\{a_1, a_2\}$	$v(\{a_1, a_2\}) = 50$
$\{a_1, a_3\}$	$v(\{a_1, a_3\}) = 60$
$\{a_1, a_4\}$	$v(\{a_1, a_4\}) = 80$
$\{a_2, a_3\}$	$v(\{a_2, a_3\}) = 55$
$\{a_2, a_4\}$	$v(\{a_2, a_4\}) = 70$
$\{a_3, a_4\}$	$v(\{a_3, a_4\}) = 80$
$\{a_1, a_2, a_3\}$	$v(\{a_1, a_2, a_3\}) = 90$
$\{a_1, a_2, a_4\}$	$v(\{a_1, a_2, a_4\}) = 120$
$\{a_1, a_3, a_4\}$	$v(\{a_1, a_3, a_4\}) = 100$
$\{a_2, a_3, a_4\}$	$v(\{a_2, a_3, a_4\}) = 115$
$\{a_1, a_2, a_3, a_4\}$	$v(\{a_1, a_2, a_3, a_4\}) = 140$

.

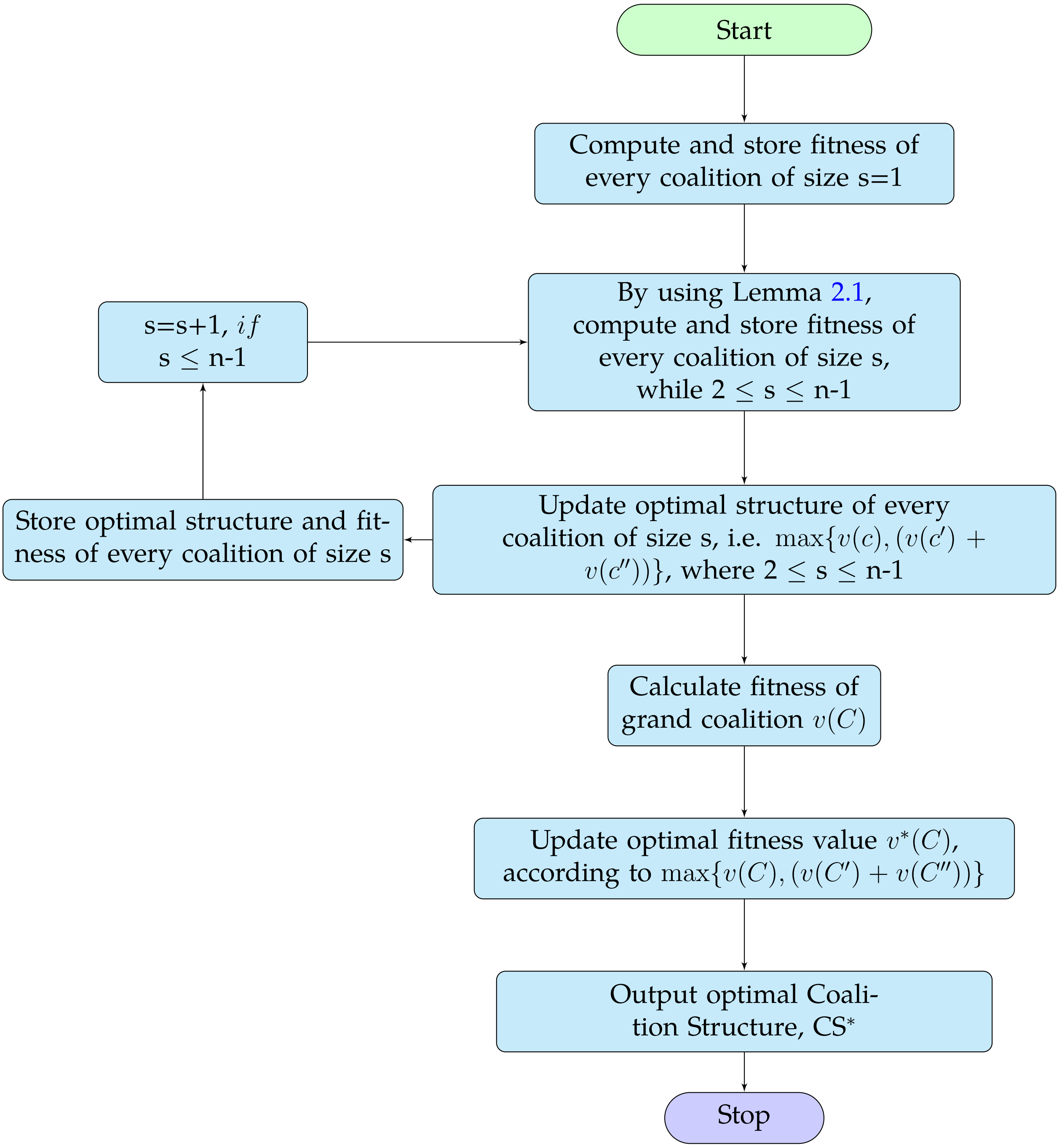

Start
Compute and store fitness of every coalition of size s=1
By using Lemma 2.1, compute and store fitness of every coalition of size s, while 2 $\leq$ s $\leq$ n-1
s=s+1, if s $\leq$ n-1
Update optimal structure of every coalition of size s, i.e. $\max\{v(c),(v(c')+v(c''))\}$, where 2 $\leq$ s $\leq$ n-1
Store optimal structure and fitness of every coalition of size s
Calculate fitness of grand coalition $v(C)$
Update optimal fitness value $v^*(C)$, according to $\max\{v(C),(v(C')+v(C''))\}$
Output optimal Coalition Structure, CS*
Stop

Figure 2.3: Flowchart of a Dynamic Programming Search

Table 2.4: Best fitness and optimal coalition according to coalitions in Table 2.3

Best Fitness v^*	Optimal c^*
$v^*(\{a_1,a_2\}) = \max\{\{v(\{a_1\}) + v(\{a_2\})\}, v(\{a_1,a_2\})\}$ $= \max\{(30+40), 50\} = 70$	$\{a_1\},\{a_2\}$
$v^*(\{a_1,a_3\}) = \max\{\{v(\{a_1\}) + v(\{a_3\})\}, v(\{a_1,a_3\})\}$ $= \max\{(30+25), 60\} = 60$	$\{a_1,a_3\}$
$v^*(\{a_1,a_4\}) = \max\{\{v(\{a_1\}) + v(\{a_4\})\}, v(\{a_1,a_4\})\}$ $= \max\{(30+45), 80\} = 80$	$\{a_1,a_4\}$
$v^*(\{a_2,a_3\}) = \max\{\{v(\{a_2\}) + v(\{a_3\})\}, v(\{a_2,a_3\})\}$ $= \max\{(40+25), 55\} = 65$	$\{a_2\},\{a_3\}$
$v^*(\{a_2,a_4\}) = \max\{\{v(\{a_2\}) + v(\{a_4\})\}, v(\{a_2,a_4\})\}$ $= \max\{(40+45), 70\} = 85$	$\{a_2\},\{a_4\}$
$v^*(\{a_3,a_4\}) = \max\{\{v(\{a_3\}) + v(\{a_4\})\}, v(\{a_3,a_4\})\}$ $= \max\{(25+45), 80\} = 80$	$\{a_3,a_4\}$
$v^*(\{a_1,a_2,a_3\}) = \max\Big\{\max\Big\{\{v^*(\{a_1,a_2\}) + v(a_3)\},$ $\{v^*(\{a_1,a_3\}) + v(a_2)\}, \{v^*(\{a_2,a_3\}) + v(a_1)\}\Big\}, v(\{a_1,a_2,a_3\})\Big\}$ $= \max\{\max\{(70+25), (60+40), (65+30)\}, 90\} = 100$	$\{a_1,a_3\},\{a_2\}$
$v^*(\{a_1,a_2,a_4\}) = \max\Big\{\max\Big\{\{v^*(\{a_1,a_2\}) + v(a_4)\},$ $\{v^*(\{a_1,a_4\}) + v(a_2)\}, \{v^*(\{a_2,a_4\}) + v(a_1)\}\Big\}, v(\{a_1,a_2,a_4\})\Big\}$ $= \max\{\max\{(70+45), (80+40), (85+30)\}, 120\} = 120$	$\{a_1,a_2,a_4\}$
$v^*(\{a_1,a_3,a_4\}) = \max\Big\{\max\Big\{\{v^*(\{a_1,a_3\}) + v(a_4)\},$ $\{v^*(\{a_1,a_4\}) + v(a_3)\}, \{v^*(\{a_3,a_4\}) + v(a_1)\}\Big\}, v(\{a_1,a_3,a_4\})\Big\}$ $= \max\{\max\{(60+30), (80+25), (80+30)\}, 100\} = 110$	$\{a_1\},\{a_3,a_4\}$
$v^*(\{a_2,a_3,a_4\}) = \max\Big\{\max\Big\{\{v^*(\{a_2,a_3\}) + v(a_4)\},$ $\{v^*(\{a_2,a_4\}) + v(a_3)\}, \{v^*(\{a_3,a_4\}) + v(a_2)\}\Big\}, v(\{a_2,a_3,a_4\})\Big\}$ $= \max\{\max\{(65+45), (85+25), (80+40)\}, 115\} = 120$	$\{a_2\},\{a_3,a_4\}$

Since there are 7 partitions, i.e. CS, in a 4 agents set that have 2 subsets, we get

$$\begin{aligned}
v^*\{a_1,a_2,a_3,a_4\} =& \max\Big\{ \max\Big\{\{v^*(\{a_1,a_2,a_3\})+v(a_4)\},\{v^*(\{a_1,a_2,a_4\})+v(a_3)\},\\
&\{v^*(\{a_1,a_3,a_4\})+v(a_2)\},\{v^*(\{a_2,a_3,a_4\})+v(a_1)\},\\
&\{v^*(\{a_1,a_2\})+v^*(\{a_3,a_4\})\},\{v^*(\{a_1,a_3\})+v^*(\{a_2,a_4\})\},\\
&\{v^*(\{a_1,a_4\})+v^*(\{a_2,a_3\})\}\Big\}, v(\{a_1,a_2,a_3,a_4\})\Big\}\\
=& \max\Big\{ \max\{(100+45),(120+25),(110+40),(120+30),(70+80),\\
&(60+85),(80+55)\},140\Big\}\\
=&150
\end{aligned}$$

Finally, we obtain the maximal fitness value $v^*(C) = 150$. From the above C^* list, we can deduce that $\text{CS}^* = \{\{a_1\},\{a_2\},\{a_3,a_4\}\}$.

Even though the running time $O(3^n)$ of DP seems faster than the $O(n^n)$ of ES (Chalkiadakis et al., 2011), DP still has an exponential number of CSs, i.e. 2^n-1, which is too slow in the application of CSG with a large number of agents. What is worse, the program still needs to store the current maximal value and its CS, but the memory requirement of DP is similar to its running time, about 2^n-1 of memory, for storing the values and their CSs. If $n=40$, then $2^n-1=2^{40}-1\approx 1.01\times 10^{12}$. Hence, it will exhaust all the memory of a computer. Therefore, the DP algorithm is quite slow in practice, though the computation of DP is less than ES.

In the applications of the CSG approach, one essential application domain for CSG is disaster management (Chalkiadakis et al., 2011). Researchers such as Kitano (2000) developed a series of technologies based on DP that can rescue people in the event of large scale disasters. Pechoucek et al. (2002) proposed a knowledge-based approach to coalition formation by using DP algorithms to plan humanitarian and peace-keeping missions. Note that the number of agents is 20 in Pechoucek et al.'s (2002) study, which matched our maximum number of agents for the DP algorithm, as mentioned in Chapter 5.

Many variants of research, based on the hybrid of DP and other search techniques have been proposed. They are dedicated to improve the efficiency of solving a CSG problem (Rahwan et al., 2015). For instance, Sandholm et al. (1999) presented a partial search algorithm which guarantees the solution to be within a bound from the optimum. Changder et al. (2016) used heuristics to select the optimal values from sub-problems and chose the remaining unassigned agents from other sub-problems. Michalak et al. (2016) combined DP with a tree-search algorithm to reduce the redundant process in DP and claimed that the approach was the fastest exact algorithm for complete set partitioning with $O(3^n)$. In addition to the above studies, the DP procedures have also been improved and continued by many researchers, including Dang and Jennings (2004), Service and Adams (2010), and Banerjee and Kraemer (2010). However, these DP variant algorithms still consume the running time exponentially and restrict their application to a limited number of agents.

Further to the above approaches, we have shown in Section 5.5 that a reasonable maximum of 20 agents can be used to apply DP in our experiments. However, we have proposed a hierarchical approach in Subsection 6.4.2, which divides 80 agents into four sub-areas (each has 20 agents) in order to obtain near optimisation.

2.3.2 Stochastic Optimisation (SO) methods

Genetic Algorithm (GA)

GA was inspired natural evolutionary processes with computer operation by using recombination operators such as mating, crossover, mutation, elitism, and tournaments to generate the new offspring (Holland, 1975).

Nowadays, GA has become one of the most popular SO algorithms with plentiful applications in the literature. There are many applications of GA in smart grids. For example, both Logenthiran et al. (2012), and Bharathi et al. (2017) used the demand side management technique in smart grids to minimise power utilisation during the electricity rush hour by effectively distributing the power available during the off-peak hour. Arabali et al. (2013) proposed a new strategy to meet the controllable heating, ventilation, and air conditioning load with a hybrid-renewable generation and energy storage system.

For the application of CSG, Sen and Dutta (2000) used an order-based GA as a stochastic search process to identify the optimal coalition structure in randomly generated optimal CS search problems. A genotype for a particular problem is represented as a fixed length of binary code. Hence, there are two different binary codes (Sen & Dutta, 2000). Two different binary codes with the same length are interpreted as two individuals with specified characteristics. Every individual in the population is evaluated by a pre-defined fitness function.

The fitness typically corresponds to the value of the objective function in the optimisation problem, which is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the knapsack problem, one wants to maximise the total value of objects that can be put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of values of all the objects in the knapsack if the representation is valid, or thr value is otherwise 0.

In some problems, it is hard or even impossible to define the fitness expression. In these cases, a simulation may be used to determine the fitness function value of a phenotype (e.g. computational fluid dynamic is used to determine the air resistance of a vehicle whose shape is encoded as the phenotype), or even interactive genetic algorithms are used.

The regular procedures of a typical GA are described below:

I. Initialisation: A GA process starts with the generation of an initial set of n probability vectors, denoted by $X_{(ini)}$, which is an array of random variables of size n rows, *aka* vectors, and length m columns. The value $x_{i,j}$ of each element is bounded by $[0 \leq x_{i,j} \leq 1]$, such that $X_i = [x_{i,1}, ...x_{i,m} | 1 \leq i \leq n, 1 \leq j \leq m]$ (Koza, 1992). Therefore, a vector of threshold p is set for filtering the elements of each row. While $x_{i,j} \leq p$, the mapped binary code will be 1, otherwise it will be 0. Consequently, an initial population, denoted by $S_{(ini)}$, of binary sets with size n rows and length m columns has been born. Accordingly, every sample s_i of the binary code with length m is represented as its gene of a possible individual. The fitness value v_i, in terms of every individual, is computed by the fitness function.

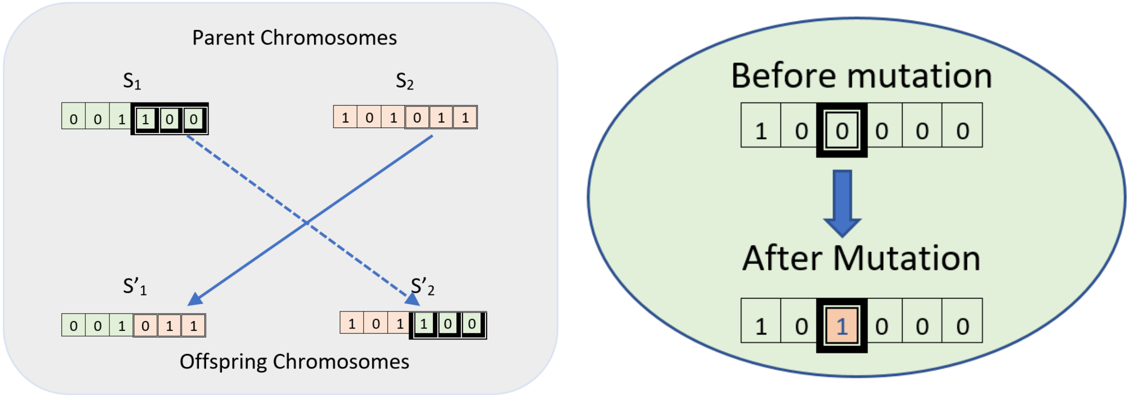
II. Selection (mating): After the first population $S_{(ini)}$ with fitness values has been created, the population keeps evolving by an iterative evolution process. The process for propagating each new generation starts at the step of mating. In the step, each pair of individuals (parents) is selected for producing the offspring (children). In general, there are many selection methods available for GA. As Saini (2017) mentioned, the most commonly used selection methods include the roulette wheel selection, the rank selection, the tournament selection, and the Boltzmann selection. The method used in our study is the roulette wheel selection. The selection by this method is proportionate to the fitness of the individual. Thus, the higher the level of fitness of an individual, the greater the chances of being selected. Therefore, the fittest candidates of parents have likely been chosen from the current population. In other words, individual solutions are selected through a fitness-based process, where fitter solutions are more likely to be selected. During the selection process, each pair of parents is selected repeatedly until the number of pairs is equal to $n/2$ and the population has n individuals.

III. Crossover: After the selection, the genotypes of any selected pair of individuals are modified to create a new pair using crossover. The pair of parents and the new pair of offspring are then compared, and two of them are selected according to better fitness. The procedure of crossover keeps iterating until the number of individuals reaches n. Note that the 'crossover' is an operation that exchanges some portion of the genotype of one chosen sample with the other, and which performs the global exploration of GA as the new sample is different from its parents. An example of crossover is shown in Figure 2.4 (a). The chromosomes of the two individuals are $S1 = (001100)$ and $S2 = (101011)$. After the crossover operation, the new individuals are $S'1 = (001011)$ and $S'2 = (101100)$.

IV. Mutation: A small number of the new generations are picked to modify some bits of genotype using mutation. Typically, less than 0.1% will be selected, and only one bit of the genotype will be changed by mutation. An example of mutation is shown in Figure 2.4 (b). Note that the 'mutation' is an operation that changes a small segment, and typically, just a bit of genotype has been changed (Marsland, 2015). Like crossover, mutation is an essential mechanism for GA optimisation for escaping local optima.

V. Elitism: The elitism operation is proceeded after crossover and mutation in the iteration, which will keep the n best from the old population to substitute n individuals in the new population. In general applications, n is set to 1. The new generation of candidates is then used in the subsequent iteration of the algorithm.

Finally, the algorithm terminates when the termination condition is met. This condition can be a predefined maximum number of iterations, or an adequate fitness value has been found from the population. For demonstrating the process of GA, a flowchart is shown in Figure 2.5.



(a) Crossover

(b) Mutation

Figure 2.4: Fig: Crossover and Mutation of Chromosomes

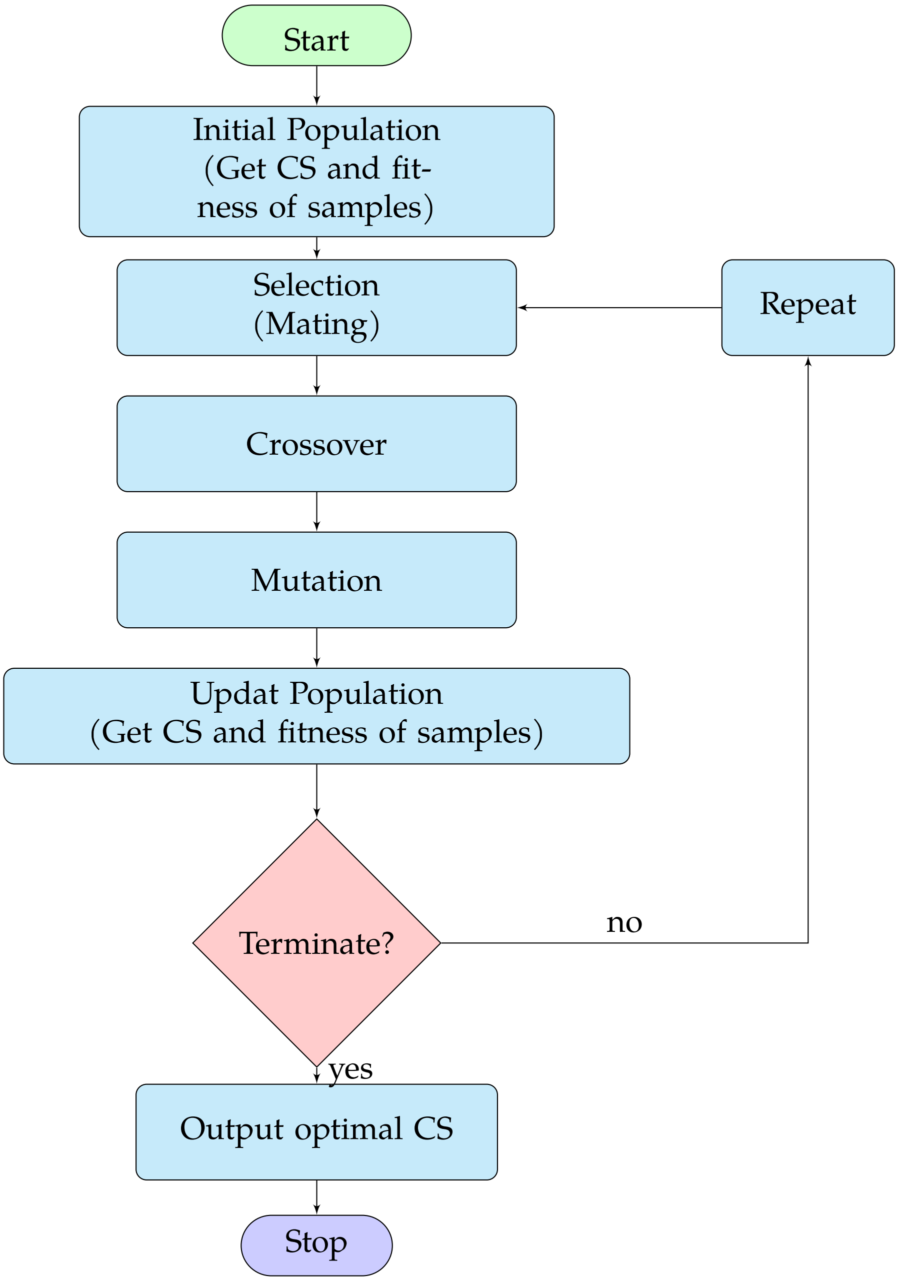

Start
Initial Population (Get CS and fitness of samples)
Selection (Mating)
Repeat
Crossover
Mutation
Updat Population (Get CS and fitness of samples)
Terminate?
no
yes
Output optimal CS
Stop

Figure 2.5: Flowchart of a Genetic Algorithm

GA is to maintain the best population (by way of crossover and mutation) from breeding new generations, to realise the evolution process for searching for the optimal solution. However, some researchers, such as Baluja and Caruana (1995), have mentioned that when there are many local optima in the searching space, the GA might be trapped in some local optima.

Furthermore, Larranaga et al. (1999) reviewed the different attempts made in their experiments to solve the travelling salesman problem (TSP) by the GA. This article has shown that for a TSP problem, there are many parameters provided for the experiment in the choosing and tuning process and accordingly, towards improving the result in terms of that specific TSP case. The TSP is known to be an NP-hard problem and its running time has been proven to be bounded by $O(n!)$ (Bonomi & Lutton, 1984). Intuitively, the TSP is more straightforward than the CSG problem, which has an order $O(n^n)$ in computational complexity theory (Chalkiadakis et al., 2011). Consequently, we argue that the GA approach might not be practical to solve serial and periodical CSG problems, which are the problems in this thesis for GA that always require choosing a set of suitable parameters before running the program. The modification of GA in solving the CSG problem will be presented in Chapter 4.

Particle Swarm Optimisation (PSO)

PSO is another popular SO method that optimises a problem by iteratively working to improve a candidate solution concerning a given fitness function. PSO was first introduced by Kennedy and Eberhart in 1995. It resolved a problem by creating a population of competitor solutions, namely particles, and driving these particles throughout the search-space in terms of a simple mathematical function over the particle's position and velocity.

PSO is widely utilised in the domains of smart grids and multi-agent systems. For instance, Pedrasa et al. (2010) proposed a decision-support tool to optimise energy services provision by enabling end-users first to assign values to the desired energy services and then schedule their available distributed energy resources to maximise net benefits. After that, they used PSO to solve the optimisation problem of the decision-support tool because of its straightforward implementation and demonstrated ability to generate near-optimal schedules within manageable computation times. Saber and Venayagamoorthy (2012) used PSO in their paper to minimise the cost and emissions problem. The problem is formalised in the resource scheduling under uncertainty in a smart grid with RE and plug-in vehicles. Mohamed et al. (2016) used PSO to maximise the system energy production and meet the load demand with a minimum cost and the highest reliability.

The particular strength of PSO is that it has both better abilities of exploration and exploitation to search for the optimisation in a continuous solution space. However, PSO is not very suited for problems in a discrete space such as CSG in our study. However, Grisales-Noreña et al. (2018) proposed to use the Population-Based Incremental Learning algorithm to locate distributed generators, i.e. a discrete number of locations, and uses PSO to define the size of those devices, i.e. a continuously adjustable amount. That article proposes a parallel implementation of PBIL and PSO, which combines the strength of the two algorithms, and consequently, has outperformed other algorithms.

However, in our unpublished research, we have used the PSO to solve the problem of minimising power loss. This problem is in the form of Coalitional Game Theory for the co-operative prosumers and is compared to Saad et al. (2011).

In this thesis, we have adopted the binary PSO (BPSO) to solve the CSG problems.

Population-Based Incremental Learning (PBIL)

In machine learning, PBIL is an estimation of distribution algorithm. In essence, PBIL is a variant of GA where the genotype (probability vector) of the entire current population is evolved, rather than the evolution of individual members (Karray et al., 2004). This algorithm is proposed by Baluja (1994). Because PBIL focuses only on searching for a better estimation of genotype distribution for an optimisation problem, they are without the need of some GA processes such as offspring and crossover, and are consequently more resilient and flexible in many cases, which leads to better and faster results than a standard GA (Baluja, 1994; Baluja & Caruana, 1995; Lee et al., 2017).

In PBIL, each gene in the probability vector is represented as a random value in the range of [0,1], representing the chance, compared to a given threshold value, that any specific allele will appear in that gene. Figure 2.6 shows a flowchart to demonstrate the process of PBIL. During the iterations, the algorithm only updates the initial probability vector of the threshold, which is different from GA, and the following generations are evolved, based on a new initial threshold vector throughout the process. The regular procedures of PBIL are described below:

1. Initialise the threshold vector (genotype): generate an initial threshold vector of a fixed value of L-length genotypes. In typical cases, an equal opportunity 0.5 is used, while in other cases the threshold may be chosen in the range of [0,1].

2. Generate a population of n random values of m length random genotypes: for each sample, compare the m length value to the threshold to form a binary vector.

3. Evaluate and rank the fitness of every sample.

4. Update the threshold vector based on some selected individual samples.

5. Repeat evolution procedure, i.e. steps 2 to 4, until a termination condition is met.

Because of the characteristics of PBIL, there are some applications using the PBIL algorithm in the domain of smart grids. For instance, Folly and Venayagamoorthy (2013) applied a multi-population-based PBIL approach to a power system controller design. The results of their simulation showed that the multi-population PBIL approach performed better than the standard PBIL and was as effective as PBIL where adaptive learning was used.

Grisales-Noreña et al. (2018) proposed a parallel implementation of using PBIL to locate distributed generators and using PSO to define the size of those devices. This approach utilised the parallel implementation of PBIL and PSO algorithms to search for the optimal location and sizing of distributed generation for improving the operation of electric systems. Their results demonstrated that the proposed parallel PBIL-PSO method provided the best balance between processing time, voltage profiles and reduction of power losses.

As far as we know, our proposed research of PBIL algorithms is the first publication using the PBIL method to solve a CSG problem (Rahwan et al., 2015; Lee et al., 2017). A more detailed explanation of how the process of searching for the optimisation by PBIL and other improved algorithms will be discussed in Chapters 4 to 6.

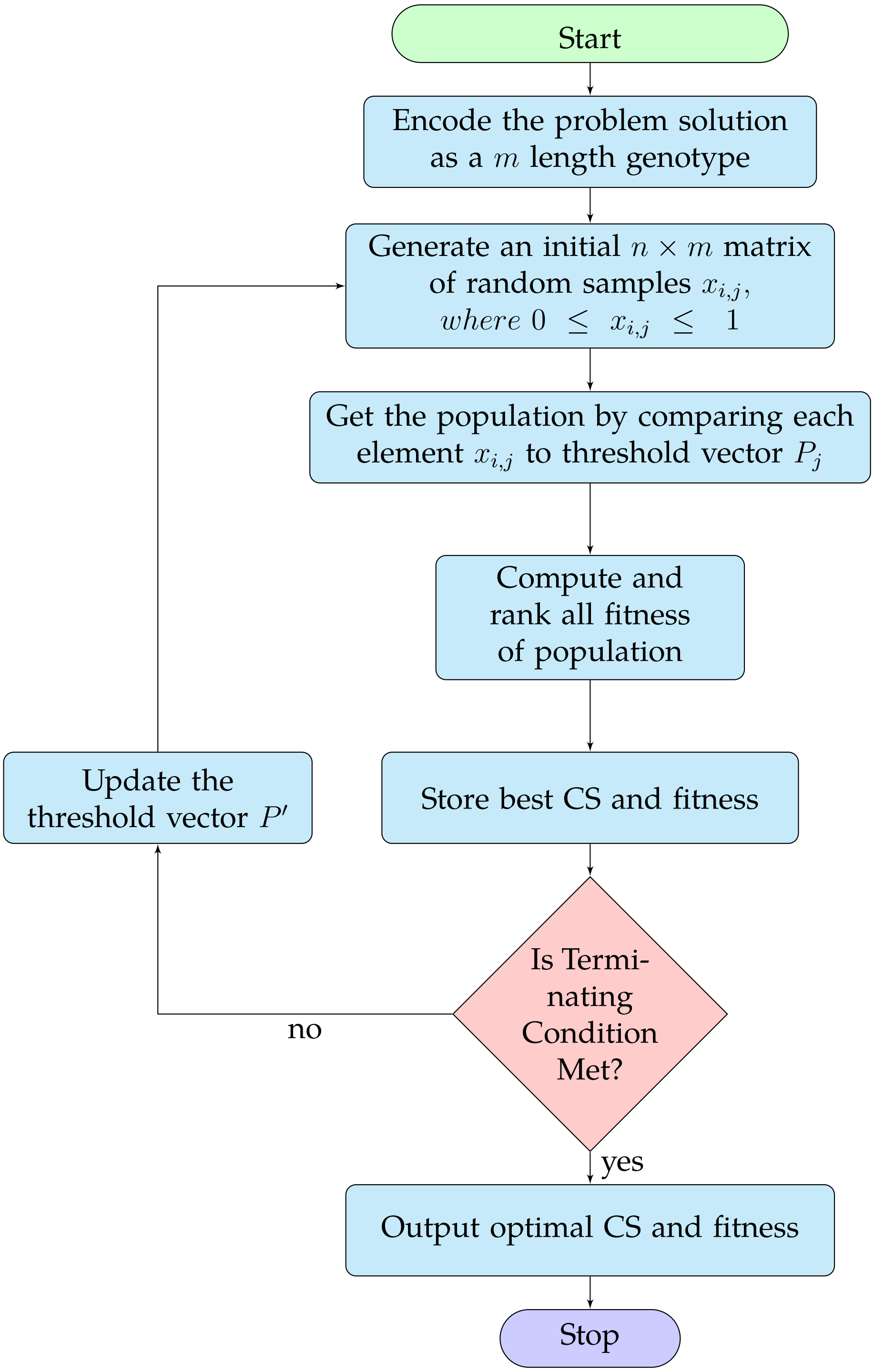

Start
Encode the problem solution as a m length genotype
Generate an initial $n \times m$ matrix of random samples $x_{i,j}$, $where\ 0 \leq x_{i,j} \leq 1$
Get the population by comparing each element $x_{i,j}$ to threshold vector P_j
Compute and rank all fitness of population
Store best CS and fitness
Update the threshold vector P'
Is Terminating Condition Met?
no
yes
Output optimal CS and fitness
Stop

Figure 2.6: Flowchart of a PBIL algorithm

2.4 Summary

As far as we know, Rahwan et al. (2015) have the most comprehensive survey for approaches to solve CSG problems. However, only a few SO approaches have been shown in this paper, and the main reason is that such algorithms can not guarantee finding an exact solution and not many papers have been published in this regard. Our purpose is to find better SO algorithms to solve CSG problem, but it is not possible to examine all the SO approaches. Therefore, a full review of all SO algorithms is beyond the scope of this research. However, some popular SO algorithms have been used to solve the CSG problems (Rahwan et al., 2015). For instance, Shehory and Kraus (1998) proposed a decentralised greedy algorithm which iteratively overlaps small coalitions, starting from a singleton, and growing to a larger size of coalitions. Although the greedy algorithm is comparably easy to implement, like other greedy applications, its searching result is easily to be trapped in a locally optimal solution.

In another study, Keinänen (2009) proposed a Simulated Annealing algorithm to solve the CSG problem. In that paper, starting from the initial CS, the algorithm recursively keeps searching for a better CS from their neighbourhood, until the termination of the program. Mauro et al. (2010) proposed another SO algorithm to solve the CSG problem. They present a greedy adaptive search procedure (GRASP) algorithm to efficiently search the space of coalition structures in order to find an optimal one. This new variant of the greedy algorithm shows a much better result than Shehory and Kraus (1998) as Rahwan et al. (2015) mentioned.

To sum up, the DP and its many variant algorithms are not able to accommodate the requirements for solving optimisation problems with a high-dimensional search space (Beheshti & Shamsuddin, 2013). In these problems, the search space grows exponentially with the problem size. Clearly, using 80 agents as we do in Chapter 6 is beyond the practical scope of DP. As a result, for a large scale of CS, it remains impractical to search for a global optimum using these approaches.

Although SO algorithms might only provide a near optimal solution, they have been proven to be the faster algorithms in running time and have fewer computer resource requirements. Moreover, they could still reach a reasonable and acceptable optimal solution in applications with a large number of agents. The above algorithms will be tested and compared with our proposed new algorithm in the following chapters.

Chapter 3

Co-operative Model for Prosumers

This chapter will construct and implement the coalition structure model for the co-operative RE agents of an alliance in smart grids. Firstly, we introduce the concept of forming coalitional models accompanied by the related literature. Following the discussion, the research design is illustrated and constructed, which includes the constraints of the model and fitness evaluation in different considerations. Based on the proposed model, we create two scenarios: one for local applications and the other for multi-region applications, respectively. The players of the CSG games, i.e. agents of the prosumer, are depicted according to their power consumption and production.

3.1 Models and Incentives in Forming Coalitions

On account of the intermittent nature of renewable energy, a prosumer will need to solve the balance of production and consumption frequently. One possible, but not worthwhile, solution is to expand the capacity of backup storage, such as increasing the battery size, to meet the needs during the deficit hours. Consequently, this will raise much more investment expense and discourage the prosumer from upgrading the RE facilities. Alternatively, most prosumers chose to trade with the utilities so they can sell the excess to them and purchase the shortage from them.

In general, many of the current utilities in power trading markets have offered RE prosumers different price rates to sell and buy-back the power (Alderfer et al., 2000). For example, the buy-back rates of solar power[1] are much lower than the

[1] https://mysolarquotes.co.nz/about-solar-power/residential/solar-power-buy-back-rates-nz/ Accessed: 31 Dec. 2019.

selling prices[2] in New Zealand. Similarly, the buy-back rates are much lower than the purchase prices in some other countries (Philpott et al., 2019; IEA, 2017). Consequently, it is becoming less profitable for a prosumer to sell the excess to the utility.

Similar to the current RE prosumers, a usual consumer who wishes to install a RE device will face the quandary of choosing a moderate combination of generator and backup system. For example, for a new prosumer with solar energy, the larger area of PV panel will bring more energy in sunshine hours. However, the prosumer will also need to consider investing in more extensive but costly backup storage to store the excess energy as much as possible or installing a small size of storage and selling the excess to the utility with the low buy-back price.

Therefore, suppose we can provide a co-operative mechanism for the prosumers to form an alliance that allows those surplus prosumers to share their excess energy with others, and in return receive power when they have a shortage. On a financial basis, the prosumer will firstly incline toward sharing the power in the alliance for increasing a higher profit and achieving a more significant RE utilisation percentage.

Following the above concept, this mechanism not only can mitigate the expense but also may bring more economic incentive for prosumers. As a result, this mechanism will attract more consumers to become prosumers and encourage current prosumers to enlarge their facilities for fulfilling their long-term power requirements.

Based on the concept above, under the emerging techniques, such as smart grids, smart meters and the internet of things, the agents of prosumers can now communicate and deliver power with others willingly, and accordingly, can coordinate instantly in order to form co-operative coalitions.

Theoretically, these approaches are known as Co-operative Games, *aka* 'Coalitional Games' (Chalkiadakis et al., 2011), which allow players, i.e. agents, to establish agreements in terms of choosing the strategies and distributing their payoff. In addition to the co-operative games of smart grids, there are two major topics which are related to our problems.

The first is how to build a reasonable, co-operative model to formalise the cooperation, such as 'Induced Sub-graph Games' (Deng & Papadimitriou, 1994), 'Net-

[2] https://www.switchme.co.nz/ Accessed: 31 Dec. 2019.

work Flow Games' (Kalai & Zemel, 1982), 'Weighted Voting Games' and 'Coalition Structure Generation Games' (Chalkiadakis et al., 2011).

The second topic is how to evaluate different coalitions by the characteristic function, *aka* utility, for comparing various coalitions and searching for the optimal solution. For example, Pitt et al. (2012) propose a voting function with satisfaction value for agents to achieve a better balance of utility and fairness. Bourazeri and Pitt (2018) suggest a mechanism by attributing rewards of social capital for agents. Saad et al. (2011) use strategies of a power loss function in distributed smart grids in order to minimise the total power losses.

Based on the discussion above, the following section will explain the reasons in selecting our model and bringing forth a price scheme for coalitions evaluation in our proposed model.

3.2 Design of Coalition Model

3.2.1 Related Work of Forming Coalitions

In terms of the forms of coalitions, some researchers such as Yasir et al. (2015) propose a dynamic coalition formation for an agent to join any coalition with its own decision which depends on the consideration of moderating the discomfort level; Bourazeri and Pitt (2018) suggest agents pursue rewards of social capital from coalitions. Nevertheless, it would be hard to measure their quantity while forming coalitions by such abstract types of concept. For example, the perception of discomfort level, like the feeling of hunger, might have a significant difference between agents; the reward of social capital is an abstract concept based on many principles of sociology which need lots of studies and consensus concerning common-pool resources (Pitt et al., 2012). As a result, such schemes could be favourable in the power grids of island modes (IEEE, 2011) which frequently lack the support of public utilities. However, they might not be the most attractive strategies in coalition formation when the agents are connected to the public grids. Alternatively, assuming that the economic factor is the primary consideration for most of the prosumers, the strategy of price incentive might be the most efficient method.

3.2.2 Proposed CSG model to Form Coalitions

Following the related work mentioned above, this thesis proposes a new co-operative model by using CSG to form coalitions. Additionally, notwithstanding that the prosumer wishes to be self-sustaining, every agent may occasionally face a shortage because of the intermittent outcomes of the RE generators. When there still exists a notable price difference in power trading markets, it is apparent that having a scheme of sharing the power with a lower price for exchanging power among the prosumers may provide a better alternative solution. Typically, by forming an alliance or group under an agreement to exchange energy with others in the alliance will provide an adequate mechanism to achieve the scheme. Besides, such a scheme will present a favourable, plausible and more economical method in comparison with enlarging the facility (such as by adding wind turbines or solar panels), installation of more backup storage, or trade with the power company. So, our goal is to improve the power-sharing among agents by introducing more economic incentive and, as a result, to stimulate more consumers to become prosumers and to support current prosumers to expand their facilities for fulfilling their long-term power sustainability. We then consider several factors to design our model for experiments as follows:

- Equity and Fairness: We assume that every agent who is willing to participate in the coalitional alliance should generate sufficient RE to meet its requirement in the long term. For example, it is compulsory to have a proper solar panel installed to generate adequate energy in supporting the annual power consumption, and likewise a fair yearly balance for wind turbine agents and other types of prosumers as well.

- Sharing before trading: For the mechanism to keep working adequately and efficiently, every agent in the alliance needs to provide the power information internally to the alliance. Thus, the alliance will estimate and form optimal coalition structures based on the information. Accordingly, all the "excess" agents should provide power for the "deficit" agents in the same chosen coalition before selling to the utility.

- Stability and steadiness: The proposed model can be working in on-grid or off-grid mode. Even if the model is running by an on-grid alliance, however, during a disaster period while the connection to the public distribution system is disconnected, the agents with excess may keep supporting the deficit agents while the internal connection still exists. Note that while in the off-grid condition, if the demand is higher than the supply, it will cause a voltage drop which will make the network unstable. Given this reason, the model requires that the alliance should provide stability and steadiness while the mechanism keeps running. Therefore, the alliance can form any feasible coalition which must have a net power surplus in order to prevent the network voltage from dropping and resulting in unstable fluctuation.

- Minimised power demand before request: Furthermore, we require that a deficit agent should have an obligation to minimise its essential power need at the given period such that the remaining surplus agents can benefit as many deficit agents as possible.

Definition of Alliance: In our proposed co-operative model, an alliance is defined as a group of participating agents under the agreement of the approvals mentioned above.

According to the shifting requirements in various conditions, the alliance might form several different coalitions to achieve an optimal solution to maximise the total benefits of the alliance. More importantly, to avoid causing unstable fluctuation within the smart grid, we require that a feasible coalition should still be a coalition with a net surplus.

Following the rules of forming coalitions, each formed coalition cannot freely transfer the utility value, i.e. profit, to other coalitions. Therefore, our model is a coalitional game with non-transferable utility (NTU) (Chalkiadakis et al., 2011). Shoham and Leyton-Brown (2008) mentioned that these NTU games 'explicitly list all the divisions that are possible and prohibit the rest'. Furthermore, from the definition of CSG in Subsection 2.2.1, we know that our proposed model is the model of a CSG problem.

For our CSG problem, we will explain the constraints of forming coalitions and the methods of computing the utility values of coalitions in the following subsection.

Constraints of Coalitions

For every period (i.e. per hour in this study), any agent a_i with extra power can share its surplus with others who are experiencing a shortage within the co-operative alliance. The goal of the alliance is to maximise the total benefit by forming coalitions, as shown in Eq. 2.2. Furthermore, we require that a deficit agent should have an obligation to minimise its essential power need at the given period such that the remaining surplus agents can benefit as many deficit agents as possible. Based on the obligation and to avoid causing unstable fluctuation within the smart grid, the alliance can form any feasible coalition which must have a net power surplus.

For instance, suppose the agents a_1 and a_2 each has 1.5 and 0.9 kWh excess accordingly, but a_3 has a shortage of 1.2 kWh. According to the requirement, a_1 and a_3 can team up as a feasible coalition $\{a_1, a_3\}$. On the other hand, a coalition such as $\{a_2, a_3\}$ is not accepted.

Under the constraint above, we further know that once the whole alliance has a net power surplus at a given hour, such that all the deficit agents can obtain their power requirement, this will lead to a grand coalition (Shoham & Leyton-Brown, 2008) and become a trivial solution for the CSG game. Thus, we consider only cases with a net power deficit for the alliance where a grand coalition is unfeasible, and a game of CSG should be constructed and needs to be resolved.

Further to the proposed power-sharing scheme, some power loss will happen with the power transportation among the agents. However, when the distance between agents is short (which is the local scenario described in Section 3.3), then the loss is trivial and can be neglected. On the other hand, for a multi-region scenario, the power loss constraint will be taken into account in our model as described in Chapter 6.

To explain the constraints applied to local and inter-region models, we will present the evaluation function and discuss the difference in the following subsection.

Coalition Evaluation

I. **Profit Evaluation Functions for Cooperation approach:**

For a set of k prosumers, denoted by $C_{(k)}$, at a given hour, let Q_s^i and Q_b^j denote the amount of power an agent has in excess or shortage respectively. The prices for a prosumer to purchase from or sell to the utility are denoted by P_b and P_s severally. We know that P_b is higher than P_s, such that $P_b > P_s$. Let $Q(s)$ denote the sum of power for agents with excess, and $Q(b)$ denote the sum of power for agents with shortage accordingly. For demonstrating the difference for the agent to cooperate or not, we need to compute the sum of currency $v(C_{(k)})$ for the set respectively.

A. Non-cooperation:

In this circumstance, $v(C_{(k)})$ is the sum of surplus agents selling Q_s^i power to the utility minus the sum of deficit agents purchasing Q_b^j power from the utility, and are given by

$$\begin{aligned} v(C_{(k)}) &= \sum_{i=1}^{l} (Q_s^i \times P_s) - \sum_{j=1}^{m} (Q_b^j \times P_b), \qquad 0 \leq l, m \leq n \text{ and } l + m = n, \\ &= \left(\sum_{i=1}^{l} Q_s^i \right) \times P_s - \left(\sum_{j=1}^{m} Q_b^j \right) \times P_b \\ &= [Q(s) \times P_s] - [Q(b) \times P_b] . \end{aligned} \tag{3.1}$$

B. Cooperation:

Suppose the prosumers can share the power within the set, then the net amount for the set to purchase from or sell to the utility, denoted by $Q(k)$, are given by

$$\begin{aligned} Q(k) &= \left(\sum_{i=1}^{l} Q_s^i - \sum_{j=1}^{m} Q_b^j \right) \\ &= Q(s) - Q(b). \end{aligned} \tag{3.2}$$

Therefore, the prosumers will share the power internally before they trade with the utility. Hence, there are two conditions that need to be considered.

a. Net Deficit:

While there is not enough power to meet the demand, i.e. $Q(k) < 0$, then the set still needs to be purchased from the utility. In this status, the sum of currency $v'(C_{(k)})$ is the net amount the set needs to purchase from the utility with price "P_b", and is given by

$$\begin{aligned} v'(C_{(k)}) &= \left(\sum_{i=1}^{l} Q_s^i - \sum_{j=1}^{m} Q_b^j \right) \times (P_b) \\ &= [Q(s) - Q(b)] \times (P_b) \\ &= [Q(s) \times P_b] - [Q(b) \times P_b] \, , \end{aligned} \tag{3.3}$$

since we know $P_b > P_s$, such that $[Q(s) \times P_b] > [Q(s) \times P_s]$. From Equations 3.3 and 3.1 we get

$$v'(C_{(k)}) = [Q(s) \times P_b] - [Q(b) \times P_b] > [Q(s) \times P_s] - [Q(b) \times P_b] = v(C_{(k)}), \tag{3.4}$$

which means the cooperation will bring a benefit, i.e. pay less to the utility. Consequently, let $v^*(C_{(k)})$ denote the net profit for the agents with cooperation in comparison with the agents without cooperation. Let Q_d represent the amount that the set can share internally, and this is equal to $Q(s)$ in this condition. And let $P_r = (P_b - P_s)$ denote the price difference by way of trading with power utility. Again, from Equations 3.3 and 3.1 we know

$$\begin{aligned} v^*(C_{(k)}) &= v'(C_{(k)}) - v(C_{(k)}) \\ &= \{[Q(s) \times P_b] - [Q(b) \times P_b]\} - \{[Q(s) \times P_s] - [Q(b) \times P_b]\} \\ &= [Q(s) \times P_b] - [Q(s) \times P_s] \\ &= Q(s) \times (P_b - P_s) \\ &= Q_d \times P_r \, . \end{aligned} \tag{3.5}$$

b. Net Surplus:

While there is enough power to support the demand, i.e. $Q(k) \geq 0$, then the set needs no purchase from the utility. In this situation, the sum of currency $v''(k)$ is the net amount sold to the utility with price "P_s", and is given by

$$\begin{aligned} v''(C_{(k)}) &= \left(\sum_{i=1}^{l} Q_s^i - \sum_{j=1}^{m} Q_b^j \right) \times (P_s) \\ &= [Q(s) - Q(b)] \times (P_s) \\ &= [Q(s) \times P_s] - [Q(b) \times P_s] \; . \end{aligned} \tag{3.6}$$

Since we know $P_b > P_s$, such that $(-1) \times [Q(b) \times P_b] < (-1) \times [Q(b) \times P_s]$. From Equations 3.6 and 3.1 we get

$$\begin{aligned} v''(C_{(k)}) =& [Q(s) \times P_s] - [Q(b) \times P_s] > \\ & [Q(s) \times P_s] - [Q(b) \times P_b] = v(C_{(k)}), \end{aligned} \tag{3.7}$$

which mean the cooperation will bring more profit. Consequently, the Q_d is the amount that the set can be share internally, and is equal to $Q(b)$ in this condition. Again, from Equations 3.6 and 3.1 we know

$$\begin{aligned} v^*(C_{(k)}) &= v''(C_{(k)}) - v(C_{(k)}) \\ &= \{[Q(s) \times P_s] - [Q(b) \times P_s]\} - \{[Q(s) \times P_s] - [Q(b) \times P_b]\} \\ &= (-1) \times [Q(b) \times P_s] - (-1) \times [Q(b) \times P_b] \\ &= Q(b) \times (P_b - P_s) \\ &= Q_d \times P_r \end{aligned} \tag{3.8}$$

In summary, no matter what the net amount $Q(k)$ is, the net profit $v^*(C_{(k)})$ is always better than agents without cooperation. Thus, we know

$$v^*(C_{(k)}) = Q_d \times P_r, \tag{3.9}$$

for all situations.

To demonstrate the profit of Eq. 3.9, let us examine two short examples, both having a set of two agents $\{a_1, a_2\}$. The prices for agents to sell and purchase power with the utility are 20 and 50 $(¢/kWh)$. In the first example, let us assume a_1 and a_2 have surplus 0.5 and -0.4 (kWh), separately. For non-co-operative mode, the sum for trading with utility is given by $0.5 \times 20 + (-0.4) \times 50 = -10$ $(¢)$. In our co-operative mode, the agents can share power by the price of the selling price. Since the net power $(0.5 - 0.4 = 0.1)$ is positive, the sum is given by $(0.5 - 0.4) \times 20 = 2$ (¢). Hence, the net profit is given by $2 - (-10) = 12$. Furthermore, by using Eq. 3.9, we can compute the profit which is given by $0.4 \times 30 = 12$, where $Q_d = 0.4$ and $P_r = 50 - 20 = 30$ (¢).

Again, let us have a second example. Suppose the two agents a_1 and a_2 have surpluses 0.3 and -0.4 (kWh) separately. For non-co-operative mode, the sum for trading with the utility is given by $0.3 \times 20 + (-0.4) \times 50 = -14$ (¢). In the co-operative mode, the agent a_1 can share 0.3 (kWh) power to a_2 by the price of the selling price. Since the net power $(0.3 - 0.4 = -0.1)$ is negative, the sum is given by $(-0.1) \times 50 = -5$ (¢). Hence, the net profit is given by $(-5) - (-14) = 9$. Furthermore, by using Eq. 3.9, we can compute the profit which is given by $0.3 \times 30 = 9$, where $Q_d = 0.3$ (kWh) and $P_r = 30$ (¢).

II. **Coalition Fitness Function:**

Following the Definition 2 in Subsection 2.2.1, for any coalition C_k in a $\mathcal{CS}$, i.e. $C_k \subseteq \mathcal{CS}$, the **value** of a $\mathcal{CS}$ is denoted by $v(\mathcal{CS})$ and is given by:

$$v(\mathcal{CS}) = \sum_{C_k \in \mathcal{CS}} v(C_k). \tag{3.10}$$

Besides, let us assume the prices for any prosumer to purchase from or sell to the utility are denoted by P_b and P_s respectively. Further we know that P_b is higher than P_s, such that $P_b > P_s$. For convenience, let $P_r = (P_b - P_s)$ represent the price difference by way of trading with the power utility. Therefore, in our study, $v(C_{(k)})$ is the fitness function of $C_{(k)}$ given by

$$v(C_{(k)}) = \begin{cases} 0 & \text{if } \; k = 1, \\ Q_d \times P_r & \text{if } \; k > 1 \text{ and } \; Q(C_{(k)}) \geq 0, \\ -9999 & \text{otherwise,} \end{cases} \tag{3.11}$$

where k denotes the size of coalition $C_{(k)}$, Q_d is the total power need for deficit agents in the $C_{(k)}$, and $Q(C_{(k)})$ is the net surplus within the $C_{(k)}$ accordingly. Furthermore, for giving penalty to an unfeasible coalition, we let $v(C_{(k)}) = -9999$.

Therefore, we aim to find the global best CS* and its optimal profit, as discussed in Chapter 2. For instance, let a_1 and a_3 each have 1.5 and 0.6 kWh excess respectively, but say a_2 has a shortage of 1.1 kWh. In terms of the requirement, a_1 and a_2 can form a feasible coalition $\{a_1, a_2\}$. On the contrary, a coalition such as $\{a_2, a_3\}$ is not rewarding. Therefore, a grand coalition cannot always be feasible and a model of CSG (Shoham & Leyton-Brown, 2008) should be constructed and needs to be resolved.

III. **Advanced Coalition Evaluation Function with Power Loss Constraint:**

Besides the price scheme, there is another factor we need to put into consideration, and that is the power transmission loss. Theoretically, there are two types of power loss; the first one is the loss caused by the transmission lines, and the other is due to the voltage changing process via transformers. Following is a brief discussion of the power losses concerned in this study.

The transmission of power P_{ij} between any two locations i and j will always be accompanied by a power loss (L) which can be given by

$$\begin{aligned} L_{ij} &= I_{ij}^2 R_{ij} + \beta P_{ij} \\ &= I_{ij}^2 \cdot \alpha \cdot d_{ij} + \beta P_{ij}, \end{aligned} \tag{3.12}$$

where i and j stand for the locations of power sent and received respectively, R_{ij} is the resistance of transmission lines between i and j, I_{ij} is the current flowed, β is the fraction of power lost in the transformer at a substation during the voltage changing process, α is the resistance of wire per unit length, and d_{ij} is the distance between i j. For a power distribution line with fixed transmission voltage V, the current I_{ij} can be substituted by $I_{ij} = \frac{P_{ij}}{V}$ and Eq. 3.12 can be expressed as

$$L_{ij} = \alpha d_{ij} \left[\frac{P_{ij}}{V}\right]^2 + \beta P_{ij}. \tag{3.13}$$

In connection with Eq. 3.13, we choose the typical values of $\alpha = 0.2\ _{Ohm/Km}$, $V = 22_{kV}$ and $\beta = 0.02$ (Saad et al., 2011) as the parameters in the power transmission function. For a co-operative model, there are two conditions under consideration:

(i) While the surplus power is shared locally, there is no power transferred via the transformer, hence the term of β can be omitted and the power loss for i is

$$L_{ij} = \alpha d_{ij} [\frac{P_{ij}}{V}]^2 = \frac{1}{242} \times 10^{-6} \times d_{ij} \times P_{ij}^2 \approx 4.13 \times 10^{-9} \times d_{ij} \times P_{ij}^2. \tag{3.14}$$

A distance in the local area is usually less than $10\ km$, and the hourly generated and consumed energy for a prosumer is usually less than 10 kWh in our experiments. For example, suppose agent a_1 has 10 kWh to share with a_2. The distance between them is 10 (km). Thus, from Eq 3.14 we can compute the power loss, which is given by $4.13 \times 10^{-9} \times 10 \times 10^2 = 4.13 \times 10^{-6} kWh$. Therefore, we know the value obtained from Eq 3.14 is trivial and can be neglected in our local scenario, which is described in the next section. Hence we can directly obtain the total power needs, Q_d, for deficit agents from the fitness function Eq. 3.11.

(ii) When the surplus is shared with other locations by way of transformers and transmission lines, we still need to use Eq. 3.13 for computing the power loss. Similar to the local model, the distance between inter-region is usually less than 50 km, and the bound for a prosumer remains the same as well. Then Eq. 3.13 becomes

$$\begin{aligned} L_{ij} &= \frac{0.2}{(22 \times 10^3)^2} \times d_{ij} \times P_{ij}^2 \quad + \beta \times P_{ij} \\ &= \frac{1}{242} \times 10^{-6} \times d_{ij} \times P_{ij}^2 \quad + \beta \times P_{ij}. \end{aligned} \tag{3.15}$$

Again, the power loss caused by transmission lines is still trivial and neglected in our inter-region scenario. Thus, Eq. 3.15 shifts to

$$L_{ij} \approx \beta \times P_{ij}, \tag{3.16}$$

where β represents the power loss caused by the number of transformers, denoted by n, between the two agents; therefore, β is equal to $0.02 \times n$. Consequently, we put the constraint of Eq. 3.16 into the calculation of Q_d when the two agents belong to different regions in Chapter 6.

3.3 Coalitional Scenarios Setup

To compare the accuracy and efficiency, and to demonstrate the ability of our proposed algorithms compared with other algorithms, we need to put forward various experiments on different scales of CSG. Thus, there are two scenarios of coalitions suggested in this thesis, namely local coalition structure and regional coalition structure.

3.3.1 Scenario of Local Coalitions

Figure 3.1 shows the power connection of local coalition structure. In this scenario, we assume that the agents of the RE prosumers are located within a distribution grid, e.g. a community or district, which could provide power-sharing directly by the distribution network. From Eq. 3.14 we know that the transmission cost and power loss are trivial; thus the loss is ignored in this scenario (Lee et al., 2017, 2018a).

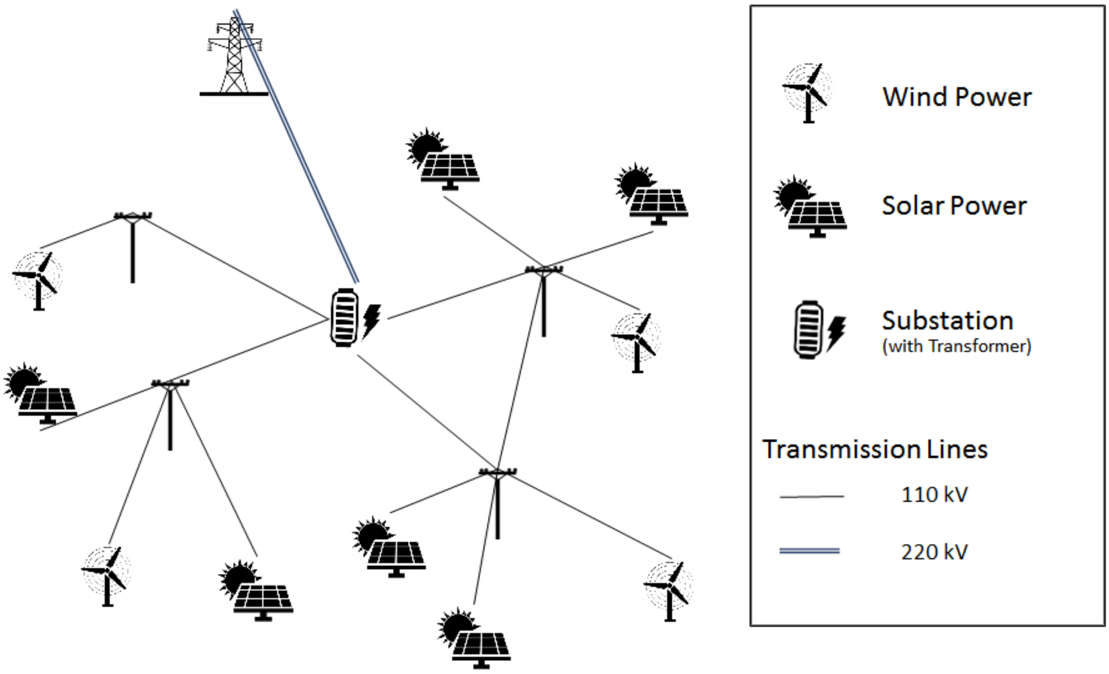


Figure 3.1: Power Connection for Scenario of Local Coalitions

3.3.2 Scenario of Regional Coalitions

In terms of the previous scenario, to demonstrate the ability on a large regional scale, such as multi-city, we further extend the distribution of agents to four regional areas as shown in Figure 3.2. Thus, the power-sharing within an inner area remains the same as the local scenario. However, power-sharing across inter-areas needs to be sent by a high voltage transmission network by way of transformers for exchanging power between distribution and transmission lines. From Eq. 3.16 we know that the power loss caused by the short distance electrical lines is trivial. So, to simplify the coalition model, only the power losses of transformers, β=2% each, per step up or down via a transformer, are consequently calculated.

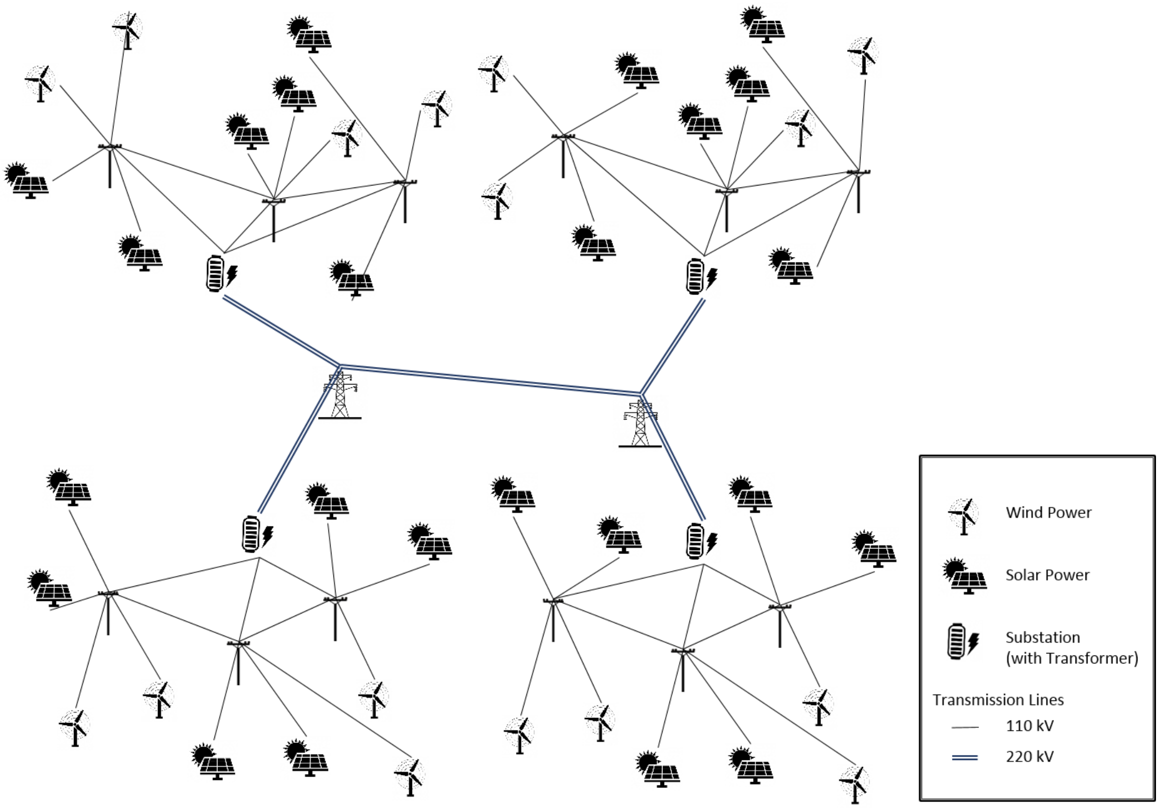


Figure 3.2: Power Connection for Scenario of Regional Coalitions

3.4 Modelling of Agents in Smart Grids

Whether the current energy status of an agent is in excess or shortfall depends on the prediction of power conditions which the agent will generate or consume in the next period. This study utilises real readings of smart meters and weather records from the data of New Zealand (NIWA, 2019).

In the following subsections, the process for obtaining and analysing the consumption profiles and the generating power of RE will be presented and demonstrated separately. Finally, we will explain how we integrate the consumption and generation into hourly condition data for the prosumers used in these experiments.

3.4.1 Power Consumption Data

To examine the electricity consumption data of households, we obtained half-hourly household readings (48×366 records) of 240 smart meters from New Zealand smart grids in 2008 (Nair & Zhang, 2009).

Figure 3.3 shows diverse variations, and an average of 40 randomly chosen households in 24 hours. Generally, it is clear to see that there are some daily peak hours of average power consumed, such as morning and afternoon activities.

Figure 3.4 shows the histogram of half-hourly consumption of these 240 agents. From the figure, it is clear that the distribution is a log-normal distribution which has coincided with some human behaviours (Galli, 2009).

The data of power profiles will provide hourly consumption for the agents in the following experiments. Meanwhile, the accumulated annual power usage for each agent will provide the reference when we choose the proper size of the PV panel and the wind turbine for coupling the agent to become a prosumer in the experiments.

3.4.2 Renewable Energy Data

Among those resources of RE, wind and solar are widely utilised and have the most commercialised devices for electricity supply to meet the consumption of private houses. Therefore, this study has chosen wind turbines and PV solar panels to be the renewable resources in experiments. Furthermore, both the wind speed and solar radiation hourly data are obtained from the New Zealand National Institute of Water and Atmospheric Research (NIWA)[3].

[3] Meteorological data obtained from NIWA's National Climate Database (https://cliflo.niwa.co.nz/ Accessed: 31 Dec. 2019.).

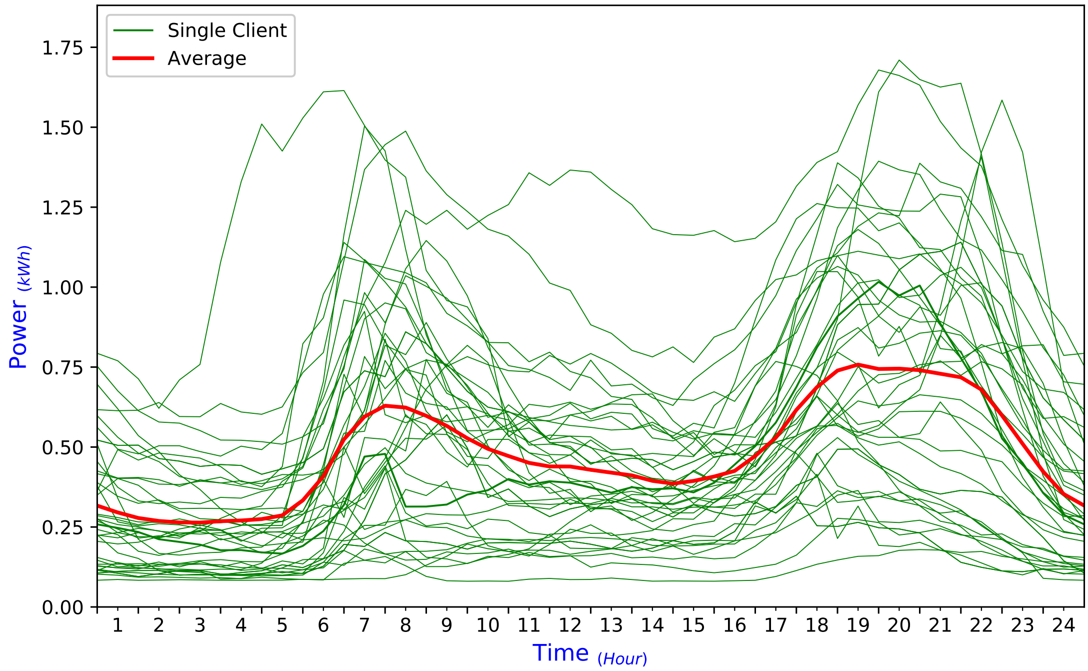


Figure 3.3: Half-hourly consumption profiles of 40 households.

Furthermore, for simplicity, the meteorological data used in our experiments are from four urban areas located in the South Island of New Zealand, namely, Dunedin, Balclutha, Middlemarch and Ranfurly. The first two urban areas are located in a windy coastal zone, and the other two are based in the sunny Central Otago region. The distance between these urban areas is approximately 40 to 80 km. Therefore, both wind speed and solar radiation have $24 \times 366 \times 4$ recordings in our experiment.

Wind Energy

The general method to convert wind power into electricity is by using wind-turbines. Therefore, for estimating the achievable wind energy that could be generated, one needs to measure the power output of commercialised wind turbines in terms of the wind-speed. The approaches of obtaining wind energy for prosumer to utilise are listed below:

I. Wind-speed data collection and analysis:
Figures 3.5 and 3.6 represent the various characteristics, e.g. peak hours and wind-speed spread range, between urban areas, which reveal the potential for cooperation between urban areas as well.

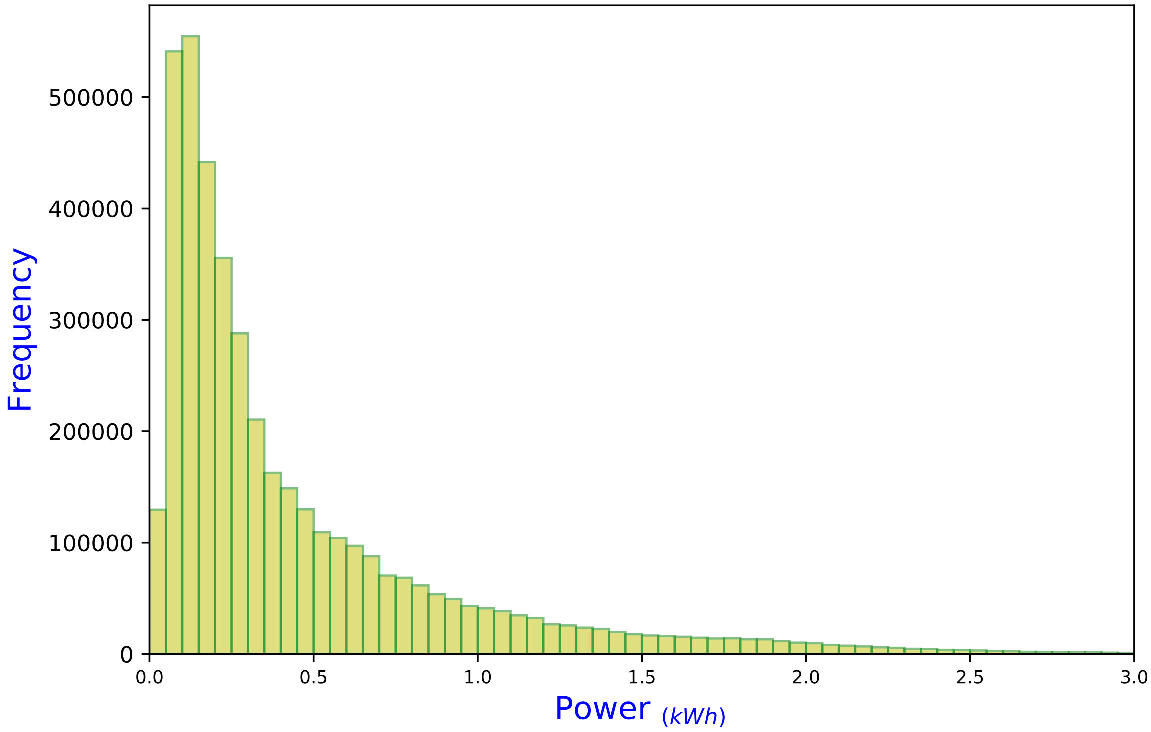


Figure 3.4: Histogram of half-hourly consumption data.

The wind-speed data of NIWA were usually measured at heights of both 5 and 10 meters[4]. To simplify the estimation, we assume that the wind-speeds were all measured at the height of 10 meters, which is much lower than the heights of commercialised wind turbines, we assume they are all installed at the height of 30 meters. To estimate the expected wind-speed at the height of wind turbines from NIWA's measured speed, we adopt the logarithmic profile equation(Oke, 2002). The logarithmic profile equation is generally used in the lowest 100 meters to describe the vertical distribution of horizontal mean wind-speeds within the lowest portion of the planetary boundary layer (Oke, 2002; Nfaoui, 2012). The equation to estimate the mean wind speed V_z at height z (meters) above the ground is:

$$V_z = \frac{V_{fr}}{k}\Big[ln(\frac{z}{z_0}) + \Psi(\frac{z}{L})\Big], \tag{3.17}$$

where v_{fr} is the friction velocity, k is the Von Kármán constant $(k \approx 0.41)$, Ψ is a stability term, and z_0 is the surface roughness (in meters), and, L is the

[4] https://www.niwa.co.nz/atmosphere/research-projects/estimating-design-wind-speeds-in-complex-terrain Accessed: 31 Dec. 2019.

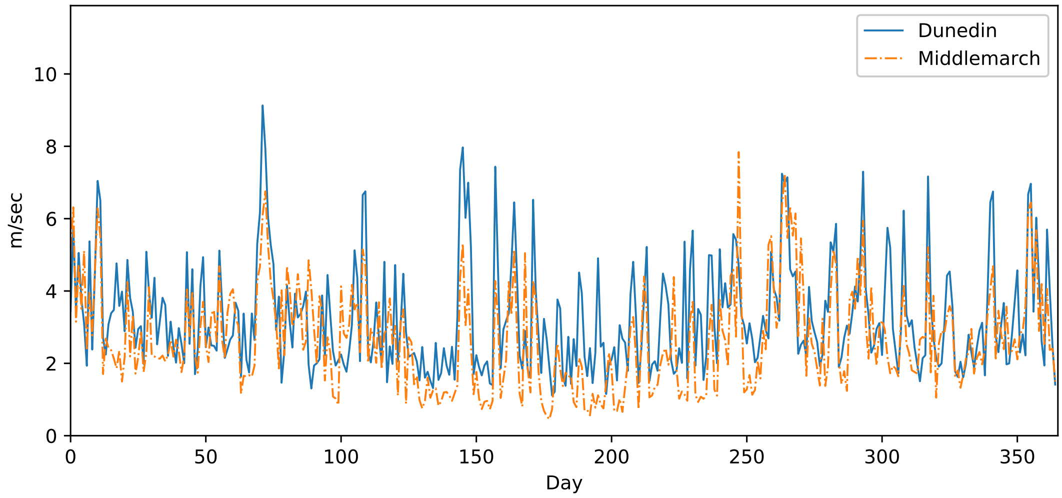


Figure 3.5: Hourly Wind-speed profile in a year

Obukhov length from Monin-Obukhov similarity theory. As mentioned by Pasquill (1961), under neutral stability conditions, $z/L = 0$ and Ψ drops out, and the equation is simplified to

$$V_z = \frac{V_{fr}}{k}\Big[ln(\frac{z}{z_0})\Big]. \tag{3.18}$$

In Equation 3.18, roughness length z_0 is a corrective measure to account for the effect of the roughness of a surface on wind flow and is between 1/10 and 1/30 of the average height of the roughness elements on the ground. Over smooth, open water, we expect a value around $0.0002\,m$, while over flat, open grassland $z_0 \approx 0.03\,m$, cropland $z_0 \approx 0.1-0.25\,m$, and brush or forest $z_0 \approx 0.5-1.0\,m$, where values above $1\,m$ are rare and indicate excessively rough terrain. Again, to simplify the estimation, we assume that the default $z_0 = 0.4\,m$ in the estimation.

By using Equation 3.18, the measured wind-speed data, i.e. $10\,m$ above ground, are transferred to the expected wind-speed data, i.e. $30\,m$ above ground, where the turbines are to be installed. Figure 3.7 shows the measured and expected wind-speed within a set of 24-hour data for the selected urban areas.

II. Wind-turbines types and fitness:

Currently, there are many commercial wind-turbines available for small scale generation, namely micro-generation, in terms of assorted individual household power consumption. From these, this study chose three types, i.e. 1, 3

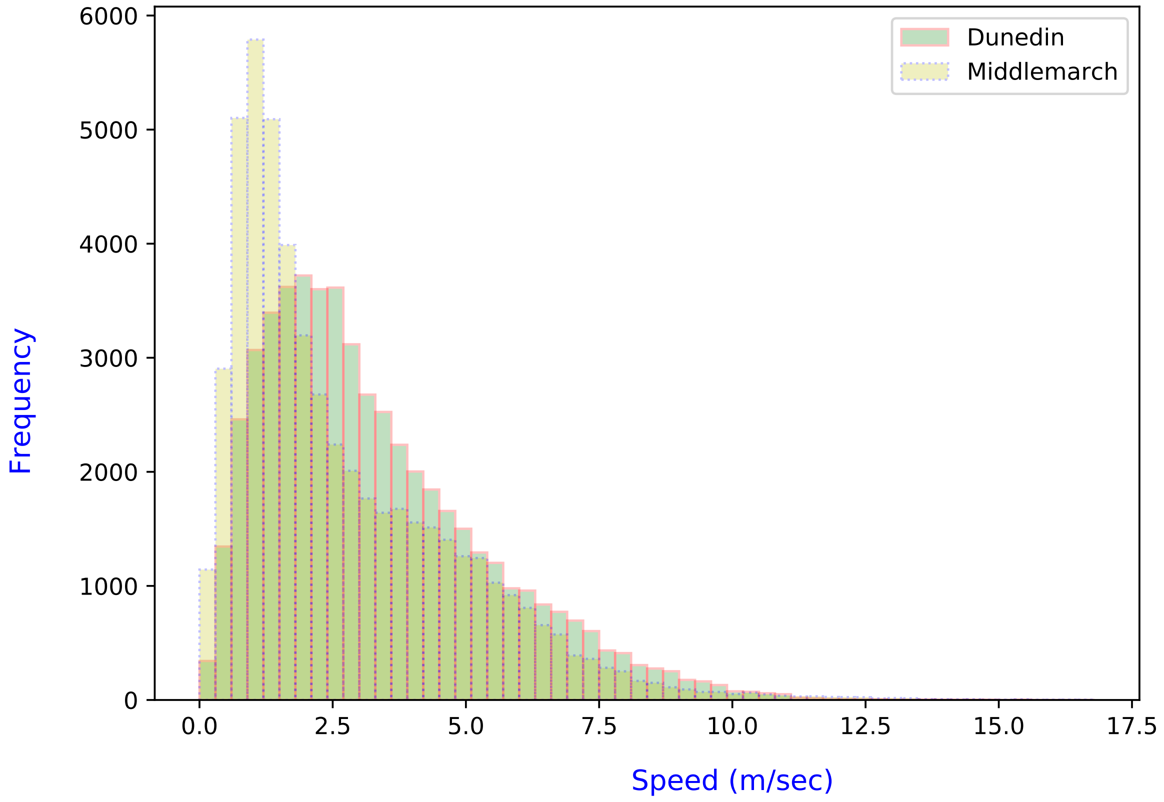


Figure 3.6: Wind-speed Histogram

and 6 kW wind turbines, to meet the needs of private annual electricity consumption while integrating with the wind-power available in its area (Olsen & Preus, 2015). Figure 1.3a and 1.3b show pictures of two of the selected types of wind turbine. Theoretically, each turbine has its specified power generating curve depending on various wind-speed and usually has been provided by the manufacturer with experimental data. For our experiments, these data were gathered from the manufacturers (Bergey, 2012; Suneco, 2015) and have been converted into the fitness function. Figure 3.8 shows the fitted power output curves in terms of various wind speed according to the three types of commercialised wind turbines.

III. Wind-power estimation by wind-speed:
In terms of the four selected urban areas, the possible generated wind energy can be estimated by using the wind-speed at the turbines, respectively. Consequently, this study establishes the hourly wind power data of the three types of turbines in four urban areas. Figure 3.9 shows the profiles of hourly wind power in one randomly selected week according to the four urban areas and

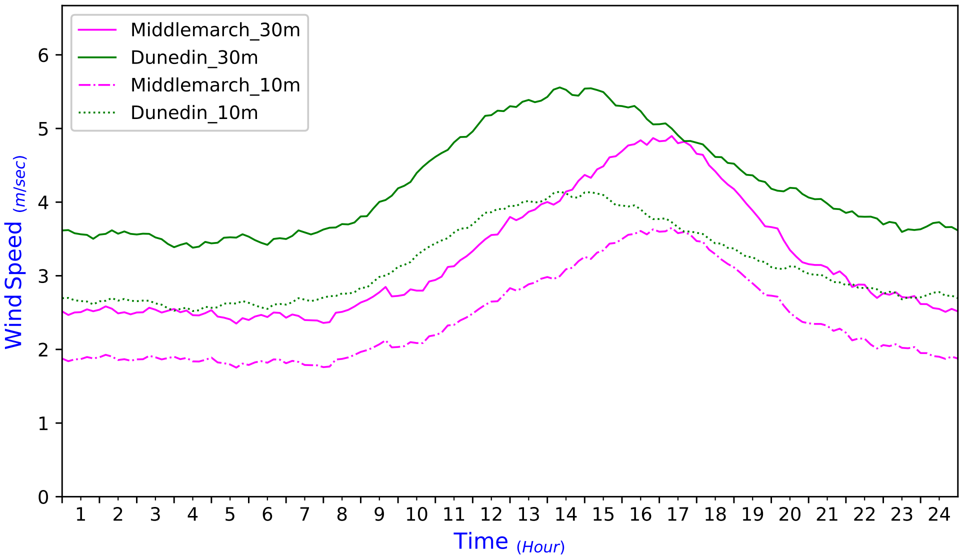


Figure 3.7: Wind-speed profile of measured (at 10 m) and estimated (at 30 m).

the three turbines. Table 3.1 shows the estimation of the total wind-power generated in one year for each of the four urban areas and three turbine types.

Table 3.1: Annual wind-power being generated by different turbines.

Turbine Type	Urban Area			
	Dunedin	Middlemarch	Ranfurly	Balclutha
Bergey 1_kW	1362.59	897.01	968.17	1016.06
ECO 3_kW	3515.23	2448.68	2486.68	2607.00
Bergey 6_kW	7271.90	4832.17	5139.52	5433.97

Unit:kWh

Solar Energy

The most popular method for harvesting solar radiation is to install the solar PhotoVoltaic (PV) panel (Branker et al., 2011). Accordingly, for estimating the achievable solar energy that could be produced, one needs to evaluate the power output of the commercialised PV panel in terms of the solar radiation. The approaches of gathering solar energy for prosumers to utilise are listed below:

I. Solar data collection and analysis:
Same as for the wind-speed data, our radiation source was received from the NIWA NZ for the same chosen urban areas. Figure 3.10 and 3.11 depict the different solar profiles of the four urban areas and the characteristics of the

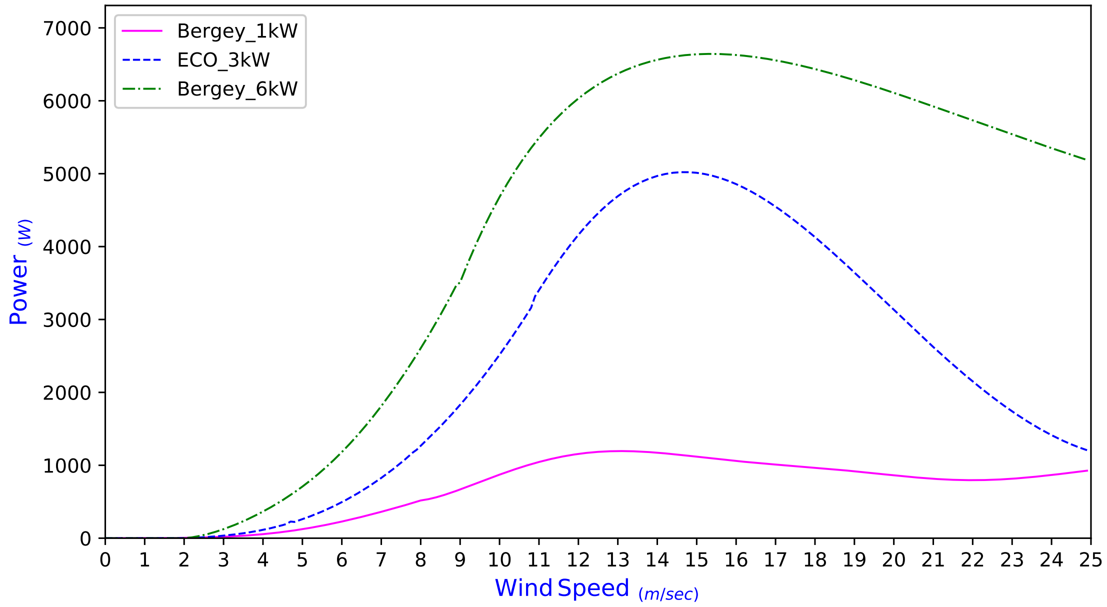


Figure 3.8: Power output by 3 types of commercialised Wind-Turbines

10-minute average profile calculated in one year. Again, the different solar extents encourage these regional areas to share energy among them. Comparable to the wind energy estimation, since the hourly radiation has been given for every city, we can compute the annual sunshine hours and the total solar radiation accordingly, as shown in Table 3.2.

Table 3.2: Annual sunshine hours and total solar radiation.

	Urban Area			
	Dunedin	Middlemarch	Ranfurly	Balclutha
Sunshine hours	4145.00	4222.33	4263.67	4223.67
Total radiation (kWh/m^2)	1209.31	1345.41	1561.32	1285.65

II. PV panel types and fitness:

Comparable to wind turbines, there are many commercialised types of PV panel available for small scale generation in terms of miscellaneous private power consumers. However, unlike the wind generators, the PV prosumer has more flexibility to choose the appropriate panel size, which is based on the modularised panel unit, to meet their needs. Therefore, instead of assigning the turbine types, we estimate the area (m^2) of the PV panel to be installed based on the number of PV units the prosumer required. The required number of PV units is evaluated according to the locations and the annual electrical

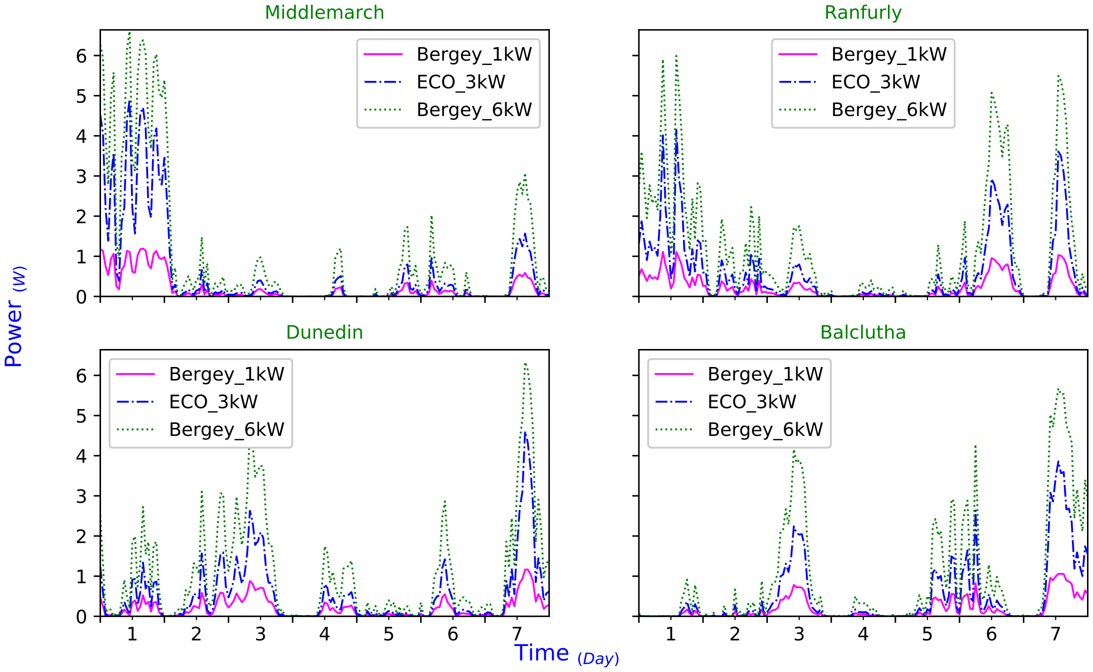


Figure 3.9: Hourly wind-power generated by types of turbines in one week.

consumption for each prosumer. Figure 1.2 shows a typical roof with solar-panel installed.

III. Solar-power estimation by radiation:
Again for our estimation, we choose a global formula to estimate the electricity generated for the output of a photovoltaic system (Photovoltaic-software, 2018). The equation of the formula is:

$$E = A \times r \times H \times PR, \tag{3.19}$$

where E = Energy (kWh), A = Total solar panel Area (m^2), r = solar panel yield or efficiency (%), H = Annual average solar radiation on tilted panels (shadings not included), and PR = Performance ratio, coefficient for losses (range between 0.5 and 0.9, default value = 0.75). Furthermore, to standardise the computation concerning each solar prosumer's number of installed panels, we select a commercialised PV panel[5] to be the unit. Thus, the prosumer only needs to calculate the number of the panels based on the power consumption and the annual radiation of the city where it is located. The parameters used for the estimation are $A = 1.956m^2$, $r = 0.177$, H may refer to Table 3.2

[5] https://www.solartradesales.co.uk/Cache/Downloads/Data-Sheet-SunEdison-Mono-Silver-SE-R335BMC-39.pdf Accessed: 31 Dec. 2019.

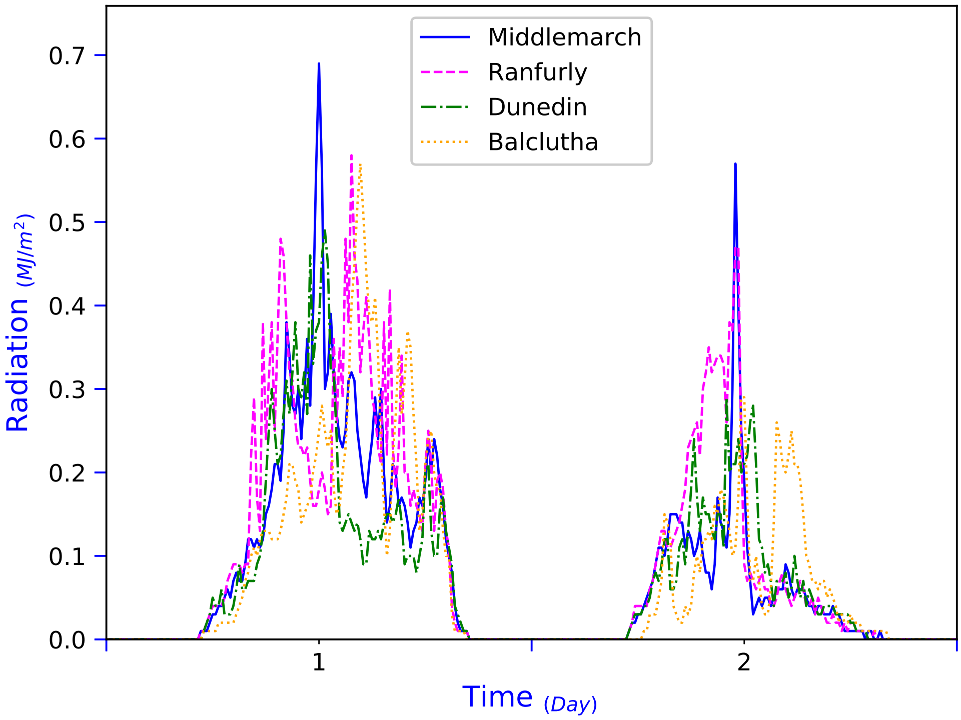


Figure 3.10: Profiles of 10-minute solar radiation in two days.

Table 3.3: Annual solar radiation and unit generated power.

	Urban Area			
	Dunedin	Middlemarch	Ranfurly	Balclutha
Radiation (kWh/m^2)	1209.31	1345.41	1561.32	1285.65
Generated power (kWh)	313.85	349.18	405.21	333.67

and PR is given by

$$PR = 1 - I_l - T_l - DC_l - AC_l - S_r - L_w - L_d - L_o, \tag{3.20}$$

where I_l (Inverter losses) = 0.08, T_l (Temperature losses) = 0.08, DC_l (DC cables losses) = 0.02, AC_l (AC cables losses) = 0.02, S_r (Rate of Shading) = 0.03, L_w (Losses at weak radiation) = 0.03, L_d (Losses due to dust, snow...) = 0.02, L_o (Other Losses) = 0.0. Consequently, we get the annual generated power by PV in the urban areas, as shown in Table 3.3.

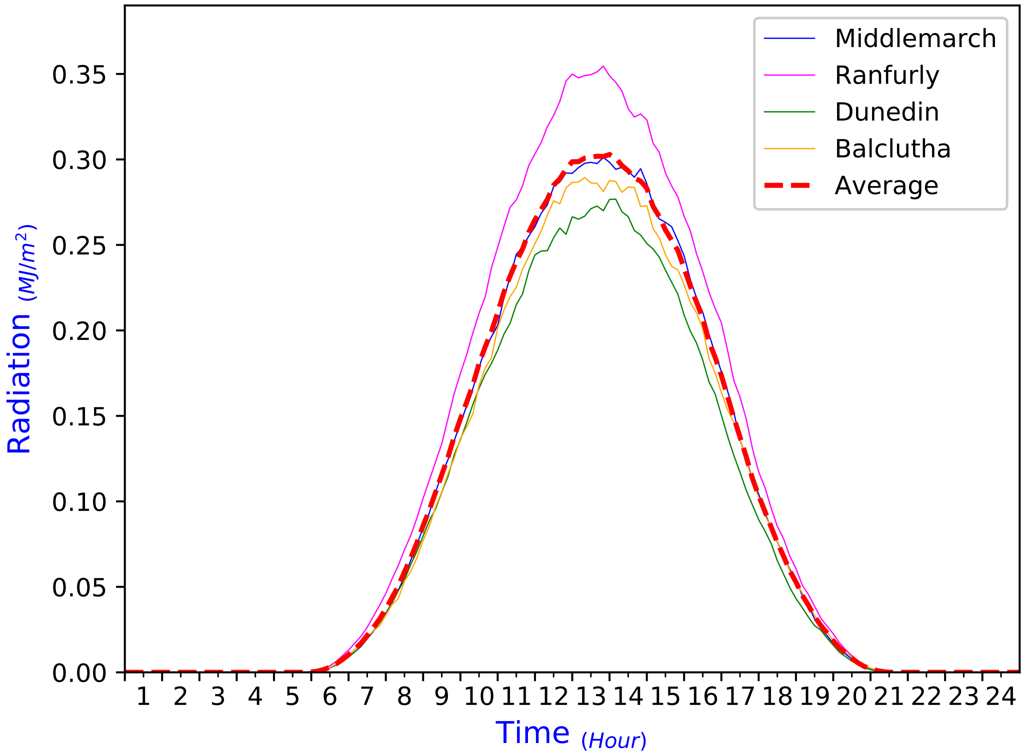


Figure 3.11: 10-minute average radiation profiles calculated in one day.

3.4.3 Modelling of Prosumer Agents

Following the treatment above, to couple the generated RE with electrical power expenditure for forming the dataset of prosumers, we have matched hourly consumed and generated electricity of the 240 households in the year 2008, respectively. The approaches of creating the data of prosumer agents are listed below:

I. Partitions of Prosumers:
We have a set of 240 hourly consumption data, and we have divided the data into two 120-agent groups as the wind and solar energy prosumers separately via random selection. Again we randomly split each 120-prosumer set into four equal partitions in terms of the urban areas.

II. Wind-turbines assignment for the 120 prosumers:
For every prosumer, the type of wind turbine is assigned according to its entire power consumption balanced to the total generated wind power of the turbine types at the specific location. Thus, the type is chosen so that its output is the closest one to meet the consumption. For example, for an agent a_1 with 3280 (kWh) consumption in Dunedin city, from Table 3.1 we identify that

the proper wind-turbine type is ECO 3_kW. Accordingly, we have computed the hourly power conditions for every wind-turbine agent based on the hourly power profile, wind-speed and the assigned turbine type. Consequently, we have obtained a set of 120 prosumers with power conditions data. The condition of each agent at a given time indicates whether it is in the state of excess or shortage, the related power amount and the located city.

III. PV panel number estimation for the other 120 prosumers:
For each prosumer, the number of panels, denoted by n, is estimated in terms of its full generated solar power by the n panels at the specific location matched to total power consumption. Hence, the number is decided so that its output is the closest one to match the consumption. For instance, for an agent a_2 with 2560 (kWh) consumption in the town of Ranfurly, from Table 3.3 we compute that the proper panel number is 6. Subsequently, we have calculated the hourly power conditions for every solar agent based on the hourly power profile, solar radiation and the number of panels. As a result, we have obtained the other set of 120 prosumers with power conditions data accordingly.

IV. Prosumer agents for experiments:
For all the CSG experiments in this thesis, every set of power conditions for the n agents, denoted by $S = \{a_1, ..., a_n\}$, is randomly selected within the hourly 240-agents data in the year 2008. The only requirement, as mentioned in Subsection 3.2.2, is that any feasible coalition must have a net power surplus, i.e. $\{a_1, ..., a_n\} \geq 0$.

Figure 3.12 shows the integration process of obtaining the hourly net-power profiles of one prosumer with Turbine (left) and one prosumer with solar PV (right) in 24 hours. Figure 3.13 shows the integration process of obtaining the hourly net-power profiles of 40 randomly selected prosumers in 24 hours. The upper part of Figure 3.13 shows the power consumption profiles. The middle part in the figure shows the power generated by wind turbines and solar PV panels. Consequently, we can get the hourly net-power profiles by adding these two parts' data. The resulting net-profiles are shown in the lower part of the figure. Following these hourly results for all agents, the generated data of conditions for every agent, i.e. the status of surplus or deficit and its electricity amount respectively, will be used in the following chapters.

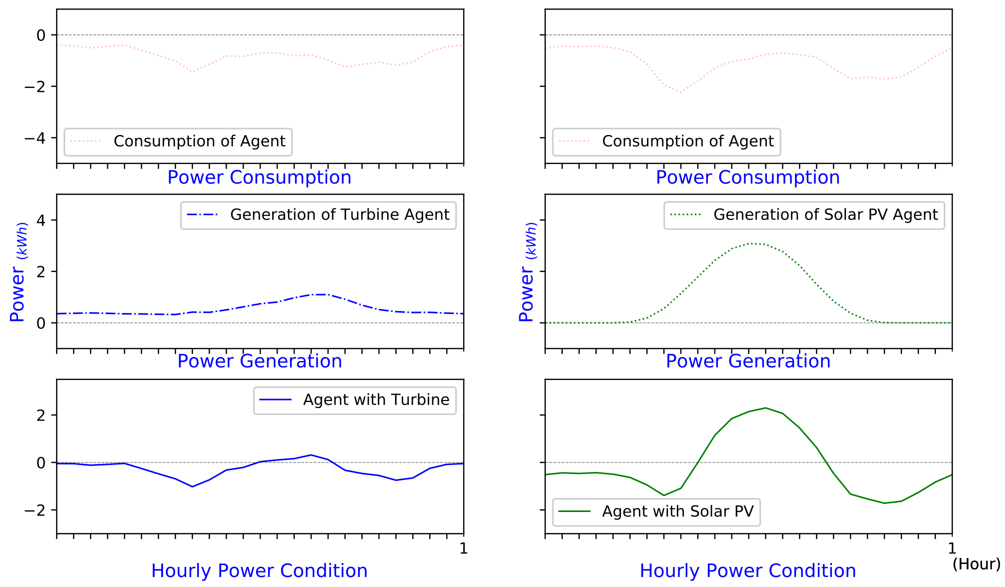


Figure 3.12: Hourly net-power profiles of prosumer (left: Turbine, right: Solar PV)

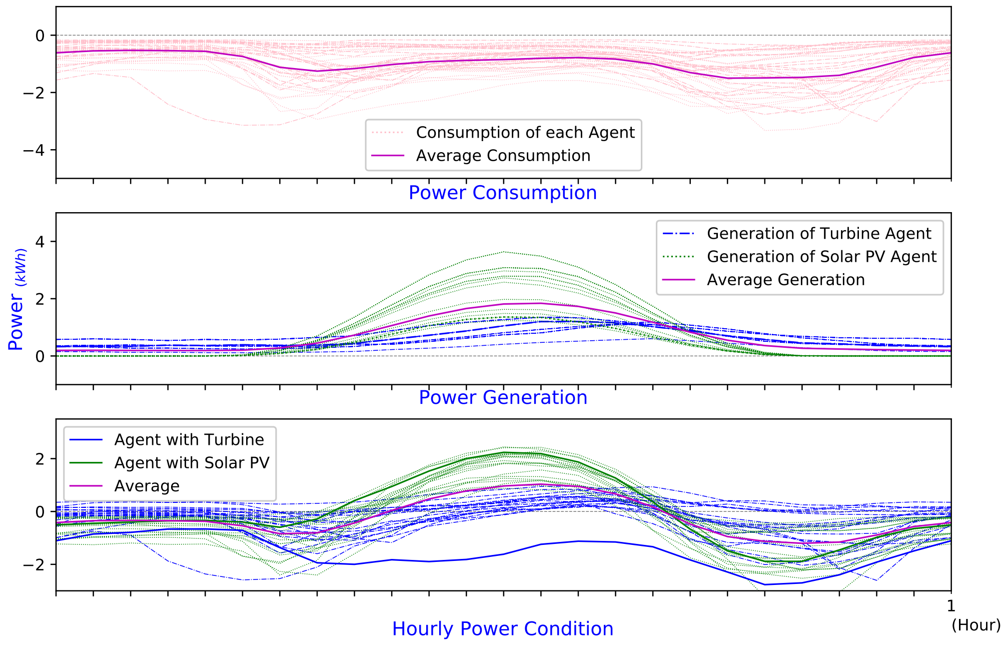


Figure 3.13: Hourly net-power profiles of 40 prosumers in 24 hours.

3.5 Outline of Experiment Processes

I. Binary scheme specification:

   In order to find the optimal Coalition Structure (CS), the ES and DP algorithms can directly compute the values of any CS. However, for our selected SO algorithms, we need to create binary encoding schemes to map the CSG to the corresponding bit-length, such as the connection scheme described in Chapter 4 and the Set-ID scheme applied in the next two chapters.

II. SO algorithms selection and improvement:

   To solve the binary optimisation problems, we have adopted some popular binary SO algorithms, such as GA, Binary PSO and PBIL. Furthermore, when applying the SO algorithm, we have attempted to improve efficiency and accuracy by using enhanced algorithm, such as algorithms of the PBIL-MW and the Hierarchical PBIL-MW.

III. Parameters and sample size selection:

   In order to apply the SO algorithms, we have done some preliminary analyses to choose moderate parameters (such as crossover and mutation-probability), initial probabilities (i.e. thresholds) and population size before we proceed with the experiments. Note that we have used the same set of initial probabilities and population size in all adopted SO algorithms to compare the results.

IV. Experiment and comparison:

   In the experiments in the following three chapters, the deterministic search algorithms (i.e. ES and DP) only need to compute once. Furthermore, while applying the SO algorithms, we have used the same population size and initial probabilities for distinct observations and to obtain mixed results.

3.6 Summary

In this chapter, we have presented our co-operative model and incentives along with some literature reviews, explained the process of designing the proposed CSG problems, described the scenarios from a local scale to a regional scale, and demonstrated the process of obtaining the power conditions data. As the outline in Section 3.5, all these sections will provide the fundamental models, alternative scenarios, and underlying hourly RE data for our further exploration in the following experiments.

Chapter 4

PBIL-based solutions for CSG

In the previous chapter, we have proposed the coalitional model of energy-sharing agents in smart grids. Consequently, we will start the implementation of the model within a local scale in this chapter. Firstly, we explain the form and constraints of the co-operative model in Section 4.1. Later in Section 4.2, several algorithms, including our proposed approach PBIL-MW, for solving the CSG problems are introduced. The experiments and results are shown in Section 4.3. In the experiments, four chosen algorithms are examined and compared using several sets of agents, which are randomly selected from the data obtained in Section 3.4. In the last section, a summary and conclusion of this chapter are provided.

In general, our proposed PBIL-MW could obtain the exact solutions the same as the DS methods while outperforming DS in the efficiency. The result of this chapter demonstrates notable performance enhancement over the traditional methods. Some components of this chapter are extensions of our paper (Lee et al., 2017).

4.1 Coalition Model in Local Smart Grids

By following our proposed coalitional model mentioned in Section 3.2, we know that our model is a CSG problem. For instance, let us assume an alliance of 4 agents, from Equation 2.9 and Table 2.1 we know that there are 15 possible CS. Therefore, the goal of the alliance is to search for an optimal CS, which has the maximal fitness value.

In this chapter, we will explore the feasibility of an alliance to form exclusive coalitions for maximising the profit. Therefore, we start the experiment by employing the alliance in the scenario of a local smart grid, which was mentioned in Subsection 3.3.1. In this scenario, we assume that the agents of the prosumers are located within a distribution grid, e.g. a community or district, which could provide power-sharing directly by the distribution network.

4.1.1 Constraints of Coalition

For consistency, this chapter adopts the constraints of coalition mentioned in Subsection 3.2.2. Thus, we consider only cases with a net power deficit for the alliance where a grand coalition is unfeasible, and a game of CSG should be constructed and needs to resolve accordingly.

For every period (e.g. per hour), any agent a_i with a surplus can share its energy with deficit ones within the team. The goal of that team is to optimize the maximum benefit by forming a coalition. Also, for the stability of a smart grid, we demand that every feasible sub-coalition must have a net power surplus.

For example, a_1 and a_2 each has 1.2 and 0.7 kWh surplus respectively, but a_3 has a deficit of 0.9 kWh. With a net joint surplus, a_1 and a_3 can form a feasible coalition $\{a_1, a_3\}$. On the contrary, a coalition such as $\{a_2, a_3\}$ is unfeasible.

4.1.2 Coalition Evaluation

Same as the constraints of a coalition, the coalition evaluation follows the one in Subsection 3.2.2. Furthermore, this chapter is focused on a local coalition, such as within a community. Therefore, from Equation 3.14 we know that the transmission cost and power loss are trivial, thus the loss is ignored in this scenario. For convenient reference, the fitness function for coalition evaluation is summarised as below.

Let $\mathcal{S}$ denote an alliance of n co-operative agents, and C_k represent a coalition which is a subset of $\mathcal{S}$. A coalition structure CS is a partition of coalitions, where $\mathrm{CS} = \{C_1, C_2, \cdots, C_m\}$ such that $\cup_k C_k = \mathcal{S},\ C_k \neq \emptyset$ and $C_k \cap C'_k = \emptyset,$ if $k \neq k'$. The goal of our model is the optimisation problem of searching for an optimal CS, denoted by CS*, over $\mathcal{S}$ whose value is optimal. Mathematically, a coalition structure CS* is said to be globally optimal, if CS* gives the maximal overall characteristic value:

$$\mathrm{CS}^* = \arg \max_{\mathrm{CS} \in \mathcal{F}(\mathrm{CS})} v(\mathrm{CS}), \tag{4.1}$$

where $\mathcal{F}(\mathrm{CS})$ denotes the whole collection of all possible CS into which the set $\mathcal{S}$ of n agents can be formed. Thus the size of $\mathcal{F}(\mathrm{CS})$ for $\mathcal{S}$ is denoted by $|\mathcal{F}(\mathrm{CS})|$. From Equation 2.6 we know that $|\mathcal{F}(\mathrm{CS})|$ is the Bell's number $B(n)$.

Furthermore, let us assume the prices for any prosumer to purchase from or sell to the utility are denoted by P_b and P_s respectively. The values of prices used in the experiment are $P_b = 40$ and $P_s = 20$ $(¢/kWh)$. Thus we know that P_b is higher than P_s, such that $P_b > P_s$. For convenience, let $P_r = (P_b - P_s) = 20$ $(¢/kWh)$ represent the price difference by way of trading with the power utility. Therefore, in our study, $v(C_{(k)})$ is the fitness function of $C_{(k)}$ given by

$$v(C_{(k)}) = \begin{cases} 0 & \text{if } k = 1, \\ Q_d \times P_r & \text{if } k > 1 \text{ and } Q(C_{(k)}) \geq 0, \\ -9999 & \text{otherwise,} \end{cases} \tag{4.2}$$

where k denotes the size of coalition $C_{(k)}$, Q_d (≥ 0) is the total power need for deficit agents in the $C_{(k)}$, and $Q(C_{(k)})$ is the net surplus within the $C_{(k)}$ accordingly. Furthermore, for giving penalty to an unfeasible coalition, we let $v(C_{(k)}) = -9999$.

4.2 Algorithmic Solutions

4.2.1 ES

As discussed in Section 2.3.1, the ES systematically enumerate all achievable candidates of CS and discover the optimal CS* with the best fitness throughout the entire solution space. As we know that an ES is not an efficient way to solve the CSG problem with a large number of agents. However, in the preliminary experiment of our proposed model, we need to compare the accuracy and efficiency between stochastic algorithms and the direct search method. Therefore, for obtaining the exact solution in the experiment, this chapter will use ES to find the optimal CS and set its value to be the ground truth as the algorithmic paradigm in results comparison.

To carry out the ES computation, let S denote the set, i.e. the alliance, of n agents, and CS$_i$ denote one possible CS in whole CS$^{(n)}$ space of S. In addition, let us represent the fitness values of the current CS$_i$ by $v(\mathrm{CS}_i)$, and denote the best CS and its fitness so far by CS' and $v'(\mathrm{CS})$ respectively. The procedure of ES is executed by systematically searching for the optimal CS, denoted by CS^*, which has the optimal fitness $v^*(\mathrm{CS})$. The pseudocode of ES is shown in **Algorithm 1**:

Algorithm 1 Exhaustive Search

```
Evaluate the fitness of the grand coalition, $v(S)$, let $\mathrm{CS}' = S$ and $v'(\mathrm{CS}) = v(S)$
repeat
    Systematically select one achievable CS$_i$ in the $\mathrm{CS}_{(n)}$
    Compute fitness $v(\mathrm{CS}_i)$
    IF $v(\mathrm{CS}_i) > v'(\mathrm{CS})$, set $\mathrm{CS}' = \mathrm{CS}_i$ and $v'(\mathrm{CS}) = v(\mathrm{CS}_i)$
until every achievable CS$_i$ is evaluated
Update $\mathrm{CS}^* = \mathrm{CS}'$ and $v^*(\mathrm{CS}) = v'(\mathrm{CS})$
Output $v^*(\mathrm{CS})$ and CS*
```

4.2.2 Genotype Encoding for Stochastic Optimisation

Before we can enable the SO algorithms to proceed in a CSG problem correctly, the first procedure is to represent the CS by a length of a binary vector.

For an alliance of n co-operative agents, let a_i denote an arbitrary agent and $C_{i,j}$ denote the connection between a_i and a_j. The graph of the connections between agents is shown in Figure 4.1. Numerically, when a_i and a_j are connected then $C_{i,j} = 1$, otherwise $C_{i,j} = 0$.

Let l denote the total length of connection vector required for indicating whether any two agents are connected or not. Hence,that l is given by:

$$l = (n-1) + (n-2) + ... + 2 + 1 = \frac{n \times (n-1)}{2}. \tag{4.3}$$

Furthermore, let s denote the number of samples, namely population, used for SO algorithms. Consequently, the binary matrix needs for computing SO algorithm is a $s \times l$ matrix. For further details, the reader may refer to Example 4.2.

4.2.3 GA

The program and encoding scheme of GA is thoroughly explained in Subsection 2.3.2. Since there are many variants of GA in real applications, this study follows the method of Marsland (2015). Again, let s denote the number of samples and l denotes the total length of connections. The parameters used here are crossover=uniform, mutation-probability=$1/l$. The running process, i.e. pseudocode, of GA, is shown in **Algorithm 2**:

Algorithm 2 Genetic Algorithm

```
Generate an initial population of $s \times l$ random samples
Filter the population by a threshold vector to form $s$ binary samples
Compute CS$_i$ from every sample $s_i$ of the population
Compute and rank fitness $v(\text{CS}_i)$ of every sample $s_i$
Get best $v(\text{CS}_i)$, let $v^*(\text{CS})$=$v(\text{CS}_i)$ and CS$^* = $ CS$_i$
repeat
    Select $s/2$ pairs of parents by the method mentioned in Subsection 2.3.2
    Proceed Crossover and Mutation according to Subsection 2.3.2
    Breed a new generation of population from the process above.
    Compute CS$_i$ from every sample $s_i$ of the new population
    Compute and rank fitness $v(\text{CS}_i)$ of every sample $s_i$
    Get best $v(\text{CS}_i)$, let $v'(\text{CS}) = v(\text{CS}_i)$ and CS$' = $ CS$_i$
    IF $v'(\text{CS}) > v^*(\text{CS})$, update CS$^* = $ CS$'$ and $v^*(\text{CS}) = v'(\text{CS})$
until termination condition has been met
Output $v^*(\text{CS})$ and CS$^*$
```

Example 4.1 Crossover and Mutation

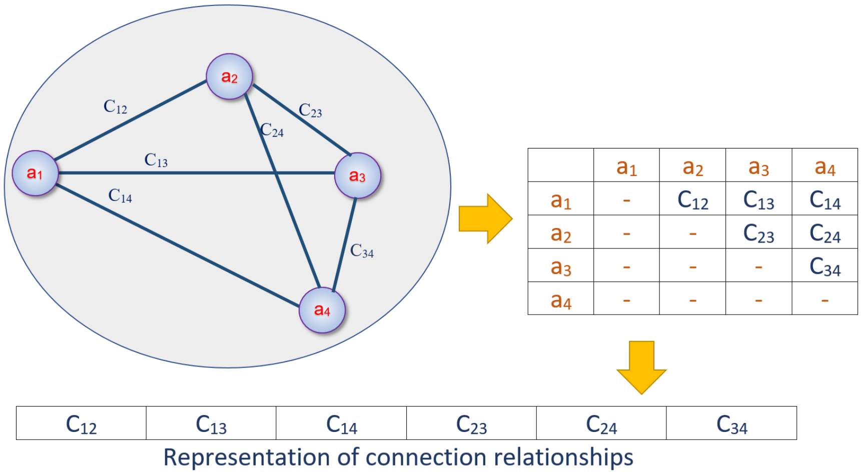
(a) Crossover:

Suppose we have two samples with 6 digits of code, say S_1 and S_2. Let's randomly choose 3 to be a fixed crossover point; then the new sample S_1' will keep the first 3 digits of S_1 and substitutes the rest with S_2, and S_2' as well. The process is shown in Figure 2.4 (a).

(b) Mutation:
Let S be the chosen sample with 6 digits of code for mutation. Again, if we randomly pick 3 as a mutation point, which value is 0, the new sample after mutation will be 1, as shown in Figure 2.4 (b).

Example 4.2 Demonstration of the first iteration of GA with the connection encoding

To simplify the demonstration, let us assume a graph of $n\!=\!4$ nodes (i.e. agents). For easy explanation, the connection index mapping of the four agents is shown in Figure 4.1.



	a1	a2	a3	a4
a1	-	C12	C13	C14
a2	-	-	C23	C24
a3	-	-	-	C34
a4	-	-	-	-

C12	C13	C14	C23	C24	C34

Representation of connection relationships

Figure 4.1: The vector represents the connections of the four agents.

While considering all possible connections, from the figure we know that the total number of connections is given by $l\!=\!(n-1)+...+1\!=\!n(n\!-\!1)/2 = 4(4\!-\!1)/2 = 6$. Therefore, l is the element length of a sample, and represents any candidate gene's binary vector in a GA population. Hence, in this example, a single sample vector, denoted by s_i, will have a length of six binary bits, such that $s_i\!=\![c_{12}, c_{13}, c_{14}, c_{23}, c_{24}, c_{34}]$, where c_{ij} denotes the connection condition, thus c_{ij} =“1” or “0”, and represents whether any two agents a_i and a_j are connected or not accordingly.

Furthermore, let $P\!=\![p_1, ..., p_l]$, $0 \leq p_j \leq 1$, denote the probability vector for any two agents being connected, which is generally called the “threshold”. In advance, let m denote the given number of samples to represent the size of the population, and let S denote the population matrix which has the size of $m\times l$. Thus any $s_i \in S$, $_{1\leq i\leq m}$, represent the i^{th} row sample which has an l length bits as well.

In the GA computation, we need to obtain m samples with an l binary length to represent the first generation of the population. Firstly, let us generate an $X = m \times l$ matrix of random values, such that every element $x_{i,j} \in X$, $0 \leq x_{i,j} \leq 1$, represents any j^{th} element in the i^{th} row. Hence, if $x_{i,j} \leq p_j$, $x_{i,j}$ will be set to 1, otherwise $x_{i,j}$ will be set to 0. Thus, by comparing every element $x_{i,j}$ in the i^{th} row with p_j, we get the binary vector of the i^{th} row that represents the connection structure of that i^{th} sample.

For instance, let $P = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$ denote the probability vector. Firstly, we need to generate four vectors ($x_{i,\ 1 \leq i \leq 4}$) of random numbers as shown in Table 4.1. After each x_i vector is compared to the threshold P, we get binary vector of sample s_i.

Table 4.1: Random number, binary vector and CS of the Four Samples.

Sample	Random Number (x_i)	Binary Vector (s_i)	CS (C_i)
No. 1	[0.22, 0.87, 0.21, 0.92, 0.49, 0.61]	[1, 0, 1, 0, 1, 0]	$\{\{a_1, a_2, a_4\}, \{a_3\}\}$
No. 2	[0.77, 0.52, 0.30, 0.19, 0.08, 0.74]	[0, 0, 1, 1, 1, 0]	$\{\{a_1, a_2, a_3, a_4\}\}$
No. 3	[0.44, 0.16, 0.88, 0.27, 0.41, 0.30]	[1, 1, 0, 1, 1, 1]	$\{\{a_1, a_2, a_3, a_4\}\}$
No. 4	[0.63, 0.58, 0.60, 0.27, 0.28, 0.25]	[0, 0, 0, 1, 1, 1]	$\{\{a_2, a_3, a_4\}, \{a_1\}\}$

*Initial Threshold $P^0 = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$

Again, from Figure 4.1 and s_i we get the CS of each C_i. The first, third and fifth bits $(c_{12} = c_{14} = c_{24} = 1)$ of vector $s_1 = [1, 0, 1, 0, 1, 0]$ mean that there are connections between (a_1, a_2), (a_1, a_4) and (a_2, a_4). The remaining three bits (=0) show that there are no connections for the three pairs of agents (a_1, a_3), (a_2, a_3) and (a_3, a_4) separately. Hence s_1 represents the CS of $C_1 = \{\{a_1, a_2, a_4\}, \{a_3\}\}$. Consequently, s_2, s_3 and s_4 represent CS of $C_2 = C_3 = \{\{a_1, a_2, a_3, a_4\}\}$ and $C_4 = \{\{a_2, a_3, a_4\}, \{a_1\}\}$ accordingly. The real connections maps of the four samples are shown in Figure 4.2.

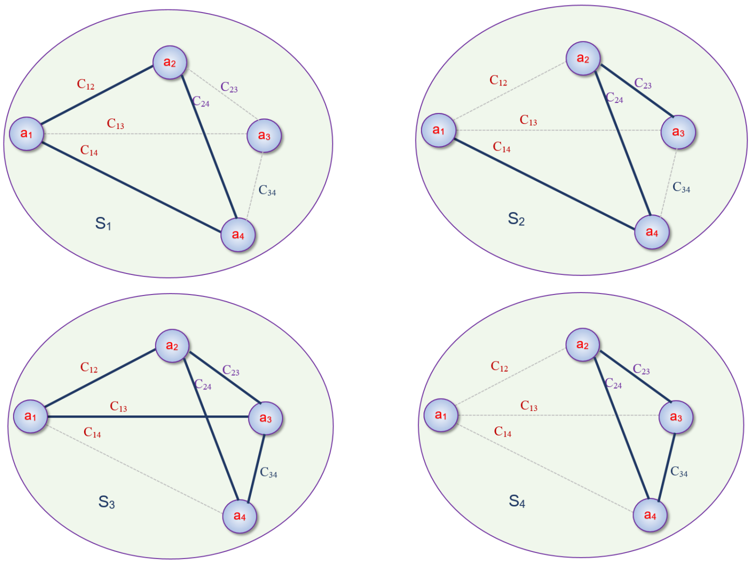


Figure 4.2: Connections Map of the Four Samples.

From the above initial population and the corresponding fitness values show in Table 2.3, we get $v(C_1) = 120 + 25 = 145,\ v(C_2) = v(C_3) = 140$ and $v(C_4) = 115 + 30 = 145$. By following the GA process, the evolution steps keep iterating from then on. For example, a second population is generated after the following steps:

- Crossover: Let's select s_1 and s_4 for the first pair of parents. Following example 4.1, after the crossover, the offspring will be $s'_1 = [1, 0, 1, 1, 1, 1]$ and $s'_2 = [0, 0, 0, 0, 1, 0]$. Their represented CS are $C'_1 = \{\{a_1, a_2, a_3, a_4\}\}$ and $C'_2 = \{\{a_2, a_4\}, \{a_1\}, \{a_3\}\}$, and respectively have the fitness values $v(C'_1) = 140$ and $v(C'_2) = 70 + 30 + 25 = 125$. Thus, by selecting the best two randomly, we keep s_1 and s_4 to be the new pair after crossover. Again, by repeating the crossover, let's select s_1 and s_2 for the second pair of parents. (Note that, since the fitness of $v(C_2) = v(C_3) = 140$ is lower than others, therefore, they are less likely to be selected.) After repeating the crossover for the second pair, we get $s'_3 = [1, 0, 1, 1, 1, 0]$ and $s'_4 = [0, 0, 1, 0, 1, 0]$. Their represented CS are $C'_3 = \{\{a_1, a_2, a_3, a_4\}\}$ and $C'_4 = \{\{a_1, a_2, a_4\}, \{a_3\}\}$, and each have the fitness values $v(C'_3) = 140$, and $v(C'_4) = 120 + 25 = 145$. Again, by selecting the best two randomly, we get s_1 and s'_4 to be the new pair after crossover. Consequently, the new population, s_1, s_4, s_1, s'_4, are created after the two crossover processes.

- Mutation: Following the mutation process mentioned above, let us select s_1 to be the chosen sample. After mutation, the new sample $s''_1 = [1, 0, 0, 0, 1, 0]$ represents $C''_1 = \{\{a_1, a_2, a_4\}, \{a_3\}\}$ and $v(C''_1) = 120 + 25 = 145$.

As a consequence, we get the new population, $[s_1, s_4, s''_1, s'_4]$, after the first iteration.

4.2.4 PBIL

For demonstrating the potential ability of our proposed algorithm, the procedure of the original PBIL is described as follows.

Let **P** denote the probability vector with real values in the range [0, 1] for each component P_j. **P**'s length, denoted by l, which is equal to that of the genotype and is given by Equation 4.3. In PBIL, every component P_j represents the threshold of gaining a value 1 in the j^{th} component of the genotype. Thus we know that **P** $= (P_1, P_2, \cdots, P_l)$.

Initially, the probability vector, denoted by $\mathbf{P}^{(0)}$, is simply set with every P_j=0.5. For each iteration, a s rows by l columns matrix R with entries of random numbers inside [0, 1] is drawn, where s is the size of the population's samples. Let **G** denote the population matrix, and $\mathbf{G}_i$, where $1 \leq i \leq s$, represent its i^{th} genotype, corresponding to a CS. Since **P** and R are given, every entry G_{ij} in $\mathbf{G}_i$ is assigned as 1 if $R_{ij} < P_j$; otherwise, $G_{ij} = 0$. Let V_i denote the fitness value of $\mathbf{G}_i$, which is evaluated by Equation 4.2 with G_{ij} given, such that $V_i = \sum_{C_k \in CS_{(i)}} v(C_k)$.

After that, $P_j^{(t+1)}$ is assessed upon the fitness of maximum V_{max} and minimum V_{min} with a pair of parameters named learning rate γ and negative learning rate ϵ, respectively. Since V_{max} and V_{min} are obtained from the current step, there are two updating rules for assigning the next $\mathbf{P}^{(t+1)}$ vector given by

$$P_j^{(t+1)} = \begin{cases} (1-\gamma)P_j^{(t)} + \gamma\,\acute{G}_j & \text{if } \acute{G}_j = \H{G}_j\ , \\ (1-\gamma)P_j^{(t)} + \epsilon(\acute{G}_j - P_j^{(t)}) + \gamma\,\acute{G}_j & \text{if } \acute{G}_j \neq \H{G}_j\ , \end{cases} \tag{4.4}$$

where $\acute{\mathbf{G}}$ and $\H{\mathbf{G}}$ represent the vector $\mathbf{G}_i$ when $V_i = V_{max}$ or $V_i = V_{min}$, respectively.

After successive $\mathbf{P}^{(t+1)}$-updating iterations, the procedure can be stopped by a given condition. This condition can be a maximum number of iterations has been reached, or an adequate fitness value has been found from the population. The pseudocode of PBIL is shown in **Algorithm 3**.

Example 4.3 Demonstration of steps for PBIL threshold vector updating:

Following Example 4.2, the first procedure of PBIL is the same as GA until the first population is generated. For instance, let $P = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$ denote the probability vector. Firstly, we need to generate four vectors ($x_{i,\ 1 \leq i \leq 4}$) of random numbers as shown in Table 4.2. After each x_i vector is compared to the threshold P, we get binary vector of sample s_i as well.

Algorithm 3 PBIL

```
Initialize probability vector $\mathbf{P}^{(0)}$
Set CS$^*$ $= \emptyset$, $v^*$(CS)$= 0$
repeat
    Generate a population $\mathbf{G}_s$ from $\mathbf{P}^{(t)}$
    Compute CS$_i$ from every sample $\mathbf{G}_i$ of the new population
    Evaluate the fitness $V_i$ of each member $\mathbf{G}_i$ for all $i \in n$
    Find $\acute{\mathbf{G}}$, $\text{\textbf{\H{G}}}$ and update $P_j^{(t+1)}$ according to Equation 4.4
    IF $V_{max} > v^*(\mathrm{CS})$, update CS$^*$ $=$ $\acute{\mathbf{G}}$ and $v^*(\mathrm{CS}) = V_{max}$
until termination condition has been met
Output $v^*(\mathrm{CS})$ and CS$^*$
```

Table 4.2: Random number, binary vector and CS of the Four Samples.

Sample	Random Number (x_i)	Binary Vector (s_i)	CS (C_i)
No. 1	[0.08, 0.78, 0.44, 0.72, 0.98, 0.54]	[1, 0, 1, 0, 0, 0]	$\{\{a_1, a_2, a_4\}, \{a_3\}\}$
No. 2	[0.51, 0.07, 0.27, 0.50, 0.68, 0.80]	[0, 1, 1, 1, 0, 0]	$\{\{a_1, a_2, a_3, a_4\}\}$
No. 3	[0.38, 0.07, 0.29, 0.91, 0.21, 0.45]	[1, 1, 1, 0, 1, 1]	$\{\{a_1, a_2, a_3, a_4\}\}$
No. 4	[0.93, 0.02, 0.60, 0.95, 0.23, 0.55]	[0, 1, 0, 0, 1, 0]	$\{\{a_1, a_3\}, \{a_2, a_4\}\}$

*Initial Threshold $P^0 = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$

Thus we have s_1 representing the CS of $C_1 = \{\{a_1, a_2, a_4\}, \{a_3\}\}$, likewise $s_2,\ s_3$ and s_4 representing CS of $C_2 = C_3 = \{\{a_1, a_2, a_3, a_4\}\}$ and $C_4 = \{\{a_1, a_3\}, \{a_2, a_4\}\}$ accordingly.

From Table 2.3 we get $v(C_1) = 120 + 25 = 145,\ v(C_2) = v(C_3) = 140$ and $v(C_4) = 60 + 70 = 130$ as well. By following the PBIL process, we select the genotype, $s_1 = [1, 0, 1, 0, 0, 0]$, of C_1 which has the best fitness to update the probability vector P^0. Let us set $\gamma = 0.02$ to be the learning rate for updating the vector and simply define the equation of updating the new threshold vector (P^{i+1}) by

$$P^{i+1} = (1 - \gamma)P^i + \gamma s^*, \tag{4.5}$$

where s^* represents the sample with best fitness value.

Hence, we can compute the new threshold vector $P^1 = (1 - 0.02)P^0 + 0.02 \times s^* = [0.98 \times 0.5, 0.98 \times 0.5, 0.98 \times 0.5, 0.98 \times 0.5, 0.98 \times 0.5, 0.98 \times 0.5] + 0.02 \times [1, 0, 1, 0, 0, 0] = [0.51, 0.49, 0.51, 0.49, 0.49, 0.49]$. After that, by following the procedure of Figure 2.6, PBIL will keep updating the threshold vector and renewing the best result according to the best fitness found in the iteration processes.

Finally, assume that the iterations have reached to 250 times, and the new threshold vector becomes $P^{250} = [0.78, 0.01, 0.87, 0.02, 0.75, 0.02]$. For the new P^{250}, let us generate another new set of 4 samples randomly. The result of random numbers (x_i), binary vectors (s_i) and CS (C_i) are shown in Table 4.3. The connections maps of the last four samples are shown in Figure 4.3.

Table 4.3: Random number, binary vector and CS of the Four Samples.

Sample	Random Number (x_i)	Binary Vector (s_i)	CS (C_i)
No. 1	[0.68, 0.75, 0.83, 0.85, 0.07, 0.85]	[1, 0, 1, 0, 1, 0]	$\{\{a_1, a_2, a_4\}, \{a_3\}\}$
No. 2	[0.36, 0.61, 0.30, 0.37, 0.26, 0.56]	[1, 0, 1, 0, 1, 0]	$\{\{a_1, a_2, a_4\}, \{a_3\}\}$
No. 3	[0.26, 0.74, 0.12, 0.57, 0.94, 0.00]	[1, 0, 1, 0, 0, 1]	$\{\{a_1, a_2, a_3, a_4\}\}$
No. 4	[0.54, 0.19, 0.81, 0.98, 0.85, 0.53]	[1, 0, 1, 0, 0, 0]	$\{\{a_1, a_2, a_4\}, \{a_3\}\}$

*Threshold $P^{250} = [0.78, 0.01, 0.87, 0.02, 0.75, 0.02]$

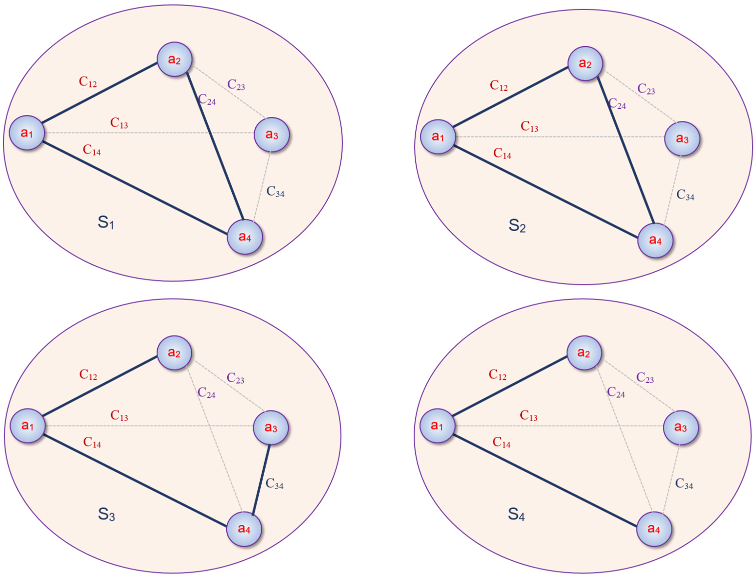


Figure 4.3: Connections Map of the Four Samples Obtained from PBIL.

During the PBIL optimisation process, the threshold has kept updating in each iteration as shown in Table 4.4. The result of this simple example reveals that the P^{250} will be much more likely to find the same global solution as the exact solution of ES.

Table 4.4: The Threshold of the four samples in some selected Iiterations.

No. of Iteration	Threshold Vector (P^i)
Initial	[0.50, 0.50, 0.50, 0.50, 0.50, 0.50]
1	[0.51, 0.49, 0.51, 0.49, 0.49, 0.49]
50	[0.53, 0.36, 0.59, 0.36, 0.56, 0.44]
100	[0.51, 0.17, 0.56, 0.33, 0.70, 0.34]
150	[0.64, 0.07, 0.73, 0.12, 0.77, 0.14]
200	[0.68, 0.03, 0.87, 0.04, 0.75, 0.06]
250	[0.78, 0.01, 0.87, 0.02, 0.75, 0.02]

Further to the original PBIL, there is a variation of PBIL also proposed by Baluja & Caruana (1995), namely the Top-k PBIL (T-PBIL). The updating process of the probability vector in T-PBIL includes all the genotypes of the k^{th} highest fitness rather than considering the best and worst. Thus, the method for updating the new $P^{(t+1)}$ can be given by

$$P_j^{(t+1)} = (1-\gamma)P_j^{(t)} + \frac{\gamma}{k}\sum_{m=1}^{k} G_m \ , \tag{4.6}$$

where G_m is the m-th genotype sorted by the fitness values. The pseudocode of T-PBIL is shown in **Algorithm 4**.

Algorithm 4 T-PBIL

```
Initialize probability vector $\mathbf{P}^{(0)}$
Set CS$^* = \emptyset$, $v^*$(CS)$= 0$
repeat
    Generate a population $\mathbf{G}_s$ from $\mathbf{P}^{(t)}$
    Compute CS$_i$ from every sample $\mathbf{G}_i$ of the new population
    Evaluate and rank the fitness $V_i$ of each member $\mathbf{G}_i$ for all $i \in n$
    Sort all $G_m$'s according to their fitness $V_m$
    Update $\mathbf{P}^{(t+1)}$ according to Equation 4.6
    IF $V_{max} > v^*(\mathrm{CS})$, update CS$^* = \acute{\mathbf{G}}$ and $v^*(\mathrm{CS}) = V_{max}$
until termination condition has been met
Output $v^*(\mathrm{CS})$ and CS$^*$
```

4.2.5 Top-k Merit Weighting PBIL (PBIL-MW)

As we know from the algorithm of PBIL, the probability vector $\mathbf{P}^{(t+1)}$ is updated depending on whether each pair of elements G_{ij} drawn from $\acute{\mathbf{G}}$ and $\H{\mathbf{G}}$ are equal or not. Hence, only V_{max} and V_{min} are utilized to choose the two alternative learning rules for updating $P_i^{(t+1)}$. Further to the T-PBIL approach, the probability vector $\mathbf{P}^{(t+1)}$ is updated by the kth best with equal weighting $1/k$. Thus we know that the fitness values of samples are used only for the selection of genotypes for updating the probability vector.

Intuitively, we know that the fitness value of a sample represents whether the sample is closer to the optimal value both in quality and quantity. However, there is no concern for the quantity of fitness both in the updating computation of Equation 4.4 and 4.6.

For instance, suppose we want to update the vector by the best two samples, s_1 and s_2. Let G_1 and G_2 denote their genotype vectors. And let V_1 and V_2 denote their fitness values as well. From Equation 4.6 we know that the new $P^{(t+1)}$ is given by

$$P_j^{(t+1)} = (1-\gamma)P_j^{(t)} + \frac{\gamma}{2}(G_1 + G_2)\ . \tag{4.7}$$

Further, let us assume that $V_1 \gg V_2 > 0$. However, we know that there is no difference in Equation 4.7. Therefore, we argue that the fitness values may contribute towards updating $\mathbf{P}^{(t+1)}$ to improve the approximation of the new probability vector to the optimal genotype. Hence, the new $P^{(t+1)}$ can be given by

$$P_j^{(t+1)} = (1-\gamma)P_j^{(t)} + \frac{\gamma}{V_1 + V_2}(V_1 \times G_1 + V_2 \times G_2)\ . \tag{4.8}$$

Therefore, based on the reason above, we propose an adaptive algorithm that incorporates the aforementioned concept by using a weighted average mechanism, and works as follows. The initial steps are the same as the original PBIL until every fitness individual V_i has been computed in the first iteration. Then V_i is ranked, and the weights are given by

$$w_i = \frac{V_i'}{\sum_{i=1}^{k} V_i'}\ , \qquad 2 \le k \le n \tag{4.9}$$

where $V_i' = V_i - V_{min}$, and k is the number of chosen samples with the highest fitness values. Now that every w_i has been obtained, the probability vector $\mathbf{P}^{(t+1)}$ is given by

$$\mathbf{P}^{(t+1)} = (1-\gamma)\ \mathbf{P}^{(t)} + \gamma\sum_{i=1}^{k}(w_i\mathbf{G}_i)\ . \tag{4.10}$$

Note that, since every fitness V_i is considered and its weight w_i is given accordingly, there is only one learning rate γ that needs to be chosen in our approach. The pseudocode of PBIL-MW is shown in **Algorithm 5**.

Algorithm 5 PBIL-MW

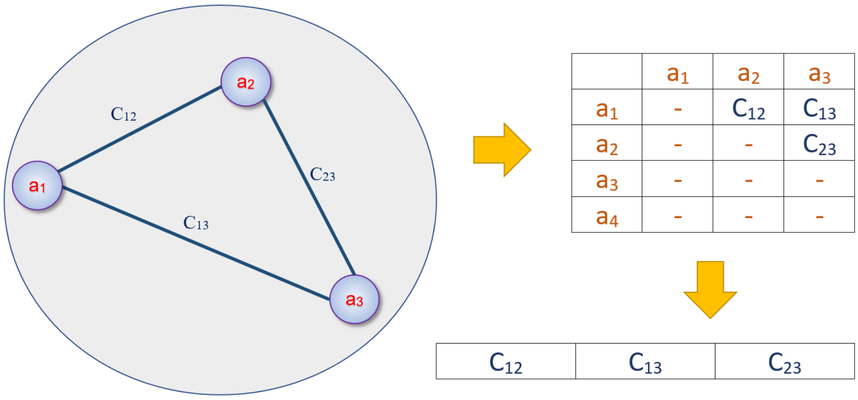
```
Initialize probability vector $\mathbf{P}^{(0)}$
Set CS$^* = \emptyset$, $v^*$(CS)$= 0$
repeat
    Generate a population $\mathbf{G}_n$ from $\mathbf{P}^{(t)}$
    Compute CS$_i$ from every sample $\mathbf{G}_i$ of the new population
    Evaluate and rank the fitness $V_i$ of each member $\mathbf{G}_i$ for all $i \in n$
    Obtain $w_i$ from Equation 4.9
    Update $\mathbf{P}^{(t+1)}$ according to Equation 4.10
    IF $V_{max} > v^*(\mathrm{CS})$, update CS$^* = \mathbf{G}_{max}$ and $v^*(\mathrm{CS}) = V_{max}$
until termination condition has been met
Output $v^*(\mathrm{CS})$ and CS$^*$
```

4.2.6 Treatment of Initialisation for Coalition Formation

As described in Subsection 2.3.2, we use the initial probability, i.e. the vector of a threshold, to filter the elements of each sample. Typically, the value is set to 0.5 in the applications both for GA and PBIL. However, in our genotype encoding scheme in Subsection 4.2.2, we found that a traditional initial value, i.e. 0.5, is not preferable.

For instance, let us assume a graph of $n = 3$ nodes (i.e. agents). From Equation 4.3 we know that there are $\frac{3\times(3-1)}{2} = 3$ connections among these three agents. According to the three connections, there are $2^3 = 8$ combinations to represent the connections. These 8 combinations are [1,1,1], [1,1,0], [1,0,1], [0,1,1], [1,0,0], [0,1,0], [0,0,1], and [0,0,0]. For easy explanation, the mapping of the connection index for the three agents is shown in Figure 4.4.

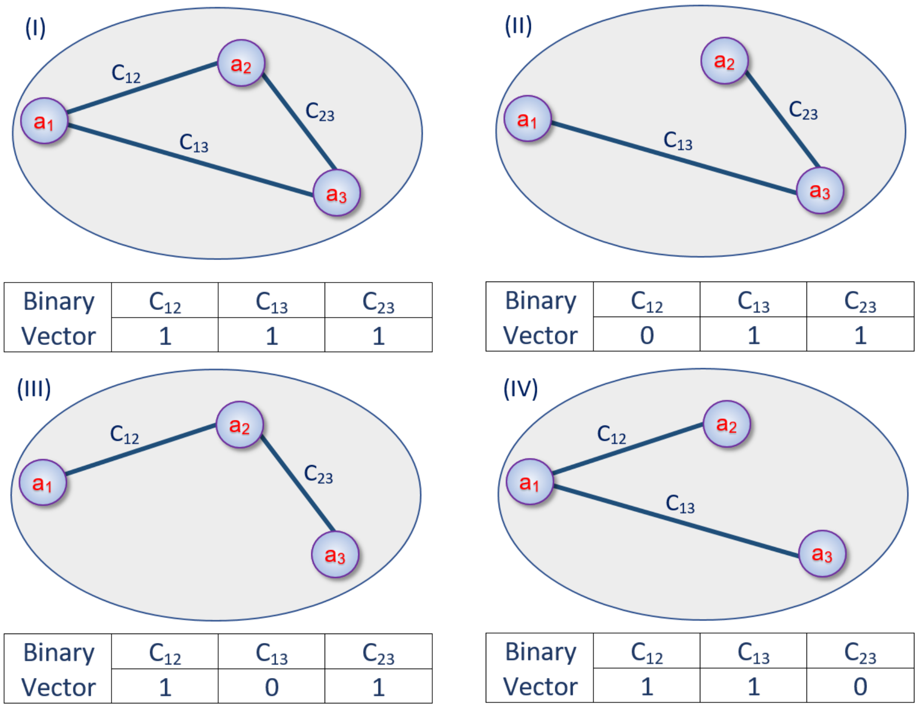
Here we can find that there are four out of eight binary vectors which all represent the grand coalition. As shown in Figure 4.5, these four binary vectors are [1,1,1], [0,1,1], [1,0,1], and [1,1,0]. For a singleton, there is only one binary vector, [0,0,0], as shown in Figure 4.6.



	a_1	a_2	a_3
a_1	-	C_{12}	C_{13}
a_2	-	-	C_{23}
a_3	-	-	-
a_4	-	-	-

C_{12}	C_{13}	C_{23}

Figure 4.4: The vector represents the connections of the four agents.



(I)

Binary Vector	C_{12}	C_{13}	C_{23}
	1	1	1

(II)

Binary Vector	C_{12}	C_{13}	C_{23}
	0	1	1

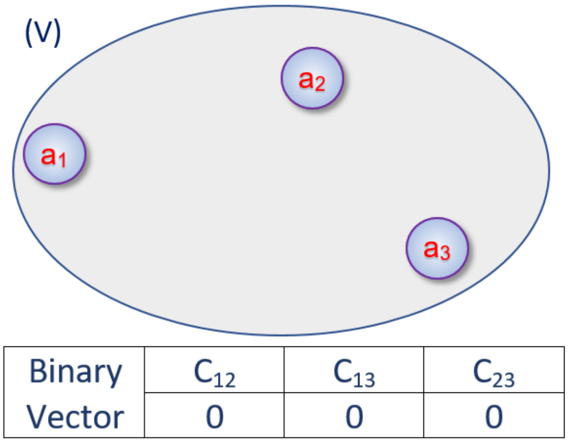
(III)

Binary Vector	C_{12}	C_{13}	C_{23}
	1	0	1

(IV)

Binary Vector	C_{12}	C_{13}	C_{23}
	1	1	0

Figure 4.5: The four vectors represent the connections of grand coalition.



(V)

Binary Vector	C_{12}	C_{13}	C_{23}
	0	0	0

Figure 4.6: The vector represents no connection of all singletons.

Furthermore, let $P^{(i)}$ denote the initial probability, which is equal to i. For a given initial probability $P^{(i)}$, let $P^{(i)}_{grand}$ and $P^{(i)}_{singleton}$ denote the probabilities of having a grand coalition or all singletons separately. Suppose the initial probability is set to $P^{(0.5)}$, which means that there is 50% chance of a binary code to be 1 or 0. Therefore, the probability of having any vector $[C_{12}, C_{13}, C_{23} | C_{ij} = 1 \text{ or } 0]$ is $0.5^3 = 0.125$. Since we know that there are four vectors which represent the grand coalition, therefore, we can find that there is $P^{(0.5)}_{grand} = 0.5^3 \times 4 = 0.5$ chance of obtaining a vector to represent a grand coalition. Further to above, we know that there is one vector, [0,0,0] to represent the singleton, and the probability of having a singleton is $P^{(0.5)}_{singleton} = 0.5^3 = 0.125$.

Moreover, let us set the initial probability to $P^{(0.3)}$. Thus we know that there 30% or 70% to have a binary code to be 1 or 0 accordingly. Therefore, the probability to have a vector [1,1,1] is $0.3^3 = 0.027$. The probability to have a vector [0,1,1], [1,0,1], and [1,1,0] is all equal to $0.3^2 \times 0.7 = 0.063$. Thus we know that there is $P^{(0.7)}_{grand} = 0.027 + 0.063 \times 3 = 0.216$ chance to obtain a vector to represent a grand coalition. Again, there is one vector, [0,0,0] to represent the singleton, and the probability to have a singleton is $P^{(0.7)}_{singleton} = 0.7^3 = 0.343$.

Subsequently, we have summarised the attributes and probabilities of different initial probabilities for several agents in Table 4.5. It is clear from the table that an initial probability of $P^{(0.5)}$ will make the binary vector more likely to represent a grand coalition in the connection scheme. For example, from Table 1, we know that $P^{(0.5)}_{grand} \approx 0.937$ for an alliance of 8 agents. The extremely higher probability will lead the searching process away from the smaller subset. Therefore we know that $P^{(0.5)}$ is not a good value for stochastic optimisation methods.

Therefore, to make the stochastic optimisation algorithms work for the coalition formation problem, we have proposed an essential treatment for giving the initial probability of the coalition formation as described below.

Assume a graph of n nodes. We consider all possible connections, and full mesh: $N = n(n-1)/2$, which is the length of gene probability vector in stochastic optimisation methods. Suppose for each of the N-connections the probability for its being " 1" (connected) is p. $\pi^{(0)}$ is the probability of having all singletons:

The probability of having m connections is

$$\pi^m = C^N_m (1-p)^m p^{N-m} \tag{4.11}$$

which gives a binomial distribution.

Table 4.5: The attributes and probabilities for different initial probabilities and agents.

No. of agents		2	3	4	5	6	7	8
No. of connections		1	3	6	10	15	21	28
Total combinations		2	8	64	1024	32768	2097152	268435456
No. of grand coalitions		1	4	38	728	26704	1866256	251548592
No. of all singletons		1	1	1	1	1	1	1
$P^{(0.5)}$	$P^{(0.5)}_{grand}$	0.5	0.5	≈0.594	≈0.711	≈0.815	≈0.890	≈0.937
	$P^{(0.5)}_{singleton}$	0.5	0.125	≈0.016	≈0.001	$\approx 3.05\times10^{-5}$	$\approx 4.77\times10^{-7}$	$\approx 3.73\times10^{-9}$
$P^{(0.3)}$	$P^{(0.3)}_{grand}$	0.3	0.216	≈0.219	≈0.256	≈0.317	≈0.394	≈0.480
	$P^{(0.3)}_{singleton}$	0.7	0.343	≈0.118	≈0.028	≈0.005	≈0.001	$\approx 4.60\times10^{-5}$
$P^{(0.1)}$	$P^{(0.1)}_{grand}$	0.1	0.028	≈0.013	≈0.008	≈0.006	≈0.006	≈0.005
	$P^{(0.1)}_{singleton}$	0.9	0.729	≈0.531	≈0.349	≈0.206	≈0.109	≈0.052

* $P^{(i)}$ denote the initial probability which is equal to i.

** $P^{(i)}_{grand}$ denote the probabilities to have a grand coalition.

*** $P^{(i)}_{singleton}$denote the probabilities to have all singletons.

Similar to Table 4.5, it can be shown from Equation 4.11 that for a conventional probability threshold set as 0.5, the probability of having $n-1$ connections in the network forming a grand coalition is potentially approaching to 1. Basically, for applying the stochastic optimisation methods, we wish that the chance to obtain any size of subset should be as even as possible. Therefore, we set a threshold such that the mean value of the number of connections is $E = (n-1)/2$. Now that the binomial distribution's mean is given by $E = Np$, we have

$$p = \frac{E}{N} = \frac{(n-1)/2}{n(n-1)/2} = \frac{1}{n}\,, \tag{4.12}$$

suggesting $1/n$ as the threshold for initializing the probability vector.

In short, our suggested threshold, i.e. $1/n$, has clearly made the optimisation process more likely to reach a better solution, as shown in the next section.

4.3 Experiment

4.3.1 Data

The data we used in this study are composed of two different sources. The first part is power consumption of smart-meter readings in New Zealand. The other part is power generated by commercialized facilities of wind turbines and solar panels which are coupled with New Zealand meteorological data[1]. The power status of all members, i.e. agents of experiment data, is then given by subtracting consumption from power generated. Eventually, we use the power data (in kWh) on an hourly basis, and the price $P_r = 20$ $(¢/kWh)$. Furthermore, we know that once the whole group has a power surplus at a given hour, then the grand coalition will make a trivial solution. Thus, we only consider cases with power deficits. Four cases are shown in Table 4.6.

Table 4.6: Power status for 12 agents in 4 case studies.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
Case I	-0.348	-1.558	-0.074	0.569	0.523	1.122	0.184	0.261	-0.450	-0.376	-0.222	-0.384
Case II	-2.020	-0.762	0.082	0.552	0.570	-0.309	-1.435	-0.889	-0.193	0.129	-0.808	2.074
Case III	-1.541	-1.084	-0.160	-0.196	0.596	-0.435	-1.611	-1.790	-0.167	0.337	1.791	3.131
Case IV	-3.414	-0.774	0.244	0.842	0.844	-1.881	-0.479	-0.901	-1.452	-0.324	0.121	0.287

4.3.2 Results

In our experiments, the cases with a group size of 12 usually will take over 200 seconds to complete the ES computation, while the size growing to 13 the running time will exponentially increase up to 20 minutes more, which means ES will become unfeasible for a more substantial number of agents. Meanwhile, the running time of SO algorithms growth nearly linearly with the number of agents. Figure 4.7 shows the running time of the four algorithms in terms of the number of agents.

Furthermore, from assessing the effectiveness of using different k values for top-k weightings in T-PBIL and PBIL-MW, the result is shown in Figure 4.8. The k value for each algorithm is appended after the algorithm name, e.g., PBIL-MW-2 denotes PBIL-MW with top 2 for weighted averaging. From the figure, it is clear that all

[1] Meteorological data obtained from NIWA's National Climate Database (https://cliflo.niwa.co.nz/ (accessed: 31/Dec/2019)).

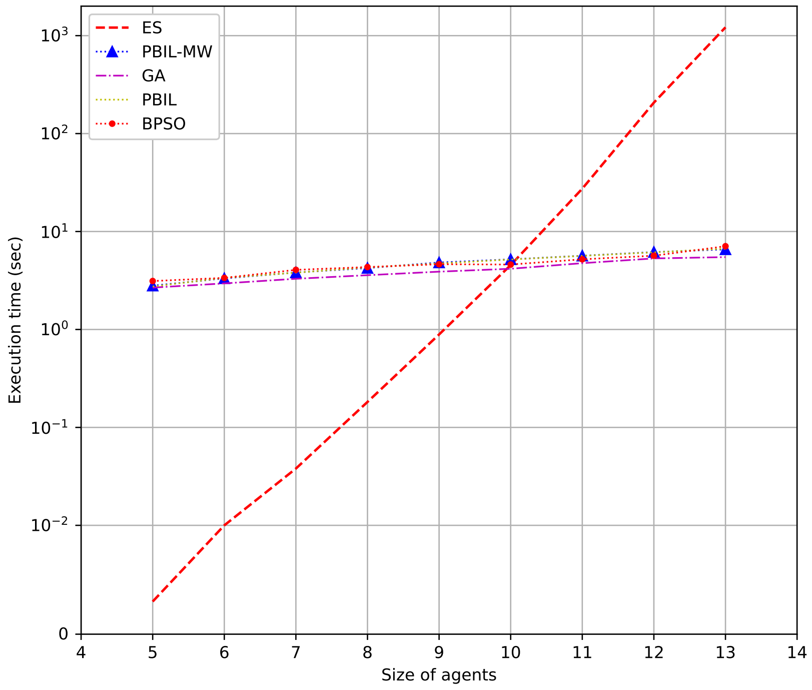


Figure 4.7: Execution time of different agent sizes

k values manage to achieve the global optimum within about 120 iterations. Although the converging speeds differ with various k values been used, PBIL-MW seems to converge faster than T-PBIL.

As the discussion in Subsection 4.2.6, the initial probability for SO may influence the converging speed. Therefore, we use PBIL-MW-10 with some probabilities to demonstrate the results of different initial probabilities, as shown in Figure 4.9. From the Figure, all the PBIL variants reach to the global optimum, and PBIL-MW-10 has the fastest converging speed.

Besides the initial probability, (Liu et al., 2016) mention that for BPSO, a smaller inertia weight enhances the exploration capability while a higher inertia weight encourages exploitation. Thus, we have compared the different inertia for BPSO, as shown in Figure 4.10. From the Figure, w=0.9 is the best that has the fastest converging speed and is the only one that hits the global optimum.

Furthermore, for comparing the different SO algorithms, we have used the same initial probability p=1/12 for all the SO algorithm. In addition to BPSO, the inertia w=0.9 is selected. The results are shown in Figure 4.11. From the Figure, PBIL-MW-10, PBIL-MW-2 and BPSO all hit the global optimum. Among the three, PBIL-

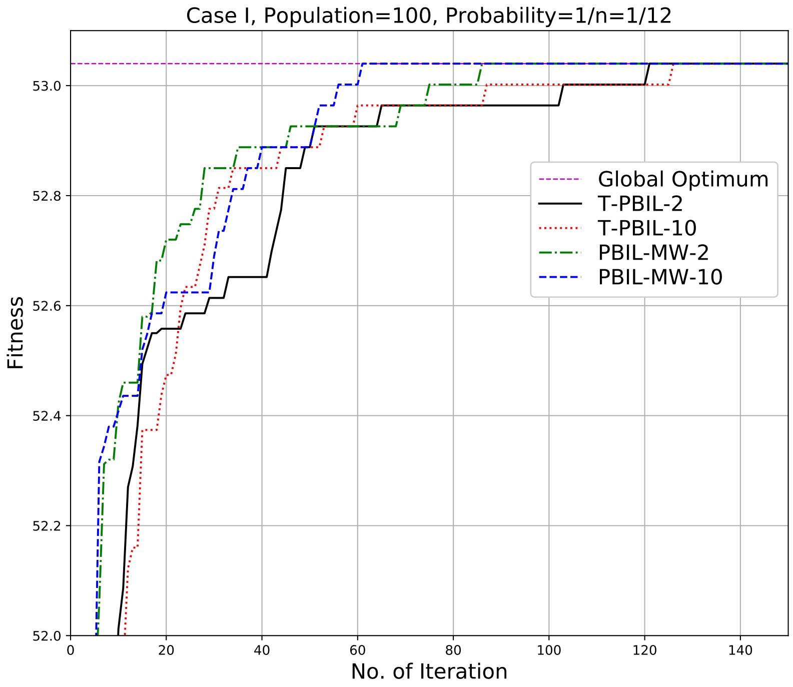


Figure 4.8: Comparisons of optimums and convergent speeds among different top-k weights of PBIL-MW for initial probability=(1/n)=(1/12)≈ 0.083.

MW-10 has the fastest converging speed, and BPSO is a bit slower than the other two.

Hereafter, a four-case study in terms of PBIL-MW, PBIL and GA are carried out. For simplicity, the experiments use all particles in merit weighting for PBIL-MW. Figure 4.12 is an example of optimized connections and profits found for samples in Case I.

After using ES to obtain the ground-truths of the four cases, the three algorithms, which using various initial probability threshold settings, are testing and comparing with the ground-truth. In order to address the potentiality of gaining the optimal structure by heuristic algorithms, a size of 100 populations, a maximum of 200 iterations and 4 initial probabilities p, $(=0.5, 0.1, 1/n \approx 0.083, 0.05)$, are given to a 12-agents coalition game as the standard setup in the experiment. For considering all possible connections, therefore, let l denote a full mesh of all agents' connections, $(l = 12 \times (12-1)/2 = 66)$, which has been given as the length of gene's probability vector for all three algorithms. Each of the three approaches is examining with six given initial probabilities and 20 runs for the four cases.

The implementation of GA follows the standard operation procedure of Mars-

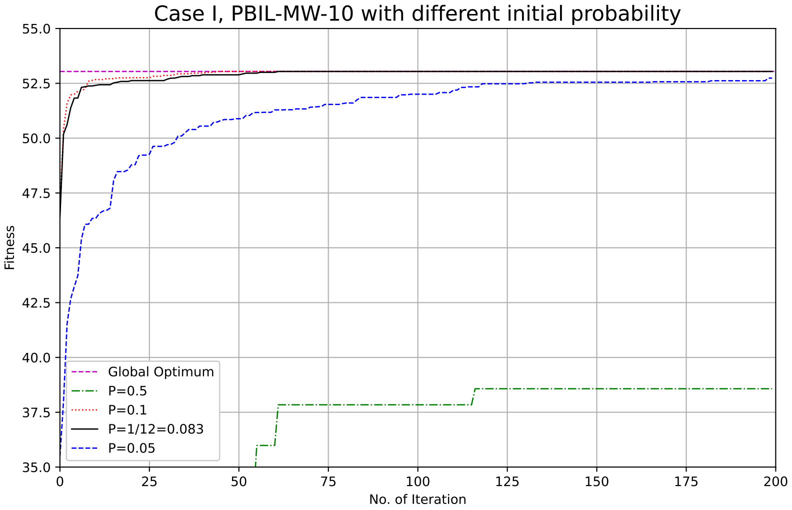


Figure 4.9: Comparisons of optimums and convergent speeds of PBIL-MW-10 among different initial probabilities (P)

land (2015). The parameters used here are mutation-probability=$1/l$, crossover=uniform. The implementation of BPSO follows the procedure of Liu et al. (2016). The parameters used here are: w=0.9, $C_1 = C_2 = 2$.

For PBIL, $\gamma = 0.1$, $\epsilon = 0.075$, mutation-probability= 0.02 and mutation-shift= 0.05 are used. While for PBIL-MW, only one parameter $\gamma = 0.05$ is used.

Table 4.7 is a statistical summary which shows the values of mean, variance and p-value of results obtained by different methods. To give the p-value of a student t-test in hypothesis testing (Wasserstein & Lazar, 2016), the significance level α is set to 0.05 and the result of PBIL-MW with $p = 1/n$ is chosen to be the principal sample while the test was performed. Therefore, if the calculated p-value is below the threshold α chosen for statistical significance (i.e. 0.05), then the result is significantly different from the result of principal sample.

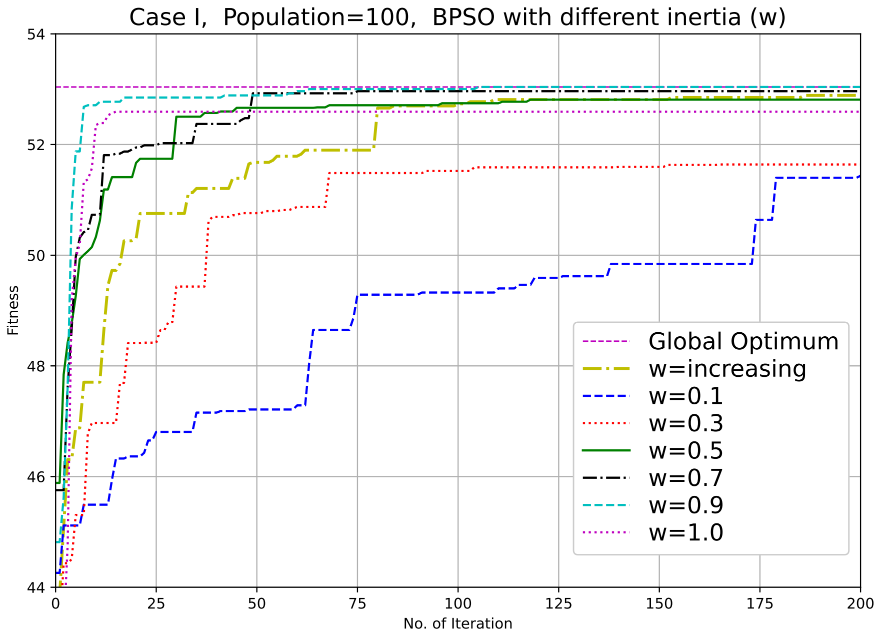


Figure 4.10: Comparisons of optimums and convergent speeds of PSO among different inertia.

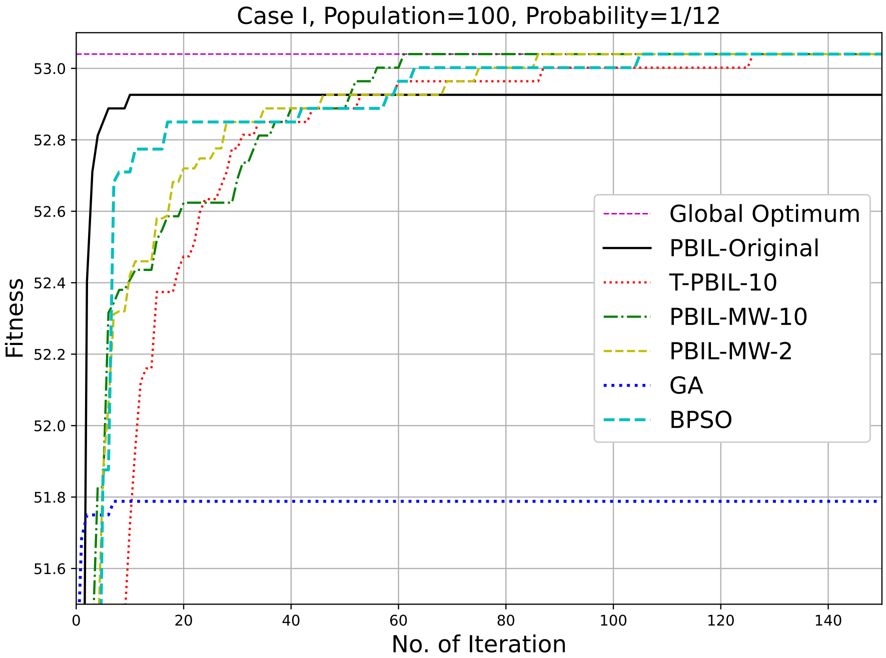
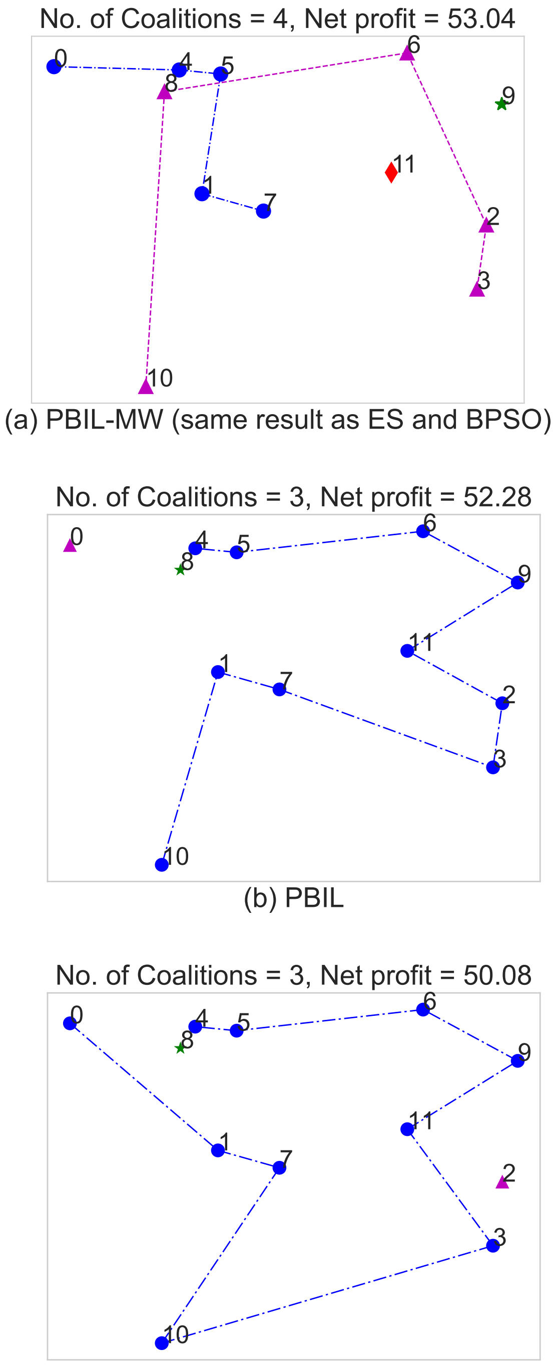


Figure 4.11: Comparisons of optimums and convergent speeds among PBIL-original, TPBIL-2 and PBIL-MW-2 for initial probability=1/n≈ 0.083.


No. of Coalitions = 4, Net profit = 53.04

(a) PBIL-MW (same result as ES and BPSO)

No. of Coalitions = 3, Net profit = 52.28

(b) PBIL

No. of Coalitions = 3, Net profit = 50.08

(c) GA

Figure 4.12: Different connections and profits found in terms of PBIL-MW, PBIL and GA.

By comparing the results obtained by the same methods in Table 4.7, it is clear that our suggestion $p = 1/n$ consistently leads to the best outcome, regardless the algorithm is PBIL, PBIL variants, GA or BPSO.

Moreover, by comparing the probabilities given in the experiment, it is easy to know that p=$1/n$ is apparent a better choice for the initial probability while searching for the optimal CS. Hence, the result proves that the suggestion of initializing probability by $1/n$ is a reasonable and effective strategy.

Furthermore, the result of the SO algorithms with $p = 1/n$ shows in Figure 4.13 to 4.16 suggests that although GA can find a better local optimum at the first few iterations, it is trapped there for the rest of the iterations. BPSO works fine in Case I but not as good in the others. The same as the PBIL, which can get a better local optimum than PBIL-MW initially, but again, it is restricted. Even though the converging speed of PBIL-MW is the least, it has a better diversity which helps to find the global optimum of fitness.

The time consumed for experiments[2] of the three algorithms is similar and approximate to 6 sec, which is faster than ES.

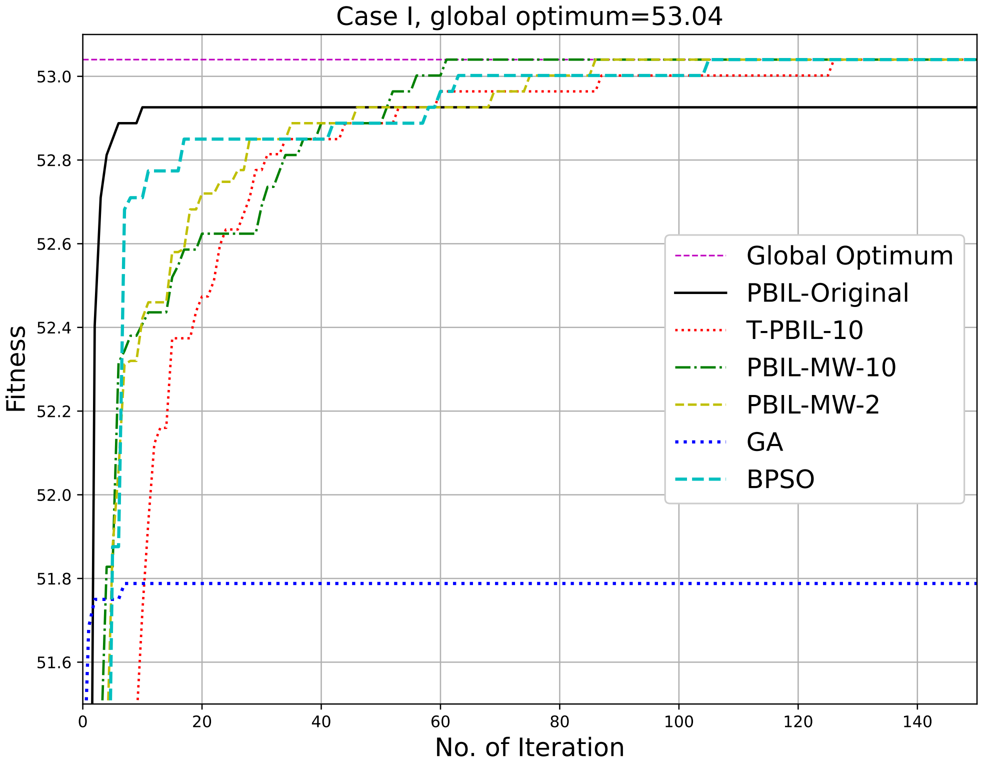


Figure 4.13: Optima and converging speeds of the algorithms for Case I.

[2] The code of the experiments is written, and testing in Python 3.6 on Windows 7 PC with Intel Core i5-4570 CPU and 16 GB RAM.

Table 4.7: Performance of PBIL-MW compared with PBIL and GA.

Algorithm	PBIL-MW				PBIL				Genetic Algorithm				BPSO
Init. Prob.	0.5	0.1	$1/n$	0.05	0.5	0.1	$1/n$	0.05	0.5	0.1	$1/n$	0.05	$1/n$
Case I Ground-truth=53.04													
Mean	39.44	53.04	53.04	53.04	36.54	52.93	52.96	52.89	3.07	52.52	52.51	52.26	53.04
Std Dev.	5.61	0	0	0	19.49	0.27	0.23	0.30	102.02	0.77	1.19	0.57	0
p-value*	**0.00**	0.5	-	0.5	**0.00**	**0.04**	0.08	**0.02**	**0.02**	**0.00**	**0.03**	**0.00**	0.5
Stat. sign.†	True	False	-	False	True	True	False	True	True	True	True	True	False
Case II Ground-truth=67.90													
Mean	2.75	67.90	67.90	67.71	54.29	67.30	67.04	67.11	-995.5	66.73	66.04	67.03	67.45
Std Dev.	11.97	0	0	0.46	23.09	0.66	0.64	0.74	83.89	1.13	1.98	0.97	0.64
p-value*	**0.00**	0.5	-	**0.04**	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Stat. sign.†	True	False	-	True	True	True	True	True	True	True	True	True	True
Case III Ground-truth=114.80													
Mean	109.2	114.8	114.8	114.8	109.2	114.7	114.7	114.4	100.9	114.0	113.4	114.4	114.5
Std Dev.	1.26	0	0	0	2.77	0.16	0.03	1.6	33.50	1.86	2.19	1.29	1.33
p-value*	**0.00**	0.5	-	0.5	**0.00**	0.17	0.17	0.17	**0.04**	**0.04**	**0.01**	0.09	0.16
Stat. sign.†	True	False	-	False	True	False	False	False	True	True	True	False	False
Case IV Ground-truth=45.10													
Mean	0	45.10	45.10	45.02	20.86	44.50	44.75	44.43	-3784.85	43.40	42.87	42.79	44.2
Std Dev.	0	0	0	0.25	21.02	0.70	0.73	1.25	673.04	2.64	3.11	2.58	1.31
p-value*	**0.00**	0.5	-	0.09	**0.00**	**0.00**	**0.03**	**0.02**	**0.00**	**0.01**	**0.00**	**0.00**	**0.00**
Stat. sign.†	True	False	-	False	True	True	True	True	True	True	True	True	True

† Stat. sign. is the abbreviation of statistical significance.

* p-values with statistical significance (i.e.$p < 0.05$) against PBIL-MW $(1/n)$ are highlighted in bold.

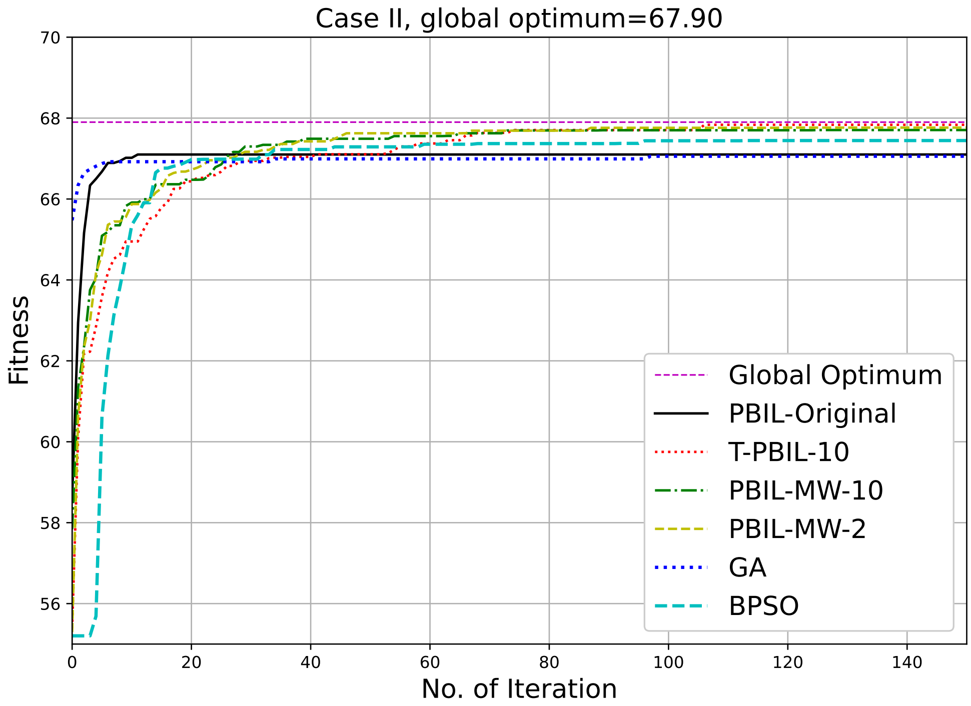


Figure 4.14: Optima and converging speeds of the algorithms for Case II.

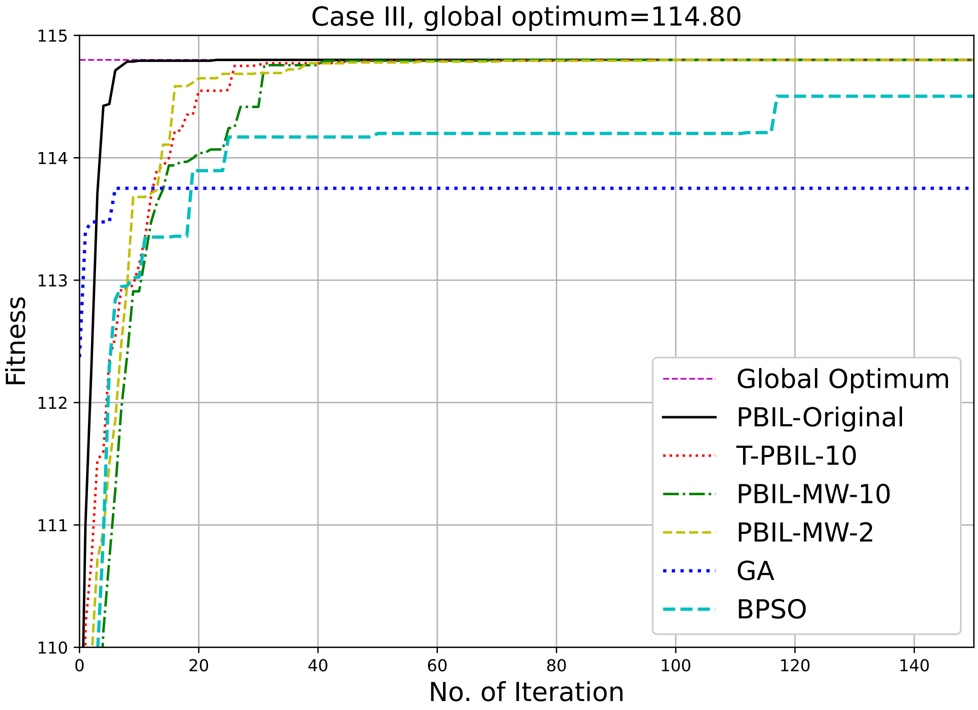


Figure 4.15: Optima and converging speeds of the algorithms for Case III.

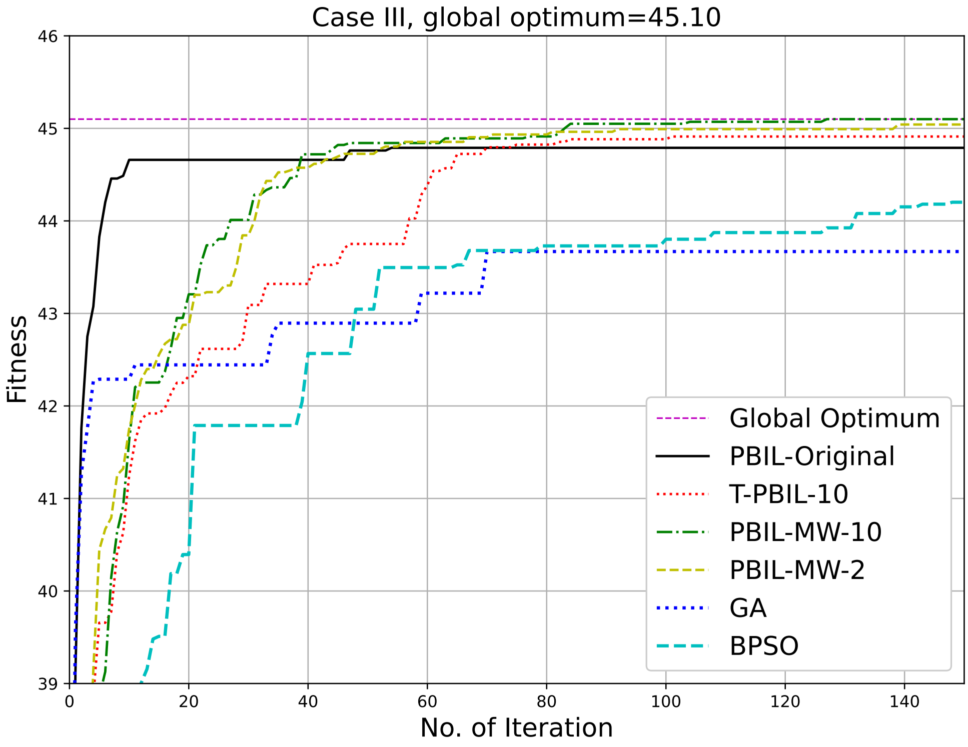


Figure 4.16: Optima and converging speeds of the algorithms for Case IV.

4.4 Summary

In this chapter, we propose a novel heuristic PBIL-MW algorithm and a strategy for choosing the initial probability for solving a coalition game of agents in partition form.

The recommended approach, which is based on SO algorithms, distributes participated agents to collaborate and act upon coalitions, to obtain the maximum profit hourly, and therefore could bring considerable payoff for members within a group in the long-term.

Furthermore, the proposed constraint by limiting the agents to form net power-surplus coalitions, one can secure the power distribution network from fluctuated demand and supply. Meanwhile, the scheme of the approach could enable a decentralised power system and improve the penetration of renewable energy.

We have shown the potential of PBIL-MW in solving the problem of CSG, such that a large number of agents in a group can form various feasible coalitions with an optimisation solution found within a limited time. The solution reveals here is a primary objective in the research of CSG. This result shows that PBIL-MW outperforms the PBIL and GA when searching for the optimal solution.

For future work, we will improve the PBIL-based algorithm based on some new coalition schemes and compare them with more SO methods by using larger-scale agents in the next two chapters.

Chapter 5

Improved CSG using Set-ID coding

Following on the previous chapter, to expend the number of agents in a CSG problem, this chapter proposes a new genotype encoding scheme, namely Set-ID coding, which has shortened the binary length of the sample. As a result, the new scheme has improved the performance of SO algorithms. Besides, for coping with the more large number of agents, this chapter has introduced a DP algorithm to substitute the ES in order to obtain exact solutions to compare the results with other SO algorithms used in this chapter.

The proposed algorithm has been examined by four sizes of agents set, including 12, 13, 16 and 20 agents, and each size has four sets for experiments. All of the agents are randomly picked from the database obtained in Chapter 3. The results demonstrate notable performance enhancements over the existing state-of-the-art methods. The approach and experiment in this chapter are extensions of one of our previous papers (Lee et al., 2018a).

The rest of this chapter is organised as follows. In Section 5.3, firstly, we review the connection scheme in previous chapter, and show the disadvantage of connection scheme for the SO methods. Then, we propose a new Set-ID encoding scheme. Furthermore, a comparison of the two as been discussed. In Section 5.4, to find the best initial probability suits for the Set-ID scheme, we have examined several initial values with respect to all the SO methods in this Chapter. Some results of the experiment are shown in Section 5.5, with the algorithm's performance, compared with DP, PBIL and GA in terms of convergence speed and computational efficiency for regional-scale optimisation. In the end, we conclude this chapter and point to some possible further directions.

5.1 Coalition Model in Local Smart Grids

In this chapter, we continue to use the scenario of local coalitions, as shown in Subsection 3.3.1 and Figure 3.1. However, in order to explore the feasibility and potential advantages of SO algorithms to solve the CSG, we will enlarge the scale from 12 agents to 20 agents in incremental steps.

Similar to Chapter 4, we assume that the prosumer agents are located within a distribution grid, e.g. a town or district, which could directly provide power-sharing through the distribution network.

5.2 Algorithmic Solutions

5.2.1 Deterministic Direct Search Algorithms

As we know from Chapter 4, the ES is an extreme low-efficiency algorithm because it directly searches the entire CSG space. Again, we have shown its limitation in the experiment in Section 5.5. Therefore, we have used the DP algorithm in this chapter to obtain the optimum solution to compare its accuracy and efficiency with the SO algorithms. The detailed procedure for using DP to solve the CSG problem has been shown in Subsection 2.3.1.

5.2.2 Stochastic Optimization (SO) methods

The SO algorithms we use in this chapter are the same as those in Chapter 4, which are GA, BPSO, PBIL and PBIL-MW. However, to improve efficiency, we have introduced a new scheme, named Set-ID, which will be demonstrated in the following Section.

5.3 Set-ID Encoding Scheme for SO Algorithms

The algorithms of SO used in this chapter are GA, BPSO, PBIL and PBIL-MW. To incorporate SO algorithms for a CSG problem, this chapter proposes a new encoding scheme for giving the initial probability of population which is described below.

In the previous Chapter 4, we proposed a connection scheme of binary encoding to represent the connection status among agents as a probability vector. Furthermore, we have shown that for a conventional probability threshold set as 0.5, the possibility of having $n-1$ connections in the network to form a grand coalition is potentially very imaginable. To avert this, we suggest $1/n$ as the threshold for initializing the probability vector.

The shortcoming of connection encoding is its vector length is a quadratic function, $n(n-1)/2$. Hence, when the number of agents becomes larger, the length of the probability vector will overgrow. Therefore, this chapter proposes a novel encoding scheme by using the coalition ID to allocate agents into different groups in the process of searching for an optimal coalition structure.

For example, in an 8 agents scenario, the bit-length is $\lceil \log_2 8 \rceil = 3$, the binary vector $[0, 0, 0]$ represents the no. 0 set, and $[0, 0, 1]$ represents no. 1 etc. Hence, if the ID array for agents 1 to 8 is $[3, 2, 3, 7, 2, 2, 4, 0]$ then the coalitions will be
Coalition 0: $\{a_8\}$;
Coalition 2: $\{a_2, a_5, a_6\}$;
Coalition 3: $\{a_1, a_3\}$;
Coalition 4: $\{a_7\}$;
Coalition 7: $\{a_4\}$.
Therefore, the CS for this ID set is
$\{\{a_1, a_3\}, \{a_2, a_5, a_6\}, \{a_4\}, \{a_7\}, \{a_8\}\}$.

Consequently, agents with the same ID suggest that they are in the same coalition. For a set of n agents, the maximum coalition is n (all singletons), and the minimum coalition is 1 (the grand coalition). Therefore, $n \times m$ bits $(m = \lceil \log_2 n \rceil)$ will be a sufficient length for the probability vector to represent all possible coalition structures. According to the Set-ID vector, any agent in the set can be indexed by any ID number within m bits. Therefore, the scheme also could provide a more effective and flexible for matching various $\mathcal{CS}$ which reveals the formalisation of CSG in terms of the number of $\mathcal{CS}$ and the number of total subsets as described in Section 2.2.

Figure 5.1 shows the Probability vector length of connect and set-id schemes. From the figure, we can see that when the number of agents growing to large size, the vector length of connection becoming a huge bit. Alternatively, the vector length of Set-ID scheme grows much lower. For instance, for 80 agents the vector length grows to 3160 bits, and the length Set-ID only grows to 560 bits, which is only 18% of the vector length of the connection scheme. Therefore, the Set-ID will consume less computer's memory and run faster, as demonstrated in Figure 5.3 in Section 5.5. Furthermore, since the Set-ID scheme demands less memory which will allow a more large size of the agent still be feasible, as described in Chapter 6.

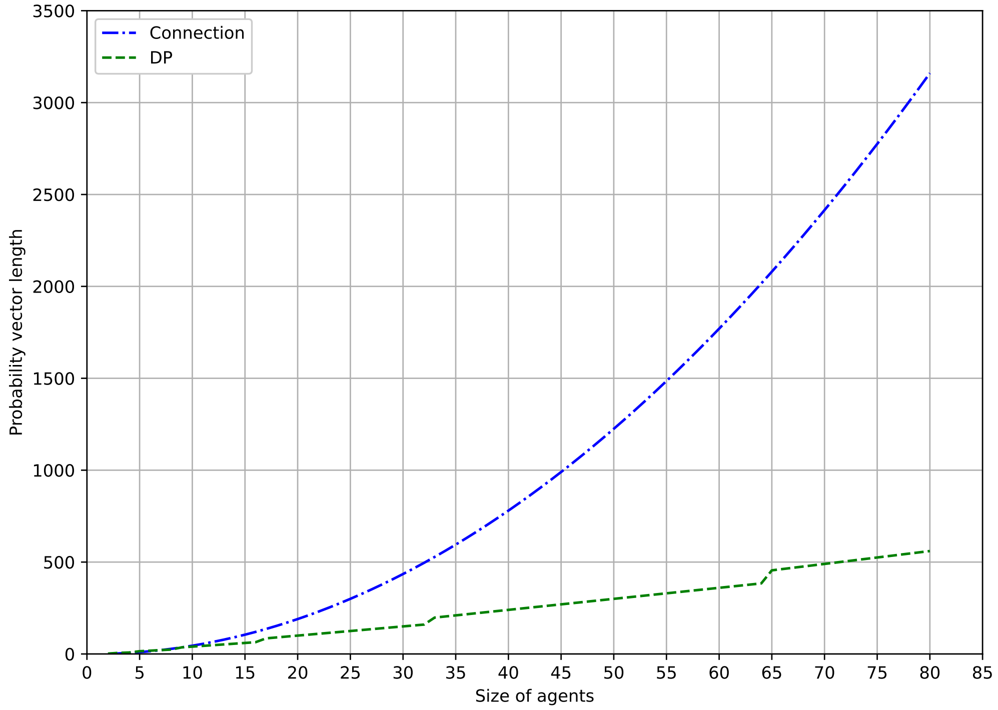


Figure 5.1: Probability vector length of connect and set-id schemes.

5.4 Initial Probability Threshold for Set-ID Encoding

According to our Set-ID scheme, we have found that an initial threshold of 0.5 will make the algorithms' iterations tend to choose coalition structures with a combination of smaller grouping sizes as coalitions, which lead to much longer iterations to find a better solution. Therefore, we have examined a series of initial probabilities from 0.5 to 0.05 with a 0.05 interval. The results suggested that a 0.1 threshold is the best initial value for 20-agent cases, which have led to optimal or good solutions with fewer iterations. Consequently, it is reasonable that we set a threshold away from 0.5 so that the $|C_k|$ (the size of coalition C_k) will form many versatile coalition structures. Figure 5.2 shows the different converging speed concerning different thresholds. In the figure, it is clear that both 0.1 and 0.9 are better than others.

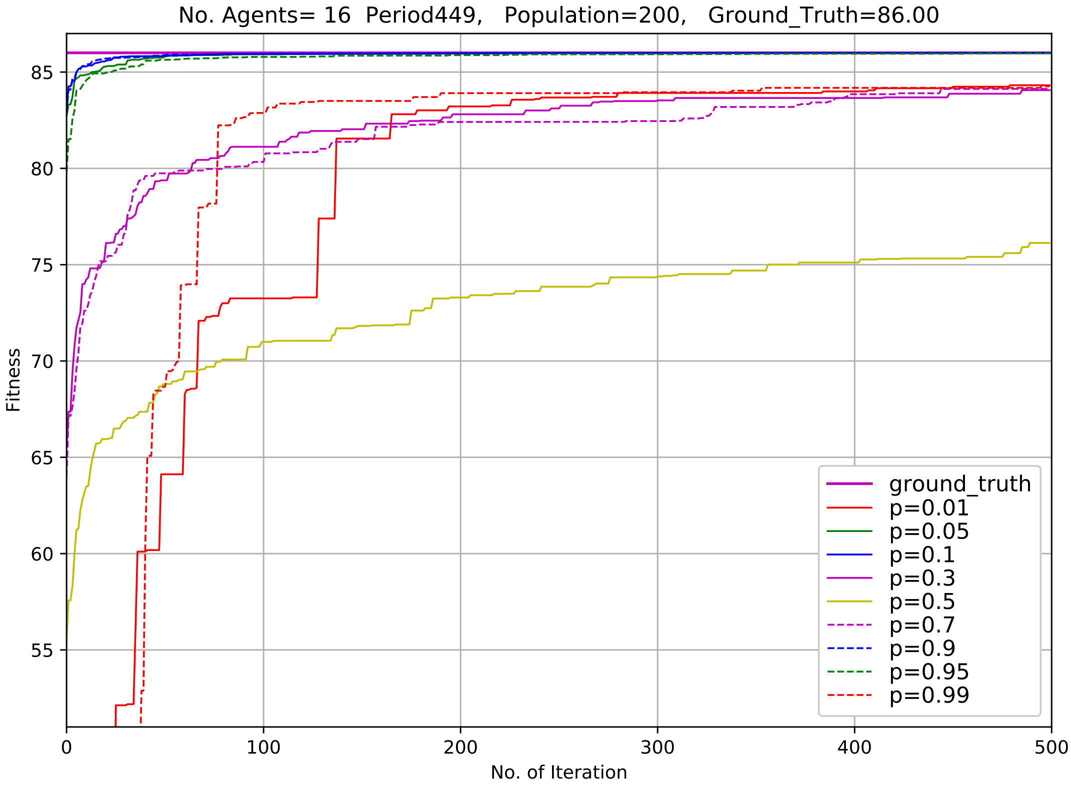


Figure 5.2: the different converging speed concerning different thresholds

5.5 Experiment

5.5.1 Setup

Aiming potential application in real-world smart grids, we follow our prior research to construct a realistic dataset, which are composed of two different sources. The first part is power consumption of smart-meter readings in New Zealand. The other part is power generated by commercialized facilities of wind turbines and solar panels which are coupled with meteorological data of New Zealand[1]. The power conditions of all agents are then given by subtracting demand from supply. The power(kWh) used are on an hourly basis, and the price $P_r = 20$ $(¢/kWh)$. Moreover, we know that once the union has a power surplus at a given hour then the grand coalition will make a trivial solution. Thus, we consider only cases with overall power deficits. In our data, we have one year of hourly power demand and supply for 240 agents. Among those, four cases with power deficits are randomly chosen. The statistics of these four cases are shown in Table 5.1. The detail process of generating the dataset is described in Subsection 3.4.3.

Table 5.1: Statistics of Power status for 20 agents in 4 cases.

Case	S_{no}	Surplus	Mean-S_{no}	D_{no}	Deficit	Mean-D_{no}
I	7	5.394	0.771	13	-12.741	-0.980
II	5	2.659	0.532	15	-7.948	-0.530
III	10	4.646	0.465	10	-10.87	-1.087
IV	10	5.024	0.502	10	-7.663	-0.766

S_{no}: No. of agents with Surplus power, D_{no}: No. of agents with Deficit power.

5.5.2 Results

To demonstrate the out-performance of Set-ID Scheme in PBIL-MW, we compare its efficiency and accuracy with different approaches.

Effectiveness

In this comparison, as shown in Table 5.2, there are seven approaches which are ES, DP, PBIL-MW with the connection scheme (Lee et al., 2017), BPSO, GA, and PBIL-MW with the Set-ID scheme. The number of agents in the experiments are 12,

[1] Meteorological data obtained from NIWA's National Climate Database (https://cliflo.niwa.co.nz/ (accessed: 31/Dec/2019)).

13, 16, 20, respectively. For SO-based algorithms, the population size and number of iterations used are 200 and 500 accordingly.

From Table 5.2 and Figure 5.3, it is evident that a group size of 12 takes over 200 sec to complete the ES, while for 16-agents the time required increases up to more than 140 hours, which becomes impractical for larger smart grids.

Even though DP runs faster than ES, however, DP still needs over 2.8 hours to reach an exact solution for a 20-agents case, which also becomes impractical in a large number of agents.

In this test, we use one connection encoding scheme for PBIL-MW. It is clear that this approach is much faster than DP, but when comparing it with the Set-ID scheme, it may consume more memory and take longer while the size becomes more significant.

Table 5.2: Average Time needed by different approaches.

Number of Agents Methods	12	13	16	20
ES	207.1	1,215.3	504,125.2	na
DP	8.7	65.3	136.5	22,367.7
PBIL-MW(C)	6.2	12.4	30.2	44.6
BPSO(ID)	21.9	24.7	29.4	43.5
GA(ID)	7.5	10.2	13.5	30.6
PBIL(ID)	5.3	6.5	**9.6**	**19.2**
PBIL-MW(ID)	**5.1**	**6.3**	9.7	**19.2**

-The best results are highlighted in bold. Unit: sec

* (C) indicates the connection scheme, (ID) indicate the SET-ID scheme.

Table 5.3 shows the average best results by different approaches compared to DP. From the Table, it is clear that PBIL-MW(ID) has the best result. For Case I, II, and IV, each case has run 20 times, and all reached the optimum get by DP. For Case III, there are 80% (16 in 20) still reached to the optimum, as shown in Table 5.4. Furthermore, if we increase the number of iterations to 2000, as shown in Figure 5.4, it is clear that after over 500 iterations, all the 20 runs reached to the optimum.

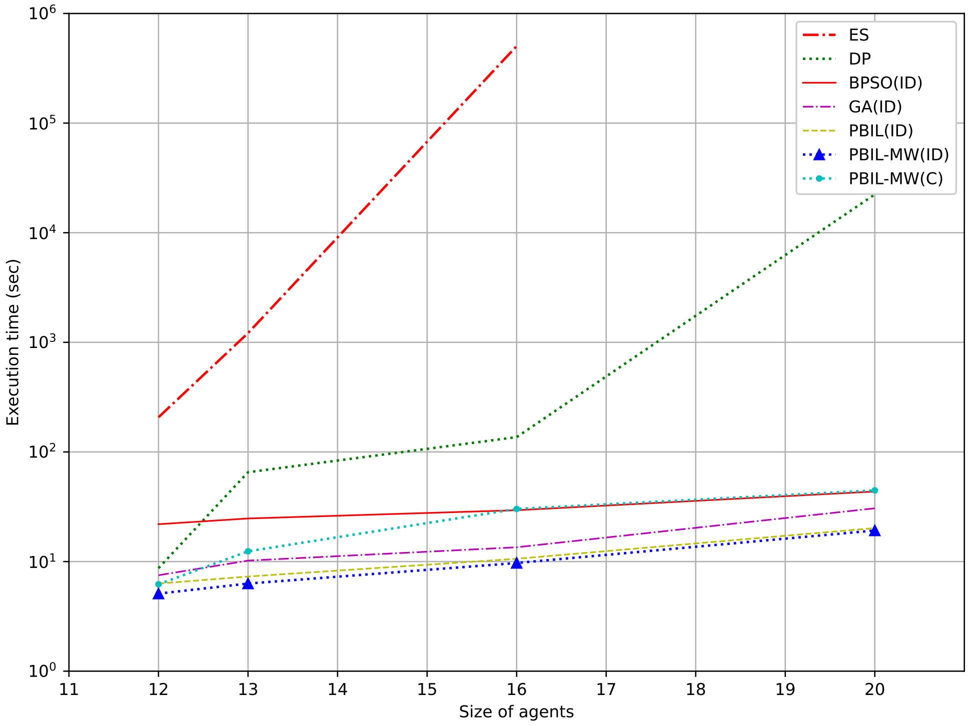


Figure 5.3: Execution time of different agent sizes

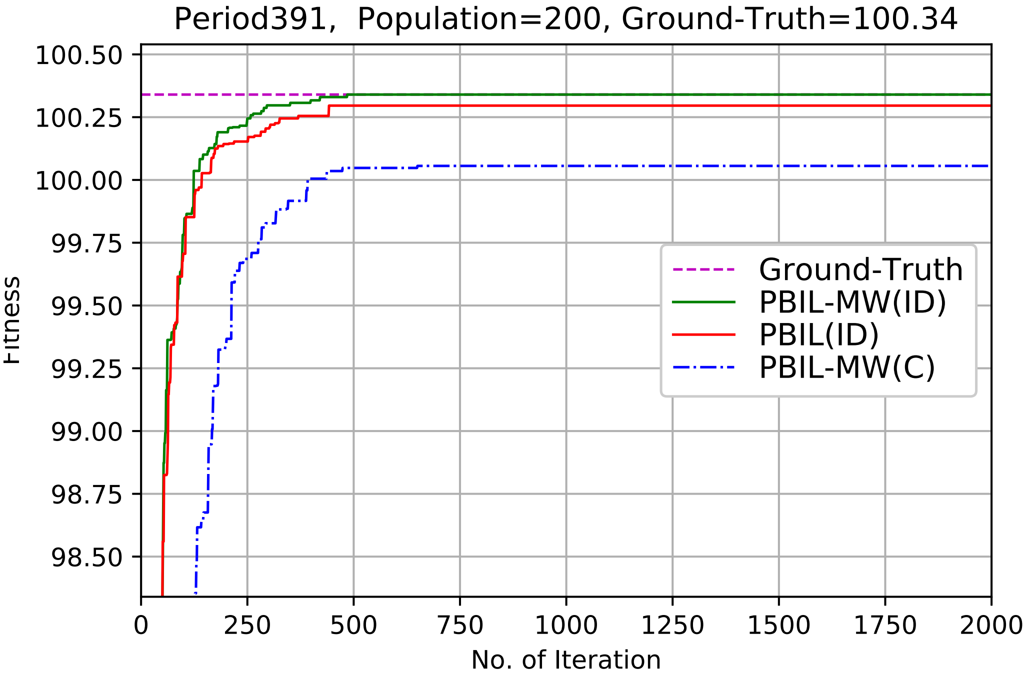


Figure 5.4: Comparative results of PBIL-MW for Case-III after 2000 iterations

Table 5.3: Average best results by different approaches compared to DP.

Case	**DP**	PBIL-MW(ID)	PBIL-(ID)	GA	PBIL-MW(C)
I	**53.16**	**53.16**	**53.16**	51.8	**53.16**
II	**92.86**	**92.86**	**92.86**	81.79	92.85
III	**100.34**	100.3	100.29	87.192	99.9
IV	**107.86**	**107.86**	**107.86**	100.79	107.85

-The best results are highlighted in bold. Unit: ¢

Table 5.4: Average iterations which hit the ground-truth.

Case	PBIL-MW(ID)	PBIL-(ID)	GA	PBIL-MW(C)
I	**98**	115	na	171
II	196	**184**	na	345(85%)*
III	**228**(80%)*	276(80%)*	na	Na
IV	**159**	186	na	283(90%)*

-The best results are highlighted in bold.

* Average Iterations for the Percentage, denotes by (%), of 20 runs which hit the ground-truth.

Accuracy

Although DP takes more time to obtain the result, it is still an essential approach because it guarantees to reach the global optimum. In comparison with other algorithms, we choose 4 cases of 20 agents run by DP to be our ground-truth. The accuracy in terms of SO-based algorithms is described below separately.

For an extensive feasible study of the application of GA to CSG, we use three crossover parameters (single-point, uniform and segment), four mutation probabilities (0.05, 0.08, 0.1, 0.2), 200 samples, 500 iterations, 20 runs and three initial probabilities (0.1, $1/n$ (=1/20) and 0.03). Among these, we found the best one is with these parameters: segment crossover, mutation = 0.03, and initial probability = 0.05.

Similar to the GA study, PBIL and PBIL-MW are run by the same parameters of population, iterations and runs. Besides this, there are only three parameters (number of top-best, learning rate and initial probability) that need to be assigned for both approaches. To summarise the best results, we have fixed the number of top-best at 2, and the learning rate at 0.005. The initial probability used in the Connection scheme is 1/N (=0.05) and 0.1 for the Set-ID scheme.

Figure 5.5 to 5.8 present the results of the four cases, where (ID) and (C) in the legend denote the schemes of Set-ID and Connection accordingly.

From the four figures, we can see that PBIL-MW(ID) outperforms all others both in converging speed and accuracy. For the Cases of I, II and IV, the optima of these cases hit the ground-truth with fewer iterations. Except for Case III, the mean of best fitness for the 20 runs is 100.32, which is slightly less than the ground-truth which is 100.34. Among those 20 runs, there are 17 experiments whose best results meet the optimum. Therefore, we can say that the accuracy to find the exact CS is 85% in this case. However, we also tried to increase the number of iterations to 1000 for case III and found that the PBIL-MW(ID) always reached the optimum for all 20 runs. The average iterations for new experiments to hit the ground-truth are 238, and the shortest and longest iterations are 82 and 512 respectively.

Furthermore, we also use a measurement, called Solution Quality (Michalak et al., 2016), to examine the "current" best solution found at a given time while running an optimisation. The solution quality is given by $v(cs)/v(cs^*)$, where $v(cs)$ and $v(cs^*)$ denote the current and the optimal solution respectively. The results are shown in Table 5.4. The quality solution for the PBIL-MW(ID) scheme after over 500 iterations all the four cases reached to the optimum. As a result, PBIL-MW(ID) has shown its superiority among the SO we used.

In general, our experiment results show that PBIL-MW(ID) is a fast and reliable approach for solving the CSG problem in smart grids.

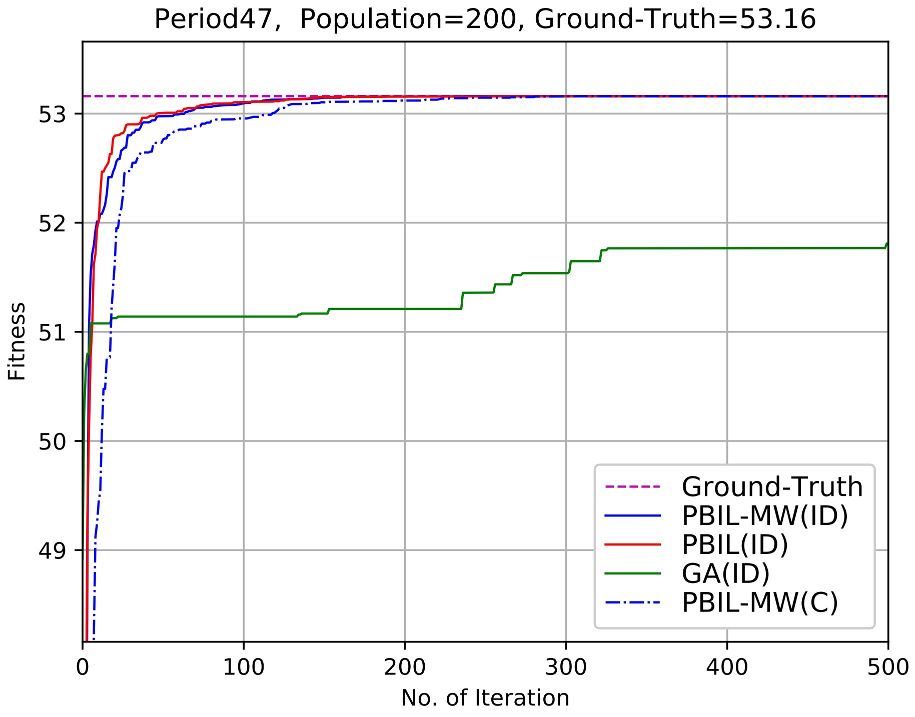


Figure 5.5: Comparative results of PBIL-MW for Case-I

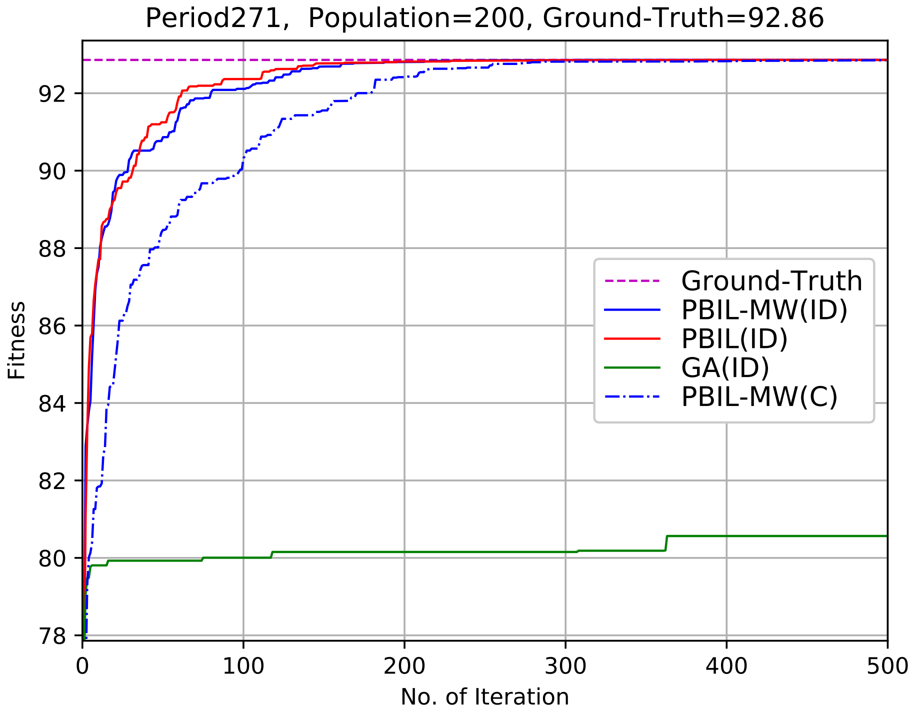


Figure 5.6: Comparative results of PBIL-MW for Case-II

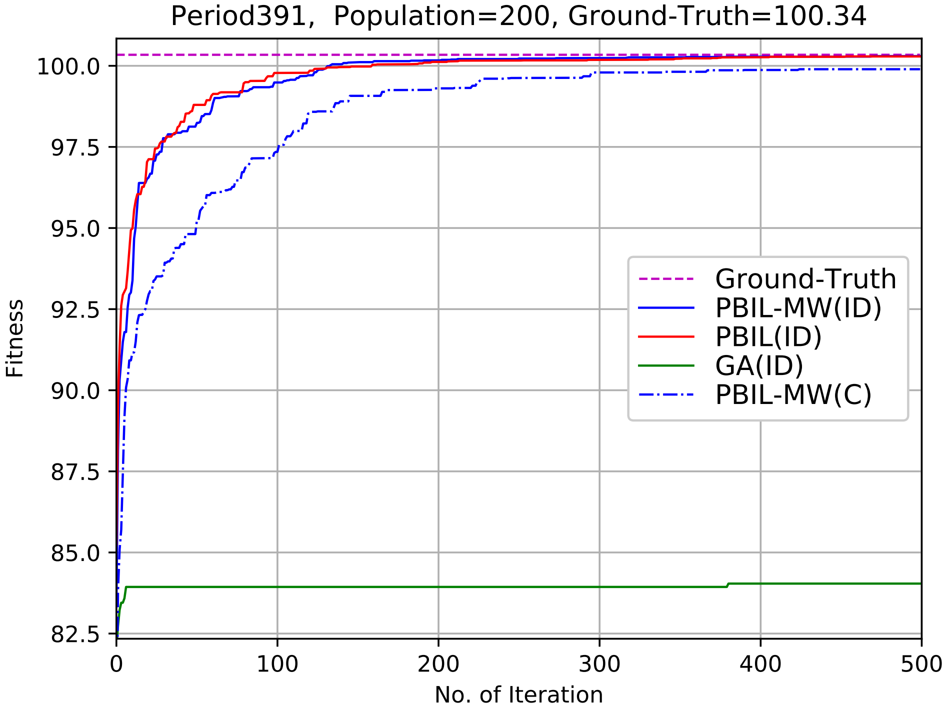


Figure 5.7: Comparative results of PBIL-MW for Case-III

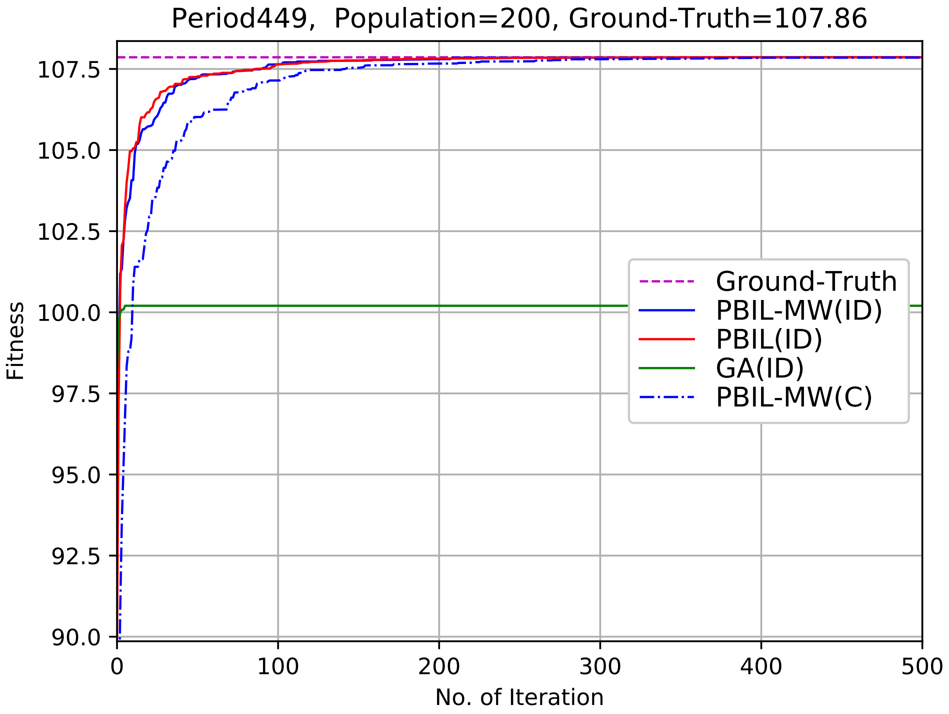


Figure 5.8: Comparative results of PBIL-MW for Case-IV

5.6 Summary

In this chapter we prove that the new Set-ID outperform the previous connection scheme both in shortening running time and less PC memory consumption. For a large number of agents, the ES will become unfeasible[2]. However, our proposed new PBIL-MW algorithm could obtain the nearly exact solutions as the DS methods and outperform other DS in efficiency and accuracy.

When the number of agents reaches to 20, there are 51.72 trillion CS need to be evaluated. Even using the state-of-the-art DS methods, namely the DP approach, will take over 2.8 hours to find the exact solution on average. However, the ordinary running time of the previous PBIL-MW with connection scheme is around 45 seconds. Moreover, the new Set-ID approach of PBIL-MW is the fast one, which is exceeding two times faster than the connection scheme.

Our results have brought evidence to prove that once the appropriate population size and the number of iterations have been adequately set up, the optimum will be reachable by the new algorithm. The faster response of the new approach meets the requirement of intermittent power exchange for smart grids operating in a co-operative mode, which can enable a decentralised power system and improve the penetration of renewable energy.

According to the quick timing of our new approach, the coalitional structure model can be applied in the off-grid (island) mode. Also, the model can be provided to a local area for power-sharing at emergence while the power supply is unavailable, which may be caused by natural hazard or improper operation.

However, this study is aimed at agents of the local smart grid without power transmission via the transformer. Hence, the consideration of power loss has been neglected. In the next chapter, we will advance the study to multi-regional applications with scenarios of more agents and long-distance power-sharing, which will be taking into account the power loss on transmission lines.

[2] From the experiment in Section 5.5, we know that a size of 12 agents has spent over 200 sec to complete the ES, and while the size expends to 16-agent the time required increases up to more than 140 hours, which makes ES becoming impractical for larger smart grids.

Chapter 6

Hierarchical solutions for a large-scale CSG

Upon the previous two chapters, for enlarging the CSG problem into a multi-regional area, this chapter proposes the Global PBIL-MW and the Hierarchical PBIL-MW approaches, which bring the power loss caused by transformers into account and enable the problem to be solved more efficiently. What is more, the Hierarchical PBIL-MW approaches improves the performance of Global PBIL-MW algorithms, while it only causes a minor lower accuracy in comparison with the latter approach.

The proposed algorithm has been examined by five sizes of agents set, including 20, 32, 48, 64 and 80 agents, and each size has four sets for experiments. All of the agents are randomly picked from the database obtained in Chapter 3. The results demonstrate that these two algorithms can overcome the limitation of the state-of-the-art DS methods in agents' number. The approach and experiments in this chapter are extensions of one of our previous papers (Lee et al., 2018b).

6.1 Related work

In Chapter 4 we have suggested an improved algorithm, namely Top-k Merit Weighting PBIL (PBIL-MW) (Lee et al., 2017), for solving CSG problems. The result has shown a promising ability, both in accuracy and efficiency in comparison with other algorithms. Advancing upon the previous approach, Chapter 5 designs a new genotype encoding scheme (Lee et al., 2018a), and the new results outperform a few SO counterparts, such as Genetic Algorithm and the original PBIL. Moreover, in comparison with DP, our approach has largely reduced the memory consumed in computation and shortened the running time significantly. In this study, we further proposed two approaches, a Hierarchical PBIL-MW algorithm and a termination scheme, and both have shown greater advantages for dealing with large scale CSG applications which the DP variants hardly achieve.

The rest of this chapter is organised as follows: In Section 6.2, we first give structure of Coalition Model for Large Scale Smart Grids. In Section 6.4, we then propose our SO-based solutions, especially the hierarchical approach and the termination scheme, for investigating the optimal partition. Some results from the experiment are shown in Section 6.5, with the algorithm's performance compared to the DP and our previous PBIL-MW method in terms of convergence speed and computational efficiency for more considerable scale optimisation. Finally, we conclude the chapter and point to some possible further directions.

6.2 Coalition Model for Large Scale Smart Grids

6.2.1 Coalition Model for Smart Grids

In comparison with other approaches for CSG in smart grids, we adopt the previous model (Lee et al., 2017; 2018a) which requires that every agent in the coalition should have adequate RE to support its own demand in general. However, according to the intermittency of RE, the agent may frequently face a shortage. Accordingly, keeping an agreement to share the surplus energy among others is a more effective method in comparison to measures such as expanding the facility, installing a larger backup capacity or dealing with power companies. Since the demand and supply are both dynamic, the model needs to continuously engage a flexible mechanism to obtain the optimal CS. In the studies, we assume all agents exchange their power generation surplus and their consumption demands regularly, e.g. on an hourly basis in this study. Thus, a faster algorithm, e.g. within 10 minutes, with acceptable accuracy will be an essential prerequisite for providing decision-makers, such as agents or the coalition organizer, with sufficient time to allow agents to adjust their power demand and supply, and to reach a confirmed agreement among the agents.

6.2.2 Distributed Agents in Regional Smart Grids

Based on our previous studies which focused on local coalition, to demonstrate the ability on a large regional scale, such as cities, we have extended the local model to four regional areas as shown in Figure 3.2. The power transports among the inter-area need to be sent by a high voltage transmission network by way of transformers to exchange power between distribution and transmission lines. The power loss caused by the electrical lines is trivial in a short distance; therefore, for a simplified coalition model, only the power losses of transformers, β=2%, per step up or down via a transformer, are calculated.

6.3 Algorithmic Solutions

6.3.1 Deterministic Direct Search Algorithms

As we know from Chapter 5, the DS algorithm has reached its limitation to solve a CSG problem with 20 agents. Therefore, as shown in the experiment in Section 6.5, the DP will not be used to obtain the optimum solution when the number of agents is more than 20.

6.3.2 Stochastic Optimization (SO) methods

From the results in the previous two chapters, we know that our proposed PBIL-MW algorithm outperforms the other SO Algorithms. Furthermore, as mentioned in Section 6.2, we have extended the model to four regional areas in this chapter. Therefore, we have skipped the SO algorithms in the previous chapters, and propose two improved algorithms of the PBIL-MW.

6.3.3 Set-ID Encoding Scheme for SO Algorithms

To incorporate SO algorithms for a CSG problem, we adopt the Set-Id encoding scheme as mentioned in Section 5.3.

6.4 PBIL-MW Approaches for Solving a Large-Scale CSG

6.4.1 Global PBIL-MW

To distinguish the organisational paradigms used in this study, we name "Global PBIL-MW" as the approach to search the optimal CSG in a single PBIL-MW computation as in previous studies (Lee et al., 2017; 2018a).

Coalition Evaluation

Following Eq. 4.2, the net power $Q(C_k)$ of coalition C_k with respect to intra-area and inter-area are calculated separately. For example, a_1 and b_1 are surplus agents in different areas, and a_2 is a deficit in the same area as a_1. For Coalition $\{a_1, a_2\}$ the net power $Q(C_k) = Q(a_1) - Q(a_2)$, and for Coalition $\{b_1, a_2\}$ the net power $Q(C_k) = Q(b_1) \times (1 - 2\beta) - Q(a_2) = Q(b_1) \times (1 - 4\%) - Q(a_2)$.

Initial Probability Threshold

During our preliminary study, we found that different initial thresholds for numerous agents and their power statuses will lead the algorithms' iterations to find a better solution at different speeds. Some results are shown in Figure 6.1. Therefore, we have examined a series of initial probabilities from 0.5 to 0.01, the suggested initial thresholds for numerous agent size are shown in Table 6.1.

Table 6.1: Different initial probabilities used in Global PBIL-MW

Period	I	II	III	IV
Agent's size	Initial probability			
20	0.1	0.1	0.1	0.1
32	0.1	0.1	0.1	0.1
48	0.1	0.2	0.1	0.1
64	0.1	0.2	0.1	0.1
80	0.1	0.2	0.05	0.1

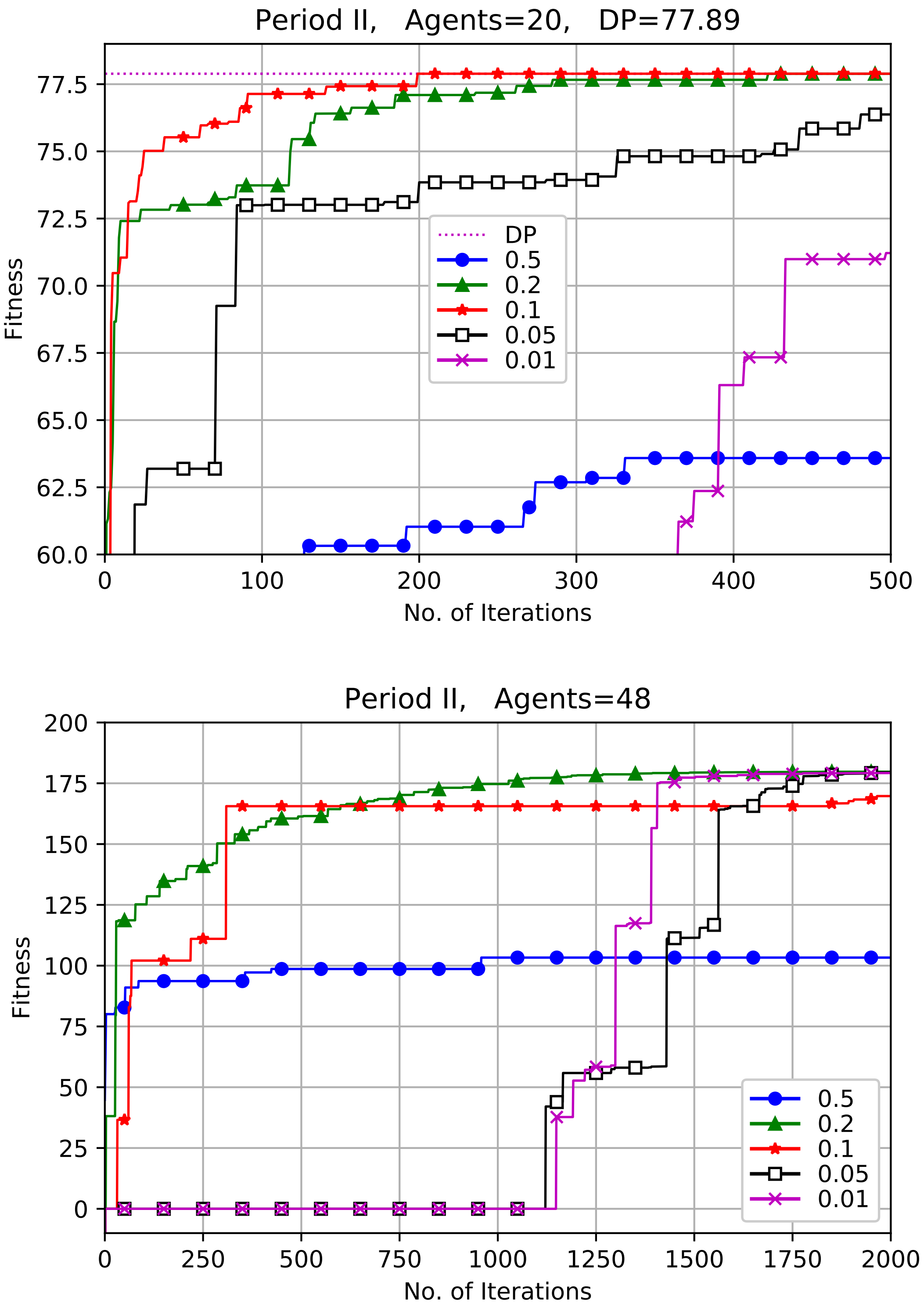


Figure 6.1: Comparative results of different initial probabilities for period II

6.4.2 Hierarchical PBIL-MW

Once the number of agents becomes larger, the size $\mathcal{F}(\mathcal{CS})$ will be very large. For instance, in the 80 agents experiment, $\mathcal{CS}_{80} = B(80) \approx 9.913 \times 10^{86}$, any program written to implement DP variants might be unfeasible. Although the vector length $L = 80 \times \lceil \log_2 80 \rceil = 560$ bits will be moderate for PBIL-MW to search for the better CS, it will consume longer iterations to reach a global or local optimal solution, e.g. over 27 minutes and 1580 iterations on average will be needed in our experiment. Consequently, we propose a hierarchical structure of PBIL-MW to accelerate the process of exploring the CS.

Our hierarchical approach uses two steps of PBIL-MW iterations. First, it searches every local area separately to form a local optimal CS. Note that, since there is no transformer within the local area, the power losses are ignored. Secondly, we then employ all the local coalitions to explore the CS for the whole region. For instance, in the first step, a_1 to a_4 and b_1 to b_4 can form a local optimal CS such as $la_1 = \{a_1, a_2\}$, $la_2 = \{a_3, a_4\}$, $lb_1 = \{b_1, b_3\}$ and $lb_2 = \{b_2, b_4\}$. According to the local CS, the second PBIL-MW can look for the hierarchical optimal CS, such as $h_1 = \{la_1, lb_2\} = \{a_1, a_2, b_2, b_4\}$ and $h_2 = \{la_2, lb_1\} = \{a_3, a_4, b_1, b_3\}$.

From the framework, we realize that after forming the local coalitions, then any agent in a local coalition would only be able to cooperate with other agents in a different local coalition by joining the two coalitions. Consequently, the best fitness will be less than or equal to the one by global PBIL-MW. However, by sacrificing the accuracy, this approach may shorten the time in return.

6.4.3 Termination Scheme

In the experiments of SO algorithms, the number of iterations and the termination conditions have to be predefined; otherwise, some program will keep running endlessly. The termination conditions could be the one or a combination of the following options:

- A cut-off criterion or condition.
- A fixed iteration number.
- A fixed running time.
- A fixed repeating condition.

As seen in Figure 6.2, we might find that some PBIL-MW can reach a maximum fitness with just a few iterations. For speeding up the running time, we have improved the algorithm by adding a threshold check to count the latest best repeating times during the iterations. Throughout the process, any latest global best fitness values and repeat times have been recorded. If a further new global best value is found, then the repeat time will return to one; otherwise, this duplicated time will be accumulated until its number meets the termination threshold, e.g. 10 times in our experiments. On that occasion, the program will be terminated earlier.

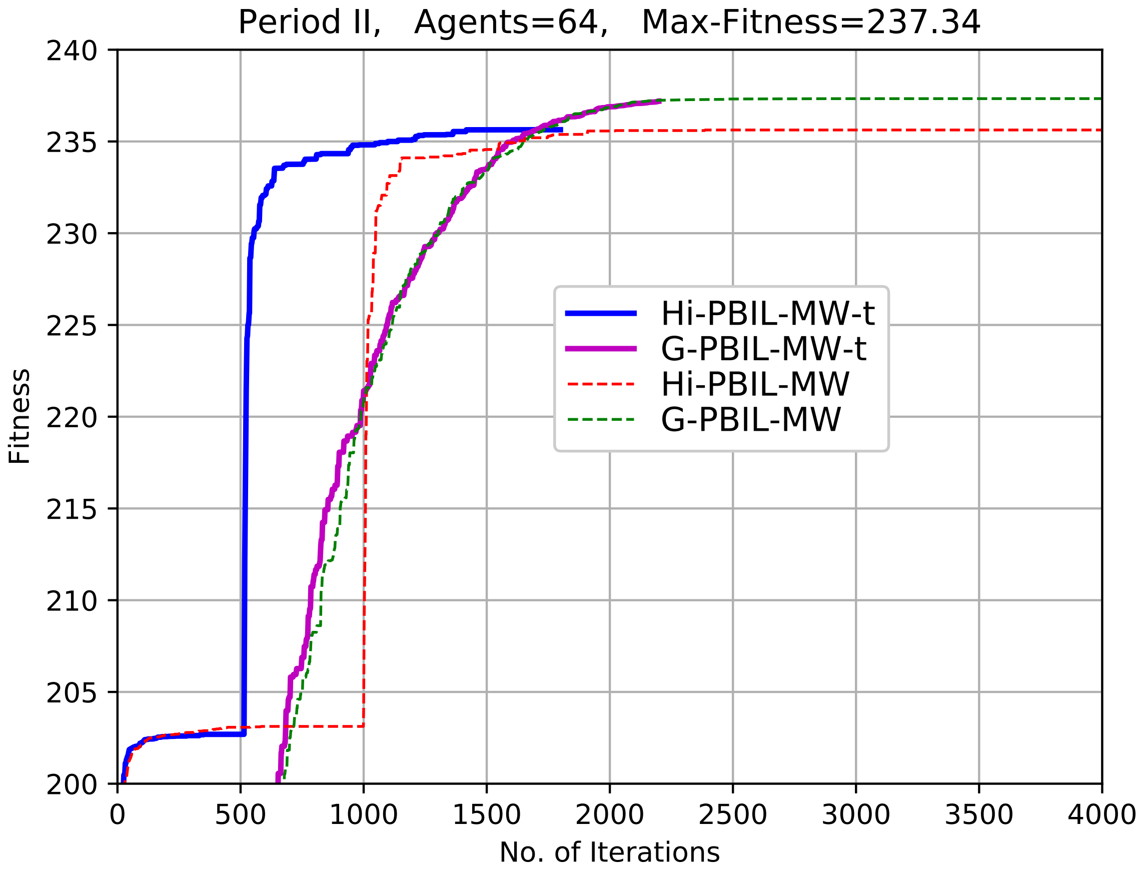


Figure 6.2: Fitness VS. iteration for Period II using different approaches

6.5 Experiment

6.5.1 Data

To demonstrate the potentials for our approach to be utilized in real-world smart grids, we follow the same approach as in our previous works to construct a realistic dataset, which is composed of two diverse sources. The first part is power consumption of smart-meter readings in New Zealand. The second source is power generated by commercialized facilities of wind turbines and solar panels which are coupled with New Zealand meteorological data from NIWA[1].

To assess the ability of sharing power in a regional area, the weather data of the four local cities, Dunedin, Balclutha, Middlemarch and Ranfurly, have been gathered from NIWA. The first two cities are located in a windy coastal zone, and the others are in sunny central Otago. The distance between cities is approximately 40 to 80 km. The power conditions of all agents are then given by subtracting demand from supply. The power(kWh) used is on an hourly basis, and the price $P_r = 20$ $(¢/kWh)$.

However, we know that while the union has a net power surplus at some given hours, then the grand coalition will have trivial solutions. Thus, we consider only periods with overall power deficits. In our data, we have one year of hourly power demand and supply for 240 agents. Among those, four hourly periods with net power deficits are randomly chosen. Furthermore, in comparison with the efficiency of different approaches, we pick five sizes of union at random which are 20, 32, 48, 64 and 80 agents for the all four periods accordingly.

6.5.2 Setup

In our understanding, the algorithms of DS categories are the ones which can guarantee an exact optimal for CSG (Rahwan et al., 2015). Figure 6.3 shows the DP running time for four periods with each size from 4, 8, 12, 16 to 20 agents[2], and all the 20 agents' cases will spend more than 6 hours to get an exact solution.

Our further experiments have utilized 20, 32, 48, 64 and 80 agents for large scale CSG evaluation, and only the result of 20 agents obtained by DP could be available as the ground truth in comparison with other algorithms. Accordingly, instead of DP we will use max-fitness obtained from all approaches in each period to be the

[1] Meteorological data obtained from NIWA's National Climate Database (https://cliflo.niwa.co.nz/ (accessed: 31/Dec/2019)).

[2] The code of the experiments is written and tested in Python 3.6 on Windows 10 PC with *Intel Core i5-4570 PU* and 16 *GB* memory.

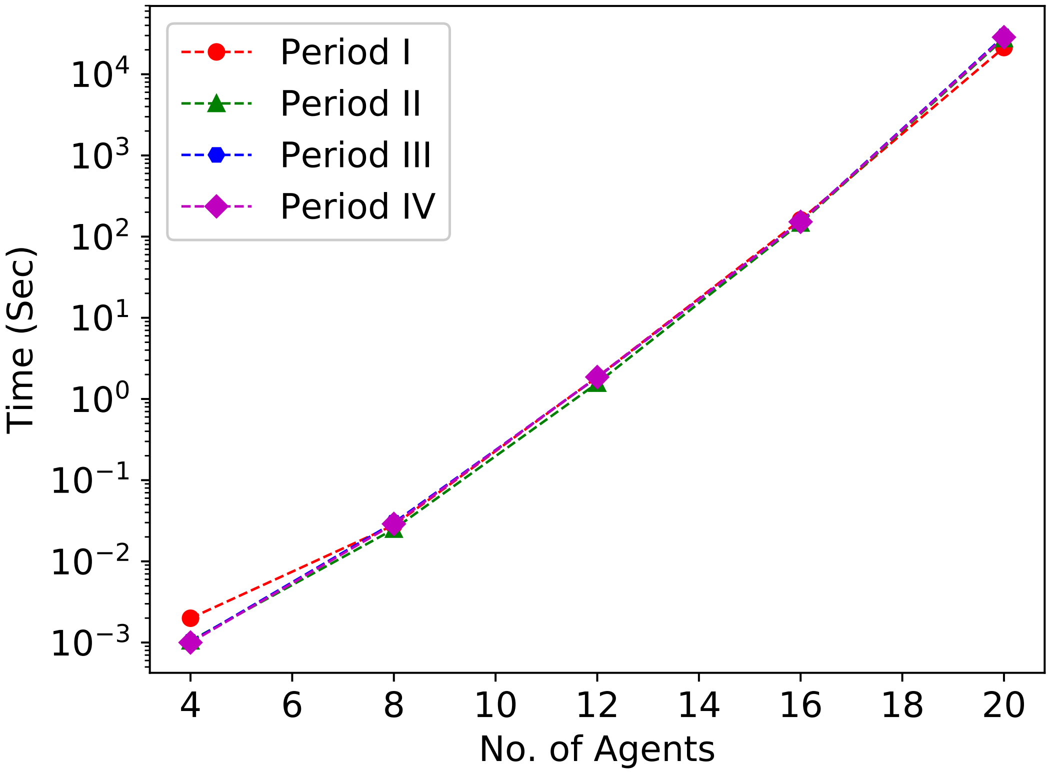


Figure 6.3: DP with log-scale running time according to different number of agents.

comparison index.

The algorithms used in experiments are Global PBIL-MW (G-PBIL-MW), Hierarchical PBIL-MW (Hi-PBIL-MW) and Hierarchical DP (Hi-DP). For two PBIL-MW algorithms we also use fixed iteration and a termination scheme for comparison. Consequently, G-PBIL-MW-t and Hi-PBIL-MW-t will represent algorithms with a termination scheme. Note that DP and Hi-DP are run once, and all PBIL-MW experiments are repeated 20 times to obtain the average. Figure 6.4 shows the running time for all PBIL-MW approaches with each size from 20 to 80 agents accordingly.

6.5.3 Results

Case study of 20 agents

For the four periods, all the algorithms can reach the exact solution as DP's results. The optima of these periods are 111.047, 77.889, 98.845, 124.895 respectively. Table 6.2 shows the time required for each approach. It is clear from the table that excludes DP, that all others are more than 200 times faster, and especially they could

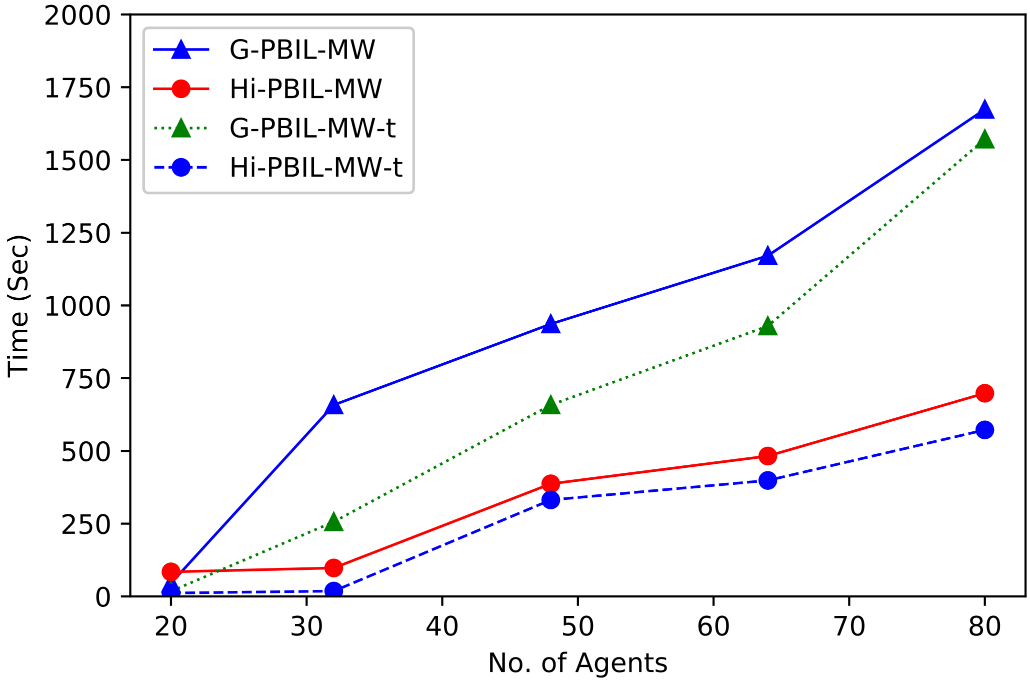


Figure 6.4: PBIL-MW running time according to different number of agents.

Table 6.2: Running time of different approaches for 20 agents

Period	DP	Hi-DP	G-PBIL-MW	Hi-PBIL-MW	G-PBIL-MW-t	Hi-PBIL-MW-t
I	21328.98	16.63	45.05	100.91	14.38	2.11
II	27565.86	14.50	43.94	76.94	36.54	39.91
III	29372.56	0.22	44.75	77.77	17.28	2.70
IV	28605.46	0.22	47.31	81.98	2.77	1.46
Mean	26718.21	**7.89**	45.26	84.40	17.74	11.54

Unit: sec

all reach the same optima as DP.

Case study from 32 to 80 agents

For the large-scale experiments, we select 32, 48, 64 and 80 agents to be the size for the whole four local areas. Hence, from Section 5.3, we know that the bits length of the probability vector are 160, 288, 384 and 560 accordingly. The population and number of iterations are consistently 500 and 4000. All PBIL-MW algorithms have run 20 times, and the results are summarised in Table 6.4. Since DP is unfeasible in these experiments, we follow the hierarchical structure of PBIL-MW to compute the fitness of CSG by hierarchical DP (Hi-DP), although it can be executed under 20 agents, while in some cases the local coalitions will exceed 20 which leads to Hi-DP

Table 6.3: Running time of different approaches for 32 agents

Period	Hi-DP	G-PBIL-MW	Hi-PBIL-MW	G-PBIL-MW-t	Hi-PBIL-MW-t
I	5608.41	622.88	111.58	171.01	43.68
II	423.81	624.66	101.97	752.01	25.34
III	0.16	676.16	89.33	20.28	2.03
IV	0.10	707.83	87.95	82.15	2.35
Mean	1508.12	657.88	97.71	256.36	**18.35**

Unit: sec

becoming unfeasible again.

Furthermore, Table 6.3 shows the running time of different approaches for 32 agents. It is obvious from the table that Hi-DP is becoming slower than others. Thus, we know that Hi-DP will still be unfeasible while the agents' size becomes larger.

In general, we find that G-PBIL-MW and G-PBIL-MW-t always have the max-fitness values, although in some cases the hierarchical approaches will have the same ones. We have explained in Subsection 6.4.2 that the best values of Hi-PBIL-MW will sometimes be less than the global approach. It is clear from Table 6.4 that even in the worst case, e.g. Period II of 64 agents (as shown in Figure 6.2), its best fitness is still close to the maximum.

Besides the accuracy, we have computed the average running time of all approaches as shown in Figure 6.4. The faster converge speed of H-PBIL-t will be an appropriate alternative, while running time is an essential concern in some applications. Like the data of our study which are based on an hourly exchange, in the 80 agents' cases, the G-PBIL-MW or G-PBIL-MW-t will demand nearly half an hour to obtain the better result, and on the contrary, Hi-PBIL-MW and Hi-PBIL-MW-t take only 10 minutes less to gain a plausible solution. However, for the large case, e.g. 200 agents in an hourly-based case, the Hi-PBIL-MW and Hi-PBIL-ME-t could be the only two possible solutions to meet the time requirement.

Table 6.4: Max-Fitness of different approaches

Size	Period	Hi-DP	G-PBIL-MW	Hi-PBIL-MW	G-PBIL-MW-t	Hi-PBIL-MW-t
32	I	**147.75**	**147.75**	**147.75**	**147.75**	**147.75**
	II	164.04	**164.12**	163.70	**164.12**	163.58
	III	**119.29**	**119.29**	**119.29**	**119.29**	**119.29**
	IV	**199.43**	**199.43**	**199.43**	**199.43**	**199.43**
48	I	NA	**218.03**	**218.03**	**218.03**	**218.03**
	II	NA	**179.76**	179.35	**179.76**	179.38
	III	**270.07**	**270.07**	**270.07**	**270.07**	**270.07**
	IV	**234.85**	**234.85**	**234.85**	**234.85**	**234.85**
64	I	NA	**288.16**	**288.16**	**288.16**	**288.16**
	II	NA	**237.34**	235.65	**237.34**	235.65
	III	**384.21**	**384.21**	**384.21**	**384.21**	**384.21**
	IV	**289.82**	**289.82**	**289.82**	**289.82**	**289.82**
80	I	NA	**369.94**	**369.94**	**369.94**	**369.94**
	II	NA	**291.48**	291.39	**291.48**	291.29
	III	NA	**462.95**	462.94	**462.95**	462.94
	IV	**372.40**	**372.40**	**372.40**	**372.40**	**372.40**

unit: cent(¢)/hr

6.6 Summary

In this chapter we know that the new Global PBIL-MW and Hierarchical PBIL-MW demonstrate the superior speed, and a comparable accuracy with other SO approaches for solving a large scale CSG.The proposed Hierarchical PBIL-MW algorithm could obtain the nearly exact solutions as the Global PBIL-MW while outperforming it in the efficiency.

When the number of agents exceeds in a large number there is no other DS method could be used to find an exact solution, then both the Global PBIL-MW and Hierarchical PBIL-MW could provide a viable alternative. Our results have brought evidence to confirm that once the appropriate population size and the number of iterations have been adequately set up, the optimum will be reachable by the new algorithms. The faster response of the new approaches meet the requirement of periodic power exchange for smart grids operating in a co-operative mode in an extensive inter-cities, which can enable the decentralized power systems and improve the penetration of renewable energy.

In this study, we demonstrate the superior speed and comparable accuracy of hierarchical PBIL-MW approaches for solving a large scale CSG. Although the stochastic optimisation algorithms cannot guarantee finding the exact solution, since no other method could be used, then it could provide a viable alternative.

Renewable energy sharing in a large region is a critical component. Some areas have abundant wind power, while others have longer sunshine hours. Consequently, the cooperation among those areas may result in flourishing energy utilisation and reduce the demand for a backup system. Our study can provide a solution. While the subsidy of RE is trivial, the prosumer could still be profitable by utilising the mechanism provided in this study.

Chapter 7

Conclusion and Future Work

7.1 Conclusions

This thesis concerns developing methods to improve the effectiveness and increase the adoption of renewable energy usage. As has been discussed at the outset of this work, the widespread replacement of fossil-fuel energy usage by renewable energy usage is essential in order for our global environment to have a sustainable future. However, most renewable energy sources are dependent on unpredictable weather conditions. The intermittent nature of this production means that a renewable energy prosumer may sometimes produce more energy or less energy than needed. In order for renewable energy to be more economically viable, there needs to be a scheme for sharing energy among prosumers so that those with excess energy can give it to those with energy deficiency. That is the task addressed in this thesis.

The way to deal with this problem is to setup local coalitions of renewable energy prosumers so that energy is shared within the coalitions in an optimally efficient manner. As explained at the beginning three chapters in this work, finding such an optimal coalition arrangement is an example of a Coalition Structure Generation (CSG) problem. The easiest way to find an optimal solution for a given pool of prosumer agents in these circumstances is to examine every possible coalition partition (coalition structure) and evaluate its comparative utility. This is known as an "exhaustive search" (ES) and can be computationally expensive. As shown earlier, the number of such evaluations in ES, even for a pool of twenty agents, can be in the tens of trillions.

The problem for us in the renewable energy domain is that because of the constantly changing weather conditions among the scattered prosumers, the CSG optimisation calculation must be carried out every hour of the day. This means that the ES approach in the CSG optimisation calculation is computationally intractable for a reasonable number of prosumer agents. Therefore, a more computationally feasible stochastic optimisation method must be used, which searches the coalition structure search space in order to find a reasonably good solution, even if it is not the global optimum.

To this end, a number of stochastic optimisation search methods have been investigated in this thesis, including some of our own novel extensions to existing approaches. These search methods have been examined with respect to two different connection arrangements – (1) when the local prosumer networks have a connection to a public utility power grid and can therefore buy needed energy (at a high price) from the grid and sell excess energy (at a low price) to the grid and (2) when the local prosumer networks are isolated from any public utility, which is referred to as "island mode". The overall goal of these investigations has been to find an optimisation approach that is near-optimal (near the global optimum of the given search space) and is computationally efficient (i.e. it does not require a vast amount of computer memory or running time).

Based on these empirical examinations, which have employed realistic parameters drawn from existing consumption and renewable energy data sets, the following conclusions concerning renewable energy can be drawn:

- It is feasible to employ ordinary computer resources to obtain near-optimal energy-sharing coalition structures on an hourly basis that will lead to a more effective and economical use of renewable energy.
- This energy-sharing approach will contribute to a more rapid adoption of existing renewable energy equipment and infrastructure.

The principal contributions of this thesis to these ends are as follows:

- A new modelling framework has been set up that can be used for extensive empirical determinations of near-optimal energy-sharing coalition structures. Based on the condition of every prosumer with either a deficit or a surplus, the proposed model can have profits for prosumers in the alliance by repeatedly forming coalitions. In such an approach, every prosumer may make more profit by sharing energy in the alliance before trading with utilities.

- A detailed empirical study has examined the relative capabilities in this context of various optimal coalition structure search methods, including genetic algorithms (GA), dynamic programming (DP), particle swarm optimisation (PSO), population-based incremental learning (PBIL), and several variants to PBIL. Since the exhaustive search and its improved methods are inefficient and not scalable for the application of our coalition model, we have proposed an improved stochastic algorithm, named Top-k Merit Weighting Population-Based Incremental Learning (PBIL-MW), to generate an optimal coalition model and maximise the benefit of the alliance. When comparing existing state-of-the-art methods with the proposed PBIL-MW algorithm, the latter not only optimises power-sharing within the alliance in our experiments, but also outperforms the state-of-the-art methods by delivering faster running speeds and consuming fewer computer resources.

- Furthermore, we have also proposed a new scalable approach, namely, Hierarchical PBIL-MW (H-PBIL-MW), to solve a coalition formation with a large number of agents in an extended region. Our results have revealed the promising potential of applying the H-PBIL-MW algorithm to deal with co-operative coalitions across extensive areas where there are constraints, such as landform restrictions, in multi-areas which power-share.

- The proposed method for sharing renewable energy among the members (prosumers) in an alliance can be directly applied to smart grids by repeatedly forming coalitions. Consequently, prosumers can share renewable energy with other members and may make more profit in comparison with trading energy with the utilities.

7.2 Limitations

Some of the limitations of this study are listed below:

- The model assumes that prosumers will voluntarily submit to the rules of the trading society, and will not cheat when reporting energy production and consumption.
- The model may still approach computational infeasibility when the number of agents increases to tens of thousands.
- Utility associated with energy is measured on a linear scale in this model similar to money. Thus, larger amounts of energy excess or deficit, which could have non-linear impacts on prosumers, are still only considered in accordance with linear proportions.

7.3 Future Work

Future work concerning this study may involve examining a wider range of energy-producing mechanisms and approaches. There could also be more practical tests of this model involving the interconnections among real wind farms and various prosumers who have solar panels. The model could involve variable rate schemes that could be applied when there are severe energy shortages in some areas. One could also investigate the possibilities of prosumers storing energy credits that could be cashed in at a later time if suitable energy storage devices were available.

References

Alderfer, B., Eldridge, M., & Starrs, T. (2000, jul). *Making connections: Case studies of interconnection barriers and their impact on distributed power projects* (Tech. Rep.). National Renewable Energy Lab., Golden, US. doi: 10.2172/755953

Arabali, A., Ghofrani, M., Etezadi-Amoli, M., Fadali, M. S., & Baghzouz, Y. (2013, jan). Genetic-algorithm-based optimization approach for energy management. *IEEE Transactions on Power Delivery*, *28*(1), 162–170. doi: 10.1109/tpwrd.2012.2219598

Baluja, S. (1994). *Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning* (Tech. Rep. No. No. CMU-CS-94-163). Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science.

Baluja, S., & Caruana, R. (1995). Removing the genetics from the standard genetic algorithm. In *Machine learning: Proceedings of the twelfth international conference* (pp. 38–46).

Banerjee, B., & Kraemer, L. (2010). Coalition structure generation in multi-agent systems with mixed externalities. In *Proceedings of the 9th international conference on autonomous agents and multiagent systems: volume 1-volume 1* (pp. 175–182).

Beheshti, Z., & Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl*, *5*(1), 1–35.

Bellman, E., Richard, & Dreyfus, E., Stuart. (1962). *Applied dynamic programming* (Vol. 2050). Princeton university press.

Bergey. (2012). *EXCEL Wind Turbines Calculations* [Performance Models]. Retrieved from http://bergey.com/wp-content/uploads/excel-1-battery-charging.xls;http://bergey.com/wp-content/uploads/excel-5-grid-intertie.xls

Bharathi, C., Rekha, D., & Vijayakumar, V. (2017, jan). Genetic algorithm based demand side management for smart grid. *Wireless Personal Communications*, *93*(2), 481–502. doi: 10.1007/s11277-017-3959-z

Bodansky, D. (1993). The united nations framework convention on climate change: a commentary. *Yale J. Int'l l.*, *18*, 451.

Bogomolnaia, A., Bramoullé, Y., Carraro, C., & Karos, D. (2019). *Coalition theory network (https://www.coalitiontheory.net/research-areas/coalition-formation-theory)*. Retrieved from https://www.coalitiontheory.net/research-areas/coalition-formation-theory

Bonomi, E., & Lutton, J.-L. (1984, oct). The n-city travelling salesman problem: Statistical mechanics and the metropolis algorithm. *SIAM Review*, *26*(4), 551–568. doi: 10.1137/1026105

Bourazeri, A., & Pitt, J. (2018). Collective attention and active consumer participation in community energy systems. *International Journal of Human-Computer Studies*, *119*, 1–11.

Branker, K., Pathak, M., & Pearce, J. (2011, dec). A review of solar photovoltaic levelized cost of electricity. *Renewable and Sustainable Energy Reviews*, *15*(9), 4470–4482. doi: 10.1016/j.rser.2011.07.104

Capellán-Pérez, I., Mediavilla, M., de Castro, C., Carpintero, Ó., & Miguel, L. J. (2014, dec). Fossil fuel depletion and socio-economic scenarios: An integrated approach. *Energy*, *77*, 641–666. doi: 10.1016/j.energy.2014.09.063

Chakraborty, S., Nakamura, S., & Okabe, T. (2014). Scalable and optimal coalition formation of microgrids in a distribution system. In *Innovative smart grid technologies conference europe (isgt-europe), 2014 ieee pes* (pp. 1–6).

Chalkiadakis, G., Elkind, E., & Wooldridge, M. (2011). Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *5*(6), 1–168.

Changder, N., Dutta, A., & Ghose, A. K. (2016). Coalition structure formation using anytime dynamic programming. In *International conference on principles and practice of multi-agent systems* (pp. 295–309).

Commission, E. U. (2010). *E. U. commission task force for smart grids* (Tech. Rep.). European Union Commission. Retrieved from http://www.ieadsm.org/publication/functionalities-of-smart-grid-and-smart-meters-eutf/

Cuellar, J. (Ed.). (2013). *Smart grid security*. Springer Berlin Heidelberg. doi: 10.1007/978-3-642-38030-3

Dang, V. D., & Jennings, N. R. (2004). Generating coalition structures with finite bound from the optimal guarantees. In *Proceedings of the third international joint conference on autonomous agents and multiagent systems-volume 2* (pp. 564–571).

Deng, X., & Papadimitriou, C. H. (1994). On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, *19*(2), 257–266.

Derksen, C., & Weber, C. (Eds.). (2017). *Smart energy research. at the crossroads of engineering, economics, and computer science*. Springer International Publishing. doi: 10.1007/978-3-319-66553-5

Dulău, L. I., Abrudean, M., & Bică, D. (2016). Smart grid economic dispatch. *Procedia Technology*, *22*, 740–745. doi: 10.1016/j.protcy.2016.01.033

Ellabban, O., Abu-Rub, H., & Blaabjerg, F. (2014). Renewable energy resources: Current status, future prospects and their enabling technology. *Renewable and Sustainable Energy Reviews*, *39*, 748 - 764. Retrieved from http://www.sciencedirect.com/science/article/pii/S1364032114005656 doi: https://doi.org/10.1016/j.rser.2014.07.113

Fadlullah, Z. M., Nozaki, Y., Takeuchi, A., & Kato, N. (2011). A survey of game theoretic approaches in smart grid. In *Wireless communications and signal processing (wcsp), 2011 international conference on* (pp. 1–4).

Folly, K. A., & Venayagamoorthy, G. K. (2013). Power system controller design using multi-population pbil. In *2013 ieee computational intelligence applications in smart grid (ciasg)* (pp. 37–43).

Galli, S. (2009, mar). A simplified model for the indoor power line channel. In *2009 IEEE international symposium on power line communications and its applications.* IEEE. doi: 10.1109/isplc.2009.4913396

Graham, R. L., Knuth, D. E., & Patashnik, O. (1989). *Concrete mathematics: a foundation for computer science*. Addison–Wesley.

Grisales-Noreña, L., Montoya, D. G., & Ramos-Paja, C. (2018, apr). Optimal sizing and location of distributed generators based on PBIL and PSO techniques. *Energies*, *11*(4), 1018. doi: 10.3390/en11041018

Gungor, V. C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., & Hancke, G. P. (2013, feb). A survey on smart grid potential applications and communication requirements. *IEEE Transactions on Industrial Informatics*, *9*(1), 28–42. doi: 10.1109/tii.2012.2218253

Höök, M., & Tang, X. (2013, jan). Depletion of fossil fuels and anthropogenic climate change—a review. *Energy Policy*, *52*, 797–809. doi: 10.1016/j.enpol.2012.10.046

Hoel, M., & Kverndokk, S. (1996, jun). Depletion of fossil fuels and the impacts of global warming. *Resource and Energy Economics*, *18*(2), 115–136. doi: 10.1016/0928-7655(96)00005-x

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* Oxford, England: U Michigan Press.

Hubbert, M. K. (1949). Energy from fossil fuels. *Science*, *109*(2823), 103–109.

Hussain, A., Arif, S. M., & Aslam, M. (2017, may). Emerging renewable and sustainable energy technologies: State of the art. *Renewable and Sustainable Energy Reviews*, *71*, 12–28. doi: 10.1016/j.rser.2016.12.033

IEA. (2011). *Technology roadmap - smart grids* (techreport). International Energy Agency.

IEA. (2017). *Analysis and Forecasts to 2022* (Tech. Rep.). International Energy Agency (IEA).

IEEE. (2011, July). IEEE Guide for Design, Operation, and Integration of distributed resource Island Systems with electric power systems. *IEEE Std 1547.4-2011*, 1-54. doi: 10.1109/IEEESTD.2011.5960751

IRENA. (2019). *Renewable capacity statistics 2019* (Tech. Rep.). Abu Dhabi: The International Renewable Energy Agency [IRENA].

Kádár, P., & Varga, A. (2012). The role of the smart meters in the energy management systems. *IFAC Proceedings Volumes*, *45*(21), 121–125. doi: 10.3182/20120902-4-fr-2032.00023

Kalai, E., & Zemel, E. (1982). Generalized network problems yielding totally balanced games. *Operations Research*, *30*(5), 998–1008.

Karray, F., Karray, F. O., & De Silva, C. W. (2004). *Soft computing and intelligent systems design: theory, tools, and applications.* Pearson Education.

Keinänen, H. (2009). Simulated annealing for multi-agent coalition formation. In *Agent and multi-agent systems: Technologies and applications* (pp. 30–39). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-01665-3_4

Kelly, G. (2011, jun). History and potential of renewable energy development in new zealand. *Renewable and Sustainable Energy Reviews*, *15*(5), 2501–2509. doi: 10.1016/j.rser.2011.01.021

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - international conference on neural networks.* IEEE. doi: 10.1109/icnn.1995.488968

Kitano, H. (2000). RoboCup Rescue: a grand challenge for multi-agent systems. In *Proceedings fourth international conference on MultiAgent systems.* IEEE Comput. Soc. doi: 10.1109/icmas.2000.858425

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection* (Vol. 1). MIT press.

Larranaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, *13*(2), 129–170.

Lee, S. H.-S., Deng, J. D., Peng, L., Purvis, M. K., & Purvis, M. (2017). Top-k merit weighting pbil for optimal coalition structure generation of smart grids. In *International conference on neural information processing* (pp. 171–181).

Lee, S. H.-S., Deng, J. D., Purvis, M. K., & Peng, M., Lizhiand Purvis. (2018a). An improved PBIL algorithm for optimal coalition structure generation of smart grids. In *Workshop on data ming for energy modelling and optimization, damemo), the 22nd pacific-asia conference on knowledge discovery and data mining* (pp. 1–8).

Lee, S. H.-S., Deng, J. D., Purvis, M. K., & Purvis, M. (2018b). Hierarchical Population-Based Learning for Optimal Large-Scale Coalition Structure Generation in Smart Grids. In T. Mitrovic, B. Xue, & X. Li (Eds.), *AI 2018: Advances in artificial intelligence* (pp. 16–28). Cham: Springer International Publishing.

Liu, J., Mei, Y., & Li, X. (2016, oct). An analysis of the inertia weight parameter for binary particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *20*(5), 666–681. doi: 10.1109/tevc.2015.2503422

Lobell, D. B., & Field, C. B. (2007, mar). Global scale climate–crop yield relationships and the impacts of recent warming. *Environmental Research Letters*, *2*(1), 014002. doi: 10.1088/1748-9326/2/1/014002

Logenthiran, T., Srinivasan, D., & Shun, T. Z. (2012, sep). Demand side management in smart grid using heuristic optimization. *IEEE Transactions on Smart Grid*, *3*(3), 1244–1252. doi: 10.1109/tsg.2012.2195686

Manolopoulos, D., Kitsopoulos, K., Kaldellis, J. K., & Bitzenis, A. (2016, aug). The evolution of renewable energy sources in the electricity sector of greece. *International Journal of Hydrogen Energy*, *41*(29), 12659–12671. doi: 10.1016/j.ijhydene.2016.02.115

Marsland, S. (2015). *Machine learning: an algorithmic perspective*. CRC press.

Mauro, N. D., Basile, T. M. A., Ferilli, S., & Esposito, F. (2010). Coalition structure generation with GRASP. In *Artificial intelligence: Methodology, systems, and applications* (pp. 111–120). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-15431-7_12

Michalak, T., Rahwan, T., Elkind, E., Wooldridge, M., & Jennings, N. R. (2016, January). A hybrid exact algorithm for complete set partitioning. *Artif. Intell.*, *230*(C), 14–50. Retrieved from https://doi.org/10.1016/j.artint.2015.09.006 doi: 10.1016/j.artint.2015.09.006

Mohamed, M. A., Eltamaly, A. M., & Alolah, A. I. (2016). Pso-based smart grid application for sizing and optimization of hybrid renewable energy systems. *PloS one*, *11*(8), e0159702.

Mondal, A., & Misra, S. (2013). Dynamic coalition formation in a smart grid: a game theoretic approach. In *2013 ieee international conference on communications workshops (icc)* (pp. 1067–1071).

Nair, N.-K. C., & Zhang, L. (2009). Smartgrid: Future networks for new zealand power systems incorporating distributed generation. *Energy Policy*, *37*(9), 3418–3427.

Nfaoui, H. (2012). Wind Energy Potential. In *Comprehensive renewable energy* (pp. 73–92). Elsevier. doi: 10.1016/b978-0-08-087872-0.00204-3

NIWA. (2019). *CliFlo: NIWA's national climate database on the web.* Retrieved from https://cliflo.niwa.co.nz

Oke, T. R. (2002). *Boundary layer climates*. Routledge. doi: 10.4324/9780203407219

Olsen, T., & Preus, R. (2015). *Small wind site assessment guidelines*. National Renewable Energy Laboratory.

Pasquill, F. (1961). The estimation of the dispersion of windborne material. *Met. Mag.*, *90*, 33.

Pechoucek, M., Marik, V., & Barta, J. (2002, may). A knowledge-based approach to coalition formation. *IEEE Intelligent Systems*, *17*(3), 17–25. doi: 10.1109/mis.2002.1005627

Pedrasa, M. A. A., Spooner, T. D., & MacGill, I. F. (2010, sep). Coordinated scheduling of residential distributed energy resources to optimize smart home energy services. *IEEE Transactions on Smart Grid*, *1*(2), 134–143. doi: 10.1109/tsg.2010.2053053

Peters, H. (2015). *Game theory*. Springer Berlin Heidelberg. doi: 10.1007/978-3-662-46950-7

Philpott, A., Read, G., Batstone, S., & Miller, A. (2019, jan). The new zealand electricity market: Challenges of a renewable energy system. *IEEE Power and Energy Magazine*, *17*(1), 43–52. doi: 10.1109/mpe.2018.2871705

Photovoltaic-software. (2018). *global formula to estimate the electricity generated in output of a photovoltaic system.* https://photovoltaic-software.com/principle-ressources/how-calculate-solar-energy-power-pv-systems.

Pitt, J., Schaumeier, J., Busquets, D., & Macbeth, S. (2012). Self-organising common-pool resource allocation and canons of distributive justice. In *2012 ieee sixth international conference on self-adaptive and self-organizing systems* (pp. 119–128).

Rahwan, T., Michalak, T. P., Wooldridge, M., & Jennings, N. R. (2015). Coalition structure generation: a survey. *Artificial Intelligence*, *229*, 139–174.

Ray, D., & Vohra, R. (2015). Coalition Formation. In *Handbook of game theory with economic applications* (pp. 239–326). Elsevier. doi: 10.1016/b978-0-444-53766-9.00005-7

Rennkamp, B., Haunss, S., Wongsa, K., Ortega, A., & Casamadrid, E. (2017, dec). Competing coalitions: The politics of renewable energy and fossil fuels in mexico, south africa and thailand. *Energy Research & Social Science*, *34*, 214–223. doi: 10.1016/j.erss.2017.07.012

Rothkopf, M. H., Pekeč, A., & Harstad, R. M. (1998, aug). Computationally Manageable Combinational Auctions. *Management Science*, *44*(8), 1131–1147. doi: 10.1287/mnsc.44.8.1131

Saad, W., Han, Z., & Poor, H. V. (2011). Coalitional game theory for cooperative micro-grid distribution networks. In *Communications workshops (icc), 2011 ieee international conference on* (pp. 1–5).

Saber, A. Y., & Venayagamoorthy, G. K. (2012, mar). Resource scheduling under uncertainty in a smart grid with renewables and plug-in vehicles. *IEEE Systems Journal*, *6*(1), 103–109. doi: 10.1109/jsyst.2011.2163012

Saini, N. (2017). Review of selection methods in genetic algorithms. *International Journal Of Engineering And Computer Science*, *6*(12), 22261–22263.

Saleh, M., Esa, Y., Mhandi, Y., Brandauer, W., & Mohamed, A. (2016, oct). Design and implementation of CCNY DC microgrid testbed. In *2016 IEEE industry applications society annual meeting.* IEEE. doi: 10.1109/ias.2016.7731870

Sandholm, T., Larson, K., Andersson, M., Shehory, O., & Tohmé, F. (1999). Coalition structure generation with worst case guarantees. *Artificial Intelligence*, *111*(1-2), 209–238.

Sanjeev Arora, B. B. (2009). *Computational complexity.* Cambridge University Pr. Retrieved from https://www.ebook.de/de/product/8216393/sanjeev_arora_boaz_barak_computational_complexity.html

Sawin, J. L., Sverrisson, F., et al. (2014). Renewables 2012 global status report. *Paris: REN21 Secretariat REN21*.

Sen, S., & Dutta, P. S. (2000). Searching for optimal coalition structures. In *Multiagent systems, 2000. proceedings. fourth international conference on* (pp. 287–292).

Sequeira, T. N., & Santos, M. S. (2018, aug). Renewable energy and politics: A systematic review and new evidence. *Journal of Cleaner Production*, *192*, 553–568. doi: 10.1016/j.jclepro.2018.04.190

Service, T. C., & Adams, J. A. (2010, feb). Constant factor approximation algorithms for coalition structure generation. *Autonomous Agents and Multi-Agent Systems*, *23*(1), 1–17. doi: 10.1007/s10458-010-9124-7

Shehory, O., & Kraus, S. (1998, may). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, *101*(1-2), 165–200. doi: 10.1016/s0004-3702(98)00045-9

Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent systems: algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Siano, P. (2014, feb). Demand response and smart grids—a survey. *Renewable and Sustainable Energy Reviews*, *30*, 461–478. doi: 10.1016/j.rser.2013.10.022

Sørensen, B. (1991, jan). A history of renewable energy technology. *Energy Policy*, *19*(1), 8–12. doi: 10.1016/0301-4215(91)90072-v

Stoustrup, J., Annaswamy, A., Chakrabortty, A., & Qu, Z. (Eds.). (2019). *Smart grid control*. Springer International Publishing. doi: 10.1007/978-3-319-98310-3

Suneco. (2015). *Suneco Wind Turbine output power curve* [Efficiency Chart]. Retrieved from https://www.wind-turbine-generator.com/Horizontal-Wind-Turbine-Generators-ECO-3000W-ECO-3KW.htm

Toffler, A. (1980). *The third wave* (Vol. 484). Bantam books New York.

von Neumann, J. (1944). *Theory of games and economic behavior*. Princeton University Press. Retrieved from https://www.ebook.de/de/product/21174542/john_von_neumann_theory_of_games_and_economic_behavior.html

Wasserstein, R. L., & Lazar, N. A. (2016, apr). The ASA statement on p-values: Context, process, and purpose. *The American Statistician*, *70*(2), 129–133. doi: 10.1080/00031305.2016.1154108

Weber, E. U. (2006, jul). Experience-based and description-based perceptions of long-term risk: Why global warming does not scare us (yet). *Climatic Change*, *77*(1-2), 103–120. doi: 10.1007/s10584-006-9060-3

Wolsink, M. (2012). The research agenda on social acceptance of distributed generation in smart grids: Renewable as common pool resources. *Renewable and Sustainable Energy Reviews*, *16*(1), 822–835.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley and Sons Ltd. Retrieved from https://www.ebook.de/de/product/7515024/michael_wooldridge_an_introduction_to_multiagent_systems.html

Yasir, M. (2018). *Multi-agent based models for the distributed coordination of energy micro-grids* (Unpublished doctoral dissertation). University of Otago.

Yasir, M., Purvis, M., Purvis, M., & Savarimuthu, B. T. R. (2015). Dynamic coalition formation in energy micro-grids. In *PRIMA 2015: Principles and practice of multi-agent systems* (pp. 152–168). Springer International Publishing. doi: 10.1007/978-3-319-25524-8_10

Yasir, M., Purvis, M. K., Purvis, M., & Savarimuthu, B. T. R. (2013, dec). Agent-based community coordination of local energy distribution. *AI & SOCIETY*, *30*(3), 379–391. doi: 10.1007/s00146-013-0528-1

Yeh, D. Y. (1986, dec). A dynamic programming approach to the complete set partitioning problem. *BIT*, *26*(4), 467–474. doi: 10.1007/bf01935053

Zou, C., Zhao, Q., Zhang, G., & Xiong, B. (2016, jan). Energy revolution: From a fossil energy era to a new energy era. *Natural Gas Industry B*, *3*(1), 1–11. doi: 10.1016/j.ngib.2016.02.001