Fast and Scalable Electrical Demand Scheduling for a Large Number of Residential Consumers

Shan He

Master of Information Technology (Honours)

A thesis

Submitted for the degree of Doctor of Philosophy at Monash University

Department of Data Science and Artificial Intelligence, Faculty of IT

March, 2022

Supervisor: Associate Prof. Campbell Wilson

Associate Supervisor: Prof. Mark Wallace and Prof. Ariel Liebman

Fast and Scalable Electrical Demand Scheduling for a Large Number of Residential Consumers

Declaration

I declare that this thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Shan He

March 16, 2022

To Mum, Dad and Ziggy

whom have supported me in their best ways

Acknowledgements

This research was supported by an Australian Government Research Training Program (RTP) Scholarship, Australian Postgraduate Award (APA) and a NICTA (National ICT Australia Ltd, now CSIRO's Data61) top-up scholarship. I would like to express my gratitude and appreciation to all have helped me to make this thesis possible:

- To my supervisors, who have been supportive throughout my PhD and not given up on me when I was at my lowest. In particular thanks to A/Prof Campbell Wilson for his advice and ongoing encouragement to persevere with a PhD project; to Prof. Mark Wallace for the valuable discussions of technical ideas and directions; and to Prof. Ariel Liebman for the priceless opportunities to explore and be exposed to this important area in academic and industry, and the changing energy systems.

- To my milestone panellists and other academic members, including Dr. Chris Mears, Prof. Bala Srinivasan, Prof. Graham Farr, Dr. Hao Wang, Dr. James Saunderson, Dr. Guido Tack, Dr. Graeme Gange and Dr. Ross Gawler for their valuable inputs for the development and presentation of my research, and especially thanks to Prof. Maria Garcia De La Banda for her warm and solid support me at my weakest.

- To members of Faculty of IT, including Prof. Sue McKemmish who created a supportive environment for the PhDs, Helen Cridland for helping me the day I appeared on ABC radio with Robin Williams, Allison Mitchell for her friendly and welcoming administrative support, and especially many thanks to Julie Holden for making me laugh while assisting with my research communication.

- To *Level Six and Half* team, who have given me memories that I will forever remember and miss, including Kevin, Maxim, Tom, Ali, Richard, Steve, David, Liz, Peter, and Laura.

- To many friends who been an important part of my life, including Nichole, Sandra, My, Jip, Alexander, Melanie, Jacob, Rejitha and Semini, and especially thanks to Daniel for everything he has done for me.
- To my family for their unconditional love.
- To the cats that are like angels to my life, including Ragnar who had been a shining star that guided me out of my invisible cocoon, and Ziggy who always remind me that there are something more painful than whatever I am experiencing: his bites and scratches.

An accident back in April 2018 had broken down the equilibrium of my life at that time. The road to recovery has been a journey. The past three years felt like a dream: a long and colourless dream. I felt as if I was drown in an ocean, pushed away by waves to different unknown directions. Living through every wave was my main goal.

After days and days and months and months, I manage to crawl back to the shore. I can now watch the ocean roam, and find my way back whenever taken away by waves again. I feel as if I have finally woken up from a very long dream.

I have completed this thesis. I do not give up what I have started easily. Now I have a life to accomplish, with my Ziggy. Special thanks to those who have shown me the path and strength to get on my feet again, including Jane, Lahna, Claire, Karla, Katrina and Betty.

Shan He

Monash University

March 2022

Vita

Publications arising from this thesis include:

He, S.; Liebman, A.; Rendl, A.; Wallace, M. and Wilson, C. (2014). Modelling RTP-based Residential Load Scheduling for Demand Response in Smart Grids, *International Workshop on Constraint Modelling and Reformulation (Carlos Ansotegui 08 September 2014 to 08 September 2014)*, Lyon, France, pp. 36-51.

He, S.; Wallace, M.; Wilson, C. and Liebman, A. (2017), Fast Electrical Demand Optimization Under Real-Time Pricing (Extended Abstract), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, pp. 4935-4936.

He, S.; Wallace, M.; Gange, G.; Liebman, A. and Wilson, C. (2018). A Fast and Scalable Algorithm for Scheduling Large Numbers of Devices Under Real-Time Pricing, *Principles and Practice of Constraint Programming*, Lille, France, pp. 649-666.

Dayaratne, T.; Rudolph, C.; Liebman, A.; Salehi, M. and He, S. (2019). High Impact False Data Injection Attack against Real-time Pricing in Smart Grids, *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, Bucharest, Romania, pp. 1-5.

Permanent Address: Department of Data Science and Artificial Intelligence, Faculty of IT

Monash University

Australia

This thesis was typeset with LaTeX 2ε[1] by the author.

[1]LaTeX 2ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Glenn Maughan and modified by Dean Thompson and David Squire of Monash University.

Fast and Scalable Electrical Demand Scheduling for a Large Number of Residential Consumers

Shan He, Master of Information Technology (Honours)
shan.he1@monash.edu
Monash University, 2022

Supervisor: Associate Prof. Campbell Wilson
{campbell.wilson}@monash.edu
Associate Supervisor: Prof. Mark Wallace and Prof. Ariel Liebman
{mark.wallace, ariel.liebman}@monash.edu

Abstract

Electricity demand management (DM) is a set of methods that encourage consumers to change their usual consumption patterns, such as reducing the electricity consumption at peak times or balancing preferred consumption times with energy costs. Better management of electricity demand will help to reduce the costs of satisfying peak demand and better utilise the system infrastructure at off-peak usage times.

Demand response (DR) is an important DM method that encourages consumption changes through financial incentives. Real-time pricing (RTP) is a widely considered financial incentive that varies the electricity price with the demand in real time. Implementing RTP is difficult without the aid of information and communication technologies (ICTs), consequently, researchers have been working on developing ICTs and algorithms that automate the DR process for consumers under RTP. Despite positive progress, the existing research requires extensions to consider scalability to enable such algorithms to be applied to large numbers of households in real time. In addition, existing research often sacrifices solution quality or scheduling flexibility for a shorter computation time.

This thesis brings efficiency, optimality, feasibility and scalability together in a DR algorithm. We seek a DR algorithm that provides the best solutions for residential consumers, allows more flexibility for consumption requirements and preferences, and ensures high scalability of the algorithm. Specifically, this thesis tackles a demand scheduling problem for multiple households where households are equipped with shiftable appliances and batteries.

This thesis proposes a fast and scalable algorithm that schedules the appliances and batteries in an iterative and distributed manner. This algorithm uses the Frank-Wolfe (FW) method, primal decomposition, a constraint programming (CP) optimisation model and a linear programming (LP) optimisation model. We call this algorithm Frank-Wolfe-based distributed demand scheduling method (FW-DDSM).

Our experimental results on up to 10000 households show that the number of iterations required to converge to the optimal solution is independent of the number of households. This solution assumes that the appliance and battery scheduling is performed by households in parallel, so the number of households would essentially have no impact on the computation time per iteration. Our algorithm used about ten seconds on modern computers to find the optimal solution for 10,000 households with batteries, making it suitable for real-time applications.

Contents

List of Figures

List of Tables

List of Acronyms

AC air conditioner.

ADMM alternating direction method of multipliers.

AEMO Australian Electricity Market Operator.

AST actual start time.

battery battery energy storage system.

CF care factor.

CG conventional generator.

CLT central limit theorem.

CO continuous optimisation.

CP constraint programming.

CPP critical peak pricing.

CSP constraint satisfaction problem.

DG distributed generator.

DM demand management.

DOD depth-of-discharge.

DR demand response.

DRSP demand response service provider.

DSP demand scheduling problem.

DSP-MB demand scheduling problem for multiple households with batteries.

DSP-MH demand scheduling problem for multiple households.

DSP-SB demand scheduling problem for a single battery.

DSP-SH demand scheduling problem for a single household.

EST earliest start time.

EV electric vehicle.

FIP fractional linear programming.

FW Frank-Wolfe.

FW-DDSM Frank-Wolfe-based distributed demand scheduling method.

GTM game theoretic method.

HEMS home energy management system.

HVAC heating, ventilation, and air conditioning.

ICM interactive coordination method.

ICT information and communication technology.

IPM interior-point method.

job the use of an appliance.

LFT latest finish time.

LLN law of large numbers.

LP linear programming.

MHA (meta)heuristic algorithm.

MIP mixed-integer programming.

MOOP multi-objective optimisation problem.

MPC model predictive control.

MSD maximum succeeding delay.

NE Nash equilibrium.

NEM National Energy Market.

NS non-shiftable.

NSJ non-shiftable job.

OG On-site generator.

OGSA Optimistic Greedy Search Algorithm.

OWCM one-way coordination method.

PAR peak-to-average ratio.

PF power flexible.

PFJ power flexible job.

PFT preferred finish time.

PST preferred start time.

PV photovoltaic.

REG renewable energy generator.

RTE round-trip efficiency.

RTP real-time pricing.

SI shiftable and interruptible.

SIJ shiftable and interruptible job.

SNI shiftable and non-interruptible.

SNIJ shiftable and non-interruptible job.

SOC state-of-charge.

TDP total demand profile.

TJ thermal job.

TOU time-of-use.

TPCM third-party coordination method.

TPE third-party entity.

WS weighted sum.

List of Symbols

$b^{+}_{h,m}$ the charge of the battery in household h at scheduling interval m.

$b^{-}_{h,m}$ the discharge of the battery in household h at scheduling interval m.

$b^{p}_{h,m}$ the activity of the battery in household h at scheduling interval m.

$b^{soc}_{h,m}$ the state-of-charge of the battery in household h at scheduling interval m.

η_h the round-trip efficiency of the battery in household h.

$\bar{b}^{power}_{h}$ the power of the battery in household h.

$\bar{b}^{cap}_{h}$ the maximum capacity of the battery in household h.

$\mathbf{b}^{+}_{h}$ the charge profile of the battery in household h.

$\mathbf{b}^{-}_{h}$ the discharge profile of the battery in household h.

$\mathbf{b}^{p}_{h}$ the discharge of the battery in household h.

$C^{total-itr}_{z}$ the total consumption cost of all households for all scheduling intervals at iteration z.

C^{total} the total consumption cost of all households for all scheduling intervals.

C^{house} the consumption cost of households h for all scheduling intervals.

c^{total}_{n} the total consumption cost of all households at pricing period n.

λ^{c} the weight of the consumption cost.

$\bar{E}^{total}$ the maximum demand limit of all households.

$\bar{E}^{house}_{h}$ the maximum demand limit of household h.

$L_{n,z}^{total-itr-p-fw}$ the total demand of all households at pricing period n at iteration z optimised by the demand response service provider.

$L_{n,z}^{total-itr-p}$ the total demand of all households at pricing period n at iteration z.

$L_{n}^{total-p}$ the total demand of all households at pricing period n.

$L_{m}^{total-s}$ the total demand of all households at scheduling interval m.

$\mathbf{L}_{z}^{total-itr-fw}$ the total demand profile of all households at iteration z calculated by Frank-Wolfe algorithm.

$\mathbf{L}_{z}^{total-itr-p}$ the total demand profile of all households at the granuality of pricing periods at iteration z.

$\mathbf{L}^{total-p}$ the total demand profile of all households at the granuality of pricing periods.

$\mathbf{L}^{total-s}$ the total demand profile of all households at the granuality of scheduling intervals.

$\mathbf{L}^{total-actual}$ the total actual demand profile of all households after applying the probability-based scheduling method.

$\mathbf{L}_{z}^{total-itr-avg}$ the total demand profile of all households at the granuality of pricing periods at iteration z.

$\mathbf{L}^{total-expected}$ the total expected demand profile of all households optimised by the FW algorithm.

$\mathbf{l}_{h}^{house-battery-s}$ the demand profile of household h with a battery at the granuality of scheduling intervals.

$\mathbf{l}_{h}^{house-p}$ the demand profile of household h at the granuality of pricing periods.

$\mathbf{l}_{h}^{house-s}$ the demand profile of household h at the granuality of scheduling intervals.

$l_{h,m}^{house-battery-s}$ the demand of household h with a battery at scheduling interval m.

$l_{h,n}^{house-p}$ the demand of household h at pricing period n.

$l_{h,m}^{house-s}$ the demand of household h at scheduling interval m.

$l_{h,d,m}^{job-s}$ the demand of job d in household h at scheduling interval m.

$\mathbf{l}_{h,d}^{job-s}$ the demand profile of job d at all scheduling intervals in household h.

H the total number of households.

h the index of a household.

$U_z^{total-fw}$ the total inconvenience cost of all households at iteration z optimised by the demand response service provider.

$U_z^{total-itr}$ the total inconvenience cost of all households at iteration z.

U^{total} the total inconvenience cost of all households.

U_h^{house} the inconvenience cost of all jobs in household h.

λ^u the weight of the inconvenience cost.

$u_{h,d}^{job}$ the inconvenience cost of job d in household h.

Z the total number of iterations.

$\alpha_{n,z}$ the step size in time period n at iteration z.

α_z^{tent} the increase in step size at iteration z.

α_z the step size at iteration z.

$\mathbf{d}_z$ the descent direction at iteration z.

$d_{n,z}$ the descent direction in time period n at iteration z.

f_z the objective value at iteration z.

$prob_z$ the probability of the schedule calculated at iteration z to be the best start times.

z the index of an iteration.

D_h the total number of jobs of household h.

$t_{h,d}^{max-delay}$ the maximum suceeding delay of job d in household h.

d the index of a job.

$e_{h,d}$ the power of job d in household h.

$w_{h,d}^{care-factor}$ the care factor of job d in household h.

$t_{h,d}^{latest-finish}$ the latest finish time of job d in household h.

$\rho_{h,d}^{prec}$ the index of job preceding job d in household h.

$t_{h,d}^{actual-start}$ the actual start time of job d in household h.

$t_{h,d}^{earliest-start}$ the earliest start time of job d in household h.

$t_{h,d}^{preferred-start}$ the preferred start time of job d in household h.

$t_{h,d}^{duration}$ the duration of job d in household h.

λ^{max} the weight of the maximum demand objective.

MAX_h^{house} the maximum demand of all jobs in household h.

λ^{par} the weight of the PAR objective.

$PAR_h^{house-l}$ the linearised peak-to-average ratio of the demand profile in household h.

PAR_h^{house} the peak-to-average ratio of the demand profile in household h.

ζ the anxilluary variable for linearing the peak-to-average ratio objective.

K_n the total number of consumption or price levels in the pricing table at pricing period n.

$R_n(\cdot)$ the pricing table at pricing interval n.

$e_{n,1}^{level}$ the lowest consumption level at pricing period n.

e_{n,K_n}^{level} the highest consumption level at pricing period n.

$e_{n,fw}^{level}$ the consumption level for the pricing period n used in the iterative process of calculating the best step size for the pricing master problem.

$e_{n,k}^{level}$ the k^{th} consumption level at pricing period n.

$r_{n,1}^{level}$ the lowest price level at pricing period n.

r_{n,K_n}^{level} the highest price level at pricing period n.

$r_{n,k}^{level}$ the k^{th} price level at pricing period n.

k the index of a consumption or price level in a pricing table.

r_n the price at pricing period n.

M the total number of scheduling intervals.

N the total number of pricing periods.

m the index of a scheduling interval.

n the index of a pricing period.

Chapter 1

Introduction

1.1 Background

Our electrical consumption varies over the day. Typically, the consumption is low at night when household occupants are sleeping, and high in the morning and evening when household occupants are active. Traditionally, our power systems are designed and built to have the capacity to satisfy our demand at anytime and anywhere in the systems (Nicholson, 2012). This demand includes short time periods each year, when the demand becomes significantly higher than the average demand, such as in the afternoon of the hottest day when most people use air-conditioners We call the demand at these times the *peak demand*.

Although the peak demand appears for a very short period of time during a year, some power stations must be built to specifically satisfy the demand at these times. However, these power stations are very expensive to operate, which make electricity significantly more expensive to supply at these peak times. Moreover, these power stations remain idle or operate with reduced capacities for a great deal of the time (Van Den Briel et al., 2013; Mishra et al., 2013), resulting in low utilisation of these generation resources.

Currently, increasing numbers of power stations and the associated infrastructure are due to retire. In the meantime, demands are expected to be more variable due to the increasing use of batteries, electric vehicles and electronic devices (IEA, 2020). Decisions can be made to either keep following traditional approaches of building more power stations and associated infrastructure to satisfy our growing demands, which will be costly, or to seek alternative solutions to reliably supply electricity in a more cost effective and environment friendly way (Finkel et al., 2017b).

Smart grids are such alternative solutions to address the increasing challenges faced by the existing power systems. Smart grids integrate advanced information and communication infrastructure to the existing power systems that allow electricity network operators to monitor and manage the electricity supply and demand in a more efficient way. One of the important features that are enabled by smart grids is enable *demand response (DR).*

DR refers to demand management activities carried out by consumers to modify their consumption patterns in response to financial incentives. The aims of DR is to reduce the peak demand and the overall costs for both electricity suppliers and consumers. By reducing the peak demand, we will improve the reliability, affordability and safety of the electricity supply (U.S. Department of Energy, 2006, 2009; MITei, 2011; Balijepalli et al., 2011; Deng, Yang, Chow and Chen, 2015; Vardakas et al., 2015; Finkel et al., 2017a).

A widely considered financial incentive for DR is the real-time pricing (RTP) scheme which increases the price with demand in real time (Albadi and El-Saadany, 2007). Some utility companies have developed programs that adopt this RTP or similar dynamic pricing schemes to encourage changes in consumption behaviours. However, these programs are difficult to implement in practice because: firstly, they can be challenging and confusing for consumers to monitor and react to a dynamic price (Mohsenian-Rad et al., 2010; Mohsenian-Rad and Leon-Garcia, 2010); and secondly, when reacting to the same pricing information, multiple consumers may simultaneously move their demand from the same expensive time periods to the same cheaper time periods, increasing the actual demand at those cheap time periods instead (Ramchurn et al., 2011).

In addition, battery energy storage system (battery) have become popular for DR as they can not only assist in flattening the peak demand but also compensate for variability in typical renewable energy generators (REGs), making the integration of REGs more viable and our electricity supply more environmental friendly in practice (Vytelingum et al., 2010; Atzeni et al., 2013). However, realising the full benefits of batteries manually is difficult in practice, as we need to consider not only our consumption requirements, but also outputs from on-site generators such as solar panels or small wind turbines if there are any, and the dynamic electricity price and other tariffs such as feed-in tariffs.

The challenges in implementing DR programs in practice have motivated researchers to develop computer algorithms that automatically suggest or determine DR activities for consumers. These algorithms can collect pricing and demand information, learn about consumers' energy needs and preferences, decide the best time to use electricity and/or control electric appliances and devices in response to changes in prices automatically for consumers. One type of computer algorithm that has attracted significant interest from the literature is *optimisation*, which finds the best times to use electric appliances or devices in ways that maximise consumers' benefits and minimise their costs.

This thesis combines the domains of optimisation and DR to investigate a novel algorithm that automates DR activities for consumers under RTP. We aim to assist consumers with adapting the smart grid technologies to better realise the full benefits of DR and keep our energy use at lower prices.

1.2 Motivation

Managing the demand of multiple households at a regional level or even the national level is essential for realising the full potential of DR. As an independent review into the future security of the Australia Electricity Market points out, "[we need to] *manage the load over many consumers aggregating a total amount of load that can have a material impact on the reliability of the national electricity market or a specific local distribution area*" (Finkel et al., 2017a). When the demand of large populations is better managed, it is possible to better predict the behaviour of the power system with reasonable accuracy, leading to a more efficient and stable power network. (Ramchurn et al., 2011).

However, managing the demand of multiple households is a complex task (Ramchurn et al., 2011). Simply letting every household schedule appliances independently against fixed prices without any form of coordination will lead to *load synchronisation* where households schedule appliances to the same time with a low price, creating a demand peak that is higher than normal. *Load synchronisation* is a serious threat to the power system as it can lead to over-demand, causing blackouts and damage to the grid infrastructure, jeopardising the reliability of the power system (Vytelingum et al., 2010). As our society is becoming more reliant on electricity, peak demand is likely to increase significantly over time, making load synchronisation more threatening to the power system (Ramchurn et al., 2011; Van Den Briel et al., 2013).

The key to successfully avoiding load synchronisation is coordination. Loads need to be coordinated or "orchestrated" in a way that stabilises the aggregate demand. One way to coordinate loads is through direct control where a demand response service provider (DRSP), such as a utility company or a third-part service provider, directly controls or advises the operations of appliances and devices. That means, appliances are managed centrally. Let us call this direct control approach the *centralised approach*. Another way to coordinate is through smart pricing, where consumers independently manage their own appliances and devices, and a dynamic pricing scheme is used for guiding households to schedule appliances and devices in the desired way. Let us call this smart pricing approach the *distributed approach*. The distributed approach allows households to preserve a great level of autonomy while being motivated by financial benefits to avoid load synchronisation.

A significant amount of research has investigated the application of these approaches to various DR problems under RTP (Deng, Yang, Chow and Chen, 2015; Vardakas et al., 2015; Scott, 2016; Bayram and Ustun, 2017) to address the need for managing the demand of multiple households at a regional level . However, there are limitations in existing works.

Firstly, many works assume that the prices (Mohsenian-Rad and Leon-Garcia, 2010; Goudarzi et al., 2011; Barbato et al., 2011; Sou et al., 2011; Zhang et al., 2013; Anvari-Moghaddam et al., 2015; Yang et al., 2015; Ma et al., 2016; Manzoor et al., 2017; Hussain et al., 2018; Couraud et al., 2020) or the demands (Vytelingum et al., 2010; Voice et al., 2011; Atzeni et al., 2013; Worthmann et al., 2015) are known and fixed in advance, thus missing the dynamic relationships between the prices and the demands under the RTP. Some works only consider the aggregate demand profiles of households, ignoring the details of appliances and devices (Samadi et al., 2010; Kou et al., 2020). Some studies consider limited constraints for household appliances, preventing consumers from expressing more complex consumption requirements (Mohsenian-Rad and Leon-Garcia, 2010; Lee et al., 2012; Zhao et al., 2013; Van Den Briel et al., 2013; Anvari-Moghaddam et al., 2015; Ma et al., 2016; Manzoor et al., 2017). Some other studies investigate DR problems with batteries only (Vytelingum et al., 2010; Voice et al., 2011; Atzeni et al., 2013; Zhang et al., 2013; Worthmann et al., 2015; Yang et al., 2015; Pelzer et al., 2016; Ghazvini et al., 2017; He et al., 2019; Hossain et al., 2019; Couraud et al., 2020), missing the opportunities of utilising both appliances and batteries to achieve more DR benefits. In summary, most studies focus on particular aspects of a DR problem without considering all aspects.

Secondly, many existing studies apply the centralised methods to solve DR problems (Adika and Wang, 2014; Longe et al., 2017; Pooranian et al., 2018). However, these methods do not scale well with the number of households (Van Den Briel et al., 2013; Zhang et al., 2015; Kou et al., 2020), making them impractical for solving problems for hundreds and/or thousands of consumers, such as the size of a major city suburb in Australia. Some other works investigate distributed methods that distribute computation cost into each households and coordinate households to ensure the total demand profile of all households is flattened as much as possible (Mhanna et al., 2016; He et al., 2019). However, some of these works assume demands or prices are fixed and known in advance (Vytelingum et al., 2010; Atzeni et al., 2013; Worthmann et al., 2015), limiting the flexibility of their methods. Some other works incorporate a dynamic pricing scheme and shiftable jobs in their problems, however, they either require households to broadcast information to others sequentially and iteratively (Mohsenian-Rad et al., 2010; Pilz et al., 2017), which imposes extra burdens on communication networks and limiting the scalability of their methods; or require users to manually tuning some parameters to ensure the best schedules can be found (Yang et al., 2015; He et al., 2019), which makes their methods not general to all problem instances. Furthermore, some existing works increase the efficiency and the scalability of their DR algorithms by ignoring consumer preferences and requirements (Manzoor et al., 2017; Hussain et al., 2018) or finding good solutions instead of the best solutions instead. In summary, none of these existing works can provide flexibility, scalability, optimality and feasibility together with ease of use at the same time in a scheduling algorithm for DR problems.

This thesis aims to address these limitations by proposing a novel algorithm for solving *demand scheduling problems for multiple households (DSP-MHs)*. This algorithm should schedule both appliances and batteries, consider complex consumer requirements and references, and use a dynamic pricing scheme. This algorithm aims to brings efficiency, flexibility, optimality, feasibility and scalability together.

1.3 Research Question and Plan

In order to design the desired algorithm that addresses the limitations in existing works, we ask the following questions:

(Q)1 Can a demand scheduling algorithm be formulated such that the number of iterations required to converge to the best solution is independent of the number of households?

(Q)2 Can this algorithm require minimum manual tuning of parameters and no information broadcasting to achieve the best solutions?

(Q)3 Can it satisfy all of the following requirements:

(R)1 schedule both household appliances and batteries with complex constraints

(R)2 respond to RTP

(R)3 minimise the total consumption cost and the inconvenience of all households

(R)4 reduce the peak demand

(R)5 meet complex consumption requirements and preferences of consumers such as dependency between appliances

In order to answer these questions, we have divided this thesis into four parts:

1. *Literature review*: We conduct our literature review of DR problems and their solving methods in the following two steps: :

   (a) *Review of existing modelling methods*: We investigate existing methods for modelling DR problems for individual and multiple households including the existing mathematical models for appliances, batteries, pricing, costs, inconvenience and consumer requirements and preferences.

   (b) *Review of existing demand scheduling methods*: We study the optimisation methods for scheduling appliances and batteries for individual households; and investigate the algorithms for solving large-scale DR problems.

2. *Problem model*: We formulate the DR problems of this thesis.

3. *Frank-Wolfe-based distributed demand scheduling method*: We propose a novel demand scheduling algorithm for solving the DR problem formulated in Step 2 that satisfies the desired requirements listed in the research questions. We call this novel algorithm the *Frank-Wolfe-based distributed demand scheduling method (FW-DDSM)*.

4. Experiment: We evaluate the efficiency, scalability, optimality and feasibility of our FW-DDSM.

1.4 Thesis Overview

This section outlines the development of the research outcomes for each part of the thesis.

1.4.1 Literature Review

In the first part of the thesis, we compare and discuss the strengths and weakness of existing methods for modelling and solving DR problems. From the review of modelling methods, we identify various types of models for appliances, batteries, pricing, costs and consumption requirements and preferences in the literature. The DR problems studied by existing works can be divided into two types: problems for individual households and problems for multiple households. Existing methods can be categorised into two types: the centralised methods and the distributed methods. We then identify the limitations and areas to expand in existing works, which constitutes the work of this thesis. the details are presented in Chapter 2.

1.4.2 Problem Model

In the second part of this thesis, we formulate the DR problems of this thesis. First, we introduce our models of household appliances, batteries, electricity pricing, costs, consumer inconvenience, and consumption requirements and preferences. In particular, we develop a new pricing function that is created based on bid stacks used by electricity wholesale markets to determine the real cost of electricity, instead of using a generic quadratic function that is commonly considered in existing studies. Second, we identify the parameters, variables, constraints and objective functions of our DR problems, and formulate the DR problems using these models. The details are provided in Chapter 3.

1.4.3 Frank-Wolfe-based Distributed Demand Scheduling Method

In the third part of this thesis, we introduce our novel algorithm that addresses the limitations found in the literature review and satisfies all our research goals. We call this method FW-DDSM. This algorithm consists of the primal decomposition, the Frank-Wolfe (FW) algorithm, a job scheduling module and a battery scheduling module.

The FW-DDSM decomposes a *demand scheduling problem for multiple households with batteries (DSP-MB)* using the primal decomposition into a household subproblem and a pricing master problem. The method for solving the household subproblem includes a job scheduling module and a battery scheduling module. We have proposed three types of solutions for the job scheduling module: a mixed-integer programming optimisation model, a constraint programming optimisation model and the Optimistic Greedy Search Algorithm. Each optimisation model has two versions: one with a data preprocessing algorithm and one without. The battery scheduling model is a linear programming optimisation model that includes the Charnes-Cooper transformation. The method for solving the pricing master problem is based on the Frank-Wolfe (FW) algorithm. The subproblem and the master problem are solved in an iterative manner until the objective value calculated by the pricing master problem does not reduce any more (or reduce no more than 0.01) in any two consecutive iterations.

After the iterations converge to the optimal solution, the best step size calculated by the pricing master problem at each iteration is then used to construct a probability distribution for choosing the actual schedules for households. We call this method the *probability-based scheduling method*. The details are presented in Chapter 4.

1.4.4 Experiments

In the fourth part of the thesis, we demonstrate the effectiveness of our FW-DDSM proposed in Chapter 4 for solving DSP-MBs. In particular, first we compare the three types of methods proposed for solving the household subproblem; second we demonstrate the optimality and scalability of our FW-DDSM; and third we investigate the impacts of some problem parameters, such as the objective weight, the number of sequential jobs, and the battery efficiency and capacity, on the solutions and the scalability of our FW-DDSM. The results show that our FW-DDSM is highly scalable, as the number of iterations for convergence are independent of the number of households, and the scheduling time per household per iteration is minimally affected by the number of households. In particular, the actual solutions produced by our probability-based scheduling method highly approximate, if not match, the optimal solutions of FW-DDSM. Moreover, including batteries in households has limited impacts on and scalability of the FW-DDSM, and yields lower total inconvenience values and more supply cost reductions.

1.5 Contribution

To summarise, the main focus of this thesis is to investigate what method can rapidly schedule appliances and batteries of a large number of households under a dynamic pricing scheme (such as real-time pricing) in a scalable manner, such that complex constraints such as dependency between appliances are enforced, the consumer requirements and preferences are included, the peak demand is reduced, and the total consumption cost and the inconvenience are minimised. Moreover, the computation time should be minimally affected by the number of households and minimum manual tuning of parameters or information broadcasting should be required to achieve the best solutions.

This thesis contributes to the domain of DR by introducing a highly scalable algorithm that achieves the above research goals, called *FW-DDSM*. The implementation of this algorithm can be found at `https://github.com/dorahee/FW-DDSM`. The experiment results have shown that our FW-DDSM is optimised for speed so that it can be used in real-time, making it easy for consumers to take up DR programs under a dynamic pricing scheme. This is a result that has set a new standard in scheduling algorithms for DR.

There are several benefits of our FW-DDSM: 1) the primal decomposition method decomposes a DSP-MB in a straightforward way without the additional transformation steps as in the dual decomposition; 2) enabling households to schedule demands independently in parallel and eliminating the needs for iteratively broadcasting information to households one by one; 3) the achievement of high efficiency and scalability with minimum manual parameter tuning; 4) the achievement of a distributed and iterative algorithm whose convergence speed is minimally affected by the number of households, jobs and batteries; and 5) enabling households to choose from multiple feasible schedules using a probability distribution while causing very limited impacts on the optimality of the results.

We also contribute to the domain of optimisation by demonstrating that a distributed and iterative algorithm that uses primal decomposition, the FW algorithm, a data preprocessing algorithm and optimisation models can solve a mixed-integer convex optimisation problem with linear constraints efficiently with high scalability.

A further contribution of this thesis is that the FW-DDSM can be implemented as part of the software used by both utility companies and consumers to jointly manage demand, in order to reduce peak demand and costs. Electricity is essential in our everyday life.

Meeting the new challenges of power systems while keeping the cost down is crucial for supplying reliable and affordable electricity to consumers. Our research provides a solution for consumers to participate in overcoming the challenges of power systems and reducing the costs of providing electricity, which will in turn improve the benefits for consumers. The outcomes of this research will benefit both electricity suppliers and consumers.

1.6 Thesis Outline

The remainder of this thesis is outlined as follows:

- **Chapter 2 Background**: presents the background information of DR and the literature review DR problem models and solving methods.
- **Chapter 3 Problem Model**: introduces the *demand scheduling problems (DSPs)* of this thesis.
- **Chapter 4 Frank-Wolfe-based Distributed Demand Scheduling Method**: presents the details of our novel FW-DDSM.
- **Chapter 5 Experimental Results**: demonstrates the efficiency, optimality and scalability of our FW-DDSM.
- **Chapter 6 Conclusion**: concludes the findings and outcomes of this thesis and discusses future work.
- **Appendix 2 Power System Now and Then**: provides a non-technical background on power systems.
- **Appendix 3 Optimisation**: introduces an overview of optimisation techniques that support the algorithm development in this thesis.
- **Appendix 3 Additional Data, Code and Figures**: contains additional information for each chapter.

Chapter 2

Background

2.1 Introduction

This chapter provides the context of this thesis, including the background information of *demand response (DR)* in Section 2.2 and the literature review of *demand scheduling problem (DSP)* models and the solving methods in Section 2.3.

2.2 Demand Response

Historically, electricity is produced when it is needed and must be consumed when it is produced. Power stations and electricity networks are designed and built to have the capacity to satisfy electrical demand at any time and any where in the network. It is vital to have a reliable and stable electricity supply at all times, however, there is a trade-off between the reliability and the cost.

Meeting the demand at any time requires extra power stations specially built for peak demand periods. However, these power stations are expensive to operate. Moreover, the peak demand is significantly higher than the average demand and it only occurs for a very short time of a year. It follows that the majority of capacity is idle for a great deal of the time (Van Den Briel et al., 2013; Mishra et al., 2013). This means, the current way to satisfy the peak demand is costly and cause low utilisation of system infrastructure.

Furthermore, current power systems worldwide were built decades ago. The current power stations and associated infrastructure are nearing their optimum working life and require retirement. However, the increasing uses of battery energy storage systems (batteries), electric vehicles (EVs) and electronic devices are introducing more variability in our electrical demands, putting more pressure on these ageing components.

Decisions can be made to either keep building new power stations and associated infrastructure to satisfy our growing demands as the traditional approach, which will be costly, or seek alternative solutions to provide reliable electricity supply in a more economic and environment friendly way (Finkel et al., 2017b). Smart grids are the next-generation power systems that integrate modern information and communication technologies (ICTs) to enable the development of more economic and environment friendly solutions to the above challenges. One of the important goals of smart grid is to realise DR that encourages consumers to better manage their electricity consumption through financial incentives.

DR refers to activities carried out by consumers that actively modify their consumption patterns in response to financial incentives, in order to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized (Albadi and El-Saadany, 2007; Good et al., 2017). A main goal of DR is to reduce the peak demand, and allow our demands to be more predictable and economic to satisfy (Finkel et al., 2017a). The detailed explanations of power systems, smart grids and DR technologies are provided in Appendix A.

2.2.1 Financial Program

Two types of financial programs for DR have been considered in practice: incentive-based or price-based programs. Incentive-based programs reward consumers for reducing consumption at high demand times or when power systems are under emergency conditions (Albadi and El-Saadany, 2007). These programs are usually only used on a small number of hours per year (Siano, 2014), otherwise they will cause “demand fatigue” where participants decrease their responsiveness or exit from the programs if they are called too frequently. For example, interrupting air-conditioners on the hottest days too frequently can frustrate consumers to the point that they will withdraw from a direct load control program during an emergency condition when their services are needed the most (Callaway and Hiskens, 2011).

Price-based programs offer a dynamic pricing scheme that encourages consumers to reduce the consumption at peak times. Consumers can receive lower electricity prices by using energy outside the peak times. These pricing schemes include (Albadi and El-Saadany, 2007; Siano, 2014):

- *time-of-use (TOU)* : TOU scheme sets a different rate for different periods of time. For example, the simplest TOU has two rates: a higher rate for the peak period (e.g. 7am – 11pm from Monday to Friday), and a lower rate for the off-peak period (e.g. all the other times). These rates reflect the average costs of electricity at different periods, however, they change infrequently in a year.
- *critical peak pricing (CPP)*: CPP is similar to TOU, except that the peak rate is raised to several times higher than usual on days when the demand is expected to be exceptionally high relative to available supply. This critical peak rate is usually called during contingencies or high wholesale electricity prices for a limited number of days or hours per year (U.S. Department of Energy, 2006).
- *real-time pricing (RTP)*: RTP is a fully dynamic scheme where the rate varies hourly (or more often) to reflect the actual variations in the system's marginal electricity cost in the wholesale market. Participants are informed about the prices on a day-head, hour-ahead or on a more frequent basis.

These programs allow consumers to reduce their electricity bills by better determining when they use electricity, making the demand more responsive to changes in system conditions and reducing the costs of satisfying the peak demand.

2.2.2 Challenge

These programs are, however, difficult to implement in practice because: firstly, they can be challenging and confusing for consumers to monitor and react to a price that changes during the day (Mohsenian-Rad et al., 2010; Mohsenian-Rad and Leon-Garcia, 2010); and secondly, when reacting to the same pricing information, multiple consumers may simultaneously move their demand from the same expensive time periods to the same cheaper time periods, increasing the demand at those cheap time periods instead (Ramchurn et al., 2011).

These challenges in implementing DR programs have motivated researchers to develop computer algorithms that automatically suggest or determine DR activities for consumers. These algorithms can collect pricing and demand information, decide the best time to use electricity and/or control electric appliances and devices in response to changes in prices automatically for consumers based on their needs and objectives. One type of computer algorithm that has attracted significant interest from the literature is *optimisation*, which finds the best times to use appliances in ways that maximise consumers' benefits and minimise their costs. This thesis investigates such algorithms for automating the DR activities under RTP for consumers.

2.3 Literature Review

A significant amount of research has investigated the application of optimisation algorithms to various DR problems under RTP (Deng, Yang, Chow and Chen, 2015; Vardakas et al., 2015; Scott, 2016; Bayram and Ustun, 2017). Generally, the existing works study a DR problem where devices, such as household appliances or batteries, are scheduled against a pricing scheme to minimise costs or maximise benefits of consumers. A solution to a DR problem includes the best start time and/or the best operation mode for each appliance, and the best charge/discharge decisions for each battery if applicable.

This thesis calls these DR problems as *DSPs* and the algorithms for solving these problems as *demand scheduling methods/algorithms*. This section presents the state-of-the-arts in modelling and solving DSPs. First, we introduce a fundamental glossary of terms that are commonly used for modelling a DSP in Section 2.3.1. Second, we discuss the common DSPs in Section 2.3.2. Third, we analyse the existing demand scheduling methods in Section 2.3.3. The detailed explanations of optimisation algorithms involved in existing works are provided in Appendix B.

2.3.1 Preliminary

We broadly categorise the terms commonly used in the literature into time slots, household appliances, batteries, pricing schemes and cost functions. We clarify the definitions provided in the literature in terms of our project.

Time Slot

When determining the time horizon for scheduling devices or appliances, a day is divided into a finite number of time slots. In the literature, time slots are generally used for pricing and scheduling, however the frequency can be different for these purposes. For example, a time slot for pricing can be 30 minutes or 60 minutes long and a time slot for scheduling can vary from 15 minutes, 20 minutes, 30 minutes to 60 minutes long. Let us define the time slots for pricing and scheduling as Definition 2.1 and Definition 2.2.

Definition 2.1. A *scheduling interval* (or *interval* for short) is a time period when a device can be scheduled to start at the beginning of it or finished at the end of it.

Definition 2.2. A *pricing period* (or *period* for short) is a time period for which a price is calculated.

Table 2.1: Existing works on timeslot models

Length		**References**
Interval	60m	(Conejo et al., 2010; Samadi et al., 2010; Mohsenian-Rad et al., 2010; Yu et al., 2011; Fan, 2011; Joe-Wong et al., 2012; Chavali et al., 2014; Shi et al., 2015; Ogwumike et al., 2015; Mhanna et al., 2016; Ma et al., 2016)
	30m	(Voice et al., 2011)
	20m	(Lee et al., 2012)
	15m	(Barbato et al., 2011; Agnetis et al., 2013; Kuschel et al., 2015)
Period	60m	(Conejo et al., 2010; Samadi et al., 2010; Mohsenian-Rad et al., 2010; Fan, 2011; Yu et al., 2011; Chavali et al., 2014; Shi et al., 2015; Ogwumike et al., 2015; Mhanna et al., 2016; Ma et al., 2016)
	30m	(Yu et al., 2011; Voice et al., 2011)

Household Appliance

Household appliances are commonly categorised into the following types (Yu et al., 2011; Adika and Wang, 2012; Joe-Wong et al., 2012; Lee et al., 2012; Van Den Briel et al., 2013; Agnetis et al., 2013; Sheikhi et al., 2015; Zhang et al., 2015; Anvari-Moghaddam et al., 2015; Mhanna et al., 2016; Ma et al., 2016; Bharathi et al., 2017):

- *non-shiftable (NS)*: appliances that cannot be scheduled to other times and must be turned on when needed, such as lighting, cooking and entertaining appliances.

- *shiftable*: appliances that can be used at a different time of the day without unpleasantly affecting a consumer's lifestyle much.

- *power flexible (PF)*: appliances whose demand can be adjusted without damaging or affecting the functionality of the appliances, such as heating and cooling.

While NS appliances are essential in households, researchers in DR often exclude them in the model because they are inflexible and therefore have no impact on the algorithm design. However, their total demands are often estimated and included in the experiments to better evaluate the impacts of demand scheduling on the costs of households and the peak demand of the power networks.

Shiftable appliances are often further divided into two types: *interruptible* and *non-interruptible*. *Shiftable and interruptible (SI)* appliances can be safely paused and resumed later to finish the desired jobs, such as swimming pool pumps. *Shiftable and non-interruptible (SNI)* appliances must continue running until the full operation cycles are finished, such as washing machines and dryers.

Some works do not allow PF appliances to be shiftable but can vary their demand rates over time. Some other works allow PF appliances to alter their operation times in addition to demand rates. Few works consider PF appliances to have multiple operation modes and their operation modes can be altered to reduce the peak demand instead. In practice, a power flexible appliance is often a thermal appliance that controls the temperature of water or the air in a household.

Appliance Attribute All appliances are modelled by one common attribute, which is the *duration* (Mohsenian-Rad and Leon-Garcia, 2010; Barbato et al., 2011; Tsui and Chan, 2012; Lee et al., 2012; Zhao et al., 2013; Anvari-Moghaddam et al., 2015; Ogwumike et al., 2015; Ma et al., 2016), explained as follows:

- *Duration*: The *duration* is the amount of time required by an appliance to finish a job. For example, a washing machine may need 45 minutes to finish a wash. Note that the *duration* is often a whole number of some time intervals, meaning an appliance must last for at least one interval and cannot stop before an interval ends. This assumption may not always be true in practice, however, it approximates the reality and it is easier for modelling.

Shiftable and NS appliances share one common attribute, which is the *demand rate* (Agnetis et al., 2013; Mohsenian-Rad and Leon-Garcia, 2010; Zhao et al., 2013; Anvari-Moghaddam et al., 2015; Ogwumike et al., 2015; Mhanna et al., 2016; Ma et al., 2016; Nan et al., 2018) or *demand profile* (Lee et al., 2012; Van Den Briel et al., 2013; Agnetis et al., 2013; Ogwumike et al., 2015). PF appliances do not have fixed demand rates or profiles.

Instead, they have *allowed minimum and maximum demand limits* (Mohsenian-Rad and Leon-Garcia, 2010; Li et al., 2011; Nguyen et al., 2012; Tsui and Chan, 2012; Agnetis et al., 2013; Shi et al., 2015; Mhanna et al., 2016; Ma et al., 2016; Hussain et al., 2018; Nan et al., 2018). These attributes are explained as follows:

- *Demand Rate/Profile*: A *demand rate* is the amount of power required by an appliance to safely operate per hour. Some appliances require a fixed demand rate such as lighting. A *demand profile* is for appliances whose demand rates vary over time with the operation cycles, such as washing machines.

- *Allowed Demand Limits*: The allowed demand limit is the maximum or minimum power rate at which a PF appliance is allowed to operate.

Shiftable and PF appliances share several common attributes, such as the *operation time window* (Barbato et al., 2011; Goudarzi et al., 2011; Li et al., 2011; Nguyen et al., 2012; Lee et al., 2012; Ma et al., 2016; Nan et al., 2018; Mohsenian-Rad and Leon-Garcia, 2010; Barbato et al., 2011; Sou et al., 2011; Tsui and Chan, 2012; Anvari-Moghaddam et al., 2015; Ma et al., 2016), the *preferred start time (PST)* (Goudarzi et al., 2011; Ramchurn et al., 2011; Tsui and Chan, 2012; Van Den Briel et al., 2013; Chavali et al., 2014; Ogwumike et al., 2015; Anvari-Moghaddam et al., 2015; Mhanna et al., 2016; Nan et al., 2018), the *operation mode* and the *ideal operation mode* (Hatami and Pedram, 2010; Chavali et al., 2014; Mhanna et al., 2016) , and the *operation phase* (Sou et al., 2011; Ogwumike et al., 2015), which are explained as follows:

- *Operation Time Window*: A shiftable appliance typically has an operation time window that includes an earliest start time (EST) and a latest finish time (LFT). An EST is the earliest time an appliance can be shifted to and a LFT is the latest finished an appliance can be delayed to and finished by.

- *Preferred Start Time*: Some works include a PST for each shiftable appliance to specify the preferences of a consumer. A consumer may prefer an appliance to be scheduled at the PSTs and he/she will be dissatisfied if the appliance is scheduled further away from that time. Few work uses a preference rating for each time slot to specify the level of satisfaction when an appliance is scheduled at that time slot instead. For example, a time slot can have a rating from 1 — 5 for an appliance and a higher rating means the consumer is happier when the appliance is scheduled at that time slot.

- *Operation Mode*: When a demand profile is used, an appliance may also have multiple *operation modes*. Each operation mode has a different demand rate and yields a different output of the appliance. For example, a fridge can have multiple modes such as the stand-by mode, the de-froze mode, the normal mode and the eco-friendly mode. Often, an appliance can switch between modes safely, therefore when managing the demand of a household, the consumer can switch the operation mode of such appliances to reduce the total demand.

- *Ideal Operation Mode*: A consumer may specify the ideal operation mode for an appliance at any time interval.

- *Operation Phase*: When a demand profile is used, an appliance may have *multiple operation phases* instead of multiple *operation modes*. Similar to the operation modes, each operation phase require a different demand rate, however, an operation phase is an essential part of the operation cycle that an appliance requires to finish the desired task. For example, a washing machine can have multiple phases such as the pre-wash phase, the wash phase, the rinse phase and the drain phase. Often, an appliance must go through its phases in order to complete the desired task. However, some appliances allow interruptions between phases, therefore when managing the demand of a household, each phase can be rescheduled at various times to distribute the consumption over a longer period of time.

Some works see interrupting a SI appliance as running the same appliance multiple times and each run is followed by a time gap. They model a SI appliance as a set of SNI "appliance" and each SNI "appliance" is a segment of the SI appliance's operation, simplifying the modelling. For example, a swimming pool pump may need to run for three hours. For demand reduction, it has to be paused twice. Using this pool pump can be seen as running it three times in a three-hour period and each time is non-interruptible.

SI appliances have some unique attributes, such as the *maximum amount of time a SI appliance can be paused in total* (Adika and Wang, 2012; Agnetis et al., 2013) , the *maximum amount of time this appliance can paused continuously at a time* (Agnetis et al., 2013), the *minimum amount of time this appliance must stay on in total* (Agnetis et al., 2013), t*he minimum amount of time this appliance must stay on continuously at a time* (Agnetis et al., 2013), and *a time window that an appliance is allowed to be off* (Agnetis et al., 2013). When considering a SI appliance with multiple interruptible phases, a *between-phase delay* (Sou et al., 2011; Ogwumike et al., 2015), may be introduced to limit the time gap between each phase. Each phase may also has its own EST and LFT.

Appliance Constraint Household appliances are often constrained by a set of rules. We define these rules or constraints as follows:

Definition 2.3. *Scheduling time constraints*: a constraint that requires an appliance to start after an EST, last for a given amount of time and finish before a LFT.

Definition 2.4. *Sequential constraint*: a constraint that requires the operation phases to execute in a given order if an appliance has multiple operation phases.

Definition 2.5. *Between-phase delay constraint*: a constraint that limits the maximum time allowed between (any) two operation phrases if an appliance has multiple operation phases and an interruption is allowed between phases.

Definition 2.6. *Precedence constraint*: a constraint that requires an appliance to start only after a preceding appliance is finished.

Definition 2.7. *Preceding delay constraint*: a constraint that limits the maximum time allowed between (any) two dependent appliances.

Definition 2.8. *Min-max demand constraint*: a constraint that allows an appliance to vary its energy demand between a minimum level and a maximum level.

Definition 2.9. *Min consecutive ON constraint*: a constraint that requires an appliance to run for a minimum number of consecutive time periods before it can be paused or interrupted if this appliance is interruptible.

Definition 2.10. *Max consecutive OFF constraint*: a constraint that limits the total number of consecutive time periods that an appliance is allowed to be paused if this appliance is interruptible.

Definition 2.11. *Max OFF constraint*: a constraint that limits the total number of time periods that an appliance is allowed to be paused if this appliance is interruptible.

Definition 2.12. *Coupling constraint*: a constraint that involves more than one appliances such as the *precedence constraint*, the *sequential constraint*, the *between-phase delay constraint* and the *household demand limit constraint*.

Job Since an appliance can be used multiple times per day, e.g. a consumer can wash clothes twice a day. Many studies have defined a task or a job as Definition 2.13. A job inherits all attributes and constraints of the associated appliance.

Definition 2.13. A *job* is a single use of any appliance.

Demand Limit

Some works adopt *a limit for the maximum demand* of all running appliances or jobs in a household or all households in an area at any time of the day (Samadi et al., 2010; Goudarzi et al., 2011; Li et al., 2012; Muralitharan et al., 2016; Swalehe and Marungsri, 2018; Nan et al., 2018; Mohsenian-Rad and Leon-Garcia, 2010; Ahmed et al., 2017). This limit is also defined as the following constraints:

Definition 2.14. *Household demand limit constraint*: a constraint that limits the total demand of all running appliances in a household at any time.

Definition 2.15. *Area demand limit constraint*: a constraint that limits the total demand of all households served by the same utility company in a given area.

Battery Energy Storage Systems

Models of batteries have been developed to predict the battery output (energy or power) under specific load conditions over the required time. Many battery models have been proposed in the literature on battery modelling and dispatches. Generally they can be divided into two types: empirical models and mechanical models (Beard, 2019a).

Empirical models are early models that use curve-fitting to find general equations from measured data to describe the empirical relationships between measured parameters (e.g. voltage and operating temperature) and the remaining battery capacities under various operating conditions (Muenzel et al., 2015; Beard, 2019a). Mechanical models are physical-based models that employ universal laws to describe the physical processes (e.g. the chemical and electrochemical reactions) of individual components in a battery as functions of their material properties (Ramadass et al., 2004; Beard, 2019a). Mechanical models are the most accurate for predicting the battery output, however, developing such models is complicated and time-consuming due to the vast number of parameters involved in the electrochemical battery process (Muenzel et al., 2015). Empirical models are widely used instead in the demand scheduling literature to reduce the model complexity and the amount of data involved in the optimisation problem.

The level of detail in an empirical model varies with the intended use. For problems that evaluate the economic impacts of using batteries on energy costs, a fixed model that includes only fixed battery parameters is often used. For problems that evaluate the battery lifetime and the impacts of battery health on costs, a variable model that considers some battery parameters as functions of others is used instead.

Battery Attribute A fixed battery model may includes some or all of the following parameters:

- *Capacity*: The amount of energy a battery can store, commonly measured in kWh however, sometimes in Ah (amp per hour). The capacity of any battery has a maximum limit and a minimum limit.

- *Charge or discharge rate*: The amount of energy charge or discharge during a time interval, commonly measured in kW and sometimes in A. This rate has a maximum and minimum limit for any battery.

- *state-of-charge (SOC)* or *depth-of-discharge (DOD)*: The charge or discharge of a battery (or the percentage of the capacity remaining in the battery). In a fixed model, the SOC or DOD of a battery at any time step is described as a linear function of the charge and discharge rates of all times leading to this time step. To prolong the battery life, a maximum limit and a minimum limit are often imposed on the SOC or DOD at any time to avoid emptying or overcharging a battery.

- *Energy level*: The energy remaining in a battery at any give time. Similar to SOC, a maximum limit and a minimum limit are often imposed to maintain the battery health. Sometimes, an additional minimum limit is imposed at the beginning and/or the end of the day to ensure sufficient energy is left to meet the needs of consumers.

- *The charge and discharge efficiencies*: The percentage of energy that is actually absorbed by the battery during charging or the energy that is actually delivered by the battery during discharging. These efficiencies are assumed to be fixed.

- *round-trip efficiency (RTE)*: Sometimes the charge and discharge efficiencies are not considered separately but together as a RTE. This efficiency is also fixed in a fixed model and is generally chosen between 80% and 90%. For example, Pilz et al. (2017) used 0.918, Pandžić and Bobanac (2019) used 0.81 and 0.866, and Pelzer et al. (2016) used 0.8.

The references for these parameters are listed in Table 2.2.

Table 2.2: Existing works on fixed battery models

Attribute	**Citation**
Capacity	(Barbato et al., 2011; Li et al., 2011; Barbato et al., 2011; Voice et al., 2011; Nguyen et al., 2012; Tsui and Chan, 2012; Kim and Giannakis, 2013; Agnetis et al., 2013; Shi et al., 2014, 2015; Kuschel et al., 2015; Anvari-Moghaddam et al., 2015; Marzband et al., 2017; Nan et al., 2018; Zhou et al., 2018; Li et al., 2019)
Max charge or discharge rate	(Barbato et al., 2011; Li et al., 2011; Voice et al., 2011; Barbato et al., 2011; Tsui and Chan, 2012; Nguyen et al., 2012; Kim and Giannakis, 2013; Shi et al., 2014, 2015; Kuschel et al., 2015; Anvari-Moghaddam et al., 2015; Marzband et al., 2017; Zhou et al., 2018; Sperstad and Korpås, 2019)
Min charge or discharge rate	(Barbato et al., 2011; Tsui and Chan, 2012; Anvari-Moghaddam et al., 2015; Nguyen et al., 2012; Kim and Giannakis, 2013; Shi et al., 2014, 2015; Zhou et al., 2018; Li et al., 2019)
Min energy level	(Nguyen et al., 2012; Shi et al., 2014; Barbato et al., 2011; Shi et al., 2015; Marzband et al., 2017; Li et al., 2019)
Start/End-day energy level	(Nguyen et al., 2012; Atzeni et al., 2013; Marzband et al., 2017; Nan et al., 2018; Sperstad and Korpås, 2019; Li et al., 2019)
Charge/discharge efficiency	(Vytelingum et al., 2010; Voice et al., 2011; Kim and Giannakis, 2013; Atzeni et al., 2013; Zhang et al., 2013; Shi et al., 2015; Kuschel et al., 2015; Marzband et al., 2017; Sperstad and Korpås, 2019; Li et al., 2019)

While fixed battery models are widely used in the demand scheduling literature, some works argue that using linear models can leads to infeasible results for battery management and an overestimate of the battery's economic performance, due to the dynamic and non-linear nature of the electrochemical battery process (Azuatalam et al., 2019). Therefore, they are interested in investigating the impacts of including non-linearity in the battery model on the economic performance of a battery in both the short-term and the long-term.

A variable battery model may include a *variable charge rate*, a *variable charge/discharge efficiency* and or a *battery degradation cost* function.

- *Variable charge rate*: In practice, the charge rate of a battery does not remain constant in practice and it changes through stages. The charge stages of a lithium-ion battery and a lead-acid battery are illustrated in Figure 2.1 and Figure 2.2. A two-stage charging model, called the *constant current-constant voltage model*, is commonly used. This model describes the charging current to be fixed when the voltage is still low and increasing and to drop exponentially when the voltage reaches a certain level and becomes fixed until the battery is fully charged (Vagropoulos and Bakirtzis, 2013; Beard, 2019b). Pandžić and Bobanac (2019) took a step further to consider the non-linearity in the battery's actual charging ability as a piece-wise linear function.

- *Variable charge/discharge efficiency*: In practice, the efficiency of a battery increases with SOC and decreases with the charge/discharge current (Safoutin et al., 2015; Azuatalam et al., 2019). Moreover, it is affected by the operating environment of the battery such as the temperature and the humidity. One way to model a variable efficiency is to use a lookup table created through observation in a lab environment, such as the one shown in Table 2.3.

- *Battery degradation cost*: In practice, a battery loses its capacity over time due to the charge-discharge cycles and ageing. This process is called battery degradation. While ageing degradation is inevitable, cycling degradation can be slowed down by better managing the battery profiles, such as avoiding frequent and large fluctuations in the charge and discharge activities and keeping the DOD in a health range (above 20% or 50%).

  Maintaining the battery health has an affect on cost savings for battery owners. In the short term, discharging batteries more often can reduce more peak demand and energy usages, leading to more daily cost savings. However, in the long term, it will shorten the lifetime of the battery, leading to less accumulated cost savings. Pelzer et al. (2016) showed a case study where a newly purchased battery of 20 kWh would lost 21% — 34% of its capacity after one year when the battery degradation was neglected during the scheduling process but only 1% — 2% when the degradation was considered. Abdulla et al. (2018) showed another case study where a battery of 5 kWh would last 3.8 years (160%) longer when the battery degradation was considered.

The simplest way to model the degradation cost is to consider it as a linear function of the charge rate and assume the charge rate to be fixed (Abdulla et al., 2018). Some studies also include other battery parameters, such as the investment costs, the end-of-life capacity and the state-of-charge in the degradation cost function (Muenzel et al., 2015; Pelzer et al., 2016; Wankmüller et al., 2017; Abdulla et al., 2018; Hossain et al., 2019).

Table 2.3: The average charge efficiencies sampled in (Safoutin et al., 2015)

SOC	Charging Current				
	12A	30A	60A	90A	120A
90%	0.993	0.986	0.977	0.968	0.961
70%	0.993	0.986	0.976	0.968	0.960
50%	0.993	0.986	0.976	0.968	0.960
30%	0.993	0.986	0.976	0.967	0.960
10%	0.993	0.985	0.976	0.967	0.960

Battery Constraint A battery's charge and discharge behaviours are governed by a set of rules, defined as follows:

Definition 2.16. *Battery charge and discharge constraints*: constraints that limit a battery to either charge or discharge under the maximum power rate at any time.

Definition 2.17. *Battery capacity constraint*: a constraint that limits the energy level of a battery to be below the maximum allowed capacity and above the minimum allowed capacity at any time.

Definition 2.18. *Battery SOC constraints*: constraints that depict the change of energy level over time between any two consecutive time periods.

Electricity Pricing

Commonly used pricing schemes in the demand scheduling literature include TOU, CPP and real-time RTP. All of these pricing schemes are time varying, however, TOU is the

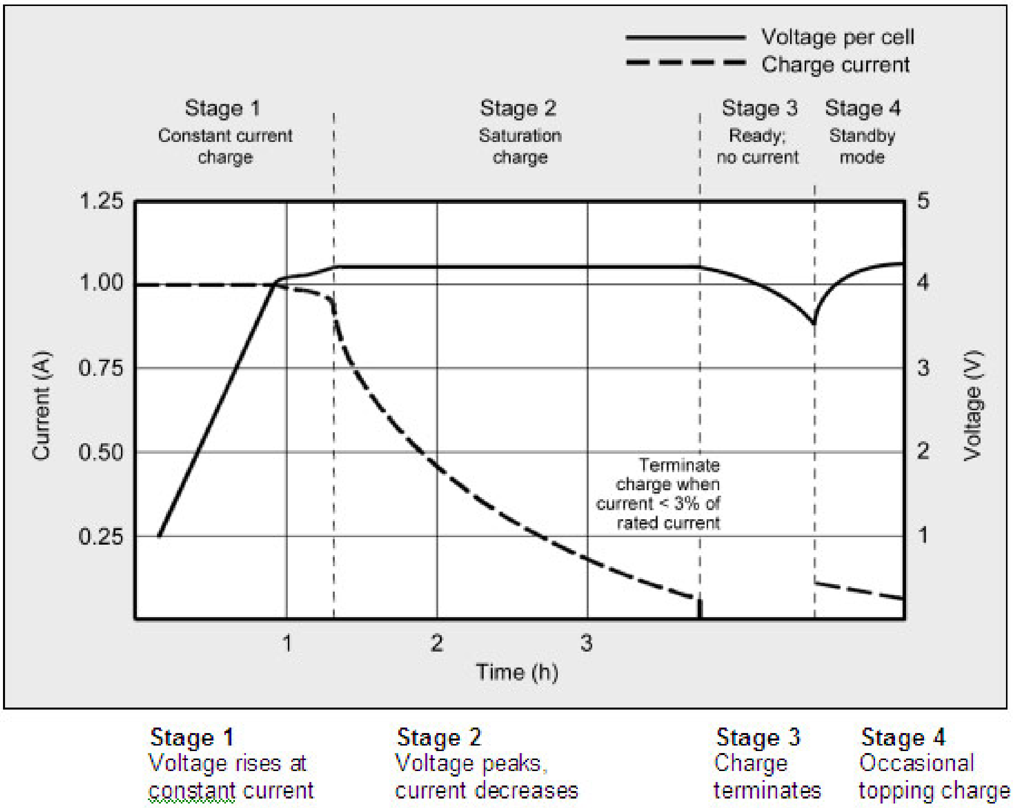


Figure 2.1: Charge states of a lithium-ion battery provided in (Battery University, 2018)

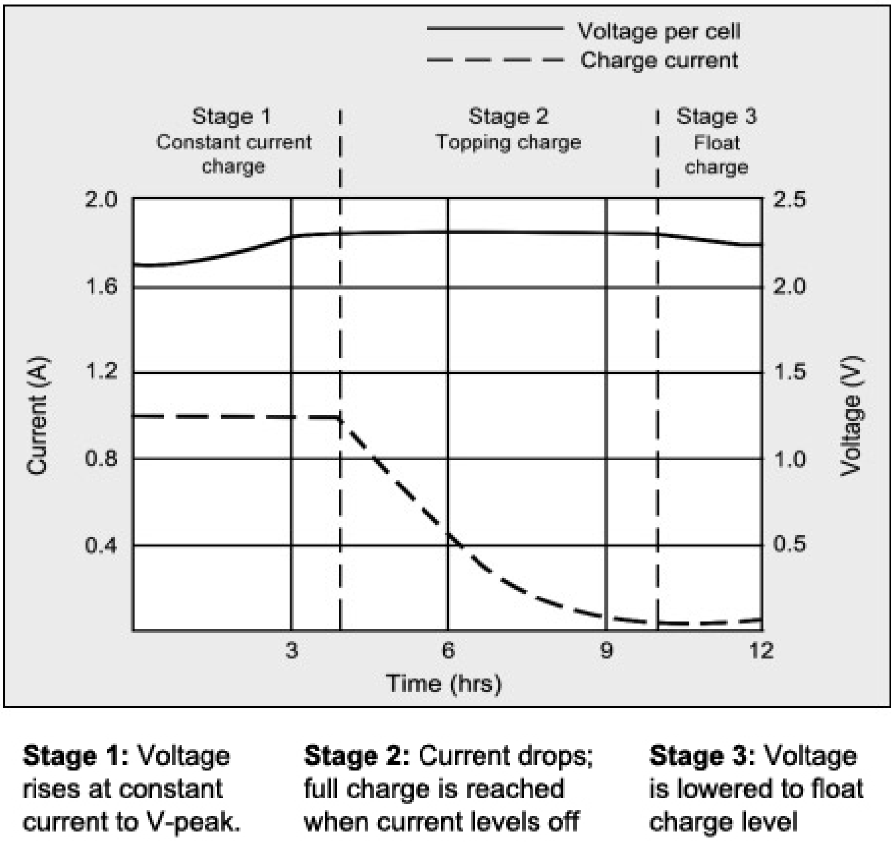


Figure 2.2: Charge states of a lead-acid battery provided in (Battery University, 2019)

most stable, CPP is semi-dynamic and RTP is the most dynamic. The references of these pricing schemes are listed in Table 2.4.

- *Time-of-Use*: The TOU pricing scheme divides a day into a few time-blocks and sets a fixed price for each of these time blocks.

- *Critical Peak Pricing*: The CPP pricing scheme can be considered as a TOU scheme with an additional dynamic price. On a day with usual demand, the CPP prices are the same as the TOU prices. On a day with higher than usual demand, the CPP scheme replaces the price at the peak period with a dynamic price that reflects the new peak demand. This dynamic price may be set a day head or in real ti me based on the expected or actual demand.

- *Real-time Pricing*: The RTP pricing scheme is the most dynamic as it offers a price that varies with the demand every (half an) hour. These prices may be set a day ahead based on the expected demand or in real time based on the actual demand. RTP has been widely recognised as the most effective at incentivising consumers to participate in DR (Ramchurn et al., 2011; Hongbo Zhu, 2018). Consequently, RTP has been adopted by many works to develop demand scheduling algorithms for reducing the costs for consumers and/or suppliers. Commonly, the RTP scheme is modelled as day-ahead prices for the next 24 hours or a pricing function for any time period. When scheduling the demand of a single household, the day-ahead prices are used for scheduling appliances or devices for the next 24 hours. When considering the demand of a wider community, the pricing function is used to decide the price for all households based on the actual demand at every time slot. The pricing function is typically a quadratic function or a step-wise function that is strictly increasing.

Cost

The goal of solving a DSP is to minimise some costs. The most popular costs considered in the literature are the monetary cost and the inconvenience cost. Some other cost-like measurements have also been considered such as the *peak demand*, peak-to-average ratio (PAR), the *load difference* and *power loss*. The references for theses costs or cost-like measurements are provided in Table 2.5.

Table 2.4: Existing works on pricing schemes

Pricing	Citation
TOU	(Hatami and Pedram, 2010; Barbato et al., 2011; Swalehe and Marungsri, 2018; Nan et al., 2018)
CPP	(Hussain et al., 2018; Nan et al., 2018)
Day-ahead RTP	(Barbato et al., 2011; Goudarzi et al., 2011; Voice et al., 2011; Atzeni et al., 2012; Ogwumike et al., 2015; Ma et al., 2016,?; Hussain et al., 2018; Nan et al., 2018)
RTP function	(Samadi et al., 2010; Mohsenian-Rad et al., 2010; Caron and Kesidis, 2010; Goudarzi et al., 2011; Li et al., 2011; Ramchurn et al., 2011; Nguyen et al., 2012; Kim and Giannakis, 2013; Chavali et al., 2014; Veit et al., 2014; Zhang et al., 2015; Shi et al., 2015; Vardakas et al., 2015; Kuschel et al., 2015; Mhanna et al., 2016)

- *Monetary Cost*: The monetary cost is paid by consumers or the utility companies to consume or produce electricity. For clarity, we call the cost paid by consumers the *consumption cost* and the cost paid by utility companies the *supply cost*. Commonly, the consumption cost is calculated as a product of demands and prices while the supply cost is modelled by a function that is strictly increasing and convex, such as a quadratic function or a piece-wise linear function (Atzeni et al., 2013).

- *Inconvenience Cost*: The inconvenience cost, also known as the discomfort or dissatisfaction in the literature, is a penalty that occurs when changing the demand patterns of consumers to reduce the peak demand. Two types of inconvenience costs have been widely considered in the literature:

  - *Start-time related*: a start-time related inconvenience cost is a penalty cost for moving appliances away from their usual consumption times. When calculating such costs, the PSTs are often involved — this cost may increase linearly or exponentially with the distance between the actual start times and the PSTs of appliances.

  - *Consumption related*: a consumption related inconvenience cost is the penalty for varying the total consumption of an appliance or a household. When calculating such costs, a measurement called *utility* is used to evaluate the satisfaction received by the consumers from consuming a good or service. In the context of electricity consumption, the less access and inconvenience to electricity, the less satisfied he/she is. Another way to calculate the convenience

Table 2.5: Existing works on costs

Cost		Citation
Monetary cost		(Conejo et al., 2010; Hatami and Pedram, 2010; Samadi et al., 2010; Mohsenian-Rad et al., 2010; Chen et al., 2011; Goudarzi et al., 2011; Ramchurn et al., 2011; Voice et al., 2011; Mohsenian-Rad and Leon-Garcia, 2010; Barbato et al., 2011; Ren et al., 2011; Tsui and Chan, 2012; Li et al., 2012; Atzeni et al., 2012; Maharjan et al., 2013; Agnetis et al., 2013; Zhao et al., 2013; Chavali et al., 2014; Song et al., 2014; Veit et al., 2014; Sheikhi et al., 2015; Zhang et al., 2015; Kuschel et al., 2015; Ogwumike et al., 2015; Anvari-Moghaddam et al., 2015; Ma et al., 2016; Mhanna et al., 2016; Muralitharan et al., 2016; Rasheed et al., 2016; Jovanovic et al., 2016; Ma et al., 2016; Fioretto et al., 2017; Bharathi et al., 2017; Hussain et al., 2018; Swalehe and Marungsri, 2018)
Inconvenience cost	Start time related	(Mohsenian-Rad and Leon-Garcia, 2010; Chen et al., 2011; Goudarzi et al., 2011; Ramchurn et al., 2011; Adika and Wang, 2012; Tsui and Chan, 2012; Agnetis et al., 2013; Zhao et al., 2013; Chavali et al., 2014; Shi et al., 2015; Anvari-Moghaddam et al., 2015; Ma et al., 2016; Muralitharan et al., 2016; Mhanna et al., 2016; Jovanovic et al., 2016; Longe et al., 2017; Bharathi et al., 2017; Hussain et al., 2018)
	Consumption related	(Conejo et al., 2010; Samadi et al., 2010; Fan, 2011; Kim and Giannakis, 2013; Maharjan et al., 2013; Rahbari-Asr et al., 2014; Zhang et al., 2015; Longe et al., 2017)
Device cost	Battery cost	(Vytelingum et al., 2010; Li et al., 2011; Zhang et al., 2013; Shi et al., 2014, 2015; Yang et al., 2015; Zhou et al., 2018)
	On-site generation	(Atzeni et al., 2013; Chaouachi et al., 2013; Shi et al., 2014)
Others	PAR	(Mohsenian-Rad et al., 2010; Nguyen et al., 2012; Ren et al., 2011)
	Peak demand	(Barbato et al., 2011; Nguyen et al., 2012; Lee et al., 2012)
	Load difference	(Logenthiran et al., 2012,?; Van Den Briel et al., 2013)
	Power loss	(Shi et al., 2014)

cost is through comparing the actual operation modes of appliances and their ideal operation modes.

- *Operation Cost*: Device costs refer to the costs occur for installing, maintaining and operating devices, including household appliances, distributed on-site generators (e.g. diesel generators, combined heat and power systems and fuel cells) and energy storage systems. While they do assist in peak demand reductions and monetary cost reductions, it is important to consider their capital and maintenance costs.

- *Other Costs*: Cost-like measurements refer to measurement that reflects the costs required for operating and maintaining the power systems, such as the PAR, the peak demand, the load difference and power loss. PAR is the ratio of the peak demand and the average demand over a period of time. The peak demand is the maximum demand of all households across a region over a period of time (e.g. a day). The load difference refers to the difference between the actual load profile of consumers and the ideal load profile expected from the utility companies. The power loss is the loss of energy due to (over)heated power lines. A high value for any of these measurements indicates a higher cost for operating and maintaining the power systems.

- *Combined Cost*: Various types of costs are often considered at the same time to balance the needs of consumers for different purposes. For example, consumers may want to reduce their consumption cost but not increase their inconvenience/discomfort too much, thus they may want to find a balance between reducing the consumption cost the inconvenience cost instead of just minimising one or maximising the other. A popular way of finding such balances is through *multi-objective optimisation* (Mohsenian-Rad and Leon-Garcia, 2010; Ramchurn et al., 2011; Chen et al., 2011; Li et al., 2011; Yu et al., 2011; Kim and Giannakis, 2013; Rahbari-Asr et al., 2014; Barbato and Capone, 2014; Deng, Yang, Hou, Chow and Chen, 2015; Anvari-Moghaddam et al., 2015; Vardakas et al., 2015).

Multi-objective optimisation is a discipline in optimisation that deals with optimising multiple conflicting objectives at the same time. The most common way to optimise multiple conflicting objectives is by combing them in a new objective using the weighted sum (WS) method and optimise the new objective instead. In the context of demand scheduling, all considered costs can be summed up together into a new cost function and each original cost can be multiplied by a weight that indicates the importance of this cost among all costs. This new cost function is then used for evaluating the best consumption times or levels for appliances. Note that finding the suitable weight for each original cost is important for achieving the desired balance between multiple needs. However, this thesis does not carry out extended experiments on evaluating the impacts of varying the choices of weights on the problem solutions, as our focus in on the algorithm design and development and the choices of weights do not affect our algorithm choice.

2.3.2 Demand Scheduling Problem

The existing DSPs can be categorised based on the number of users involved in the problem, such as demand scheduling problems for a single household (DSP-SHs) and demand scheduling problems for multiple households (DSP-MHs). First, we introduce the elements of a typical DSP-SH. Then, we present the elements of a typical DSP-MH.

Demand Scheduling Problem for a Single Household

A typical DSP-SH includes three essential elements: a household demand model, a pricing model and objectives. The models of these elements vary across different research works.

Household Demand Model A household demand model can include a job model, a battery model and/or other device model such as an on-site generator model.

Job Model A household job has been modelled:

1. using a forecast demand profile that does not change over time (Vytelingum et al., 2010; Voice et al., 2011; Atzeni et al., 2013; Worthmann et al., 2015) when batteries are considered,
2. using a minimum and a maximum demand limit of the household for any time interval with no job details (Samadi et al., 2010; Kou et al., 2020), or
3. using a set of jobs that have their attributes and constraints (Chavali et al., 2014; Mhanna et al., 2016).

A job can be NS, shiftable or PF as discussed in Section 2.3.1. A job model consists of attributes and constraints, which is described as follows:

- *Attributes*: The widely considered attributes of a job include a fixed demand rate, a duration, an EST and a LFT (Mohsenian-Rad and Leon-Garcia, 2010; Lee et al., 2012; Zhao et al., 2013; Van Den Briel et al., 2013; Anvari-Moghaddam et al., 2015; Ma et al., 2016; Manzoor et al., 2017). Some works have expanded the job to include multiple operation phases and each phase has a different demand rate (Barbato et al., 2011; Sou et al., 2011; Ogwumike et al., 2015). Some other works have added a PST for each job, so consumers can specify their ideal times to run those jobs (Goudarzi et al., 2011; Agnetis et al., 2013; Anvari-Moghaddam et al., 2015). A few works have allowed the demand rate to be adjustable and a maximum and a minimum demand limits are set for any time of the scheduling horizon (Sou et al., 2011; Agnetis et al., 2013; Ma et al., 2016; Manzoor et al., 2017).

- *Constraints*: The *scheduling time constraints* are essential for all jobs. The *sequential constraint* is considered when a job has multiple operation phases (Barbato et al., 2011; Agnetis et al., 2013). Sometimes the *between-phase delay constraint* is used together with the *sequential constraint* (Sou et al., 2011; Anvari-Moghaddam et al., 2015; Ogwumike et al., 2015). The *precedence constraint* and the *preceding delay constraint* are considered when some jobs need to run in a given order (Sou et al., 2011; Anvari-Moghaddam et al., 2015; Ogwumike et al., 2015). The *min-max demand constraint* is used when a job is PF and its demand rate is adjustable (Sou et al., 2011; Agnetis et al., 2013; Ma et al., 2016; Manzoor et al., 2017). The *min consecutive ON constraint*, the *max consecutive OFF constraint* and the *max OFF constraint* are included when a job is interruptible (Agnetis et al., 2013).

Note that, the *precedence constraint*, the *preceding delay constraint*, *sequential constraint*, *between-phase delay constraint* and *household demand limit constraint* are considered as *coupling constraints* because they involve more than one job in a constraint. These constraints couple multiple jobs together, preventing them from being scheduled independently from each other and increasing the difficulties of scheduling.

In addition to constraints for job, the *household demand limit constraint* is sometimes adopted when the total demand of running jobs cannot exceed a given threshold at any time (Samadi et al., 2010; Mohsenian-Rad and Leon-Garcia, 2010; Goudarzi et al., 2011; Sou et al., 2011; Lee et al., 2012; Agnetis et al., 2013; Ogwumike et al., 2015; Hussain et al., 2018).

Battery Model A battery model also includes a set of attributes and constraints. The constraints of a battery have been discussed in Section 2.3.1.

The common attributes of a battery model include a charge or discharge efficiency, a minimum and a maximum charge or discharge rate, a maximum capacity (Vytelingum et al., 2010; Voice et al., 2011; Atzeni et al., 2013; Zhang et al., 2013; Worthmann et al., 2015; Yang et al., 2015; Pelzer et al., 2016; Ghazvini et al., 2017; He et al., 2019; Hossain et al., 2019; Couraud et al., 2020). Some works include additional attributes such as a minimum capacity or a minimum initial capacity (at the start of the scheduling horizon) (Atzeni et al., 2013; Yang et al., 2015; He et al., 2019), and an energy leakage rate (Atzeni et al., 2013). A few works consider a running cost rate (Vytelingum et al., 2010; Voice et al., 2011; Couraud et al., 2020), a storage cost rate (Zhang et al., 2013) or a battery loss rate (Yang et al., 2015) to measure the loss of battery capacity over time due to the charges and discharges.

Other studies consider a more advanced battery model where some attributes are dynamic (Hossain et al., 2019; Pandžić and Bobanac, 2019; Sui and Song, 2020). For example, Pilz et al. (2017) modelled the charging process as a two-stage process. First, the stored energy increased linearly when the battery was on charge. Second, when the battery cell voltage was beyond a terminal voltage, the charging current dropped off exponentially and the stored energy levelled off exponentially accordingly. This work also considered two types of discharging: normal discharging (when the battery was being discharged) and self-discharging (when the battery was not being used). During self-discharging, the stored energy decreased exponentially over time.

Other Device Model Some studies include on-site generators such as photovoltaic (PV) panels, fuel cell co-generation systems and micro combined head and power systems (Mohsenian-Rad and Leon-Garcia, 2010; Barbato et al., 2011; Agnetis et al., 2013; Anvari-Moghaddam et al., 2015; Hossain et al., 2019) in their DSP-SHs. The simplest model for a PV panel is a forecast of energy outputs for the next day (Barbato et al., 2011; Anvari-Moghaddam et al., 2015). A more complex model for a PV panel may include a forecast of solar irradiation, an overall efficiency, the area of the panel, the temperature around the panels and a temperature coefficient of the maximum output power (Hossain et al., 2019). The models for other fossil fuelled on-site generations can include a fuel consumption rate, an efficiency, a minimum and a maximum power output capacity and a ramping up/down rate (Agnetis et al., 2013; Anvari-Moghaddam et al., 2015).

Pricing Model The prices for DSP-SHs are often assumed to be known or forecasted by most works. The commonly used pricing scheme is the day ahead pricing or RTP called by some works (Mohsenian-Rad and Leon-Garcia, 2010; Goudarzi et al., 2011; Barbato et al., 2011; Sou et al., 2011; Zhang et al., 2013; Anvari-Moghaddam et al., 2015; Yang et al., 2015; Ma et al., 2016; Manzoor et al., 2017; Hussain et al., 2018; Couraud et al., 2020), followed by TOU (Agnetis et al., 2013; Zhao et al., 2013), which are all forecasted or given a day ahead. However, few works do not require prices for scheduling (Lee et al., 2012; Van Den Briel et al., 2013; Ogwumike et al., 2015).

Objective The most common objective of solving a DSP-SH is minimisation of the daily consumption cost (Mohsenian-Rad and Leon-Garcia, 2010; Goudarzi et al., 2011; Barbato et al., 2011; Sou et al., 2011; Agnetis et al., 2013; Zhao et al., 2013; Anvari-Moghaddam et al., 2015; Ogwumike et al., 2015; Ma et al., 2016; Manzoor et al., 2017; Pilz et al., 2017; Hussain et al., 2018) , followed by minimisation of inconvenience or dissatisfaction (Vytelingum et al., 2010; Voice et al., 2011; Mohsenian-Rad and Leon-Garcia, 2010; Goudarzi et al., 2011; Agnetis et al., 2013; Zhao et al., 2013; Anvari-Moghaddam et al., 2015; Yang et al., 2015; Ma et al., 2016; Bharathi et al., 2017; Manzoor et al., 2017; Hussain et al., 2018; He et al., 2019; Hossain et al., 2019; Couraud et al., 2020). The inconvenience or dissatisfaction is usually considered as a penalty cost that is calculated based on differences between preferred and actual start times/temperatures/operation modes. Some works consider minimisation of the battery loss (Vytelingum et al., 2010; Voice et al., 2011; Zhang et al., 2013; Yang et al., 2015; He et al., 2019; Hossain et al., 2019; Couraud et al., 2020). Some other works include minimisation of the maximum/peak demand or PAR (Barbato et al., 2011; Lee et al., 2012; Pilz et al., 2017). A few works consider minimisation of operation and maintenance costs when on-site generators and batteries are included (Zhang et al., 2013; Anvari-Moghaddam et al., 2015; Bharathi et al., 2017).

Analysis In summary, we have found from our review of literature on DSP-SHs that:

1. Household demand model: A household demand may include jobs, a battery and other devices such as on-site generators.

   (a) Job model: The simplest household job model is a fixed demand profile, or a variable demand profile that can be adjusted over time. More advanced job models include a set of essential attributes and non-coupling constraints. Complex job models have more attributes and coupling constraints.

(b) Battery model: The simplest battery model is a fixed model where attributes are all fixed. More advanced battery models have variable attributes and even battery operation or degradation costs.

(c) Other device model: Sometimes, additional on-site generators are included in the problem model.

2. Pricing model: Generally, DSP-SHs use price forecasts under various pricing schemes instead of pricing functions where the price changes dynamically with the demand in real time.

3. Objective: The most common objectives are consumption and inconvenience cost. Sometimes, battery costs, peak demand, PAR or operation and maintenance costs of on-site generators are also included.

The household job model can be continuous and linear when a demand profile is considered, or mixed-integer and linear when shiftable jobs are considered. The battery model can be linear when a fixed model is used, or non-linear when a variable model is adopted. The pricing model is non-linear when a dynamic pricing function is used. The objective functions can be linear or non-linear depending on the design of each cost function. Overall, a DSP-SH can be a continuous and linear problem or a mixed-integer non-linear problem depending on the problem models.

Demand Scheduling Problem for Multiple Households

Same as DSP-SHs, DSP-MHs include three key elements: a household job model, a pricing model and objectives. The models of these elements vary across different research works.

Household Demand, Battery and Other Device Models The household demand model is the same as those in DSP-SHs, except that an additional *area demand limit constraint* is sometimes used for restricting the total demand of all households at any time (Samadi et al., 2010).

Definition 2.19. *Area demand limit constraint*: a constraint that limits the total demand of all households served by the same utility company in a given area.

Pricing Model Different fromDSP-SHs, the prices for DSP-MHs are commonly determined by a pricing function which calculates the prices based on the actual demand A pricing function can be a generic convex and increasing quadratic function (Samadi et al., 2010; Mohsenian-Rad et al., 2010; Voice et al., 2011; Atzeni et al., 2013; Chavali et al., 2014; Mhanna et al., 2016; Ghazvini et al., 2017; Pilz et al., 2017; He et al., 2019; Kou et al., 2020), or a piece-wise linear convex function (Li et al., 2011; Kim and Giannakis, 2013). However, some studies still assume the prices are given a day ahead (Vytelingum et al., 2010; Ramchurn et al., 2011; Zhang et al., 2013; Worthmann et al., 2015; Yang et al., 2015; Fioretto et al., 2017; Couraud et al., 2020). Few works require no prices for scheduling (Van Den Briel et al., 2013).

Objective The objectives of a DSP-MH are the same as those in DSP-SHs except that the objective values are calculated from (the total demand profile of) all households instead of from one single household

Analysis In summary, we have identified the following findings from our review of literature on problem formulations for demand scheduling problems for multiple households with batteries (DSP-MBs):

1. The model of a DSP-MH is developed based on the model of a DSP-SH. The household demand model and objectives are the same as those in a DSP-SH, however, an additional constraint that limits the total demand of all households may be considered and the objective values are calculated based on the total costs or the total demand profile of all households.

2. The electricity price is generally calculated using a quadratic, piece-wise linear or step pricing function. However, price forecasts are still used in some works.

DSP-MHs with or without batteries are generally non-linear because of the pricing function, however, they can be continuous when jobs are modelled by demand profiles, or mixed-integer when jobs include a set of attributes and constraints.

2.3.3 Demand Scheduling Method

The existing demand scheduling methods for DSPs can be broadly divided into two types: centralised methods and distributed methods. Centralised methods refer to methods whose the computations are carried out on a single computation unit, while distributed methods refer to those whose computations are distributed to various computation units. Centralised methods can be used for solving DSP-SHs and small-scale DSP-MHs. Distributed methods are more efficient for solving large-scale DSP-MHs.

Centralised Method

Typically, centralised methods collect the full knowledge of appliances in households, compute the optimal schedule for these appliances on a single computation unit, and send back the schedule to each household. Most centralised methods are designed for solving DSP-SHs. However, some studies also adopt centralised methods for solving small DSP-MH.

For example, Mohsenian-Rad and Leon-Garcia (2010) studied a DSP-SH where multiple shiftable and non-interruptible jobs (SNIJs) and an EV was scheduled to minimise the daily consumption cost of all jobs and the delay of starting each job given a price forecast. Goudarzi et al. (2011) investigated a DSP-SH where multiple SNIJs were scheduled under two pricing schemes: the day ahead pricing scheme and the RTP scheme to minimise the daily consumption cost of all jobs and the inconvenience to consumers. Sou et al. (2011) investigated a DSP-SH where multiple SNIJs and shiftable and interruptible jobs (SIJs) were scheduled against known prices to minimise the daily consumption cost of all jobs. Barbato et al. (2011) studied a DSP-SH where multiple SNIJs, a battery and PVs were managed to minimise the daily consumption cost and the peak demand of all jobs against a price forecast. Van Den Briel et al. (2013) investigated a DSP-SH where SNIJs were scheduled to fit a desired load profile given by an utility company. Zhao et al. (2013) studied a DSP-SH where multiple SNIJs/SIJs (rice cookers, air conditioners, electric radiators, water heaters, dishwashers, kettles, humidifiers and clothes dryers) and non-shiftable jobs (NSJs) (lights, computers, cleaners, TVs, irons, hair dryers and fans) were scheduled against given prices to reduce the daily consumption cost of all jobs and the wait time of using each job. Agnetis et al. (2013) investigated a DSP-SH where multiple NSJs, SNIJs, thermal jobs (TJs) (water heaters or air conditioners (ACs)), SIJs, a battery and On-site generators (OGs) such as micro-CHPs and PV were managed to minimise the daily consumption cost and the deviations between the PSTs and the actual start times (ASTs) of all jobs and the climate discomfort of the household under the TOU pricing scheme. Anvari-Moghaddam et al. (2015) considered a DSP-SH where multiple SNIJs, TJs, PVs and a battery were managed against given prices to minimise the cost of purchasing electricity from the main power system, the costs of operating the co-generation generation system and the battery, and the inconvenience to consumers. Ogwumike et al. (2015) considered a DSP-SH where four SNIJs (a washing machine, a dryer, a dish washer and an electric vehicle) and two NSJs (a heater and a TV) were scheduled to reduce the daily consumption cost of all jobs. Pelzer et al. (2016) studied a DSP-SH where a battery was scheduled for energy arbitrage to maximise revenues from discharging back to power systems, and minimising costs of purchasing energy from power systems and the degradation cost from charging/discharging the battery. Ma et al. (2016) studied a DSP-SH where NSJs, SNIJs, and power flexible jobs (PFJs) were managed to minimise the daily consumption cost of all jobs and the discomfort/inconvenience to consumers against give prices. Manzoor et al. (2017) investigated a DSP-SH where three NSJs (a kettle, a toaster and a fridge), one SNIJ (a washing machine) and two PFJs flexible jobs (lights and a heating, ventilation, and air conditioning (HVAC)) were scheduled against given prices to reduce the daily consumption cost of all jobs and the discomfort/inconvenience to consumers. Marzband et al. (2017) investigated a DSP-SH where a household managed its distributed generators (DGs), batteries and jobs to reduce energy costs and protect the stability of its power supply from sudden changes in the demand or supply. Pandžić and Bobanac (2019) worked on a DSP-SH for a single battery to evaluate revenues earned from energy arbitrage using different battery charging models. Hossain et al. (2019) studied a DSP-SH in a microgrid to evaluate the impacts of including the battery degradation cost on energy costs when uncertainties in renewable energy resources, household jobs and electricity prices were considered. Couraud et al. (2020) investigated a demand scheduling problem for a single battery (DSP-SB) for a microgrid to evaluate cost savings and the battery depreciation when the battery lifespan was considered.

Adika and Wang (2014) worked on a DSP-MH where households allowed an aggregator to manage jobs and batteries, in order to reduce the total energy cost. Longe et al. (2017) studied a DSP-MH where households allowed a controller (e.g. the utility company or an aggregator) to compute the best schedules for their jobs and batteries, in order to minimise the total consumption cost and the total inconvenience cost (dissatisfaction cost). Pooranian et al. (2018) investigated a DSP-MH where a smart building scheduled the jobs of multiple homes, a solar panel and a battery to minimise the total cost of buying/selling power from/to the main power system, the cost of maintaining and operating the solar panels and the battery, and the cost of the peak demand charge. Zhou et al. (2018) considered a DSP-MH where batteries and controllable jobs of consumers were managed in ways to minimise the cost of purchasing energy from the power systems and the cost of operating the cloud energy storage, and to maximise the revenues from sharing energies between consumers.

Solving Technique The techniques that have been widely used in centralised methods are mixed-integer programming (MIP) methods, (meta)heuristic algorithms (MHAs) (e.g. genetic algorithms and evolutionary algorithms) and constraint programming (CP) methods. More explanations of these techniques are provided in Appendix B.3.

MIP methods have been widely used in early works. These methods involve implementing problems in a modelling language such as AMPL (Fourer et al., 1989), YALMIP (Lofberg, 2004) or GAMS (Bussieck and Meeraus, 2004), and sending these models to a MIP solver such as Gurobi or CPLEX to retrieve the optimal solutions (Mohsenian-Rad and Leon-Garcia, 2010; Barbato et al., 2011; Sou et al., 2011; Agnetis et al., 2013; Anvari-Moghaddam et al., 2015). Few works have considered CP methods (Fioretto et al., 2017). Similar to MIP methods, CP methods also involve implementing problems in a modelling language, however, CP methods send the models to a CP solver such as Gecode and Chuffed instead. Recent years have seen increasing interests in using MHAs such as evolutionary algorithms (Zhao et al., 2013; Manzoor et al., 2017; Bharathi et al., 2017; Hussain et al., 2018), probability distributions (Van Den Briel et al., 2013), and variants of searching algorithms (Ogwumike et al., 2015; Ma et al., 2016). After investigating each type of method, we have found that:

- *MIP methods*: MIP methods are mature optimisation methods that have been used in industrial applications for decades. They offer powerful modelling capacities, however, their computation costs are exponential and therefore these methods do not scale well in practice.

- *heuristic algorithms*: heuristics algorithms are widely used techniques that scarify optimality for computation speed. However, their capacity to incorporate coupling constraints is limited.

- *CP methods*: similar to MIP methods, CP methods provide powerful modelling capacities and optimality. However, CP and MIP methods are based on different principles. CP methods are based on search and inference while MIP methods are based on mathematical structures of the problems. More details on the differences between CP and MIP methods have been presented in Chapter B. CP methods have proven to be more efficient at solving some scheduling problems (Kelareva et al., 2012; e Silva de Oliveira and de C. Ribeiro, 2015; Maleck et al., 2018; Li and van der Linden, 2018; Laborie, 2018), however, limited studies have been done on applying CP methods in demand scheduling problems.

The strengths and weaknesses of each type of methods are summarised in Table 2.6.

Table 2.6: Strengths and weaknesses of MIP, CP and heuristic algorithms

	Strengths	**Weaknesses**
MIP	- well-known techniques - powerful modelling capacities - guarantee optimality	- exponential complexity and therefore do not scale well
CP	- powerful modelling capacities - guarantee optimality	- exponential complexity and therefore do not scale well - limited studies available
Heuristic	- widely used techniques - polynomial complexity and therefore faster	- does not guarantee optimality - less powerful modelling capacities

Analysis While centralised methods are effective for scheduling devices including jobs, batteries and on-site generators in ways to minimise some total costs while satisfying a given set of constraints, our review of literature has informed us that these methods are widely considered impractical at a large scale by many works (Samadi et al., 2010; Voice et al., 2011; Ramchurn et al., 2011; Tsui and Chan, 2012; Logenthiran et al., 2012; Van Den Briel et al., 2013; Shi et al., 2014; Chavali et al., 2014; Wang et al., 2014; Deng, Yang, Hou, Chow and Chen, 2015; Zhang et al., 2015).

Firstly, centralised methods require complete knowledge of appliances and consumption requirements/preferences across all consumers in advance, which is itself impractical and even a privacy concern to many people.

Secondly, centralised methods do not scale well with the number of consumers or households as the computational complexity often increases exponentially with the problem size (Sou et al., 2011; Van Den Briel et al., 2013; Kim and Giannakis, 2013; Shi et al., 2014; Wang et al., 2014; Mhanna et al., 2016), especially when coupling constraints and a pricing function (when the price depends on the demand in real-time) are used. Ramchurn et al. (2011) tested solving a DSP-MH using the IBM ILOG CPLEX solver on a 64-bit machine with 12GB of RAM. A pricing function was used and no coupling constraint was considered for each household. Their computer could only model and solve a DSP-MH of up to 75 households with up to three SNIJs each and no other coupling constraints. Mhanna et al. (2016) tested solving a DSP-MH using the Gurobi 6.0.5 solver on a 64-bit machine with an Intel ® 4.7GHz Core i7 and 128GB of RAM. Their problem included 1280 households with up to ten appliances each. A pricing function was used and a coupling constraint was set for all households. Their computation time was three days. Kou et al. (2020) tested solving a DSP-MH using the CPLEX solver on a laptop with Intel ® CoreTM i7-8650U 1.90GHz CPU, and 16.00GB RAM. Their problem included 35 households with an air-conditioner, an electric hot water heater, a battery and some jobs each. Their machine could not reach a solution within an hour.

Distributed Method

The drawbacks of the centralised scheduling methods have driven researchers to look for alternative methods that are more scalable, efficient and practical. The distributed scheduling methods are such alternative methods, which allow households to schedule appliances or devices independently and be coordinated through smart pricing. Households do not need to share their detailed consumption needs, addressing the privacy and practicality issue of the centralised scheduling methods; and the computation of scheduling is distributed to each household, addressing the scalability issue.

Coordination of household is a key component of distributed methods, without which will cause *load synchronization* or a *rebound peak* where the peak demand increases above normal levels at the cheaper/cheapest time intervals. For example, consider when households schedule appliances independently against the same set of prices without any coordination, which is often the case in practice, households are likely to move consumption towards the same cheaper/cheapest time intervals, causing *load synchronisation* or a *rebound* *peak* (Mohsenian-Rad and Leon-Garcia, 2010; Mohsenian-Rad et al., 2010; Vytelingum et al., 2010; Goudarzi et al., 2011; Ramchurn et al., 2011; Voice et al., 2011; Chen et al., 2011; Nguyen et al., 2012; Li and Trayer, 2012; Li et al., 2012; Van Den Briel et al., 2013; Veit et al., 2014). Some form of coordination is essential for enabling the significant benefits of managing demand for a large population.

Three types of coordination methods have been proposed in existing works. We have named them the *one-way coordination methods (OWCMs)*, the *interactive coordination methods (ICMs)* and the *third-party coordination methods (TPCMs).* OWCMs reply on information passed from utility companies to households without communication between households or feedback from households to the companies. ICMs use information iteratively broadcasted among households. TPCMs employ a third-party entity, such as a utility company or a demand response service provider (DRSP), to produce pricing signals based demand profiles of households sent from households.

One-way Coordination Method *One-way coordination methods* (OWCMs) include distributed scheduling methods that coordinate households through information passed from the utility company to households, such as day-ahead pricing signals (Ramchurn et al., 2011) or ideal shiftable load profiles (Van Den Briel et al., 2013). Only once-off communication is required between households and the utility company each day.

Vytelingum et al. (2010) studied a DSP-MB where households learnt to choose the best battery capacities for themselves. Each household was considered as an agent that selfishly scheduled its battery against predicted prices of the next day in ways to minimise its own energy cost. At the beginning of each day, each agent calculated a better battery capacity and a new storage profile. More specifically, each agent performed the following actions at the beginning of the day:

1. calculated a desired battery capacity (regardless of its own battery capacity) that would minimise its energy cost given the predicted prices,

2. applied a learning rate to update (increase) its current battery capacity towards the desired battery capacity,

3. scheduled the battery using the updated capacity, and

4. applied another learning rate to decide whether this agent would exercise the new battery schedule in practice.

Their method was tested with simulated households created from typical UK load profiles. Their results showed that when 38% of the simulated population were equipped with a battery, the algorithm would converge to the best desired battery capacity (3.55 kWh on average at which the market prices were flatten to the minimum values) after around two months. This method took months to converge because each agent rescheduled its battery once per day. Moreover, it did not consider dynamic pricing where the price varied with the actual demand, and it assumed household demands to be the same every day. Thirdly, the learning rate required manual tuning to ensure convergence. For example, both learning rates needed to be small enough, e.g. between 0.05 to 0.2, to ensure convergence.

Ramchurn et al. (2011) solved a DSP-MH using pricing signals to coordinate households and an *adaptive mechanism* to avoid load synchronization each day. This method scheduled households jobs against day-prices before a day starts, and used an adaptive mechanism to determine how to execute the schedules during the day. This adaptive mechanism included a learning rate and a probability. The learning rate was designed for moving jobs towards their scheduled time intervals part of the way instead of all the way through. The value of the learning rate affects how close jobs would be moved to their scheduled times. The probability was used for deciding whether the heater would execute the pre-calculated schedule at all. Both the learning rate and the probability were given by the utility company and the same for all households. Their method was tested on a population of 5000 households. Each household had two jobs and 7% to 25% of these households had an electric heater. When the learning rate and the probability were small enough, for example when the probability was 0.05, their method achieved the best schedules for households after 100 days under the assumption that the household demands remained more or less the same during the whole period. The peak demand was reduced by 17% when 7% of the population owned a heater and 22% when 25% of the population owned a heater. However, they assumed the consumption requirements and preferences of all households were more or less the same for all those days.

Voice et al. (2011) investigated a DSP-MB where households were guided through expected market prices to maximise the benefits. Each household was considered as an agent that selfishly scheduled its battery to minimise its own energy cost given the expected market prices from the suppliers. This work proposed a scheduling method that is repeated once every day (each day was an iteration), which is described as follows:

1. The supplier calculated the optimal amount of electricity to purchase for the next day by minimising the expected market costs given the total expected demand profile. Then the supplier calculated new market prices based on that optimal amount of electricity, and passed the market prices to households.

2. Each household scheduled battery against the prices given by the supplier to minimise its energy cost, battery running cost and a penalty cost imposed by the supplier to avoid the storage profile changing greatly between days. Note that the household demand was assumed to be fixed everyday.

The penalty cost imposed for each household was the key for achieving the equilibrium. This cost included a coefficient that needed to be small but greater than a minimum number calculated by the Lyapunov function. This work proved that the use of penalty cost guaranteed convergence and the supplier would make a profit everyday. Their method was tested with 1000 simulated households. The battery capacity of each household was on average 10 kWh. Their results showed that the unique equilibrium was achieved after a number of trading days and the wholesale cost for the supplier was reduced by 16%. Although this work employed a dynamic pricing function, it again required days to achieve convergence. Moreover, the demand was assumed to fixed.

Van Den Briel et al. (2013) solved a DSP-MH using an ideal shiftable load profile to coordinate households and a *probabilistic mechanism* to avoid load synchronisation. This ideal shiftable load profile is the total shiftable load desired by the utility company at each time interval. This method scheduled household jobs by generating a probability distribution based on the ideal load profile. A probability distribution was calculated for each job based on its EST, LFT and the ideal shiftable load profile. The actual start time of a job was randomly selected using its probability distribution. Their experiment results showed that the actual shiftable load profile of all households approximated the ideal shiftable load profile to a great extent when millions of jobs were scheduled using their probability distributions. Their method was fast to run and easy to use. However, the utility company needed to financially motivate consumers to choose ESTs and LFTs for their SNIJs in ways that were more likely to match the ideal shiftable load profile.

These methods provide guidances for households to move appliances or devices in ways that avoid load synchronisation, and have a relatively low computation cost. However, they assumed that the demand does not change over time, did not consider inconvenience to consumers and excluded coupling constraints on appliances. It is unclear that how these methods would incorporate changes in demands and coupling constraints in practice.

Interactive Coordination Method *Interactive coordination methods* (ICMs) refer to distributed scheduling methods that coordinate households through iterative communication among households. Common ICMs include *cooperative game theoretic methods* and *multiagent methods*. ICMs follow a general solving framework, which is described as follows:

1. *Initialisation*: Each household schedules appliances and broadcasts its demand profile to all other households.

2. *Rescheduling*: Upon receiving demand profiles from others, each household reschedules appliances to minimise its cost, assuming the schedules of others were fixed. Then households again broadcast their demand profiles to others.

3. *Iteration*: Households iteratively reschedule appliances based on demand profiles of others and broadcast demand profiles to others.

4. *Convergence*: The iteration continues until they reach a stopping condition, which can be the Nash equilibrium (NE) where no household can reschedule appliances to reduce its cost or some termination conditions (e.g. a timeout or a maximum number of iterations) are met.

The problem solved by each household at the *rescheduling* step is in fact a DSP-SH that can be solved by the methods introduced in Chapter **??**.

Mohsenian-Rad et al. (2010) proposed a *cooperative game theoretic method* that reduced the total cost of households. Each household was considered as a player and they played a game together to find the NE. First, this work formulated the DSP-MH as two separate problems: a PAR minimisation problem and a cost minimisation problem. Second, this work solved each problem following the solving framework of ICMs. At the *rescheduling and iteration* step, households used the interior-point method (IPM) to reschedule appliances. Only one household was allowed to reschedule and broadcast at one time, otherwise the NE will not be reached. The order of households at each iteration was determined by an ordering heuristic algorithm. Their method was tested on up to ten consumers each of which had 10–20 shiftable jobs and 10–20 non-shiftable jobs. The results showed that only minimising the total consumption cost was sufficient to reduce the PAR effectively, however only minimising the PAR did not reduce the total consumption cost effectively. When minimising the total consumption cost, this method converged in 22 iterations, which was 2 iteration per household, and achieved 17% PAR reduction and 18% cost reduction. This work proved that using a convex and strictly increasing function as the cost function was essential for the algorithm to reach the desired NE. Moreover, proportionally charging households would incentivise consumers to participate truthfully as they would not gain benefits from cheating.

Pilz et al. (2017) studied a DSP-MB to evaluate the impacts of considering efficiencies in the battery models on the overall cost reduction and the PAR reduction for multiple households. Each household was considered as an agent that determined the battery operation at each scheduling interval. The possible operations for each battery included charging for the whole or half of the scheduling interval, discharging or doing nothing. The goal of each agent was to satisfy the demand of its loads in ways to minimise the cost of purchasing energy from the main power system and to maximise the earning of selling excessive energy back to the power system. This work applied a best-response algorithm (Shoham and Leyton-Brown, 2008) to solve the problem in an iterative fashion. A maximum number of iterations was imposed to ensure the algorithm terminated within a reasonable time frame and at least a good enough solution could be found. This work applied their method on 25 simulated households created from the openei dataset (U.S. Department of Energy, 2013). They compared the impacts of including and neglecting the charge and discharge efficiencies on cost and PAR reductions. The results showed that the efficiencies affected the battery schedules and the reductions. The PAR reduction was 14% when the efficiencies were considered and 36% otherwise, and cost reduction was between 6 7% when the efficiencies were considered and 7 9% otherwise. This method required broadcasting information to all consumers iteratively, imposing extra work loads on the communication networks. Moreover, consumers were not allowed to update their battery schedules at the same time, which means only one or few consumers could update their schedules per iteration, limiting the scalability of these methods.

Fioretto et al. (2017) proposed a *multiagent* method that reduced both the peak demand and the total cost of all households. Each household was seen as an agent that acted independently and selfishly based on the information received from other agents. All agents repeated selfishly rescheduled their jobs to reduce its own cost and broadcast information to others until no agent could further reduce its own cost. Their method followed the general solving framework of ICMs and some adjustments were made at each step. At the *initialisation* step, households scheduled actuators to minimise the consumption costs using a CP solver. At the *rescheduling* step and the *iteration* step, households calculated the peak demand and the consumption cost their new schedules would save after rescheduling the actuators, and broadcast both the new demand profiles and the savings to all other households. Although after all households had rescheduled and broadcast at a iteration, only the household with the biggest saving would execute the new schedule and all other households would use their previous schedule. If more than one household had the biggest saving, then the household with the smaller identity number would execute its new schedule. At the *convergence* step, households stopped rescheduling and broadcasting when the biggest saving was zero or a termination threshold, e.g. the maximum time or iterations, was reached.

While ICMs offer households independence and the power to collectively reduce the overall cost and the peak demand, they introduce a large communication burden on communication networks as a result of the constant data exchange among households (Mhanna et al., 2016; Yu et al., 2011; Joe-Wong et al., 2012). Households must reschedule and broadcast one by one at each iteration otherwise the method will not converge, which can result in a long computation time especially when the population is large. Furthermore, some works argue that sharing demand profiles with all other households is still a privacy concern (Deng, Yang, Hou, Chow and Chen, 2015; Sheikhi et al., 2015).

Third-party Coordination Method *Third-party coordination methods* (TPCMs) refer to distributed scheduling methods that coordinate households through a third-party entity. This third-party entity can be the utility company or a DRSP. Let us call this third-party entity a DRSP for convenience. The main idea of these methods is that a

DRSP iteratively communicates with households to jointly find the best schedule through smart pricing (Chavali et al., 2014; Li et al., 2011; Mhanna et al., 2016; Chen et al., 2010; Samadi et al., 2010; Wang et al., 2014; Fan, 2011; Veit et al., 2014; Shi et al., 2015). The general solving framework of TPCMs can be described as follows:

1. *Initialisation*: Households independently schedule appliances and report the demand profiles to the DRSP.

2. *Pricing*: Upon receiving demand profiles from households, the DRSP calculates new pricing signals and sends them back to households.

3. *Rescheduling*: Upon receiving pricing signals, households independently reschedule their appliances to reduce their own consumption costs and report the updated demand profiles back to the DRSP.

4. *Iteration*: Then, as in the case of ICMs, households and the DRSP iteratively communicates with each other.

5. *Convergence*: The iteration continues until no consumer can reschedule appliances to receive lower prices from the DRSP or some termination conditions (e.g. a timeout or a maximum number of iterations) are met.

Decomposition TPCMs involve decomposing a DSP-MH into a master problem and a subproblem. The master problem is solved by the DRSP at the *pricing* step. The subproblem is solved by each household at the *rescheduling* step. Let us rename the master problem as the *pricing master problem* and the subproblem as the *household scheduling subproblem.* The pricing signals sent by the DRSP can be artificial prices that are solely for coordinating households or actual prices that reflect the true cost of electricity supply for all households.

Coupling Constraint A challenge in decomposing DSP-MHs is the handling of coupling constraints, such as the *area demand limit constraint.* Dual decomposition handled the coupling constraints by incorporating them into the objective function of the original problem as a constraint violation cost and the dual variables can be seen as the "prices" for violating those coupling constraints. Once decomposed, the subproblem can be considered as the original problem without the coupling constraints and the master problem minimises the prices for violating the coupling constraints based on the solution of the subproblem. A step size is often used for updating the dual variables at each iteration in order to find the optimal solution. More details on decomposition are presented in Section B.3.6 of Chapter B.

Convergence Another challenge of TPCMs is converging to the global optimal solution that truly minimises the total cost of households. Many TPCMs include a parameter, such as a *step size*, a coefficient or a scaling factor, that controls the number of iterations required for convergence (Chen et al., 2010; Samadi et al., 2010; Li et al., 2011; Sheikhi et al., 2015; Zhang et al., 2015; Shi et al., 2015; Mhanna et al., 2016; Kou et al., 2020). Generally this parameter needs to be small enough to guarantee convergence. However, when this parameter is too small, it can lead to more iterations or even oscillation where no convergence will be reached. When this parameter is not small enough, it can lead to premature convergence where the converged solution is sub-optimal. The value of this parameter needs to be chosen carefully and the best value of this parameter can vary from method to method and even problem instance to problem instance. Furthermore, an additional penalty cost is sometimes used to guarantee convergence (Chavali et al., 2014; Mhanna et al., 2016; He et al., 2019). This additional penalty cost is added to the objective of the household sub-problem, penalising the difference between the demand profiles of any two consecutive iterations. The value of this penalty cost may increase proportional to the number of iterations, or inversely proportional to the consumption cost or demand of a household. For example, it may be smaller in early iterations and larger in later iteration, or smaller when the cost or demand is lower and larger otherwise.

Existing TPCMs work on two levels of DSPs: the demand level and the job level.

Demand Level Method The demand level DSPs are concerned with finding the best aggregate demand level per time interval for households, ignoring the details of any appliances or jobs and assuming the demand level varies in a continuous fashion.

Samadi et al. (2010) decomposed a DSP-MH using the dual decomposition, and coordinated households using the dual variables as artificial prices. The dual decomposition was applied to incorporate coupling constraints into the original objective function, and obtain a pricing master problem and a household scheduling subproblem. The gradient projection method was used for solving the master problem and the subproblem At each iteration, the master problem updated the dual variables given the solutions of the subproblem and the subproblem maximised its social welfare given the dual variables updated by the master problem. A step size was used in updating the dual variables and it needed to be tuned manually to guarantee convergence. Their method was tested on 10 households over 24 hourly time intervals.

Atzeni et al. (2013) investigated a DSP-MB where households were equipped with DGs and/or batteries, and coordinated by an aggregator to minimise the overall supply cost. Households were divided into two groups: passive and active households. Passive households consumed electricity as usual without active demand management. Active users engaged in active demand management by using DGs and/or batteries. This work developed a day-ahead optimisation process to schedule DGs and batteries to minimise the total supply-side cost of all households. The proposed method used the proximal decomposition method to formulate the optimisation problem as a Nash game where households communicated with the aggregator iteratively to find the equilibrium together. At each iteration, each household minimised its own energy cost and a penalty cost. This penalty cost included a parameter that required manual tuning to ensure the equilibrium of the Nash game was found. The detailed optimisation process is described as follows:

1. Aggregator broadcasted the supply-side cost function, the grid coefficients and relevant parameters to all households.

2. Each household selected an initial schedule for DGs and batteries arbitrarily, and reported the demand profile (a result of the load profile minus the energy production profile plus the battery storage profile) back to the aggregator.

3. Aggregator calculated the total demand profile of all households and broadcast it to all households.

4. Each household calculated the total demand profile of other households at the previous iteration using the total demand profile of all households sent by the aggregator, rescheduled DGs and batteries to minimise the total supply-side cost and the penalty cost assuming the total demand profile of all other households were unchanged, and reported the updated demand profile back to the aggregator.

5. The aggregator and households repeated the above steps until the total demand profiles of any two consecutive iterations were (nearly) the same.

Their method was tested with 1000 simulated households where 18% of those were active households and the rest were passive. Their method converged in ten iterations, reducing around 12.6% of the average supply price. They also changed the percentage of active users to 6%, 12% and 24% of the population and found the more active users yielded a flatter total demand profile and therefore a more uniform supply price per unit of electricity. Although a dynamic pricing function and DGs are included in the problem, the household demand was assumed to be fixed. Moreover, a parameter required manual tuning to ensure the equilibrium was found and the desired results were achieved.

Worthmann et al. (2015) studied a DSP-MB where households either independently scheduled their batteries against prices given by an aggregator (named as a market maker in this work) to minimise their costs, or were coordinated by the aggregator taking into the account the local generation from solar panels. Four methods were proposed to solve their problem, which are described as follows:

1. Simple controller: This method simply charged the battery when the local generation exceeded the demand and discharged otherwise.

2. Centralised method: The aggregator collected the demands, local generation and battery information from all households, scheduled the batteries centrally and sent the optimal decisions back to households.

3. Decentralised method: Each household scheduled its battery independently without communicating with each other.

4. Distributed method: All households communicated with the aggregator iteratively to find the best schedules for the batteries without needing to communicate with each other. The convergence was achieved by the pricing setting method used by the aggregator.

Their methods were tested with a group of 20 simulated households and another group of 300 simulated households created from the real demand data obtained from Ausgrid (an Australian electricity distribution company). The results showed that the simple controller method performed the worst among the four methods, followed by the decentralised method. The distributed method was better than the decentralised method. The centralised method had the best result, although it was not scalable. It could not find a solution for 300 households due to the significant computation cost of a problem at that size. They assumed both prices and household demands were fixed and inflexible.

Kou et al. (2020) decomposed the DSP-MH using a method called the *alternating direction method of multipliers (ADMM)* (Boyd et al., 2011). Households were coordinated using the primal variables and the dual variables introduced during decomposition method as artificial prices. The ADMM was applied to obtain a pricing master problem and a household scheduling subproblem. The coupling constraint was incorporated to the objective functions of the master problem and the subproblem. This decomposition method introduced two types of ancillary variables called the primal residuals and the dual variables. Same as the dual decomposition, the dual variables in ADMM were used for minimising the violation of the coupling constraint. The primal residuals were used for updating the demand levels of households. At each iteration, the pricing master problem updated the primal residuals and the dual variables using the CPLEX solver, and the household scheduling subproblem adjusted the demand levels given the updated primal residuals and and the dual variables using the BARON and SCIP solvers. A scaling factor, called the penalty factor, was involved in solving the household scheduling subproblem and updating the dual variables in the pricing master problem. The value of this factor needed manual tuning to guarantee convergence. Their method was tested on 605 households on a scheduling horizon of 96 15-minute intervals. The method converged after 23 iterations in about 188s. The speed of convergence depended on the penalty factor, meaning that a smaller value could yield better results but led to more iterations or no convergence while a larger factor might be quick to converge but gave a sub-optimal solution.

Job Level Method The job level DSPs focus on finding the best start time and/or the demand rate for each job per time interval, considering the (coupling) constraints of jobs, and inconvenience to consumers.

Li et al. (2011) decomposed the DSP-MH using the primal decomposition. Households were coordinated using the marginal cost of supplying electricity as the pricing signals. The primal decomposition was applied to obtain a pricing master problem and a household scheduling subproblem. The pricing master problem calculated the prices for households from the supply cost function. The household scheduling subproblem was solved by a gradient descent algorithm and a step size was used in to update the schedules of households at each iteration. The step size needed manual tuning to ensure convergence. Their method was tested on up to 24 households each of which had up to six appliances on a scheduling horizon of 24 hourly time intervals. No data on convergence speed was provided.

Similar to (Li et al., 2011), Chavali et al. (2014) decomposed a DSP-MH using the primal decomposition. Households were coordinated using the marginal cost of supplying electricity to all households as the pricing signals. The primal decomposition was applied to obtain a pricing master problem and a household scheduling subproblem. The pricing master problem computed the prices for households based on the supply cost function. The household scheduling subproblem was solved by a greedy algorithm. An additional penalty cost was added to the household scheduling subproblem to achieve convergence. This penalty cost varied inversely proportional to the consumption cost of a household and proportional to number of iteration, which means households with higher consumption costs had smaller penalty costs particularly in early iterations while households with smaller costs incurred larger penalty costs especially in later iterations. A coefficient that required manual tuning was included in the penalty cost to guarantee convergence. Their method was tested on 100 households each of which had 10 jobs. They applied a game theoretic method (GTM) to their DSP-MH and compared the results of the game theoretic method with those of their methods. On average, their method converged to a solution that was very close to that of the GTM in about 30 iterations. Although the implementation of their method was much easier and cheaper than the GTM since no optimisation solver was needed in the computation.

Zhang et al. (2013) investigated a DSP-MB where a microgrid managed several conventional generators (CGs) (e.g. fossil fuel powered generators), renewable energy generators (REGs) (e.g. wind farms), DGs or batteries, and jobs to minimise the cost of running the CGs and batteries, and the worst-case transaction cost. The worst-case transaction cost was for determining the amount of energy to buy from or sell to the main power system based on predicted outputs of REGs. This work proposed an iterative and distributed algorithm, which is described as follows:

1. The dual decomposition and Lagrangian relaxation was applied to decompose the original problem into a master problem and a subproblem. The master problem updated the Lagrangian multiplier using the subgradient method. The subproblem scheduled jobs, batteries, CGs of each household using general linear programming or quadratic programming methods, and determined the amount of energy to buy or sell using the bundle method and the vertex enumerating algorithm.

2. The master problem and subproblems were solved iteratively until convergence.

Their linear programming model was implemented in CVX (Grant and Boyd, 2014) and solved by the MOSEK solver. Their method was applied on a simulated microgrid with three CGs, ten shiftable or flexible jobs, three batteries, and two REGs. The inelastic jobs were modified from the real load data provided by Midcontinent Independent System Operator (Fderal Energy Regulatory Commission, n.d.) in America in 2012. The maximum capacity of the batteries was 30kWh. The results showed that their proposed method worked as expected. The outputs of CGs increased with the total demand of inelastic jobs over time, the microgrid always purchased power from the main system when the purchase price is lower than the marginal cost of running CGs and selling activities were the most active during times with the highest selling prices. Although this work considered both flexible and inflexible demand, it assumed the prices to be fixed and known in advance. Moreover, optimising against the worst-case transaction cost would lead to very conservative battery behaviour, missing more opportunities to save money. Furthermore, a step size used in the subgradient method required manual tuning to ensure convergence.

Yang et al. (2015) focused on a DSP-MB where a building scheduled multiple appliances and batteries to minimise the total energy cost, the total battery loss cost and the total dissatisfaction cost (inconvenience cost) against the prices given by a central controller, such as the utility company. This work proposed a hybrid of Lagrangian relaxation and Benders decomposition to solve their problem in a distributed and iterative manner. Their proposed involved two step sizes that required manual tuning to ensure the convergence of the iterative process. Their method was tested with a simulated building of 5, 60 and 100 shiftable appliances. The number of batteries in the building increased with the appliances. The results showed that the number of iterations before convergence was almost the same when the appliance and battery ratio was kept the same. Although this work has incorporated flexibility in household appliances for scheduling, it assumed the prices were known. Moreover, two parameters required manual tunings to ensure the effectiveness of the proposed method.

Different from other works, Mhanna et al. (2016) aimed to achieve a near-optimal solution in exchange for a higher scalability and a shorter computation time. They smoothed the dual problem of the DSP-MH and decomposed the smoothed dual problem. Households were coordinated using multiple parameters introduced in the smoothing and the decomposition processes as artificial prices. They solved the problem in two stages:

- Stage one: First they applied the Lagrangian relaxation to obtain the dual problem of the original DSP-MH. Second, they applied a double smoothing technique to the dual problem and decomposed the double smoothed dual problem into a pricing master problem and a household scheduling subproblem. Third, they used a fast gradient algorithm to solve the master problem and the subproblem iteratively for a fixed number of iterations to obtain the suitable values for some key parameters including the smoothing parameters, the step size and the Lagrange multipliers.
- Stage two: First, they applied a single smoothing technique on the dual problem again, added a penalty term to the single smoothed dual problem, and decomposed this augmented single smoothed dual problem into a pricing master problem and a household scheduling subproblem. The penalty term was added to guarantee convergence. Second, they solved the new master problem and subproblem iteratively with the pre-computed smoothing parameters, the step size and the Lagrange multipliers using the fast gradient algorithm for 60 iterations. The number of iterations in the second step might be tuned in practice based on the problem instance.

A parameter used in the double smoothing technique required manual tuning to ensure the double smoothed dual problem was as close to the original dual problem as possible. A couple of other parameters could be tuned to further reduce the gap between the near-optimal solution and the optimal solution. Their method was tested on up to 2560 simulated households each of which had up to ten appliances on average. Their scheduling horizon included 24 hourly time slots. The compared the results of their methods with those of the centralised method (the Gurobi solver). They claimed that their method could find a near-optimal solution that was on average less than 0.5% worse than the optimal solution. The optimal solution was found by the solver in three days and the near optimal solution was found by the proposed method in 9 seconds (assuming all households solved the subproblems in parallel at each iteration), although it is unclear that how much time was needed to solve the master problem. Moreover, the smoothing parameters, the step size and the Lagrange multipliers needed to be pre-computed in stage one to boost the performance in stage two, which means those pre-computed parameters may not suit different problem instances and additional time was required to complete the whole solving process besides those 60 iterations.

He et al. (2019) decomposed the DSP-MH using the primal decomposition. Households were coordinated using the marginal cost of supplying electricity to all households as the pricing signals. Different from other works which only ran the iterative process once at the beginning of the day, this work employed a *model predictive control (MPC)* framework that repeated the iterative process at every time interval, incorporating changes during the day in near real time. The primal decomposition was applied to obtain the pricing master problem and the household scheduling subproblem. The pricing master problem simply updated the selling and purchasing prices. The household scheduling subproblem was solved by a CPLEX solver. An additional penalty cost was added to the household scheduling subproblem to achieve convergence. This penalty cost varied proportionably to the number of iteration and inversely proportional to the demand of a household. A penalty coefficient was included in the penalty cost that required manual tuning to guarantee convergence. Different from other TPCMs, this method employed a MPC framework where the iteration between the DRSP and households repeated at every time interval. Only the decision for the next time interval would be carried out in practice. This way, changes during the day, such as output from REGs and consumption requirements of households, could be incorporated timely, balancing the demand and supply in real time. Their method was tested on four households each of which had wind and PV generators, a battery and eight appliances of various types. The scheduling horizon had 24 hourly intervals. The outputs of REGs were predicted using the historical data of NSJs, wind and PV generation collected and modified from Belgium's transmission system. Similar to (Li et al., 2011), they implemented the cooperative GTM proposed by (Mohsenian-Rad et al., 2010) and compared the results of that method with the results of the GTM. Their method managed to converge in 12 iterations while the GTM converged after 44 iteration in total (11 iterations per household).

Third-party entity (TPE) methods require minimum changes to the communication networks as only pricing signals and demand profiles are exchanged between households and the TPE, which already exists in places where smart meters are installed (see Appendix A.3 for details of smart meters). However, some existing works consider only the aggregate demands of households and ignore all details of jobs. Other methods incorporate details of jobs, however, require at least one parameter to be tuned manually to ensure convergence. Otherwise, a method can converge prematurely to a sub-optimal solution or lead to oscillation where no optimal solutions can be achieved. Moreover, the best values of these parameters can vary from problem (instance) to problem (instance), making these methods less general.

Analysis In summary, we have identified the following findings from our review of literature on distributed methods for DSPs:

- *One-way coordination methods*: Their methods move appliances in ways that benefit households and the utility company, and avoid load synchronisation at a relatively lower computation cost, however, they exhibit limitations in achieving optimality, incorporating changes in demands or introducing coupling constraints on appliances.

- *Interactive coordination methods*: These methods find the best schedules for households without interventions from any third-party entity. However, they can be time-consuming for a large population since only one household is allowed to reschedule and broadcast at each iteration. Furthermore, these methods introduce large workloads on the communication networks as households broadcast information to all others iteratively. Some works argue that sharing demand profiles with all other households is a privacy concern to some consumers, therefore, these methods are not considered as practical. An optimisation solver is required for each household to schedule appliances.

- *Third-party coordination methods*: These methods find the best schedules for households with the help of a TPE. They are more scalable as each household can schedule appliances independently and simultaneously without interaction with other households. However, some existing works consider only the aggregate demands of households and ignore all the details of jobs. Other methods incorporate details of jobs, however, convergence can be a challenge for these methods as at least one parameter requires manual tuning to guarantee convergence. Moreover, the best values of these parameters can vary from problem (instance) to problem (instance), making the methods less general.

Moreover, when batteries are considered, existing works often assume the consumer demands are fixed and known in advance, ignoring the flexibility in consumers' energy needs.

2.3.4 Summary and Analysis

We summary our findings from the review of literature on demand scheduling problem models and solving methods, and the key limitations and areas to extend as follows:

Problem Model

A typical DSP includes three essential elements: a household demand model, a pricing model and objectives. The models of these elements vary across different research works. A household demand model can include a job model, a battery model and/or other device model such as an on-site generator model.

The simplest household job model is a fixed demand profile, or a variable demand profile that can be adjusted over time. More advanced job models include a set of essential attributes and non-coupling constraints. Complex job models have more attributes and coupling constraints. The simplest battery model is a fixed model where attributes are all fixed. More advanced battery models have variable attributes and even battery operation or degradation costs. Generally, a *demand scheduling problem for a single household (DSP-SH)* uses price forecasts instead of pricing functions where the price changes dynamically with the demand in real time. The most common objectives are the electricity cost and the inconvenience value. Sometimes, battery costs, the peak demand, the peak-to-average ratio (PAR) and/or the operation and maintenance costs of on-site generators are also included.

The *DSP-MH* model is developed based on the DSP-SH model. The household demand model and objectives are the same as those in a DSP-SH, however, an additional constraint that limits the total demand of all households may be considered and the objective values are calculated based on the total costs or the total demand profile of all households. Moreover, the electricity price is generally calculated using a pricing function instead of using a forecast. However, forecasts are still used in some works.

Both DSP-SHs and DSP-MHs (with or without batteries) can be continuous and linear, or mixed-integer and non-linear depending on the design of the demand model, pricing function and objective functions.

Solving Method

A significant amount of research has investigated the application of optimisation algorithms to various DR problems under RTP (Deng, Yang, Chow and Chen, 2015; Vardakas et al., 2015; Bayram and Ustun, 2017), however, there are limitations in existing works.

Many existing studies apply centralised methods to solve DSPs (Adika and Wang, 2014; Longe et al., 2017; Pooranian et al., 2018). However, these algorithms do not scale well with the number of households (Van Den Briel et al., 2013; Zhang et al., 2015; Kou et al., 2020), making them impractical for solving problems for hundreds and/or thousands of consumers, such as the size of a major city suburb in Australia.

Some other works develop distributed methods that allocate computation cost into each household and coordinate households in ways to flatten the total demand profile of all households as much as possible (Mhanna et al., 2016; He et al., 2019). However, some of these works assume demands or prices are fixed and known in advance (Vytelingum et al., 2010; Atzeni et al., 2013; Worthmann et al., 2015), limiting the flexibility of their methods. Some other works incorporate a dynamic pricing scheme and shiftable jobs in their problems. However, they either require broadcasting information to all households sequentially and iteratively (Mohsenian-Rad et al., 2010; Pilz et al., 2017), imposing extra burdens on communication networks and limiting the scalability of their methods; or manually tuning some parameters to ensure the optimal solutions will be reached (Yang et al., 2015; He et al., 2019), making their methods not general to all problem instances. Some studies prioritise scalability of their DR algorithms, however, sacrifice the optimality of solutions, and consumer preferences and requirements (Manzoor et al., 2017; Hussain et al., 2018). Table 2.7 summarises the strengths and weaknesses of these methods.

Table 2.7: Strengths and weaknesses of existing works for solving demand scheduling problem for multiple households with batteries

Method	Strength	Weakness
Centralised scheduling	Optimal, feasible	Do not scale well
One-way coordination	Fast	Near optimal, may not be feasible
Interactive coordination	Optimal	Time consuming for large populations, privacy concern, Large workloads on the communication networks
Thirty-party coordination	More scalable	Convergence can be challenging, manual parameter tuning is required.

Limitations in Existing Works

Our review of literature on demand scheduling problems and solving methods have informed us that the DSP-MHs are challenging to solve because:

C.1 The prices and demand schedules are coupled together: The optimal schedules of households depend on the prices, and the prices are calculated based on the schedules.

C.2 The DSP-MHs are mixed-integer non-linear problems with coupling constraints and multiple objectives when shiftable jobs are considered. These problems are NP-hard, which means they very hard to solve especially at a large scale.

Many works did not address both challenges at the same time or only considered particular aspects of the problems. For example, many works have assumed that the prices or the demands are known and fixed in advance, missing the correlation between the prices and the demands. Some other works exclude integer variables by modelling the household demand as a demand profile or ignore coupling constraints, removing some of the complexity. Some studies have addressed both challenges, however, they consider only job scheduling or battery scheduling, and/or require manual parameter tuning or information broadcasting to achieve convergence. Moreover, most studies schedule jobs or batteries over a sparsely granulated time horizon, e.g. 24 hourly or 48 thirty-minute periods, offering limited flexibility in scheduling.

In summary, none of the existing works has achieved all of the followings:

1. scheduling jobs with coupling constraints and batteries,

2. balancing the cost of electricity supply, the energy need of consumers, the inconvenience to consumers and constraints on household appliances,

3. coordinating households using dynamic pricing without iterative and sequential interactions among households,

4. requiring minimum parameter tuning to achieve convergence for any problem instance,

5. is optimised for speed so that it can be used in real time to incorporate changes in energy requirements and constraints during the day,

6. providing a finer granulated scheduling horizon and therefore more flexibility in scheduling appliances.

In addition, very limited works have investigated the application of CP on DSPs while CP has been shown to be effective and efficient for solving combinatorial problems such as scheduling problems (see Appendix B.3.4 for more details). There are opportunities for designing a new algorithm that will address all of the above challenges and limitations, and investigating the impacts of incorporating CP into such an algorithm. These opportunities constitute the work of this thesis.

Chapter 3

Problem Model

3.1 Introduction

This chapter presents the detailed model of the *demand scheduling problem for multiple households with batteries (DSP-MB)* of this thesis. We introduce the scheduling and pricing time horizon in Section 3.2, the household demand model including the job model and the battery energy storage system (battery) model in Section 3.3, the pricing model in Section 3.4, the objective function in Section 3.5 and the formal problem formulation in Section 3.6.

3.2 Time Horizon

This thesis have chosen 144 ten-minute intervals per day for scheduling and 48 thirty-minute periods for pricing. The ten-minute interval is shorter than the intervals chosen in many existing works. We believe that a shorter interval provides more flexibility for scheduling appliances. The thirty-minute periods are adopted to match the trading frequency currently in the Australian electricity wholesale market. Let us write a period and an interval as the following:

- $m \in [1, M] \cap \mathbb{Z}_{>0}$: the index of an interval and $M = 144$,
- $n \in [1, N] \cap \mathbb{Z}_{>0}$: the index of a period and $N = 48$.

3.3 Household Demand

This thesis considers two types of demands: household jobs and batteries. The job model is presented in Section 3.3.1, the battery model in Section 3.3.2 and the demand profiles and limits in Section 3.3.3.

3.3.1 Household Job

This subsection presents the attributes and constraints for a job model. Detailed explanations of each attribute and constraint are provided in Section 2.3.1 of Chapter 2.

Job Attribute

We have adopted shiftable appliances or jobs (see Definition 2.13 in Section 2.3.1 of Chapter 2) including interruptible and non-interruptible ones, and modelled them with the following attributes:

1. demand rate: a flat demand rate measured in KW
2. duration: the amount of time required from the start to finish, measured in the number of scheduling interval
3. actual start time (AST): the actual start time
4. earliest start time (EST): the earliest time an appliance can start running
5. latest finish time (LFT): the latest time an appliance must finish running
6. preferred start time (PST): the time that a consumer prefers this appliance to start
7. care factor (CF): the level of inconvenience to a consumer rated by the consumer when an appliance doesn't start at its PST, ranging from 0 to 10 (0 meaning highly satisfied to 10 highly dissatisfied)
8. predecessors: the device or the set of devices that must finish running before using this appliance
9. maximum succeeding delay (MSD): the maximum amount time allowed after the predecessors are finished before this appliance starts

The best AST of each job is the decision variable that our demand scheduling method needs to calculate. We denote a job with the followings:

- $h \in [1, H] \cap \mathbb{Z}_{>0}$: the index of a household,
- $d \in [1, D_h] \cap \mathbb{Z}_{>0}$: the index of a job inside the h^{th} household.
- $e_{h,d} \in \mathbb{R}_{>0}$: the power of a device in KW,
- $t_{h,d}^{duration} \in [1, M] \cap \mathbb{Z}_{>0}$: the duration,
- $t_{h,d}^{actual-start} \in [1, M] \cap \mathbb{Z}_{>0}$: the actual scheduled start time (AST),
- $t_{h,d}^{preferred-start} \in [1, M] \cap \mathbb{Z}_{>0}$: the preferred start time (PST),
- $t_{h,d}^{earliest-start} \in [1, M] \cap \mathbb{Z}_{>0}$: the earliest start time (EST),
- $t_{h,d}^{latest-finish} \in [1, M] \cap \mathbb{Z}_{>0}$: the latest finish time (LFT),
- $w_{h,d}^{care-factor} \in [0, 10] \cap \mathbb{R}_{>0}$: the care factor (CF),
- $\rho_{h,d}^{prec} \in [1, D_h] \cap \mathbb{Z}_{>0}$: the index of job preceding the job with the index d,
- $t_{h,d}^{max-delay} \cap \mathbb{Z}_{>0}$: the maximum succeeding delay (MSD).

In experiments, we assume that the EST, LFT, PST, CF, predecessors and MSD are provided by consumers. In practice, these data can be learned from historical data using machine learning methods (Siebert et al., 2017; Varghese et al., 2018; Jiang et al., 2019; Sharda et al., 2021). The care factor indicates the flexibility of a job where 0 means this job has full flexiblity can be scheduled to any time within the operation window and 10 means this job is the least flexible. The MSD for a job can be zero if this job has no preceding job.

Note that power flexible appliances are excluded from this thesis because they commonly refer to thermal appliances that are very hard to model. Their models depend on the design of the appliances and factors vary from household to household, such as the the structure of the house, the total mass of air inside the house, the behaviour of its occupiers and so on. Moreover, the data for supporting such modelling is very difficult to get if available. Most importantly, the absence of PF appliances do not affect the design of algorithms developed in this thesis. Therefore, we have decided not the include them in this thesis but in the future work instead.

Job constraint

For each household h, we have adopted the following constraints for each job:

- *Scheduling Time Constraint* A job d of any household h must run after its EST $t_{h,d}^{earliest-start}$ and finish before its LFT $t_{h,d}^{latest-finish}$, which is described as follows:

$$\forall h,\ \forall d \in [1, D_h] \cap \mathbb{Z}_{>0},\ t_{h,d}^{earliest-start} \leq t_{h,d}^{actual-start} \leq t_{h,d}^{latest-finish} - t_{h,d}^{duration} + 1 \tag{3.1}$$

- *Precedence Constraint and Preceding Delay Constraint* If a job d has a preceding job $\rho_{h,d}^{prec}$, this job must run after the preceding job is finished and the delay between these two jobs must be smaller than the MSD $t_{h,d}^{max-delay}$. The sequential and MSD constraints can be described as follows:

$$i = \rho_{h,d}^{prec} \tag{3.2}$$

$$\forall h,\ \forall d \in [1, D_h] \cap \mathbb{Z}_{>0}, \tag{3.3}$$

$$t_{h,i}^{earliest-start} + t_{h,i}^{duration} - 1 < t_{h,d}^{actual-start} < t_{h,i}^{actual-start} + t_{h,i}^{duration} - 1 + t_{h,i}^{max-delay} \tag{3.4}$$

The scheduling time constraint is non-coupling as it involves one job only. The precedence constraint and the preceding delay constraint are coupling as they involve multiple jobs. More explanations of these constraints are provided in Section 2.3.1 of Chapter 2.

3.3.2 Household Battery

This subsection presents the attributes and constraints of a battery model. Detailed explanations of each attribute and constraint are provided in Section 2.3.1 of Chapter 2.

Battery Attribute

We model a battery using a fixed model with a fixed maximum capacity (in kW), a fixed maximum charge or discharge rate (in kW/h), a fixed round-trip efficiency and a linear state-of-charge (SOC) function. The charge rate and the discharge rate are assumed to be same. As a simplified assumption, we do not consider variable models in this thesis.

This thesis assumes that each household can have at most one battery. Let us write the battery of the h^{th} household as follows:

- $\bar{b}_h^{power}$: the maximum power rate (charge or discharge rate),
- $\bar{b}_h^{cap}$: the maximum capacity,
- η_h: the round-trip efficiency,
- $b_{h,m}^{soc}$: the SOC at the m^{th} time period.

Additionally, we define a battery profile as the following:

Definition 3.1. A *battery charge/discharge profile* includes the amount of electricity charged/discharged at each time period of the day.

Definition 3.2. A *battery activity profile* is the sum of the charge and the discharge profiles of a battery.

We write these battery profiles as the following:

- $\mathbf{b}_h^+ = \{b_{h,m}^+ \mid m \in [1, M] \cap \mathbb{Z}_{>0}\}$: the charge profile, where $b_{h,m}^+$ is the amount of electricity charged at the m^{th} period,
- $\mathbf{b}_h^- = \{b_{h,m}^- \mid m \in [1, M] \cap \mathbb{Z}_{>0}\}$: the discharge profile, where $b_{h,m}^-$ is the amount of electricity discharged at the m^{th} period.
- $\mathbf{b}_h^p = \{b_{h,m}^p \mid m \in [1, M] \cap \mathbb{Z}_{>0}\}$: the battery activity profile, where $b_{h,m}^p = b_{h,m}^+ - b_{h,m}^-$.

The best charge and discharge profiles of each battery are the other decision variables that our demand scheduling method needs to compute.

Battery Constraint

For each household h, the battery is limited by the following constraints:

- *Charge and discharge constraints*: require the battery to either charge or discharge below the maximum power rate at any time, which are written as Equation 3.5 and Equation 3.6.

$$\forall m \in [1, M] \cap \mathbb{Z}_{>0},\ 0 \leq b^{+}_{h,m} \leq \bar{b}^{power}_{h} \tag{3.5}$$

$$\forall m \in [1, M] \cap \mathbb{Z}_{>0},\ \ -\bar{b}^{power}_{h} \leq b^{-}_{h,m} \leq 0 \tag{3.6}$$

$$\forall m \in [1, M] \cap \mathbb{Z}_{>0},\ b^{+}_{h,m} \times b^{-}_{h,m} = 0 \tag{3.7}$$

- *Capacity constraint*: requires a battery to maintain its energy level below the maximum capacity at all times, which is written as Equation 3.8.

$$\forall m \in [1, M] \cap \mathbb{Z}_{>0},\ 0 \leq b^{soc}_{h,m} \leq \bar{b}^{cap}_{h} \tag{3.8}$$

- *SOC constraints*: describe the change of energy level over time, which are written as Equation 3.9 and Equation 3.10.

$$\forall m \in [2, M] \cap \mathbb{Z}_{>0},\ b^{soc}_{h,m} = b^{soc}_{h,m-1} + b^{+}_{h,m-1} + b^{-}_{h,m-1} \tag{3.9}$$

$$b^{soc}_{h,\ 1} = b^{soc}_{h,\ M} + b^{+}_{h,M} + b^{-}_{h,M} \tag{3.10}$$

These constraints are non-coupling as they involve individual batteries only. More explanations of these constraints are provided in Section 2.3.1 of Chapter 2.

3.3.3 Demand Profile and Limit

This subsection defines the battery profiles, demand profiles and demand limit constraints defined in this thesis.

Demand Profile

We define a demand profile as the following:

Definition 3.3. A *demand profile* is a set of demands at at every interval/period.

We write a demand profile as the following:

- $\mathbf{l}_{h,d}^{job-s} = \{l_{h,d,m}^{job-s} \mid m \in [1, M] \cap \mathbb{Z}_{>0}\}$: the demand profile of job d at the granularity of scheduling intervals, where $l_{h,d,m}^{job-s} = e_{h,d}$ means the job is running at the m^{th} scheduling interval and $l_{h,d,m}^{job-s} = 0$ otherwise.

- $\mathbf{l}_{h}^{house-s} = \{l_{h,m}^{house-s} \mid m \in [1, M] \cap \mathbb{Z}_{>0}\}$: the demand profile of a household h without batteries at the granularity of M scheduling intervals, where $l_{h,m}^{house-s}$ is the total demand of jobs at the m^{th} scheduling interval and calculated as follows:

$$\forall h,\ \forall m,\ l_{h,m}^{house-s} = \sum_{d=1}^{D_h} \{e_{h,d} \text{ if } \ t_{h,d}^{actual-start} \leq m \leq t_{h,d}^{actual-start} + t_{h,d}^{duration} - 1\} \tag{3.11}$$

- $\mathbf{l}_{h}^{house-battery-s} = \{l_{h,m}^{house-battery-s} \mid m \in [1, M] \cap \mathbb{Z}_{>0}\}$: the demand profile of a household h with batteries at the granularity of M scheduling intervals, where $l_{h,m}^{house-battery-s}$ is the total demand of jobs and the battery at the m^{th} scheduling interval and calculated as follows:

$$\forall h,\ \forall m,\ l_{h,m}^{house-battery-s} = b_{h,m}^{+}/\eta_h + b_{h,m}^{-} \times \eta_h + l_{h,m}^{house-s} \tag{3.12}$$

- $\mathbf{l}_{h}^{house-p} = \{l_{h,n}^{house-p} \mid n \in [1, N] \cap \mathbb{Z}_{>0}\}$: the demand profile of a household h at the granularity of N pricing periods, where $l_{h,n}^{house-p}$ is the total demand of all jobs at the n^{th} pricing period and calculated as follows:

$$\forall h,\ \forall n,\ l_{h,n}^{house-p} = \sum_{m=n\times 3-2}^{n\times 3} l_{h,m}^{house-s} \tag{3.13}$$

- $\mathbf{L}^{total-s} = \{L_{m}^{total-s} \mid m \in [1, M] \cap \mathbb{Z}_{>0}\}$: the demand profile of all households, where $L_{m}^{total-s}$ is the total demand of all households at the m^{th} scheduling interval and calculated as follows:

$$\forall m,\ L_{m}^{total-s} = \sum_{h=1}^{H} l_{h,m}^{house-s} \tag{3.14}$$

- $\mathbf{L}^{total-p} = \{L_{n}^{total-p} \mid n \in [1, N] \cap \mathbb{Z}_{>0}\}$: the demand profile of a household h at the granularity of N pricing period, where $L_{n}^{total-p}$ is the total demand of all households at the n^{th} period and calculated as follows:

$$\forall n,\ L_n^{total-p} = \sum_{m=n\times 3-2}^{n\times 3} L_m^{total-s} \tag{3.15}$$

Demand Limit and Constraint

We denote the demand limit for every household h at any time as $\bar{E}_h^{house}$ and the total demand limit for all households as $\bar{E}^{total}$. These constraints are written as follows:

- *Household demand limit constraint*: The total demand of household h at time interval m: $l_{h,m}^{house-s}$ without batteries or $l_{h,m}^{house-battery-s}$ with batteries cannot exceed the given demand limit $\bar{E}_h^{house}$, which is described as Equation 3.16.

$$\forall m \in [1, M] \cap \mathbb{Z}_{>0},\ l_{h,m}^{house-s} \text{ or } l_{h,m}^{house-battery-s} \leq \bar{E}_h^{house} \tag{3.16}$$

- *Area demand limit constraint*: The total demand of all households at any time interval m cannot exceed the given demand limit $\bar{E}^{total}$, which is described as Equation 3.17.

$$\forall m \in [1, M] \cap \mathbb{Z}_{>0},\ \sum_{h=1}^{H} l_{h,m}^{house-s} \text{ or } \sum_{h=1}^{H} l_{h,m}^{house-battery-s} \leq \bar{E}^{total} \tag{3.17}$$

These constraints are coupling as they involve all jobs and batteries of one or all households. More explanations of these constraints are provided in Section 2.3.1 of Chapter 2.

3.4 Electricity Pricing

This thesis has adopted the real-time pricing (RTP) to incentivise consumers to shift demand. Two different models are used for single households and multiple households.

- *For single households*: we have adopted the day-ahead prices.
- *For multiple households*: we have adopted a pricing function to calculate the price based on the total consumption of all households.

Existing works often use a generic quadratic function to calculate the prices for multiple households based on the total demand. In this thesis, we have designed a pricing table created from bid stacks used by the Australian Electricity Market Operator (AEMO) to calculate the actual prices of electricity in real time.

3.4.1 Bid Stack

A bid stack is a table that includes generation capacities of participating power generators, prices they ask for, the actual generation of each dispatched generator and the cumulative generation of all dispatched generators. Details of a bid stack and the dispatch order are explained in Appendix A.2.2. An example of such bid stacks is provided in Table 3.1.

Table 3.1: An example of the bid stack

Fuel Type	**Price ($/MWh)**	**Quantity (MW)**	**Dispatched (MW)**	**Cumulative Generation (MW)**
Wind	-1000	159	45	45
Wind	-1000	130	114	159
Gas	65.69	138	138	1762
Gas	79.99	100	55	1817
Gas	79.99	100	54	1871

1

3.4.2 Pricing Table

We have designed our pricing table based on such bid stacks. This pricing table includes a list of consumption levels that approximates the cumulative generation, and a price level for each consumption level that approximates the dispatch price. Similar to a bid stack, the price level and the marginal cost increase with the consumption level, which can be understood as when the consumption exceeds another consumption level, a more-expensive-fuel fired generator is required and therefore the marginal cost and the price have to increase. Moreover, we allow each time period to have a pricing table with a different set of consumption levels to better simulate the fact that the generators dispatch in each time period can vary over time. An example of such a pricing table for one time period is illustrated in Table 3.2 and Figure 3.1.

This pricing function is essentially a strictly increasing step function. The electricity cost calculated using this pricing function is piece-wise linear. When setting the price for a pricing period with this table, a algorithm would calculate the total consumption of all households in that period, find the lowest consumption level that is higher than the total consumption and use the price level for that consumption level as the price for households in that period.

Let us write the pricing table at the n^{th} period as $R_n(\cdot) = \{(r^{level}_{n,k},\ e^{level}_{n,k}) \mid k \in [1, K_n] \cap \mathbb{Z}_{>0}\}$:

Table 3.2: An example of the pricing table for a pricing period

Level	1	2	3	4	5	6	7	8
Price (cent/KWh)	14.0	14.1	14.2	14.3	14.4	14.5	14.6	14.8
Consumption (KWh)	102.31	104.32	108.33	112.34	114.35	118.36	122.37	126.38
Level	9	10	11	12	13	14	15	16
Price (cent/KWh)	15.1	15.4	15.8	16.3	17	17.8	18.9	20.3
Consumption (KWh)	128.39	132.4	136.41	138.42	142.43	146.44	148.45	152.46
Level	17	18	19	20	21	22	23	24
Price (cent/KWh)	22.1	24.5	27.5	31.3	36.3	42.7	50.9	61.6
Consumption (KWh)	156.47	160.49	162.49	166.5	170.52	172.52	176.53	180.55
Level	25	26	27	28	29	30		
Price (cent/KWh)	75.2	92.8	115.5	144.6	182.1	230.4		
Consumption (KWh)	182.55	186.56	190.58	194.59	196.59	200.61		

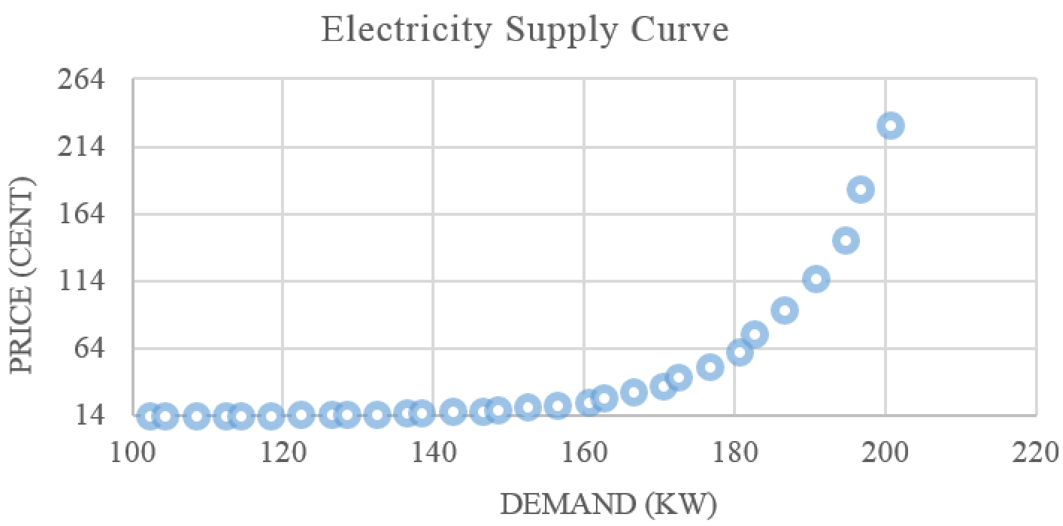


Figure 3.1: An example of the pricing function

- $r_{n,k}^{level}$: the k^{th} price level in period n,
- $e_{n,k}^{level}$: the k^{th} consumption level in period n,
- $k \in [1, K_n] \cap \mathbb{Z}_{>0}$: the index of a price or consumption level in period n.
- K_n : the total number of levels in period n.

The pricing function at the n^{th} period: r_n can be written as Equation 3.18.

$$r_n = \begin{cases} r_{n,1}^{level}, & \text{if } L_n^{total-p} \leq e_{n,1}^{level} \\ r_{n,k}^{level}, & \text{if } e_{n,k}^{level}{}_{-1} < L_n^{total-p} \leq e_{n,k}^{level} \\ r_{n,K_n}^{level}, & \text{if } L_n^{total-p} > e_{n,K_n}^{level} \end{cases} \tag{3.18}$$

Note that, we use the prices given by this table as the prices that consumers pay for in the algorithms and experiments. However, the design of such a pricing table is a complicated economic problem that is not investigated in details by this research. The future work could improve the design of pricing tables so that they indeed reflects the true value of electricity in the market. Moreover, in practice, these prices can also be seen as pricing signals that coordinate the demand scheduling of households instead of the actual prices that consumers receive.

3.4.3 Implicit Area Demand Limit Constraint

The design of our pricing table allows us to incorporate the *area demand limit constraint* implicitly by adding an extra consumption level that is the same as this demand limit and assigning a extremely high price for this consumption level. This way, the demand scheduling method will automatically satisfy this constraint by avoiding this very high price, thus reducing the problem complexity without changing the original problem. Moreover, this design allows us to convert this *area demand limit constraint* into a soft constraint, which means the violation of this constraint will incur an extremely high cost and a solution will still be found. Otherwise, no solution will be found if this constraint is violated.

3.5 Objective

We have adopted the widely used costs or cost-like measurements in the literature: the monetary cost and the inconvenience cost.

3.5.1 Monetary Cost

We compute the supply cost for electricity providers using a piecewise linear function derived from the pricing table proposed in Section 3.4.2 as follows:

$$c_n^{total} = \begin{cases} r_{n,1}^{level} \times L_n^{total-p} \times 24/N, & \text{if } L_n^{total-p} \times 24/N \leq e_{n,1}^{level} \\ \hat{p}_1 \times e_{n,1}^{level} + r_{n,k}^{level} \times (L_n^{total-p} \times 24/N - e_{n,k}^{level}{}_{-1}) & \\ + \sum_{i=1}^{i=k-2} [r_{n,k}^{level}{}_{-1} \times (e_{n,k}^{level}{}_{-i} - e_{n,k}^{level}{}_{-i-1})], & \text{if } L_n^{total-p} \times 24/N > e_{n,1}^{level} \end{cases} \tag{3.19}$$

$$C^{total} = \sum_{n=1}^{N} c_n^{total} \tag{3.20}$$

We calculate the consumption cost for a consumer using the household demand profile and the prices offered by the electricity provider. Let us write C^{house} as the cost of the household h and r_n as the price of the pricing period n. The cost for a consumer or household h is defined as follows:

$$C^{house} = \sum_{n=1}^{N} \{r_n \times l_{h,n}^{house-p}\} \times 24/N \tag{3.21}$$

The price of each scheduling interval is essentially the largest gradient of the supply cost in that interval.

3.5.2 Inconvenience Cost

We have chosen to model the inconvenience cost for a job as a linear function that increases with the difference between the PST and actual start time of that job. We believe a linear function provides sufficient flexibility for distributing some peak demand to other times of the day. Moreover, we have introduced a care factor for each job (see Section 2.3.2) to indicate the scale of inconvenience that occurs to the consumer when a job is scheduled away from its PST. Let us write:

- $u_{h,d}^{job}$: the inconvenience cost of the d^{th} job in the h^{th} household,
- U_h^{house}: the total inconvenience cost of the h^{th} household,
- U^{total}: the overall inconvenience cost of all households.

The inconvenience costs of a job and a household are calculated as the followings:

$$u_{h,d}^{job} = |t_{h,d}^{preferred-start} - t_{h,d}^{actual-start}| \times w_{h,d}^{care-factor} \tag{3.22}$$

$$U_h^{house} = \sum_{d=1}^{D_h} u_{h,d}^{job} \tag{3.23}$$

3.5.3 Cost Combination

We combine two types of costs together using the weighted sum approach. The details of this approach are provided in Appendix B.2.1. Let us write:

- λ^c: the weight of the consumption cost,
- λ^u: the weight of the inconvenience cost.

The new combined cost function is written as:

$$f = \lambda^c C^{total} + \lambda^u U^{total} \tag{3.24}$$

3.6 Problem Formulation

The formal formulation of a demand scheduling problem for a single household (DSP-SH) can be written as follows:

$$\begin{aligned} &\textsf{minimise} && f = \lambda^c C^{house} + \lambda^u U_h^{house} \\ &\textsf{subject to} && (3.1),\ (3.3),\ (3.5),\ (3.6),\ (3.7),\ (3.8),\ (3.9),\ (3.10), \\ &&& (3.16) \end{aligned} \tag{3.25}$$

The formal formulation of a demand scheduling problem for multiple households with batteries (DSP-MB) can be written as follows:

$$\begin{aligned} &\textsf{minimise} && f = \lambda^c C^{total} + \lambda^u U^{total} \\ &\textsf{subject to} && (3.1),\ (3.3),\ (3.5),\ (3.6),\ (3.7),\ (3.8),\ (3.9),\ (3.10), \\ &&& (3.16),\ (3.17). \end{aligned} \tag{3.26}$$

The decision variables include the best start time of each job $t_{h,d}^{actual-start}$ and the charge and discharge profiles $\mathbf{b}_h^+$ and $\mathbf{b}_h^-$ of the battery in each household h. The constraints of jobs are (3.1) and (3.3). The constraints of batteries are (3.5) , (3.6) , (3.7) , (3.8) , (3.9) and (3.10). The demand limit constraints of every household and all households are (3.16) and (3.17), respectively. The *demand scheduling problem for multiple households with batteries (DSP-MB)* is a mixed-integer non-linear optimisation problem as the start times are discrete, the charge and discharge profiles are continuous, the constraints are all linear, the supply cost function is increasing piece-wise linear and the inconvenience value function is linear.

3.7 Summary

This chapter presents our models for household jobs and batteries which compose of the household demand. We also introduce our method for developing a new pricing function as a step function that is derived from the bid stacks used by *Australian Electricity Market Operator (AEMO)*. Then we present the objective functions and the formal formulation for our *demand scheduling problems (DSPs)*, which will be investigated in later chapters.

Chapter 4

Frank-Wolfe-Based Distributed Demand Scheduling Method

4.1 Introduction

This chapter presents the method we proposed for solving the *demand scheduling problems (DSPs)* introduced in Chapter 3. We call our proposed method *Frank-Wolfe-based distributed demand scheduling method (FW-DDSM)*. First, this method decomposes the original demand scheduling problem for multiple households with batteries (DSP-MB) into two subproblems: a household subproblem and a pricing master problem. The household subproblem is solved by two optimisation models that schedule jobs and the battery energy storage system (battery) for each household independently. The pricing master problem is solved by the Frank-Wolfe algorithm, which is also known as the conditional gradient descent method. Second, these two problems are solved in a distributed and iterative manner until convergence. The convergence is guaranteed by the Frank-Wolfe algorithm and minimum parameter tuning is required. Third, once converged, the intermediate results calculated during the iterative process are used to construct a probability distribution for choosing the actual schedules of households.

The details of the decomposition and the iterative framework are presented in Section 4.2, the optimisation models for solving the household subproblem are introduced in Section 4.3, the Frank-Wolfe algorithm for solving the pricing master problem is explained in Section 4.4 and the steps for calculating the probability distribution and finalising the actual schedules for households are described in Section 4.5.

4.2 Decomposition

4.2.1 Primal Decomposition

Our FW-DDSM applies the primal decomposition to divide our DSP-MB into two subproblems: a household subproblem and a pricing master problem. The pricing master problem involves summing the demand profiles of all households, setting the price per period using the pricing table based on the total demand profiles of households, and minimising the total supply cost and the inconvenience cost of all households. This problem is solved by an aggregator or a demand response service provider (DRSP). The household scheduling subproblem involves scheduling jobs and a battery in ways that minimise its electricity cost and the inconvenience value against the prices given by the DRSP. The pricing master problem and the household scheduling subproblem are then solved in an iterative way until convergence.

The primal decomposition we have adopted is different from the dual decomposition that is commonly used in the literature. The dual decomposition is best for optimisation problems with coupling constraints (or complicating constraints as called in the optimisation literature) whereas the primal decomposition is best for problems with coupling variables or complicating variables. In order to solve problems with complicating variables using the dual decomposition, we need to firstly modify the original problem by converting the complicating variables into coupling constraints, secondly transform the modified problem into its dual problem, thirdly decompose the dual problem into subproblems and fourthly solve the subproblems in an iterative fashion. However, when using the primal decomposition, we can decompose the problem based on complicating and non-complicating variables straight-away without any problem transformation. More detailed explanations of the complicating constraints and variables, and decomposition methods are provided in Appendix B.3.6.

The decision variables (the best start times of jobs, and the charge and discharge profiles of batteries) are the complicating variables of our DSP-MB, because they are involved in the objective function (they are used for calculating the demand profiles which are then used for calculating the prices and the supply cost, and the total inconvenience value of all households in the objective function). Although our DSP-MB has coupling constraints, most of these constraints are applied to individual households except for the *area demand* *limit constraint* which is applied to all households. As discussed in Section 3.4.3 of Chapter 3, the *area demand limit constraint* can be satisfied implicitly by the pricing table. This way, our DSP-MB can be considered as an optimisation problem with coupling variables on the household level. Therefore, applying the primal decomposition on the household level, which is more straightforward and efficient.

4.2.2 Iteration

We solve the pricing master problem and the household scheduling subproblem iteratively as follows:

- *Initialisation*: Households schedule jobs at their preferred start times (PSTs) and send the resulting demand profiles to the DRSP.
- *Pricing*: Upon receiving the household demand profiles, the DRSP solves the pricing master problem and sends prices to households.
- *Rescheduling*: After receiving the prices, households solve the scheduling subproblems and send the resulting optimal demand profiles to the DRSP again.
- *Iteration*: Repeats the pricing step and the rescheduling step until convergence.

The iterations are showed in Figure 4.1. The details of each subproblem are presented in the following sections.

Convergence conditions We consider the convergence is reached when the objective value (the total consumption cost and the total inconvenience value of all households) calculated by the pricing master problem does not change, or change within a very small range (e.g. 0.01), in any two consecutive iterations.

4.3 Household Subproblem

The household subproblem is responsible for finding the best schedules for jobs and the battery of each household, such that the costs of the consumer are minimised. In the iterative framework of our FW-DDSM, each household receives prices from the DRSP and schedules its demand against those prices. This subproblem is equivalent to the demand

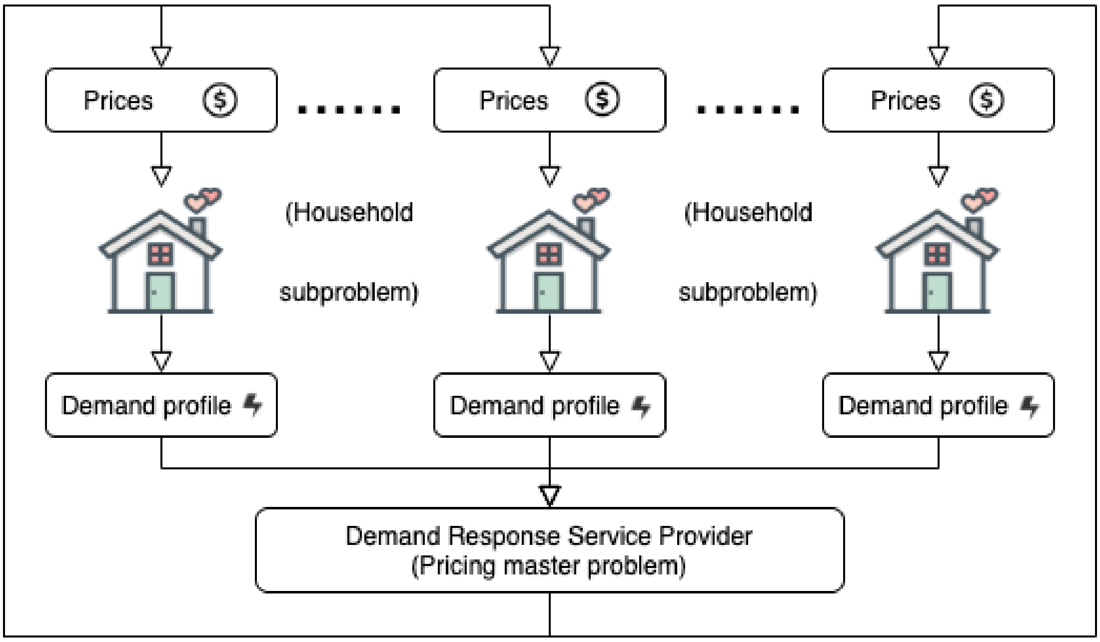


Figure 4.1: Iterations of the Frank-Wolfe-based distributed demand scheduling method

scheduling problem for a single battery (DSP-SB) we have discussed in Section 2.3.2 of Chapter 2.

These days, our homes are becoming smarter. Recently developed appliances have the capabilities of being monitored and controlled through wireless networks at any where and any time. Consumers can install smart scheduling algorithms on a computation machine that has the ability to communicate with smart appliances through the Internet. Consumers can express their consumption preferences or requirements to this machine using some interface. This machine can convert these preferences and requirements into constraints for the household subproblem, collect the pricing information from the DRSP through the Internet, calculate the best feasible schedule for their smart appliances using the scheduling algorithms, and dispatch the control commands to the smart appliances accordingly. Details of such smart homes can be found in Appendix A.3.2. Our solution to the household subproblem can be installed on such a computation machine to compute the best schedules and take advantages of the advanced features of smart appliances.

4.3.1 Subproblem Model

The decision variables of this problem are the start time of each job $t_{h,d}^{actual-start}$ and the charge and discharge profiles of the battery $\mathbf{b}_h^+$ and $\mathbf{b}_h^-$. The constraints of jobs and the battery have been defined in Equation (3.1) – (3.16) in Section 3.3 of Chapter 3. The objective function includes the consumption cost and the inconvenience cost defined as Equation (3.21) and Equation (3.23) in Section 3.5.

4.3.2 Subproblem Solving Method

We solve the household scheduling subproblem in two steps. First, we schedule the jobs against the prices to minimise the consumption cost and the inconvenience value of this household, and calculate the optimised demand profile of this household (the aggregate demand of jobs per scheduling interval when the jobs start at their best start times). Second, we schedule the battery given the optimised demand profile to further flatten the demand profile of this household.

Note that, since the scheduling of jobs and batteries are independent at each iteration (both jobs and the battery are scheduled against the same prices and the households do not update prices based on their schedules), scheduling the battery after jobs does not affect the feasibility nor the optimality of the job schedule. Moreover, scheduling the battery given the optimised demand profile of jobs can only further improve the objective value of this household.

Job Scheduling Module

In Section 2.3.3, we have discussed three types of techniques for scheduling jobs in a demand scheduling problem for a single household (DSP-SH): mixed-integer programming (MIP), constraint programming (CP) and heuristic methods. Since few exiting works, if any, has compared the optimality and efficiency of these methods, in order to select the most efficient method, this research is interested in developing three types of methods for scheduling jobs of a household with these techniques, and comparing their performances. Moreover, we will investigate the impacts of using each of these methods on the solutions to our DSP-MB in Chapter 5.

MIP Optimisation Model First, we have developed a MIP model for scheduling jobs of a household, depicted in Figure 4.2. This model includes the following elements:

- *Model input parameters*: We declare the problem parameters from line 3- 27: the number of time intervals per day (line 4), the number of time intervals per hour (line 6), the price per time interval (line 8), the number of jobs (line 10), the PSTs of all jobs (line 12), the earliest start times (ESTs) (line 13), the latest finish times (LFTs) (line 14), the durations (line 15), the consumption per interval (line 16), the care factors (CFs) (line 17), the weight of the inconvenience cost (line 18), the weight of the electricity cost (line 19), the total number of jobs that have precedences (line 21), the indices of the preceding jobs (line 23), the index of succeeding job for each preceding job (line 24) and the maximum succeeding delays (MSDs) between the preceding and the succeeding jobs (line 25).

- *Decisions*: We define the decision variables in lines 30-34. The set of variables `actual_starts` represent the time interval at which each job is scheduled. For instance, `actual_start[i, 3] = 1` means that the job `i` is scheduled at the 3^{rd} time interval while `actual_start[i, 4] = 1` means that this job is not scheduled at the 4^{th} time interval. The second set of variables, `run_costs` (line 34), represents the objective value (including the electricity cost and discomfort) of scheduling each job at each feasible time interval.

- *Constraints*: The constraints are listed from line 37-50. First, we ensure each job is scheduled after its EST and will be finished before its LFT (line 40). Second, we define the succeeding orders for jobs that have precedences (line 47). Third, we impose the *demand limit constraint* stating the scheduled jobs must not exceed the demand limit (line 50).

- **Objective**: we state the objective (line 54 and line 55).

CP Optimisation Model Second, we have developed a CP model for scheduling jobs of a household, depicted in Figure 4.3. This CP model is similar to the MIP model except for the following elements:

- *Decisions*: While it is more efficient for a MIP model to use binary values as decision variables, it is better for a CP model to use feasible values instead. Consequently, we define the decision variable `actual_starts` as an array where each element represents the actual start time of a job, instead of a matrix where each element represents if a job is scheduled at a time interval as in the MIP model. For instance, `actual_start[i] = 3` means that the job `i` is scheduled at the 3rd time interval.

Figure 4.2: MIP model for scheduling jobs of a household

```

% ---------- input parameters ---------- %

int: num_intervals;
int: num_intervals_hour;
set of int: INTERVALS = 1..num_intervals;

array[INTERVALS] of int: prices;

int: num_jobs;
set of int: JOBS = 1..num_jobs;
array[JOBS] of int: preferred_starts;
array[JOBS] of int: earliest_starts;
array[JOBS] of int: latest_ends;
array[JOBS] of int: durations;
array[JOBS] of int: consumptions;
array[JOBS] of int: care_factors;
int: inconvenience_weight;
int: cost_weight;

int: num_precedences;
set of int: PREC = 1..num_precedences;
array[PREC] of JOBS: predecessors;
array[PREC] of JOBS: successors;
array[PREC] of int: prec_delays;

int: max_demand;

% ---------- Decision variables ---------- %

array[JOBS, INTERVALS] of var 0..1: actual_starts;
array [JOBS, INTERVALS] of int: run_costs = array2d(JOBS,INTERVALS, [
care_factors[d] * abs(s - 1 - preferred_starts[d]) * inconvenience_weight *
num_intervals_hour +
sum (t in s..min(s + durations[d] - 1, no_intervals)) (prices[t] * consumptions
[d]) * cost_weight  | d in JOBS, s in INTERVALS]);

% ---------- Constraints ---------- %

constraint forall (d in JOBS) (
earliest_starts[d] + 1 <= sum(s in INTERVALS) (actual_starts[d,s] * s)
/\ sum(s in INTERVALS) (actual_starts[d,s] * s) + durations[d] - 1 <=
latest_ends[d] + 1);

constraint forall (p in PREC) (
let {JOBS: pre = predecessors[p] ;
  JOBS: succ = successors[p] ;
  int: d = prec_delays[p]; } in
sum(s in INTERVALS) (actual_starts[pre,s] * s) + durations[pre] <= sum(s in
INTERVALS) (actual_starts[succ,s] * s)
/\ sum(s in INTERVALS) (actual_starts[succ,s] * s) <= sum(s in INTERVALS) (
actual_starts[pre,s] * s) + durations[pre] + d);

constraint forall (s in INTERVALS)
(sum(d in JOBS) (actual_starts[d, s] * consumptions[d]) <= max_demand);

% ---------- Objective and search ---------- %
var int: obj= sum (d in JOBS, s in INTERVALS)
(run_costs[d, s] * actual_starts[d, s]);
solve minimize obj;

```

- *Constraints*: A major advantage of CP methods is the use of global constraints. More explanations of global constraints are provided in Appendix B.3.4. We take advantage of this benefit by calling the global constraint library (line 1) and replacing the demand limit constraint in the MIP model with a global constraint called `cumulative` (line 50) to ensure the accumulated demand is below the household demand limit at any time interval.

- *Search and reasoning strategies*: As CP methods reply on search and inference reasoning to eliminate infeasible and suboptimal values, choosing which variable to search and which value to reason next is important for the efficiency of a CP solver. Users can choose such variable selection strategies and value choice strategies for a solver to improve its solving time. Common strategies are listed in Table 4.1. Detailed explanations of these strategies can be found at the MiniZinc. Handbook (Stuckey et al., 2018). This CP model uses the *first fail* and *indomain max* strategies (line 54). Section **??** will implement more strategy combinations and compare their performances.

Table 4.1: Commonly used value choice strategies and selection strategies for CP solvers

Value Choice Strategy		**Variable Selection Strateg**	
input_order	most_constrained	indomain_min	indomain_max
first_fail	anti_first_fail	indomain_random	indomain
smallest	largest	indomain_split	indomain_reverse_split

Note that we have declared all decision variables as integers although in practice prices and consumptions are floating point numbers. We have multiplied all prices and consumptions by 100 to transform them into integers because CP solvers are most efficient for integer variables. A sample data file used for both the MIP model and the CP model is presented in Appendix C.1.

Data Preprocessing Third, we have noticed that the cost of starting each job at each time interval is independent of any constraints. We can develop a preprocessing algorithm to pre-computes the `run_costs` and even eliminate some constraints and parameters in the model by incorporating them into the precomputed `run_costs`. For example, we can incorporate the scheduling time constraints into the `run_costs` by setting an extremely

Figure 4.3: CP model for scheduling jobs of a household

```
include "globals.mzn";

% ---------- input parameters ---------- %

int: num_intervals;
int: num_intervals_hour;
set of int: INTERVALS = 1..num_intervals;

array[INTERVALS] of int: prices;

int: num_jobs;
set of int: JOBS = 1..num_jobs;
array[JOBS] of int: preferred_starts;
array[JOBS] of int: earliest_starts;
array[JOBS] of int: latest_ends;
array[JOBS] of int: durations;
array[JOBS] of int: consumptions;
array[JOBS] of int: care_factors;
int: inconvenience_weight;
int: cost_weight;

int: num_precedences;
set of int: PREC = 1..num_precedences;
array[PREC] of JOBS: predecessors;
array[PREC] of JOBS: successors;
array[PREC] of int: prec_delays;

int: max_demand;

% ---------- Decision variables ---------- %

array[JOBS] of var INTERVALS: actual_starts;
array [JOBS, INTERVALS] of int: run_costs = array2d(JOBS,INTERVALS, [
care_factors[d] * abs(s - 1 - preferred_starts[d]) * inconvenience_weight * num_intervals_hour+
sum (t in s..min(s + durations[d] - 1, num_intervals)) (prices[t] * consumptions[d]) * cost_weight | d in JOBS, s in INTERVALS]);

% ---------- Constraints ---------- %

constraint forall (d in JOBS) (
earliest_starts[d] + 1 <= actual_starts[d]
/\ actual_starts[d] + durations[d] - 1 <= latest_ends[d] + 1);

constraint forall (p in PREC) (
let {JOBS: pre = predecessors[p] ;
  JOBS: succ = successors[p] ;
  int: d = prec_delays[p]; } in
actual_starts[pre] + durations[pre] <= actual_starts[succ]
/\ actual_starts[succ] <= actual_starts[pre] + durations[pre] + d);

constraint cumulative(actual_starts, durations, consumptions, max_demand);

% ---------- Objective and search ---------- %
var int: obj= sum (d in JOBS) (run_costs[d, actual_starts[d]]);
solve :: int_search(actual_starts, first_fail, indomain_max, complete)
minimize obj;
```

high cost for starting a job outside its feasible time intervals. This way, we can further reduce the computation time of the model. This preprocessing algorithm is straight forward. We have included the pseudo code in Appendix 4.

The modified MIP model and CP model used with the preprocessing algorithm are depicted in Figure 4.4 and Figure 4.5, respectively. These modified models do not require the ESTs, PSTs, LFTs and CFs parameters. Instead, a new parameter `run_costs` is used. However, in the modified MIP model, an additional constraint is required to ensure that only one start time is selected for each job (line 33). An example of the input data file for both modified models is provided in Appendix C.2

Optimistic Greedy Search Algorithm Fourth, we have developed a heuristic algorithm based on the greedy algorithm. We call this algorithm the *Optimistic Greedy Search Algorithm (OGSA)*. Similar to the modified models, this algorithm is used after the run costs are calculated by the data preprocessing algorithm. The OGSA schedules each job independently in the following steps:

1. Find the time intervals that satisfy the *scheduling time constraints*, and the *precedence constraint* and the *preceding delay constraint* if a predecessor exists for this job. This set of intervals is called the feasible interval set.

2. Select the time intervals from the feasible interval set that have the lowest run costs (precomputed by the data-preprocessing algorithm).

3. Choose a cheapest time interval as the actual start time and calculate the maximum demand of the household.

4. Check if the maximum demand exceeds the limit:

   (a) if no, the chosen interval is the final actual start time.

   (b) if yes and the feasible interval set is not empty, remove this interval from the feasible interval set and go back to Step 2.

   (c) if yes and the feasible interval set is empty, choose the removed time interval that exceeds the limit the least as the final actual start time.

The pseudo code of this algorithm is presented at Appendix 5.

Battery Scheduling Module

The battery scheduling method is responsible for finding the best time to charge and discharge a battery given the prices and the demand profile of a household after its jobs have been scheduled. Different from the job scheduling problem, we use minimisation of the consumption cost, the maximum demand and the peak-to-average ratio (PAR) of the household demand profile as the objective, because inconvenience is not applicable to batteries and we want to flatten the demand profile to the lowest. We have observed from

Figure 4.4: Modified MIP model for scheduling jobs of a household

```

% ---------- input parameters ---------- %

int: num_intervals;
set of int: INTERVALS = 1..num_intervals;

int: num_jobs;
set of int: JOBS = 1..num_jobs;
array[JOBS] of int: durations;
array[JOBS] of int: consumptions;

int: num_precedences;
set of int: PREC = 1..num_precedences;
array[PREC] of JOBS: predecessors;
array[PREC] of JOBS: successors;
array[PREC] of int: prec_delays;

int: max_demand;

array [JOBS, INTERVALS] of int: run_costs;

% ---------- Decision variables ---------- %

array[JOBS, INTERVALS] of var 0..1: actual_starts;

% ---------- Objectives ---------- %

var int: obj= sum (d in JOBS, s in INTERVALS)
(run_costs[d, s] * actual_starts[d, s]);

% ---------- Constraints ---------- %

constraint forall (d in JOBS) (sum(s in INTERVALS) (actual_starts[d, s]) == 1);


constraint forall (p in PREC) (
let {JOBS: pre = predecessors[p] ;
  JOBS: succ = successors[p] ;
  int: d = prec_delays[p]; } in
sum(s in INTERVALS) (actual_starts[pre,s] * s) + durations[pre] <= sum(s in
INTERVALS) (actual_starts[succ,s] * s)
/\ sum(s in INTERVALS) (actual_starts[succ,s] * s) <= sum(s in INTERVALS) (
actual_starts[pre,s] * s) + durations[pre] + d);

constraint forall (s in INTERVALS)
(sum(d in JOBS) (actual_starts[d, s] * consumptions[d]) <= max_demand);

% ---------- Objective and search ---------- %
solve minimize obj;

```

Figure 4.5: Modified CP model for scheduling jobs of a household

```
include "globals.mzn";

% ---------- input parameters ---------- %

int: num_intervals;
set of int: INTERVALS = 1..num_intervals;

int: num_jobs;
set of int: JOBS = 1..num_jobs;
array[JOBS] of int: durations;
array[JOBS] of int: consumptions;

int: num_precedences;
set of int: PREC = 1..num_precedences;
array[PREC] of JOBS: predecessors;
array[PREC] of JOBS: successors;
array[PREC] of int: prec_delays;

int: max_demand;

array [JOBS, INTERVALS] of int: run_costs;

% ---------- Decision variables ---------- %

array[JOBS] of var INTERVALS: actual_starts;

% ---------- Objectives ---------- %
var int: obj= sum (d in JOBS) (run_costs[d, actual_starts[d]]);

% ---------- Constraints ---------- %


constraint forall (p in PREC) (
let {JOBS: pre = predecessors[p] ;
  JOBS: succ = successors[p] ;
  int: d = prec_delays[p]; } in
actual_starts[pre] + durations[pre] <= actual_starts[succ]
/\ actual_starts[succ] <= actual_starts[pre] + durations[pre] + d);

constraint cumulative(actual_starts, durations, consumptions, max_demand);

% ---------- Objective and search ---------- %
solve :: int_search(actual_starts, first_fail, indomain_max, complete)
minimize obj;
```

our experiments that minimising both PAR and the maximum demand yields the flattest demand profile, compared to simply minimising the PAR or the maximum demand.

However, PAR is a ratio that depends on multiple decision variables. Introducing this ratio into the objective function has turned the battery scheduling problem into a *fractional linear programming (FIP)* problem (Hooker, 2019). Solving such FIP problems requires transforming the original problem to a linear problem using the Charnes-Cooper transformation (Charnes and Cooper, 1962).

Charnes-Cooper Transformation Let us write the general form of FIP problems as Equation 4.1.

$$\begin{aligned} \text{minimise} \quad & \frac{cx + c_0}{dx + d_0} \\ \text{subject to} \quad & Ax \geq b \\ & x \geq 0 \end{aligned} \tag{4.1}$$

The Charnes-Cooper transformation linearises this FIP problem by replacing $x = x'/z$ and fixing the denominator to 1, showed as Equation 4.2.

$$\begin{aligned} \text{minimise} \quad & cx' + c_0 z \\ \text{subject to} \quad & Ax' \geq bz \\ & dx' + d_0 z = 1 \\ & x', z \geq 0 \end{aligned} \tag{4.2}$$

Particularly for FIP problems whose objective value is a relative maximum value, showed as Equation 4.3, the problems are linearised as Equation 4.4.

$$\begin{aligned} \text{minimise} \quad & \frac{u_{max}}{\bar{u}} = \frac{u_{max}}{(1/n)\sum_i u_i} \\ \text{subject to} \quad & \forall i,\ u_{max} \geq u_i \\ & \forall i,\ u_i = a_i x_i,\ 0 \leq x_i \leq b_i \\ & \sum_i x_i = B \end{aligned} \tag{4.3}$$

$$\begin{aligned} \text{minimise} \quad & u_{max} \\ \text{subject to} \quad & \forall i,\ u_{max} \geq u'_i \\ & \forall i,\ u'_i = a_i x'_i,\ 0 \leq x'_i \leq b_i \zeta \\ & \sum_i x'_i = B\zeta \\ & (1/n)\sum_i u'_i = 1 \\ & \zeta \geq 0 \\ & \forall i,\ u'_i \geq 0 \end{aligned} \tag{4.4}$$

Battery Scheduling Subproblem Linearisation Using Charnes-Cooper transformation, the battery scheduling subproblem is linearised in the following steps:

1. Reformulate the PAR minimisation objective for our battery scheduling subproblem in the form of Equation 4.3 as the following:

$$\begin{aligned} \text{minimise} \quad & PAR_h^{house} = \frac{MAX_h^{house}}{\bar{\mathbf{l}}_h^{house-battery-s}} = \frac{MAX_h^{house}}{(1/M)\sum_m l_{h,m}^{house-battery-s}} \\ \text{subject to} \quad & \forall m,\ MAX_h^{house} \geq l_{h,m}^{house-battery-s} \end{aligned} \tag{4.5}$$

2. Linearise the PAR minimisation objective in the form of 4.6 as the following:

$$\begin{aligned} \text{minimise} \quad & PAR_h^{house-l} = MAX'_h \\ \text{subject to} \quad & \forall m,\ MAX'_h \geq l'_{h,m} \\ & \forall m,\ l'_{h,m} = \zeta \times l_{h,m} \\ & (1/M)\sum_m l'_{h,m} = 1 \\ & \zeta \geq 0 \end{aligned} \tag{4.6}$$

3. Let us remove the division in the Equation 4.6 by replacing $l'_{h,m} = l''_{h,m} \times M$, and rewrite the linearised PAR minimisation objective as the following:

$$\begin{aligned} \text{minimise} \quad & PAR_h^{house-l} = MAX'_h \\ \text{subject to} \quad & \forall m,\ MAX'_h \geq l''_{h,m} \times M \\ & \forall m,\ l''_{h,m} = \zeta' \times l_{h,m} \\ & \sum_m l''_{h,m} = 1 \\ & 0 \leq \zeta' = \zeta/M \leq 1 \end{aligned} \tag{4.7}$$

4. Reformulate the battery scheduling subproblem as Equation 4.8. The differences between the transformed problem and the original problem are highlighted in bold.

$$\begin{aligned} \text{minimise} \quad & y_h = \lambda^c C^{house} + \lambda^{max} MAX_h^{house} + \lambda^{par} PAR_h^{house-l} \\ \text{subject to} \quad & (3.5),\ (3.6),\ (3.8),\ (3.9),\ (3.10),\ (3.16),\ (\mathbf{4.7}) \end{aligned} \tag{4.8}$$

The reformulated battery scheduling subproblem is a linear programming problem where the decision variables are continuous, and the constraints and the objective functions are linear. This problem can be solved by a linear programming (LP) optimisation model.

We do not use CP solve this subproblem because the battery charge and discharge rates are continuous instead of discrete, and LP is the best solving method for optimisation problems with continuous variables.

LP Optimisation Model Figure 4.6 shows the LP model developed for the battery scheduling problem, including elements which are described as follows:

- *Input parameters* from line 3 to line 18:
  - The time related parameters: the number of intervals per day (line 3) and the number of intervals per hour (line 4)
  - The battery related parameters: the minimum energy capacity (line 8), the maximum energy capacity (line 9), the maximum power rate (line 10) and the efficiency (line 11) of the battery
  - The demands and prices related parameters: the optimised demand profile of jobs (line 14), the limit of the total demand at any time (line 15) and the price per interval (line 18).
- *Decision variables* from line 21 to line 34:
  - The original decision variables: `battery_soc` represent the state-of-charge (SOC) of the battery at each interval, `battery_charge` (or `battery_discharge`)represent the amount of electricity charged (or discharged) per interval, and `demand` represent the optimised demand profile of the household after the battery is scheduled.
  - The additional variables for linearising the PAR objective: `z`, `maxz`, `minz`, `battery_soc2`, `battery_discharge2` and `demand2`.
- *Objectives* from line 37 to line 41.
  - The weight of the original PAR objective: `par_weight`.
  - The linearised PAR objective: `PAR`.
  - The maximum demand: `max_demand`.
  - the consumption cost: `cost`

Figure 4.6: LP model for scheduling the battery of a household

```

% ------ time intervals ----- %
int: num_intervals;
int: num_intervals_hour;
set of int: INTERVALS = 1..num_intervals;

% ------ battery specifications ------ %
float: min_energy_capacity;
float: max_energy_capacity;
float: max_power;
float: efficiency;

% ------ demands ------ %
array[INTERVALS] of float: existing_demands;
float: demand_limit = 9999999999.9;

% ------ prices  ------ %
array[INTERVALS] of float: prices;

% ------ decision variables ------ %
array[INTERVALS] of var min_energy_capacity..max_energy_capacity: battery_soc;
array[INTERVALS] of var 0..max_power: battery_charge;
array[INTERVALS] of var -max_power..0: battery_discharge;
array[INTERVALS] of var 0..demand_limit: demand = array1d([existing_demands[i]
+ (battery_charge[i]  / efficiency) + (battery_discharge[i] * efficiency) | i
in INTERVALS]);

% ------ additional variables for PAR linearisation  ------ %
float: minz = 0 ;
float: maxz = 1 ;
var float: z;
array[INTERVALS] of var minz * min_energy_capacity..maxz * max_energy_capacity:
 battery_soc2;
array[INTERVALS] of var 0..maxz * max_power: battery_charge2;
array[INTERVALS] of var -maxz * max_power..0: battery_discharge2;
array[INTERVALS] of var 0..maxz * demand_limit: demand2 = array1d([z *
existing_demands[i] + (battery_charge2[i] / efficiency) + (battery_discharge2[i
] * efficiency) | i in INTERVALS]);

% ------ objectives  ------ %
int: par_weight;
var float: max_demand;
var float: PAR = max(demand2) * num_intervals ;
var float: cost = sum (i in INTERVALS) (demand[i] * prices[i]);
var float: obj = cost + par_weight * PAR + max_demand;

% ------ constraints for the original decision variables ------ %
constraint forall(i in INTERVALS)(battery_soc[i] <= max_energy_capacity);
constraint forall(i in INTERVALS)(battery_soc[i] >= min_energy_capacity);
constraint forall (i in 2..num_intervals) (battery_soc[i] * num_intervals_hour
- battery_soc[i - 1] * num_intervals_hour = battery_charge[i - 1] +
battery_discharge[i - 1]);
constraint battery_soc[1] * num_intervals_hour - battery_soc[num_intervals] *
num_intervals_hour = battery_charge[num_intervals] + battery_discharge[
num_intervals];
constraint forall(i in INTERVALS)(max_demand >= demand[i]);

% ------ constraints for the additional variables ------ %
constraint z >= minz  /\ z <= maxz;
constraint sum(demand2) = 1;
constraint sum(i in INTERVALS)(battery_charge2[i] * battery_discharge2[i]) = 0;
constraint forall(i in INTERVALS)(battery_soc2[i] <= max_energy_capacity * z);
constraint forall(i in INTERVALS)(battery_soc2[i] >= min_energy_capacity * z);
constraint forall (i in 2..num_intervals) (battery_soc2[i] * num_intervals_hour
 - battery_soc2[i - 1] * num_intervals_hour = battery_charge2[i - 1] +
battery_discharge2[i - 1]);
constraint battery_soc2[1] * num_intervals_hour - battery_soc2[num_intervals] *
 num_intervals_hour = battery_charge2[num_intervals] + battery_discharge2[
num_intervals];

% ------ minimise the objective value ------ %
solve minimize obj;

```

  - the objective value: `obj`

- *Constraints* from line 44 to line 57.

  - The SOC constraints for the original decision variables: line 44 to line 47.
  - The maximum demand constraint for the original decision variables: line 48.
  - The constraints for the additional variables: line 51 to line 57

4.4 Pricing Master Problem

The pricing master problem is responsible for setting the prices for households, such that when households respond to these prices in the most selfish way (e.g. in a way that minimise their own costs), the total demand profile of all households will be flatten as much as possible and the total cost of all households will be reduced. The prices for all households are the same. No information needs to be exchanged among households. During the iterations, only the demand profile of each household is required to be sent to the DRSP and no information other than prices are sent back to households.

In countries where smart meters have been widely adopted (see Appendix A.3.2 for explanations of smart meters), electricity network operators have already recorded household consumptions at a higher frequency (e.g. every seconds or minutes) remotely using the wireless communication features of these meters. These meters have the capabilities of receiving information from network operators through the Internet. Our solution to the pricing master problem can be applied to a retailer or a network operator with minimum changes to the existing network infrastructure.

4.4.1 Naive Approach

First, we have used the simplest way to calculate prices: at each iteration, we calculate the total demand profile of all households from the optimised demand profiles sent by households, and compute the prices from the total demand profile using the pricing table.

Oscillations However, this naive approach cannot achieve convergence at all. We have observed from experiments that at one iteration, households would schedule most demands to the cheapest time intervals, making them very expensive for the next iteration; and at the next iteration, households would schedule very little demands in the those time intervals due to their high prices, making them very cheap again. Figure 4.7 and Figure 4.8 show the experimental results of this naive approach. This experiment was conducted with 90 simulated households over 48 thirty-minute time periods. At each iteration, the households scheduled devices against the prices given by the DRSP in the previous iteration, and the DRSP updated the prices based on the total demand profile of the same iteration. The figures show the total consumption of households per period and the total costs for 20 iterations. We can observe that the oscillation started just after iteration 3.

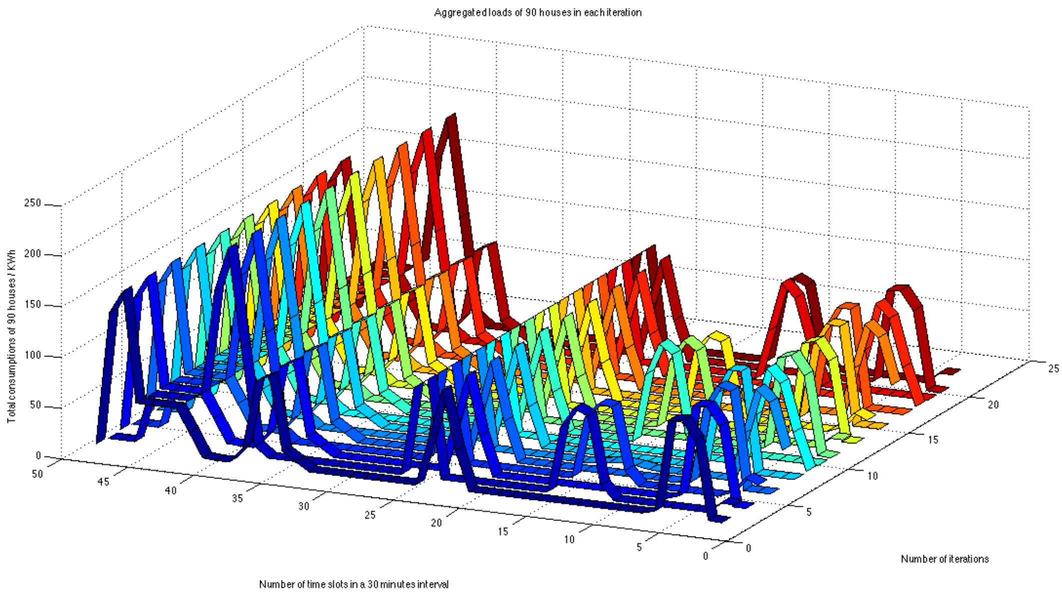


Figure 4.7: Oscillations of Consumptions as a Result From the Naive Approach

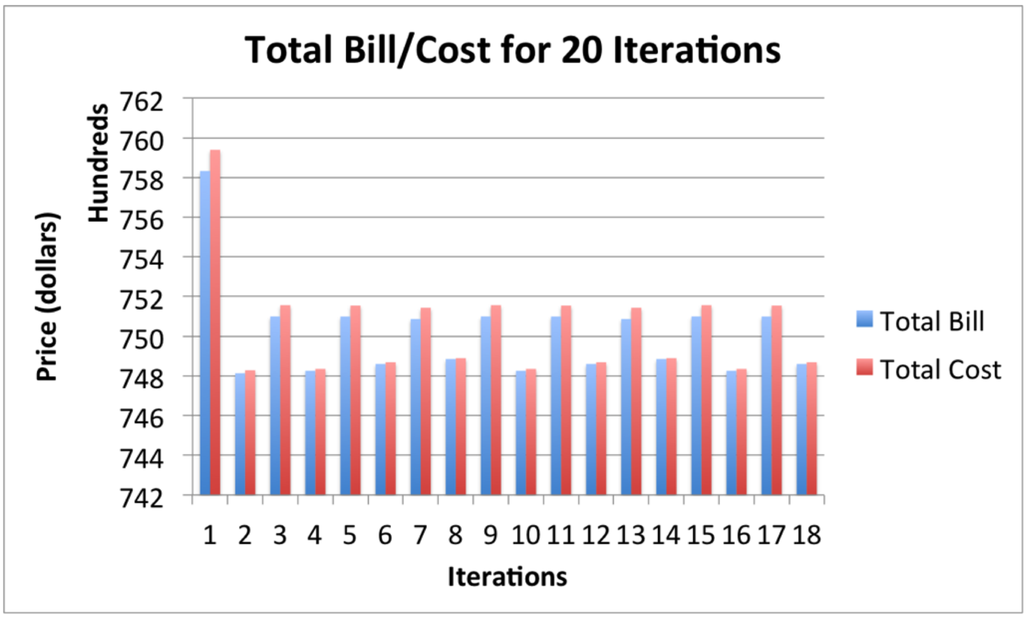


Figure 4.8: Oscillations of Costs as a Result From the Naive Approach

4.4.2 Averaging Approach

Second, we have adopted the traditional mechanism for achieving convergence while reducing the objective value: *averaging* (Chapman et al., 2011). At each iteration, we calculate the prices based on the demand profiles found at the current iteration and the average demand profiles from all previous iterations. Let us write the total demand profile of all households, calculated from the optimised demand profiles of all households, at iteration z as $\mathbf{L}_z^{total-itr-p}$. The averaged total demand profile of all iterations up to iteration z $\mathbf{L}_z^{total-itr-avg}$ can be calculated as follows:

$$\mathbf{L}_z^{total-itr-avg} = (1-\alpha) * \mathbf{L}_{z-1}^{total-itr-avg} + \alpha * \mathbf{L}_z^{total-itr-p}, \text{ where } \alpha = 1/z \tag{4.9}$$

The prices are then calculated using $\mathbf{L}_z^{total-itr-avg}$ and the pricing table. We illustrate the iterations of the primal decomposition with this averaging approach as Figure 4.9.

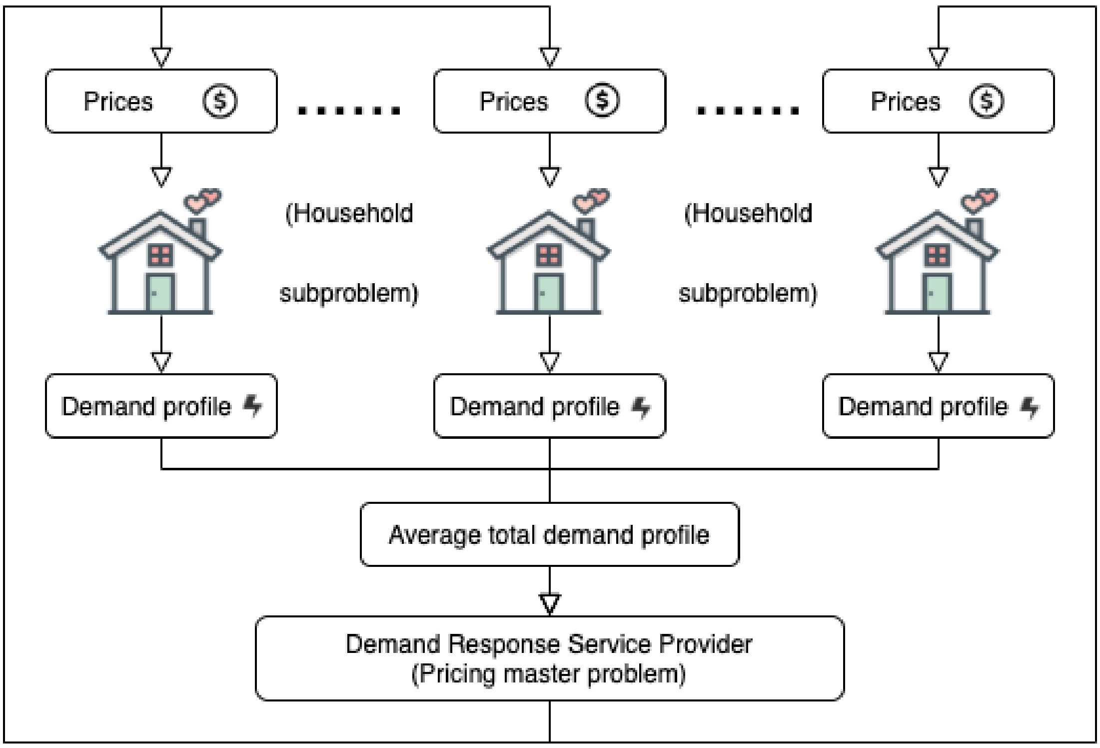


Figure 4.9: Iterations between the DRSP and households with the *averaging* method

Total Expected Demand Profile and Probability

This approach can be understood as calculating prices based on partial demand profiles from all iterations, or the expected demand profiles of households at each iteration. The form of Equation 4.9 allows us to interpret $\mathbf{L}_z^{total-itr-avg}$ as the total expected demand profile of all households at iteration z, where $1 - \alpha$ is the weight of the average total demand profile at iteration $z - 1$: $\mathbf{L}_{z-1}^{total-itr-avg}$, and α is the weight of the total demand profile at iteration z: $\mathbf{L}_z^{total-itr-p}$. Alternative, we can understand α as the probability for households to switch to their optimal schedules found at iteration z from the schedules at iteration $z - 1$; or the probability for a job to start and a battery to charge/discharge according to the optimal schedules of its household at iteration z.

Oscillations and Convergence

Calculating the prices from the total expected demand profile allows the pricing master problem to consider the possible responses of households from all iterations, preventing prices from swinging with the demands at each iteration. Although this method will achieve convergence, it is unclear whether it will converge to the global optimal solution or a local optimal. Moreover, we have observed in experiments that this approach preserves smaller ranges of oscillations, making the convergence very slow to achieve. Figure 4.10 shows the results of applying the averaging approach to the problem and data discussed in the naive approach (see Section 4.4.1). We can observe that the small ranges of oscillations appear just after iteration 3.

4.4.3 Frank-Wolfe Approach

Although the averaging approach is very slow to converge, it has shown the potential of finding an (local) optimal solution to our DSP-MB using the weighted demand profiles from all iterations or the total expected demand profile of all households. We believe that there are opportunities for a more sophisticated algorithm to achieve the optimal solution in less iterations in a similar way. The Frank-Wolfe (FW) algorithm (Sheffi, 1985; Frank and Wolfe, 1956), also known as the *conditional gradient descent method*, is such a method we have found. Moreover, this algorithm can guarantee that the global optimality of the solution to a convex optimisation problem.

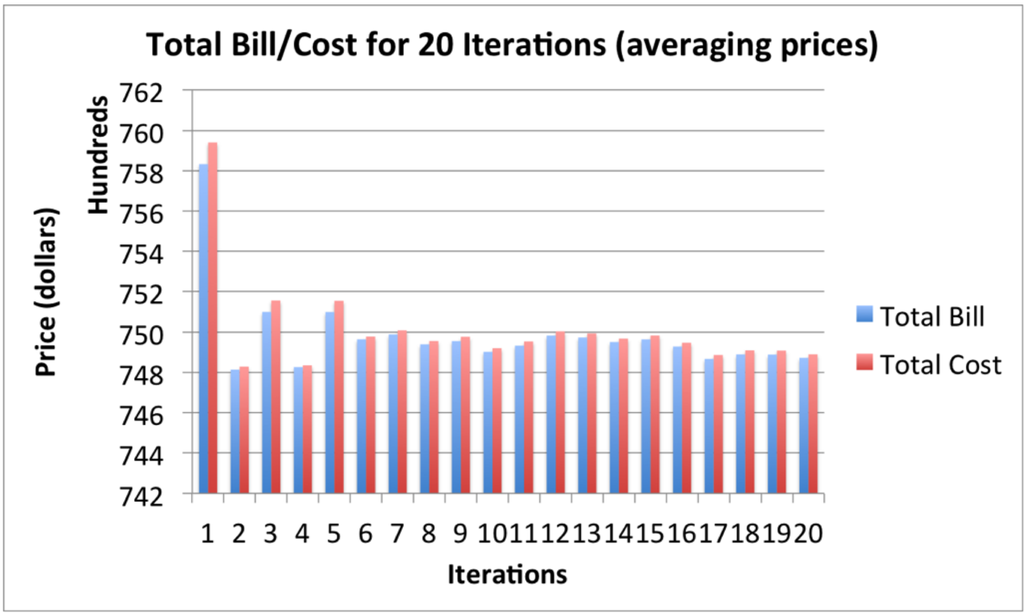


Figure 4.10: Cost per Iteration Using the Averaging Approach

Frank-Wolfe Algorithm

The FW algorithm is an optimisation method that solves constrained convex optimisation problems iteratively using the linear approximations or the first-order Taylor approximations of the problems at each iteration. It has been successfully applied to scheduling problems in the traffic assignment domain (Nakamura et al., 2020). In general, the FW algorithm works as follows:

1. Choosing an initial solution x_k $(k = 0)$ to the problem by selecting values for variables randomly or using some heuristic methods.

2. Calculating the linear approximation of the original problem which is also the gradient at the current solution x_k: $F_{x_k}(x) = f(x_k) + \nabla f(x_k)(x - x_k)$.

3. Finding a new feasible solution y_k that minimises the objective value of this linear approximation function subject to the constraints.

4. Calculating the steepest descent direction as the line function that connects the two newly calculated solutions: $d_k = y_k - x_k$.

5. Performing a line search on this descent direction to find the point x_{k+1} that minimises the objective value of the original problem. This x_{k+1} can be written using a variable called the *step size* α_k and the previous solution as:

$$x_{k+1} = x_k + \alpha_k * d_k \text{ where } \alpha \in [0, 1] \tag{4.10}$$

Since both x_k and d_k are known, this step is equivalent to determining the best value of α_k that minimises the objective value.

6. Repeating Step 2 – Step 5 until convergence.

In a multi-dimension problem like our DSP-MB, each point in the solution space has various descent directions that reduce the value of the objective function. One of these descent directions reduces the objective value in the fastest way, which is the steepest descent descent. This steepest descent direction is the linear approximation or the gradient of the objective function. The FW algorithm searches for solutions along this steepest descent that reduce the objective value the most (Step 2). In addition, this algorithm ensures the feasibility of solutions at each iteration by considering the descent direction within the feasible solution space only (Step 3 and 4).

In more details, at each iteration, the FW algorithm finds a point x_k in the solution space (Step 1) using some optimisation methods, calculates the gradient at this point $F_{x_k}(x)$ (Step 2) and finds another feasible point y_k on the gradient that minimises the value of this gradient function while satisfying all constraints (Step 3). Then we draw a line d_k between the previous point x_k and this new point y_k. Since the problem is convex and both x_k and y_k are feasible, any points on this line d_k is feasible. As y_k is found on the steepest descent direction, all points on this line have reduced objective values. So we find a point x_{k+1} on this line that has the lowest objective value of the original problem (Step 5). This point x_{k+1} is best solution we can find up to the current iteration. Then we repeat Step 2 – Step 5 iteratively until x_{k+n} does not have a lower objective value. The last x_{k+n} is the optimal solution we seek to the optimisation problem. More details of line search methods, gradients and descent directions can be found in Appendix B.3.5.

Note that the FW algorithm can solve convex problems that are not strictly convex. When a problem is non-strictly convex, in Step 5, there may be several points on d_k that have the same optimal objective value. We choose one of these points as the best solution x_{k+1} up to the current iteration. In other words, although there can be multiple optimal solutions with the same objective value in a non-strictly convex optimisation problem, we simply find one of these optimal solutions as the best final solution.

Relations Between Frank-Wolfe, Averaging, Decomposition and DSP-MB

We have found some relations between the FW algorithm and our DSP-MB, the primal decomposition and the averaging approach as follows:

- *Relations between the FW and our DSP-MB*: In our DSP-MB, the linear approximation of the objective function $F_{x_k}(x)$ is the pricing function. The gradient at any solution is the prices of that solution.
- *Relations between the FW and the primal decomposition*: In our household subproblem, we schedule jobs and the battery given the prices from the DRSP, which is equivalent to finding a new solution given a fixed the gradient in Step 3 of the FW algorithm. In the pricing master problem, we calculate new prices given the total demand profile of households, which is equivalent to calculating the new gradient at a given solution in Step 2.
- *Relations between the FW and the averaging approach*: Equation 4.10 in Step 5 of the FW algorithm has the same form as Equation 4.9 in the averaging approach.

This comparison has informed us that:

- Step 1, 2 and 3 of the FW algorithm match with the purposes of the subproblem and the master problem in our primal decomposition.
- The similarity between Equation 4.10 in the FW algorithm and Equation 4.9 in the averaging approach allows us to 1) interpret the step size α_k in the FW algorithm in the same way as that in the averaging approach: the probability for each household to adopt the new schedule y_k at iteration z; and 2) integrate Step 4 and 5 of the FW algorithm in the same way as integrating the averaging approach into our primal decomposition, which is illustrated as Figure 4.11.

Different from the averaging approach, the FW algorithm re-calculates the step size α_k at each iteration to ensure the objective value is reduced in the fastest way. Moreover, by adopting this variable step size, we transform our DSP-MB into a continuous convex optimisation problem whose goal is to find a sequence of step sizes that minimises the objective function, instead of the exact start times of jobs and charge/discharge profiles of batteries. The prices and the total objective value optimised by the FW approach will

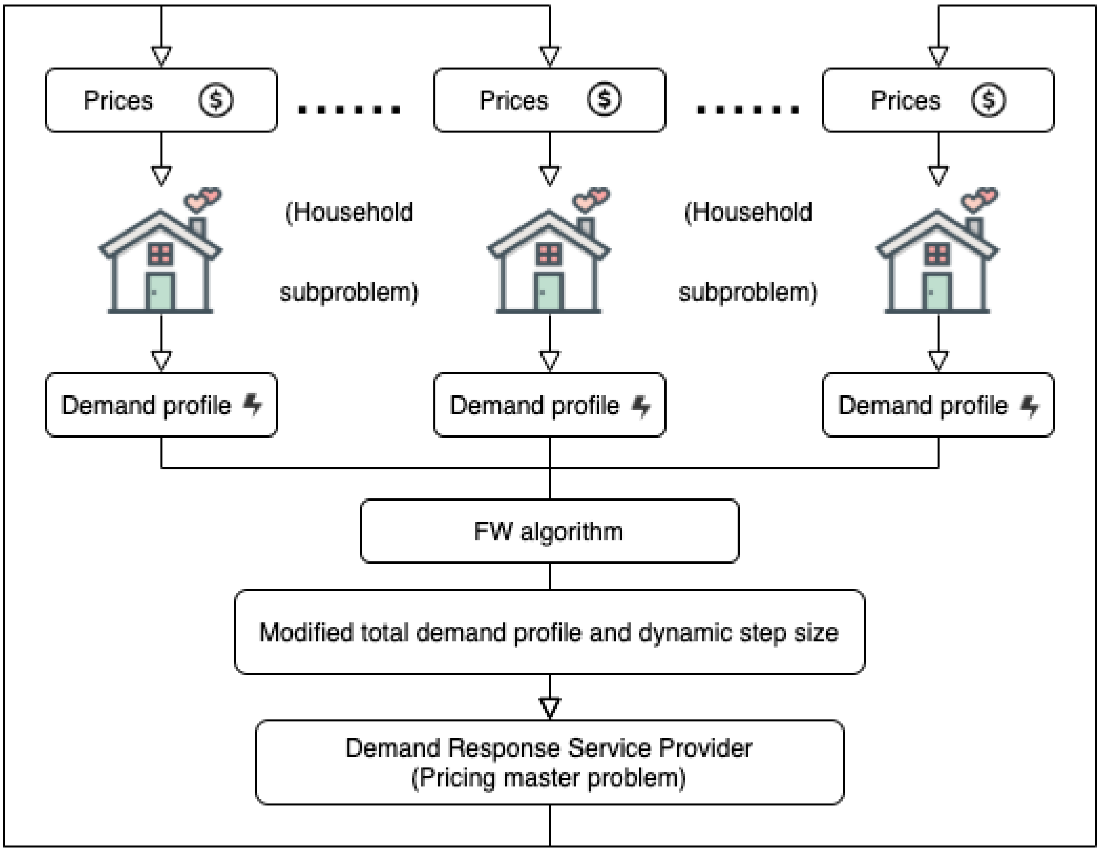


Figure 4.11: Iterations between the DRSP and households with the FW algorithm

be guaranteed to be global minimum. We explain this problem transformation in more details in the following subsection.

Problem Transformation

By integrating the FW algorithm, our pricing master problem at each iteration is now concerned with calculating prices based on the best step size and the total demand profiles in the current and previous iterations (Step 2, 4 and 5). Since a step size of the FW algorithm can be interpreted as the probability of adopting the optimal schedules at an iteration, the total demand profile used for pricing can also be seen as the total expected demand profile (see Section 4.4.2 for explanations) of all households. Consequently, the supply cost is calculated from the total expected demand profile instead of the total demand profile at any iteration.

Since the step size is optimised per iteration to reduce the objective value in the fastest way, the total expected demand profile is also optimised to find the lowest possible prices at each iteration. This means, the solution to our DSP-MB is in fact now the optimal total expected demand profile that minimises the objective value; and the decision variables are now the best step size per iteration instead of the exact start times of jobs and the charge/discharge profiles of batteries. In other words, we essentially allow a job or a battery to be used at any time and we optimise the probability of using a job or a battery at each feasible time interval, so that the total expected objective value is minimum. Note that, due to the scheduling time constraints of jobs, the probability of starting a job outside its feasible time intervals is zero.

Although this FW approach optimises the total expected demand profile instead of the total demand profile at any iteration, we show that we can achieve this optimal total expected demand profile and the minimum objective value in practice using a probability-based scheduling method, to be described in Section 4.5.

Since the step size or the probability is continuous, our DSP-MB is now a continuous convex optimisation problem instead of the mixed-integer non-linear problem discussed in Section 3.6 of Chapter 3. Therefore, the global optimality of the best solution can be guaranteed by the FW algorithm. We prove the convexity of this transformed problem in the following subsection.

Proof of Convexity

Let us write $\mathbf{L}_x^a$ and $\mathbf{L}_y^a$ as two total demand profiles (or demand vectors in mathematical terms) of households, calculated from different feasible job and battery schedules of households. Let us consider an arbitrary point $\mathbf{L}_z^a$ on the line joining these two demand vectors as follows:

$$\mathbf{L}_z^a = \mathbf{L}_x^a + \alpha(\mathbf{L}_y^a - \mathbf{L}_x^a), \text{ where } \alpha \in [0, 1] \tag{4.11}$$

Since all constraints of jobs and batteries are linear, $\mathbf{L}_z^a$ is also feasible. To prove that the objective function is convex, we must show that the objective value of $\mathbf{L}_z^a$ is less than its linear approximation as follows:

$$f(\mathbf{L}_z^a) < f(\mathbf{L}_x^a) + \alpha(f(\mathbf{L}_y^a) - f(\mathbf{L}_x^a)) \tag{4.12}$$

The objective value consists of the supply cost for all households and the total inconvenience cost. Since the inconvenience cost function is linear, it is sufficient to prove that the supply cost function satisfies the following:

$$C^a(\mathbf{L}_z^a) < C^a(\mathbf{L}_x^a) + \alpha(C^a(\mathbf{L}_y^a) - C^a(\mathbf{L}_x^a)) \tag{4.13}$$

Since for any demand vectors $\mathbf{L}_v^a$, the total supply cost is the sum of the cost at each period n:

$$C^a(\mathbf{L}_v^a) = \sum_{n=1}^{48} C^a(L_{v,n}^a) \tag{4.14}$$

It is sufficient to proof convexity by proving for each period n that:

$$C^a(L_{z,n}^a) \leq C^a(L_{x,n}^a) + \alpha(C^a(L_{y,n}^a) - C^a(L_{x,n}^a)) \tag{4.15}$$

With the use of α, we can rewrite this expression as follows:

$$\begin{aligned} &\alpha(C^a(L_{y,n}^a) - C^a(L_{x,n}^a)) \\ =&\alpha(C^a(L_{y,n}^a) - C^a(L_{z,n}^a) + C^a(L_{z,n}^a) - C^a(L_{x,n}^a)) \\ =&\alpha(C^a(L_{y,n}^a) - C^a(L_{z,n}^a)) + C^a(L_{z,n}^a) - C^a(L_{x,n}^a) - (1-\alpha)(C^a(L_{z,n}^a) - C^a(L_{x,n}^a)) \end{aligned} \tag{4.16}$$

To complete the proof we need to show that:

$$\begin{aligned} &C^a(L_{x,n}^a) + \alpha(C^a(L_{y,n}^a) - C^a(L_{x,n}^a)) - C^a(L_{z,n}^a) \\ =&C^a(L_{x,n}^a) + \alpha(C^a(L_{y,n}^a) - C^a(L_{z,n}^a)) + C^a(L_{z,n}^a) - C^a(L_{x,n}^a) \\ &- (1-\alpha)(C^a(L_{z,n}^a) - C^a(L_{x,n}^a)) - C^a(L_{z,n}^a) \\ =&\alpha(C^a(L_{y,n}^a) - C^a(L_{z,n}^a)) - (1-\alpha)(C^a(L_{z,n}^a) - C^a(L_{x,n}^a)) \\ \geq&0 \end{aligned} \tag{4.17}$$

Let $\delta = L_{y,n}^a - L_{x,n}^a$, then:

$$\begin{aligned} L_{z,n}^a - L_{x,n}^a &= \delta \times \alpha \\ L_{y,n}^a - L_{z,n}^a &= \delta \times (1-\alpha) \end{aligned} \tag{4.18}$$

Case 1: $C^a(L_{x,n}^a) < C^a(L_{y,n}^a)$ Let $p(L_{v,n}^a)$ be the price of any demand vector $\mathbf{L}_v^a$ in period n. From the price table $T_n(\cdot)$, find the level indexed by $ix < iz < iy$, with price levels $\hat{p}_{n,ix} = p(L^a_{x,n})$, $\hat{p}_{n,iz} = p(L^a_{z,n})$, $\hat{p}_{n,iy} = p(L^a_{y,n})$, and the corresponding consumption levels $\hat{e}_{n,ix}$, $\hat{e}_{n,iz}$, $\hat{e}_{n,iy}$. Accordingly:

$$\begin{aligned} &C^a(L^a_{z,n}) - C^a(L^a_{x,n}) \\ =&\hat{p}_{ix} \times (\hat{e}_{ix} - L^a_{x,n}) + \hat{p}_{iz} \times (L^a_{z,n} - \hat{e}_{iz-1}) + \sum_{ix<i<iz} (\hat{p}_i \times (\hat{e}_i - \hat{e}_{i-1})) \end{aligned} \tag{4.19}$$

The proof for this case follows, if we replace higher prices by lower prices in a positive context, and lower prices by higher prices in a negative context:

$$\begin{aligned} &\alpha(C^a(L^a_{y,n}) - C^a(L^a_{z,n})) - (1-\alpha)(C^a(L^a_{z,n}) - C^a(L^a_{x,n})) \\ =&\alpha \times (\hat{p}_{iz} \times (\hat{e}_{iz} - L^a_{z,n}) + \hat{p}_{iy} \times (L^a_{y,n} - \hat{e}_{iy-1}) + \sum_{iz<i<iy} (\hat{p}_i \times (\hat{e}_i - \hat{e}_{i-1})) \\ &- (1-\alpha) \times (\hat{p}_{ix} \times (\hat{e}_{ix} - L^a_{x,n}) + \hat{p}_{iz} \times (L^a_{z,n} - \hat{e}_{iz-1}) + \sum_{ix<i<iz} (\hat{p}_i \times (\hat{e}_i - \hat{e}_{i-1}))) \\ >&\alpha \times (\hat{p}_{iz} \times (\hat{e}_{iz} - L^a_{z,n}) + \hat{p}_{iz} \times (L^a_{y,n} - \hat{e}_{iy-1}) + \sum_{iz<i<iy} (\hat{p}_{iz} \times (\hat{e}_i - \hat{e}_{i-1}))) \\ &- (1-\alpha) \times (\hat{p}_{iz} \times (\hat{e}_{ix} - L^a_{x,n}) + \hat{p}_{iz} \times (L^a_{z,n} - \hat{e}_{iz-1}) + \sum_{ix<i<iz} (\hat{p}_{iz} \times (\hat{e}_i - \hat{e}_{i-1}))) \\ =&\hat{p}_{iz} \times (\alpha \times (L^a_{y,n} - L^a_{z,n}) - (1-\alpha) \times (L^a_{z,n} - L^a_{x,n})) \\ =&\hat{p}_{iz} \times (\alpha \times \delta \times (1-\alpha) - (1-\alpha) \times \delta \times \alpha) = 0 \end{aligned} \tag{4.20}$$

Case 2: $C^a(L^a_{x,n}) > C^a(L^a_{y,n})$ In this case the consumption levels are in reverse order, but the proof is similar.

Case 3: $C^a(L^a_{x,n}) = C^a(L^a_{y,n})$ In this case $C^a(L^a_{z,n}) = C^a(L^a_{x,n})$ and the linear approximation is equally $C^a(L^a_{x,n})$.

Q.E.D

Pricing Master Problem Model and Solving Method

Since our pricing master problem is now concerned with finding the best step size based on the Step 2, 4 and 5 of the FW algorithm, we can describe the steps of solving this pricing master problem as follows:

1. Calculate the total demand profile $\mathbf{L}_z^{total-itr-p}$ of households in the current iteration.

2. Compute the descent direction $\mathbf{d}_z$ from $\mathbf{L}_z^{total-itr-p}$ and the total expected demand profile $\mathbf{L}_{z-1}^{total-itr-fw}$ optimised for pricing in the previous iteration as follows:

$$\mathbf{d}_z = \mathbf{L}_z^{total-itr-p} - \mathbf{L}_{z-1}^{total-itr-fw} \tag{4.21}$$

   Since the demand profiles are calculated by households when the prices are fixed and the prices comprise the gradient of the objective function, $\mathbf{d}_z$ is the steepest descent direction at $\mathbf{L}_z^{total-itr-p}$.

3. Compute the best step size α_z along $\mathbf{d}_z$ that minimises f_z as follows:

$$\alpha_z = \arg\min_{\alpha\in[0,1]} \; f_z(\mathbf{L}_{z-1}^{total-itr-fw} + \alpha \mathbf{d}_z) \tag{4.22}$$

$$= \arg\min_{\alpha\in[0,1]} \; C_z^{total-itr}(\mathbf{L}_{z-1}^{total-itr-fw} + \alpha \mathbf{d}_z) + U_z^{total-itr}(\mathbf{L}_{z-1}^{total-itr-fw} + \alpha \mathbf{d}_z) \tag{4.23}$$

$$= \arg\min_{\alpha\in[0,1]} \; [\mathbf{L}_{z-1}^{total-itr-fw} + \alpha(\mathbf{L}_z^{total-itr-p} - \mathbf{L}_{z-1}^{total-itr-fw})] \tag{4.24}$$

$$\times R_n(\mathbf{L}_{z-1}^{total-itr-fw} + \alpha(\mathbf{L}_z^{total-itr-p} - \mathbf{L}_{z-1}^{total-itr-fw})) \times 24/N \tag{4.25}$$

$$+ U_z^{total-itr} + \alpha(U_z^{total-itr} - U_{z-1}^{total-fw}) \tag{4.26}$$

$$= \arg\min_{\alpha\in[0,1]} \; \sum_{n=1}^{N} [L_{n,z-1}^{total-itr-p-fw} + \alpha(L_{n,z}^{total-itr-p} - L_{n,z-1}^{total-itr-p-fw})] \times r_n \times 24/N \tag{4.27}$$

$$+ [U_{z-1}^{total-fw} + \alpha(U_z^{total-itr} - U_{z-1}^{total-fw})]\} \tag{4.28}$$

$$\tag{4.29}$$

4. Move $\mathbf{L}_{z-1}^{total-itr-fw}$ to $\mathbf{L}_z^{total-itr-p}$ along $\mathbf{d}_z$ by α_z as follows:

$$\mathbf{L}_z^{total-itr-fw} = \mathbf{L}_{z-1}^{total-itr-fw} + \alpha_z(\mathbf{L}_z^{total-itr-p} - \mathbf{L}_{z-1}^{total-itr-fw}) \tag{4.30}$$

5. Calculate new prices from $\mathbf{L}_z^{total-itr-fw}$ using the pricing table for each period. These prices are the ones that incorporate the demand profiles and their probabilities from all previous iterations.

6. Send the new prices back to households for another round of rescheduling as illustrated in Figure 4.11.

Solving this pricing problem is straightforward except for Step 3 where a linear optimisation problem is required to solve to find the best step size. Since the only constraint for this problem is that the step size must take a value in [0,1], we can solve this problem by simply finding the α_z at which the gradient of the linear approximation of the objective function $f_z(\mathbf{L}_z^{total-itr-fw}{}_{-1} + \alpha\mathbf{d}_z)$ reaches zero (or turns positive) without using a solver. The problem in Step 3 can be re-written as follows:

$$\begin{aligned}
h(\alpha) &= \nabla f_z(\mathbf{L}_{z-1}^{total-itr-fw} + \alpha\mathbf{d}_z) && (4.31)\\
&= \nabla\{\sum_{n=1}^{N}[L_{n,z-1}^{total-itr-p-fw} + \alpha(L_{n,z}^{total-itr-p} - L_{n,z-1}^{total-itr-p-fw})] \times r_n && (4.32)\\
&\times 24/N + [U_{z-1}^{total-fw} + \alpha(U_z^{total-itr} - U_{z-1}^{total-fw})]\} && (4.33)\\
&= \sum_{n=1}^{N}(L_{n,z}^{total-itr-p} - L_{n,z-1}^{total-itr-p-fw}) \times r_n \times 24/N && (4.34)\\
&+ (U_z^{total-itr} - U_{z-1}^{total-fw}) && (4.35)\\
&= 0 && (4.36)\\
& && (4.37)
\end{aligned}$$

Since the pricing table is a step function, the gradient changes in steps instead of continuously, therefore the objective value changes only when the total demand of a period exceeds the next (higher or lower) consumption level of the pricing table for that period. We propose to find the best step in an iterative manner which is described as follows:

1. Set $\alpha_z = 0$.

2. For each time period n, move the total expected demand $L_{n,z-1}^{total-itr-p-fw}$ along the descent direction $d_{n,z}$ to the next (higher or lower) consumption level $e_{n,fw}^{level}$ where the price for that period changes.

3. Calculate the step size $\alpha_{n,z}$ for each time period as follows:

$$\alpha_{n,z} = (e_{n,fw}^{level} - L_{n,z-1}^{total-itr-p-fw})/d_{n,z} \tag{4.38}$$

4. Calculate the smallest step size α_z^{tent} calculated from the step size per period calculated in Step 3 as follows:

$$\alpha_z^{tent} = min(\{\alpha_{n,z} \mid n \in [1, N]\}) \tag{4.39}$$

   This α_z^{tent} is the smallest distance that the total expected demand $L_{n,z-1}^{total-itr-p-fw}$ needs to move along the descent direction to change the price and reduce the objective value.

5. Update α_z using the smallest step size found in Step 4 as follows:

$$\alpha_z = \alpha_z + \alpha_z^{tent} \tag{4.40}$$

6. Use the updated α_z to update the total expected demand profile as follows:

$$\forall n \in [1, N],\ L_{n,z}^{total-itr-p-fw} = L_{n,z-1}^{total-itr-p-fw} + \alpha_z * d_{n,z} \tag{4.41}$$

7. Calculate the gradient of the linear approximation function at the updated total expected demand profile $h(\alpha) = \nabla f_z(\mathbf{L}_{z-1}^{total-itr-fw} + \alpha \mathbf{d}_z) = \nabla f_z(\mathbf{L}_z^{total-itr-fw})$:

   - If the gradient is less than zero: $h(\alpha) < 0$, repeat Step 2 — 7.
   - If the gradient is greater than zero: $h(\alpha) > 0$, we remove the last α_z^{tent} from α_z as follows:

$$\alpha_z = \alpha_z - \alpha_z^{tent} \tag{4.42}$$

This α_z is the optimal step size we seek for the pricing master problem at iteration z. We remove the last α_z^{tent} to ensure the total expected demand profile calculated in the next step will not increase the objective value at all.

- If the gradient is zero: $h(\alpha) = 0$, we make no change to α_z and use it as the optimal step size.

Convergence Rate

Two types of iterations are involved in our FW approach: 1) the *outer iterations* between the household subproblem and the pricing master problem for finding a sequence of step sizes that minimises the objective value, and 2) the *inner iterations* within the pricing master problem for calculating the best step size at each iteration. We do not consider the iterations involved in the solvers as those information are often unknown to users especially when using commercial solvers. We have analysed the convergence rates for these iterations as follows:

- *Outer iterations*: The details of jobs and batteries are handled by the household subproblem. The pricing master problem needs only the total demand profile and the total inconvenience value of households to calculate the objective value. This means, the number of iterations required for finding a sequence of step sizes are affected by the total demand profile and the total inconvenience value, and the number of time intervals in a demand profile is fixed regardless of the number of households. In other words, increasing or decreasing the number of households in the DSP-MB changes the values of the total demand profile and the inconvenience value, however, it has very limited impacts on the number of iterations required for convergence. We demonstrate that the convergence speed for various number of households is near constant in the experimental results in Section 5.5.3 of Chapter 5.

- *Inner iterations*: The best step size is found by iteratively moving the total expected demand per period to the nearest consumption level until the gradient of the linear approximation function reaches zero or turns positive. This means, the number of iterations depend on the number of pricing levels and the distances between the total expected demands and the consumption levels. We have observed in the experiments that the number of inner iterations is high in the early outer iterations, however, reduces significantly as the objective value reduces in the later outer iterations. Nevertheless, the computation time for the pricing master problem is very small. The experimental results show that the computation time is close to zero.

4.5 Probability-Based Scheduling

Since we optimise the total expected demand profile of all households to achieve the lowest possible prices and the minimum objective value, we cannot simply use the prices or the job and battery schedules in the last iteration as the optimal solutions. However, we need to achieve the optimal total expected demand profile and its minimum objective value using a *probability-based scheduling method*. This probability-based scheduling method calculates a probability distribution using the best step size calculated at each iteration and selects the actual schedules for households uses this probability distribution.

4.5.1 Probability Distribution

As the optimal step size at a iteration is considered as the probability for households to adopt their new schedules calculated at that iteration, when the iterations converge, we can obtain an accumulated probability for each iteration z. Let us expand the formulation of the optimal total expected demand profile (or the probability-weighted total demand profile) at the last iteration Z: $\mathbf{L}_Z^{total-itr-fw}$ as Equation 4.43.

$$\begin{aligned}
\mathbf{L}_Z^{total-itr-fw} =& \mathbf{L}_{Z-1}^{total-itr-fw} + \alpha_Z * \mathbf{d}_Z && (4.43)\\
=& \mathbf{L}_{Z-1}^{total-itr-fw} + \alpha_Z * (\mathbf{L}_Z^{total-itr-p} - \mathbf{L}_{Z-1}^{total-itr-fw}) && (4.44)\\
=& (1-\alpha_Z) * \mathbf{L}_{Z-1}^{total-itr-fw} + \alpha_Z * \mathbf{L}_Z^{total-itr-p} && (4.45)\\
=& (1-\alpha_Z) * [(1-\alpha_{Z-1}) * \mathbf{L}_{Z-2}^{total-itr-fw} + \alpha_{Z-1} * \mathbf{L}_{Z-1}^{total-itr-p}] && (4.46)\\
& + \alpha_Z * \mathbf{L}_Z^{total-itr-p} && (4.47)\\
=& (1-\alpha_Z) * (1-\alpha_{Z-1}) * \mathbf{L}_{Z-2}^{total-itr-fw} && (4.48)\\
& + (1-\alpha_Z) * \alpha_{Z-1} * \mathbf{L}_{Z-1}^{total-itr-p} + \alpha_Z * \mathbf{L}_Z^{total-itr-p} && (4.49)\\
=& \prod_{z=1}^{Z}(1-\alpha_z) * \mathbf{L}_0^{total-itr-p} + && (4.50)\\
& \sum_{z=1}^{Z-1}(\prod_{i=z+1}^{Z}(1-\alpha_i) * \alpha_z * \mathbf{L}_z^{total-itr-p}) + \alpha_Z * \mathbf{L}_Z^{total-itr-p} && (4.51)\\
& && (4.52)
\end{aligned}$$

$\mathbf{L}_0^{total-itr-p}$ is calculated before the iterations start when jobs are assumed to start at their PSTs and no batteries are required.

The accumulated probability per iteration $prob_z$ is the coefficient of $\mathbf{L}_z^{total-itr-p}$ for $z = 0...Z$ in Equation 4.43. This accumulated probability is the weight of the tentative optimal total expected demand profile at each iteration, or the chance for households to adopt their optimal schedules calculated at each iteration to be their final schedules. Let us rewrite the accumulated probability per iteration z as Equation 4.53.

$$prob_z = \begin{cases} \prod_{i=1}^{Z}(1-\alpha_i) & \text{if } z = 0 \\ \alpha_z \prod_{i=z+1}^{Z}(1-\alpha_i), & \text{if } 1 < z < Z \\ \alpha_Z & \text{if } z = Z \end{cases} \tag{4.53}$$

Note that the accumulated probability per iteration is the same for all households as these probabilities are calculated from step sizes which are independent of any consumption details within households.

These accumulated probabilities comprise a probability distribution of the tentative optimal schedules of households at each iteration. We can use this probability distribution to choose the actual job schedule and battery charge/discharge profile for each household. We describe the steps for choosing the actual schedules in the following subsection.

4.5.2 Actual Household Schedule

The steps for choosing the actual schedules for each household using this probability distribution is straightforward as follows:

1. Each household independently selects a number i from 0 to Z (the total number of iterations) using the probability distribution.
2. Each household independently chooses the optimal job and the battery schedules calculated at iteration i during the FW-DDSM iterations as its actual schedules.

According to the law of large numbers (LLN) and the central limit theorem (CLT), when we choose schedules for a large number of households independently using a probability distribution in this way, the resulting total demand profile of all households approximates the optimal total expected demand profile with a very high probability. Moreover, the schedules chosen by each household enforce the constraints of jobs and batteries, maintaining the feasibility of the final solutions. We explain the effectiveness of the LLN and the CLT in our probability-based scheduling method in the following subsection.

4.5.3 Law of Large Numbers and Central Limit Theorem

The key to the effectiveness of this *probability-based scheduling method* lies in LLN (Evans and Rosenthal, 2009) and the CLT (*Central Limit Theorem*, 2008).

The LLN states that the average of random variables in a sample converges to the true mean of the population when the sample size is large and each variable has a finite expected value. In our DSP-MB, the random variables are the schedules or the demand profiles of households. The expected demand profile of each household is finite. Let us write the total actual demand profile of all households as $\mathbf{L}^{total-actual}$, the optimal total expected demand profile as $\mathbf{L}^{total-expected}$, and the total number of households as H. The sample average is $\mathbf{L}^{total-actual}/H$ and the true mean of the population is $\mathbf{L}^{total-expected}/H$. The LLN states that:

$$P(\lim_{H\to\infty} \mathbf{L}^{total-actual}/H = \mathbf{L}^{total-expected}/H) = 1 \tag{4.54}$$

The CLT states that the sum of a large number of random variables will always have approximately a normal distribution with the mean of this normal distribution being the true mean of the population. In other words, the sample average approximates the true mean of the population with a high probability when the sample size is large. Generally, a sample size of 30 is considered sufficiently large.

Considering the statements of LLN and CLT, we claim that when a large number of households sample schedules or demand profiles using this probability distribution calculated from the sequence of step sizes that lead to the optimal total expected demand profile, the resulting total actual demand profile of all households approximates the optimal total expected demand profile with a high probability. In other words, we can achieve the optimal solution calculated by the FW approach using this probability-based scheduling method. The effectiveness of such a probabilistic scheduling method has been demonstrated by Van Den Briel et al. (2013). We will demonstrate the effectiveness of our scheduling method in Section 5.5.4 of Chapter 5.

4.6 Summary

This chapter proposes a novel distributed and iterative scheduling algorithm called *Frank-Wolfe-based distributed demand scheduling method (FW-DDSM)* for solving a *demand scheduling problem for multiple households with batteries (DSP-MB)*. This algorithm uses primal decomposition, the Frank-Wolfe (FW) algorithm, a job scheduling module and a battery scheduling module.

The FW-DDSM decomposes the DSP-MB using the primal decomposition into a household subproblem and a pricing master problem. The method for solving the household subproblem include a job scheduling module and a battery scheduling module. We have proposed three types of solutions for the job scheduling module: a mixed-integer programming optimisation model, a constraint programming optimisation model and the Optimistic Greedy Search Algorithm. Each optimisation model has two versions: one with a data preprocessing algorithm and one without. The battery scheduling model is a linear programming optimisation model that includes Charnes-Cooper transformation. The method for solving the pricing master problem is based on the FW algorithm. The subproblem and the master problem are solved in an iterative manner until the objective value calculated by the pricing master problem does not reduce any more (or reduce no more than 0.01) in any two consecutive iterations.

After the iterations converge, the best step size calculated by the pricing master problem at each iteration is used to construct a probability distribution for sampling the actual schedules for households. We call this method the *probability-based scheduling method.* The law of large numbers and the central limit theorem have showed that by independently choosing schedules for a large number of households using this probability distribution, we will achieve the optimal total expected demand profile calculated by the FW-DDSM that has the minimum objective value in the last iteration.

There are a number of benefits of using our FW-DDSM to solve DSP-MBs:

1. The primal decomposition method decomposes a DSP-MB in a straightforward way without the additional transformation steps as in the dual decomposition, simplifying the solving process.

2. The decomposition allows households to schedule demands independently in parallel, eliminating the needs for iteratively broadcasting information to households sequentially.

3. To the best our knowledge, the FW-DDSM is the first method that applies the FW algorithm to solve DSPs, which is the key to achieve high efficiency and scalability with minimum manual parameter tuning.

4. The combination of the primal decomposition and the FW algorithm achieves a distributed and iterative algorithm whose convergence speed is minimally affected by the number of households, jobs and batteries.

5. The probability-based scheduling method allows households to choose from multiple feasible schedules using a probability distribution while causing very limited impacts on the optimality of the results, offering more flexibility for consumers.

Chapter 5

Experimental Result

This chapter presents the results that demonstrate the effectiveness of our *Frank-Wolfe-based distributed demand scheduling method (FW-DDSM)* proposed in Chapter 4 for solving *demand scheduling problems for multiple households with batteries (DSP-MBs).* First, we define a list of terms that are used to describe the variations of our FW-DDSM and to name the results in Section 5.1. Second, we present the experiment environment and our methods for generating the experiment data in Section 5.2 and Section 5.3, respectively. Third, we compare the mixed-integer programming (MIP), constraint programming (CP) and Optimistic Greedy Search Algorithm (OGSA) we have proposed for solving the household subproblem in Section 5.4. Fourth, we demonstrate the scalability and efficiency of our FW-DDSM, and the effectiveness of our probability-based scheduling method in Section 5.5. Fifth, we evaluate the impacts of various problem parameters on the solutions and the scalability of our FW-DDSM in Section 5.6.

5.1 Preliminary

Let us define a list of terms for describing the variations of methods and naming the results this chapter.

Definition 5.1. *FW-DDSM-CP*: the FW-DDSM that uses the modified CP model and the date preprocessing algorithm as the method for solving the household subproblem.

Definition 5.2. *FW-DDSM-OGSA*: the FW-DDSM that uses OGSA as the method for solving the household subproblem.

Definition 5.3. *total demand profile (TDP)* : the total demand of all households per time period/interval.

Definition 5.4. *Preferred total demand profile* (preferred TDP): the total demand profile of all households where jobs are scheduled at their preferred start times before FW-DDSM is applied.

Definition 5.5. *optimal total demand profile* (optimal TDP): the total expected demand profile of all households computed at the last iteration of FW-DDSM-CP.

Definition 5.6. *improved total demand profile* (improved TDP): the total expected demand profile of all households computed at the last iteration of FW-DDSM-OGSA.

Definition 5.7. *peak-to-average ratio (PAR)*: the ratio of the peak demand to the average demand of the entire scheduling horizon (e.g. 24 hours in our experiments).

5.2 Experiment Environment

This section presents our experiment environment. All optimisation models were implemented in MiniZinc 2.2.3 (Nethercote et al., 2007). The MIP models were solved by Gurobi (Gurobi Optimization, LLC, 2021) 7.5.2 and the CP models were by Gecode (Gecode Team, 2017) 6.1.1. The experiments were performed on a virtual machine with 32 virtual CPUs and 64GB memory, provided by the NCRIS-funded Australian Research Data Commons (Department of Education, Skills and Employment, n.d.).

5.3 Data Generation Method

This section introduces our methods for generating time horizons, jobs, battery energy storage systems (batteries) and pricing data. Similar to existing works (Mhanna et al., 2016; Van Den Briel et al., 2013), this work synthetically generated data based on real-world data.

5.3.1 Time Horizon

We scheduled jobs on a time horizon of 144 ten-minute scheduling intervals and 48 thirty-minute pricing periods in all experiments. This design is different from the majority of existing works where 48 half-hourly periods or 24 hourly periods are used for both scheduling and pricing. Using a finer granulated scheduling horizon increases the number of decision variables in the DSP-MB, making it more time consuming to solve, however, it also offers more flexibility for consumers to use their devices. Moreover, our FW-DDSM can efficiently find the optimal solution to a DSP-MB despite the granularity of the scheduling horizon.

5.3.2 Household Demand Data

Each household had five to ten jobs, of which up to nine were sequential. The size of the battery in each household was assumed to be the same, however, varied in each experiment. Each job of a household was generated in the following steps:

1. used the average working-day demand profile of Victoria, Australia in 2018 obtained from the Australian Electricity Market Operator (AEMO) website to generate a probability distribution that describes the probability of a job being set to turn on by a consumer at each time interval,
2. sampled the preferred start time (PST) using the above probability distribution,
3. sampled the duration using the Rayleigh distribution,
4. sampled the power rate from a list of commonly used appliances obtained from the Ausgrid website (Ausgrid, n.d.),
5. sampled the earliest start time (EST) and the latest finish time (LFT) using the uniform distribution,
6. sampled the care factor (CF) between 1 and 10 using the uniform distribution,
7. selected from 0 and 1 randomly to decide if this job had a predecessor. If 1 (yes), then selected a predecessor and the maximum succeeding delay (MSD) using the uniform distribution.

According to the law of large numbers (LLN), when a large number of jobs were scheduled at their PSTs using the probability distribution generated in Step 1, the resulting preferred TDP would approximate the demand profile used for generating this probability distribution. Figure 5.1 demonstrates the effectiveness of this method by showing the preferred TDP of 10000 synthetically generated households, and the average working-day demand profile used for computing the probability distribution to sample the PSTs of jobs in these households. For simplicity, this work set the household demand limit as the total demand of all jobs, and used the Rayleigh distribution to sample durations as it provided a good approximation to empirically observed durations.

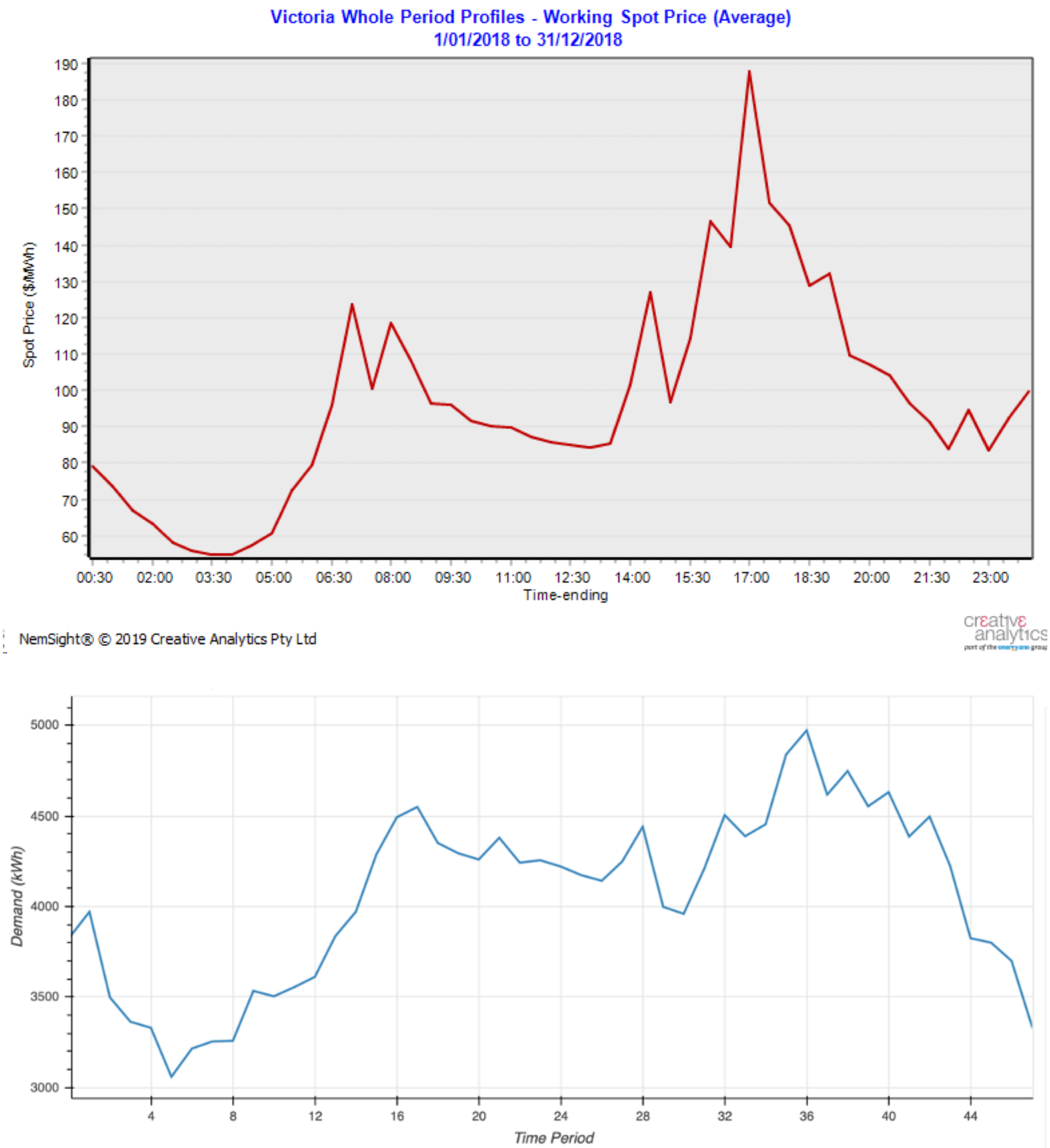


Figure 5.1: Average working-day demand profile obtained from AEMO for generating job and household data, and the preferred TDP of 10000 synthetic households

Demand Wrapping Around We allow each job to start from a time interval from the previous day and finish by another time interval the next day. When calculating the demand profile of a household, the demand of a job that occurred yesterday would be added to the end of today, and the demand that occurred tomorrow would be added to the start of today. We call this method the *demand wrapping around method*, which is illustrated in Figure 5.2. We used this method to generate a more realistic total demand profile, otherwise very little demand would appear at the beginning and the end of a day.

(a) Append the job demand from yesterday to the end of today

(b) Wrap the job demand from tomorrow to the start of today

Figure 5.2: Demand wrapping around method

5.3.3 Price Data

We generated fixed prices for evaluating the three types of solving methods for the household subproblem, and pricing tables for evaluating the FW-DDSM for the DSP-MB.

Fixed Prices

Twelve average monthly price profiles of Victoria, Australia in 2018 obtained from AEMO were used as the fixed prices for solving the household subproblem without the iterations of FW-DDSM, which is illustrated in Figure 5.3.

Pricing Table

A pricing table was created from a supply curve that was derived from one year of historical data from the Australian National Energy Market (NEM) where we used the average relationship between the wholesale market spot price and the total supply. The detailed steps are described as follows:

1. We downloaded a year of wholesale market spot prices[1], the metered demand of the whole NEM and the available electricity generation of the whole NEM from the AEMO website.

2. We calculated the percentage of the metered demand in the available electricity generation for every thirty-minute period. We called these percentages the *normalised demands*.

[1]The market spot price was the price that paid by all electricity retailers for all the bulk purchases they make on behalf of their retail customers, and all power station operators earn for power generated in the same half hour. The metered demand is the total demand of all consumers in the NEM. The available generation is total capacity of available generators. The data was recorded in thirty-minute intervals.

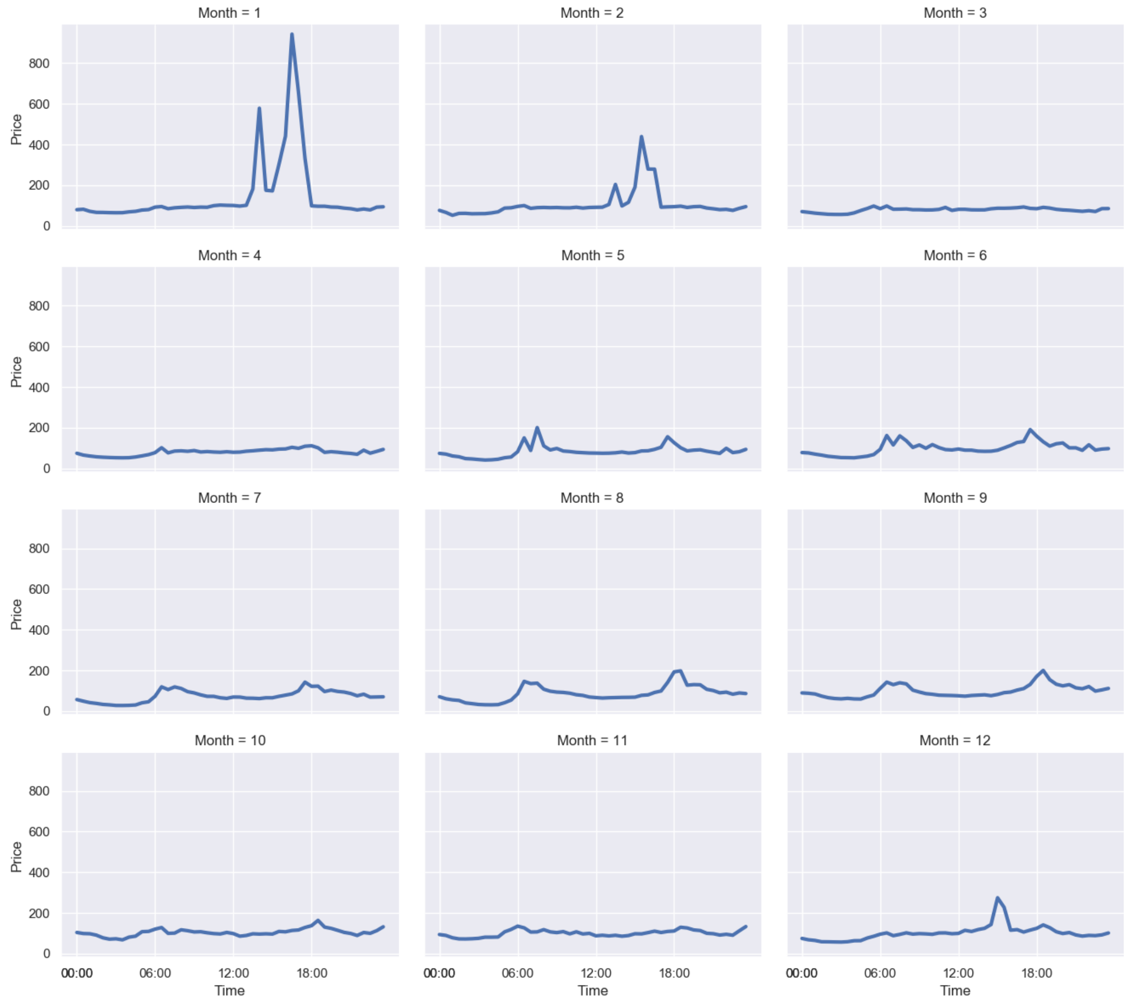


Figure 5.3: Twelve monthly average working-day price profiles of Victoria, Australia in 2018 used in the experiments

3. We derived a curve from the spot prices and the normalised demands, called the *supply curve*.

4. We discretised the supply curve evenly to retrieve a set of price levels and the corresponding normalised consumption levels.

5. We compiled a normalised pricing table from those pricing levels and the normalised consumption levels.

6. We rescaled the normalised pricing table according to the maximum demand of the preferred TDP in each problem instance as follows:

   (a) We calculated the maximum demand of the preferred TDP.

   (b) We computed new consumption levels for this problem instance by multiplying all normalised consumption levels by the maximum demand.

(c) We multiplied the new consumption levels with a scaler, such as 0.8, 1 or 1.2, to reduce or increase the consumption level for each price level. We called this scaler the *pricing table multiplier*.

For example, assuming there are 200 households to be scheduled, the highest or the second highest consumption level of the pricing table at each time period would be the maximum demand of the preferred TDP of those 200 households.

Implicit Area Demand Limit Constraint As discussed in Section 3.4.3 of Chapter 3, the design of our pricing table allows us to incorporate the *area demand limit constraint* implicitly by adding an extra consumption level that is the same as this demand limit and assigning a extremely high price for this consumption level. This implicit constraint was achieved in Step 6 where we rescaled the normalised pricing table to ensure the highest consumption level matched with the area demand limit.

5.4 Comparison of Job Scheduling Methods

This section presents the results of comparing the three types of methods (proposed in Section 4.3.2 of Chapter 4) to schedules jobs for households. Recall that in Section 2.3.4 of Chapter 2, we have discussed that few exiting works, if any, have compared the optimality and efficiency of the MIP, CP and heuristic methods for solving the demand scheduling problems for a single battery (DSP-SBs). Moreover, very limited works have investigated the application of CP on demand scheduling problems (DSPs) despite that CP has been shown to be effective and efficient for solving combinatorial problems such as scheduling problems (see Appendix B.3.4 for more details). We therefore have proposed two MIP models, two CP models, a data preprocessing algorithm and a heuristic approach (OGSA) for scheduling jobs per household in Section 4.3.2. This set of experiments first compared the run times of these optimisation models (MIP or CP) with and without the data preprocessing algorithm and the run times of the OGSA, and second compared the solutions of these three types of methods.

5.4.1 Problem Instance and Parameter Setting

For this set of experiments, we defined a problem instance as a set of synthetic jobs and a price profile for one household. In total, we created 1200 problem instances using 100 households and twelve monthly price profiles. We fixed the electricity cost weight to be 1: $\lambda^c = 1$ and the inconvenience value weight to be 900: $\lambda^u = 900$. We have chosen these weights because we observed from experiments that they are sufficiently large to produce the desired balance between these two objectives.

5.4.2 Evaluation of Run times

This experiment investigates the impacts of the data preprocessing algorithm on the run times of the optimisation models, and compares the run times of the optimisation models and those of the OGSA.

Impact of the Data Preprocessing Algorithm

- *Criteria*: In order to evaluate the impacts of the preprocessing algorithm on the optimisation models, we defined a *model run time ratio* as follows:

  Definition 5.8. *Model run time ratio* is the ratio of the initial model's run time to the modified model's run time, which is calculated as Equation 5.1. This ratio represents the number of times the initial MIP/CP model are slower than the modified MIP/CP model. A higher ratio means the modified model is faster.

$$\textit{model run time ratio} = \frac{\text{run time of the initial model}}{\text{run time of the modified model}} \tag{5.1}$$

- *Results*: The model run time ratios of the 1200 problem instances are illustrated in Figure 5.4. The aggregate model run time ratios such as the maximum, the minimum and the average ratios, and the standard deviations are presented in Table 5.1.

Table 5.1: Aggregate model run time ratios, initial models vs. modified models

Result Type	**Run time ratio**						
	Mean	**Std.**	**Min**	**25%**	**50%**	**75%**	**Max**
Ratios of the CP models	1.13	0.12	0.5	1	1.17	1.2	2
Ratios of the MIP models	1.29	0.16	0.69	1.22	1.25	1.38	2.78

Note: N% means N% of the problem instances.

Figure 5.4: Model run time ratios, initial models vs. modified models

Note: the legend *Gurobi* refers to the model run time ratios of the initial and modified MIP models, and *Gecode* refers to the model run time ratios of the CP models.

- *Analysis*: The results demonstrate that using the data preprocessing algorithm with the modified models were faster than using the models alone in most problem instances. Table 5.1 shows that the model run time ratios for the CP and MIP models were above 1.13 while the standard deviation is less than 0.2, which means the modified models with preprocessing were faster than the initial models in most instances. In the worst case, the run time of the modified CP model with preprocessing was 0.5 of the initial CP model, and the run time of the modified MIP model was 0.69 of the initial MIP model.

Comparison of MIP and CP Models

- *Criteria*: In order to evaluate the impacts of the optimisation techniques (MIP or CP) on run times, we defined a *solver run time ratio* as follows:

  Definition 5.9. *Solver run time ratio* is the run time of the MIP model to the run time of the CP model ratio, calculated as Equation 5.2. This ratio represents the number of times the initial/modified MIP model are slower than the initial/modified CP model. A higher ratio means the CP model is faster.

$$\textit{solver run time ratio} = \frac{\text{run time of Gurobi}}{\text{run time of Gecode}} \tag{5.2}$$

- *Results*: The solver run time ratios of the 1200 problem instances are illustrated in Figure 5.5. The aggregate solver run time ratios such as the maximum, the minimum and the average ratios, the standard deviations are presented in Table 5.2.

Table 5.2: Aggregate solver run time ratios, CP models vs. MIP models

Model type	**Solver run time ratio**						
	Mean	**Std.**	**Min**	**25%**	**50%**	**75%**	**Max**
Initial CP/MIP	1.87	0.37	1.08	1.67	1.83	2	4.86
Modified CP/MIP	1.64	0.28	0.75	1.5	1.6	1.8	3.17

Note: N% means N% of the problem instances.

- *Analysis*: The results show that the CP models were faster than the MIP models in most problem instances. Table 5.2 shows that the solver run time ratios for the initial and the modified models were above 1.5 while the standard deviation is less than 0.5, which means the CP models were faster in most instances. The initial CP

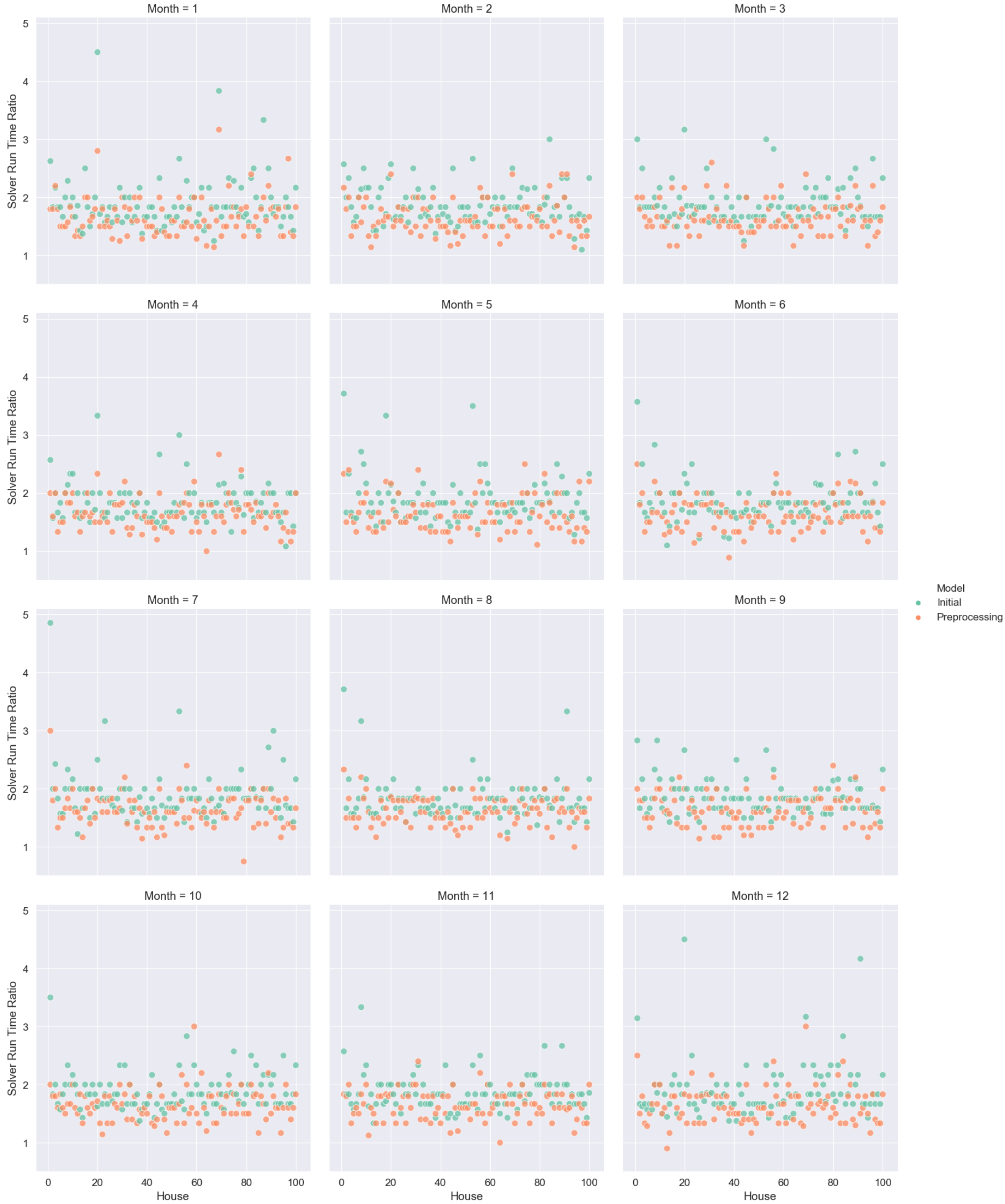


Figure 5.5: Solver run time ratios, CP models vs. MIP models

Note: the legend *Initial* refers to the solver run time ratios of the initial MIP and CP models, and *Modified* refers to the solver run time ratios of the modified MIP and CP models.

model was faster than the initial MIP in all problem instances. The run time of the modified CP model was 0.75 of the modified MIP in the worst case.

Comparison of the Optimisation Models and the Heuristic Method

- *Criteria*: We simply compared the run times of the heuristic method OGSA with those of the modified optimisation models with the data preprocessing algorithm.

- *Results*: The run times of the modified optimisation models with preprocessing, and OGSA are illustrated in Figure 5.6. The aggregate run times including the maximum, the minimum and the average values, and standard deviations are presented in Table 5.3.

Table 5.3: Aggregate run times of the modified CP/MIP models and OGSA

Method	**Run time (ms)**						
	Mean	**Std.**	**Min**	**25%**	**50%**	**75%**	**Max**
Modified CP model	58.98	8.41	50	50	60	60	140
Modified MIP	83.88	15.85	60	70	80	90	250
OGSA	0.00022	5.00E-05	0.00017	0.0002	0.00021	0.00022	0.00077

Note: N% means N% of the problem instances.

- *Analysis*: The results show that OGSA was at least four orders of magnitude faster than any optimisation models in all instances. Table 5.3 shows that the maximum run time of OGSA was 0.00077ms while the minimum run time of the modified CP mode was 50ms and the minimum run time of the modified MIP model was 60ms.

Finding from the Evaluation of Run times

From the experimental results of evaluating the run times of three types of job scheduling methods, we have validated our hypothesis that CP methods can be more efficient than MIP methods when solving *demand scheduling problems for a single household (DSP-SHs)*. Using a data preprocessing algorithm with a modified CP models can further reduce the run times. OGSA is extremely fast as expected for a heuristic method.

5.4.3 Evaluation of Solutions

This experiment presents the results of comparing the solutions of the modified optimisation models (MIP or CP) with the data preprocessing algorithm and those of the OGSA.

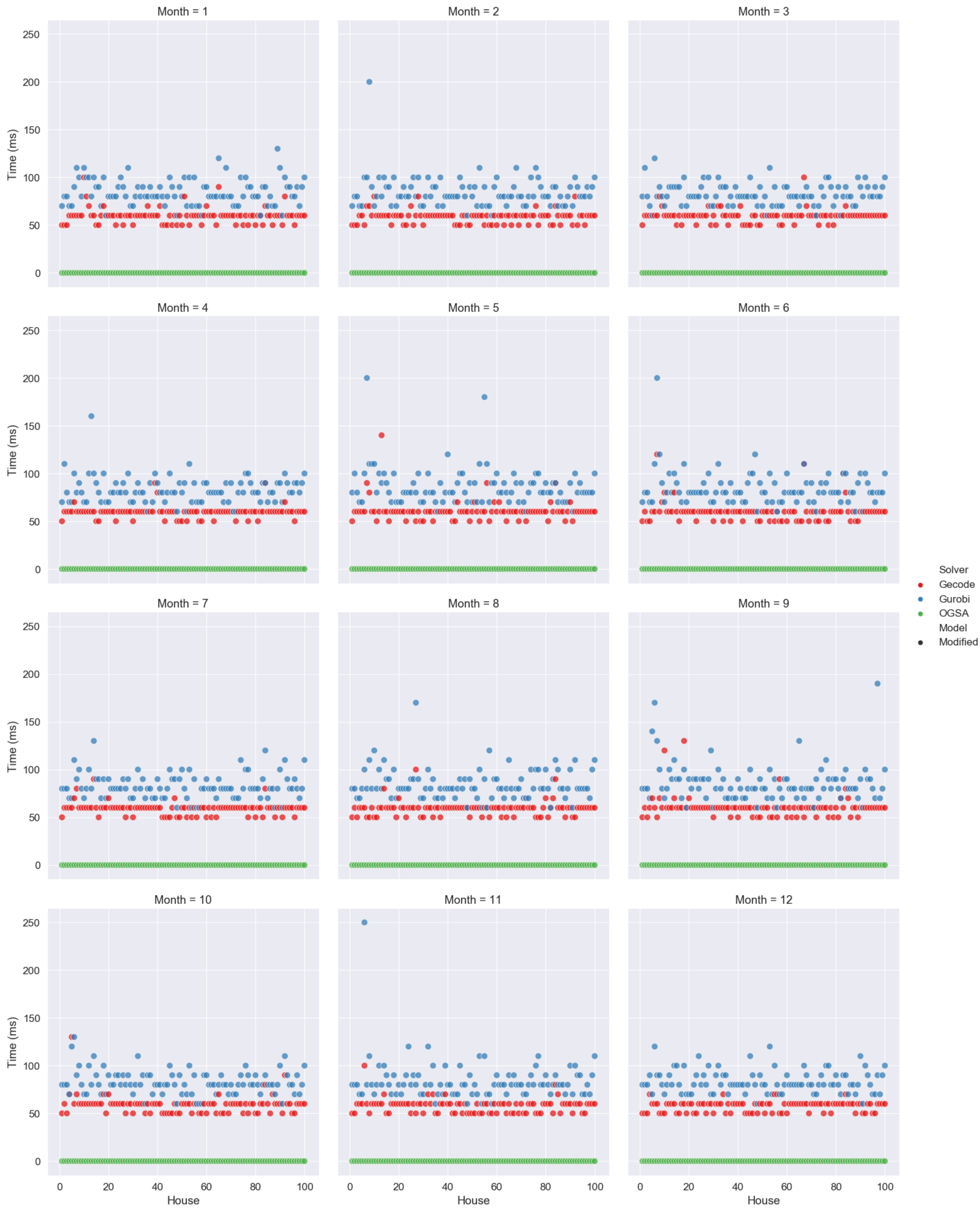


Figure 5.6: Run times of the modified CP/MIP models and OGSA

Note: the legend *Gurobi* refers to the run times of the modified MIP models, and *Gecode* refers to the run times of the modified CP models.

- *Criteria*: We compared the solutions of the optimisation models and the OGSA by evaluating their objective values. We defined an *objective ratio* as follows:

  Definition 5.10. *Objective ratio* is the ratio of the improved objective value found by OGSA and the optimal objective value found by the modified (CP or MIP) model, calculated as Equation 5.3. This ratio represents the number of times that the OGSA solution is worse than the modified model solution. A higher ratio means the objective value of the modified model is better.

$$\textit{objective ratio} = \frac{\text{suboptimal objective calculated by OGSA}}{\text{optimal objective calculated by a solver}} \tag{5.3}$$

- *Results*: The original objective values of the problem, the optimal objective values calculated by the optimisation models and the improved objective values computed by the OGSA are illustrated in Figure 5.8. The objective ratio of each problem instance is illustrated in Figure 5.7. The aggregate objective ratios such as the maximum, the minimum and the average ratios, the standard deviations are presented in Table 5.4. The average cost reductions for each method are presented in Table 5.5.

Table 5.4: Aggregate objective ratios for the OGSA and optimisation models

Ratio Type	**Mean**	**Std.**	**Min**	**25%**	**50%**	**75%**	**Max**
Objective ratio	1.05	0.16	1	1	1	1.01	4.17

Note: N% means N% of the problem instances.

Table 5.5: Average daily consumption cost reductions of the MIP, CP and OGSA solutions

Month of the Price Profile	**Method**		
	OGSA	**CP/MIP**	**Difference**
1	48%	48%	0%
2	35%	35%	0%
3	17%	17%	0%
4	22%	22%	0%
5	39%	39%	0%
6	38%	39%	1%
7	49%	51%	2%
8	53%	55%	2%
9	33%	34%	0%
10	24%	25%	1%
11	19%	19%	0%
12	31%	32%	1%

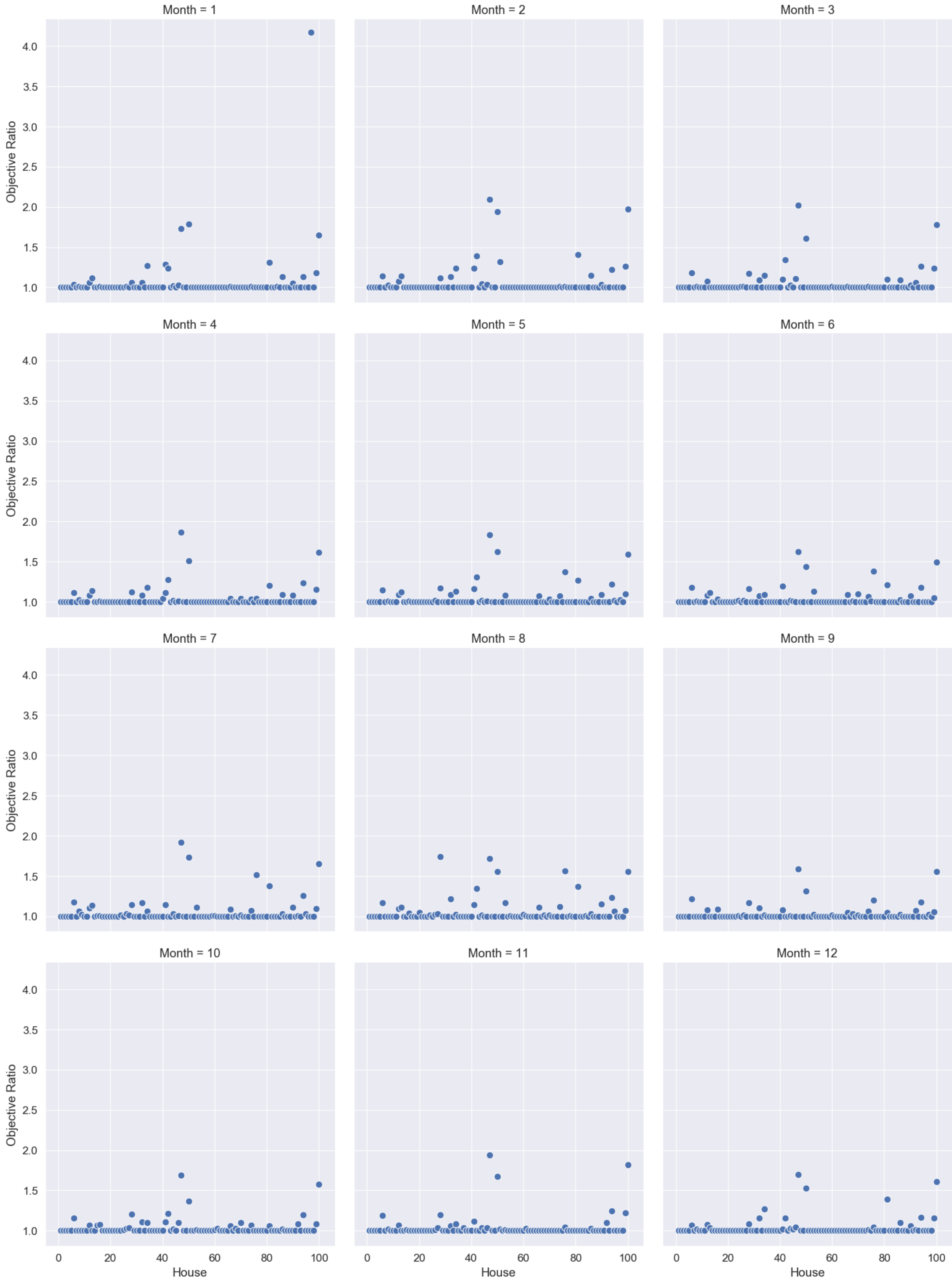


Figure 5.7: Objective ratios of the OGSA and optimisation models

Figure 5.8: Objective values of the MIP, CP and OGSA solutions

Note: the legend *Gurobi* refers to the run times of the modified MIP models, and *Gecode* refers to the run times of the modified CP models. A number of CP solutions are identical to MIP solutions and therefore not visible

- *Analysis*: The results demonstrate that the improved solutions of the OGSA were close to the optimal solutions of the optimisation models in many instances. Table 5.4 shows that the objective ratio was within 1.01 in 75% of the instances, the average objective ratio was 1.05 and the standard deviation was 0.16, which means the OGSA solutions were close to the optimal solutions (less than 1.21 times worse) in most instances. Moreover, Table 5.5 shows that the OGSA solutions achieved the same average cost reductions for seven price profiles as those of the optimal solutions, and were at most 2% worse than the optimal solutions for other price profiles.

Finding from the Evaluation of Solutions

The results have informed us that the OGSA was very fast and its solutions were close to the optimal solutions in many instances. Moreover, the OGSA is simple to implement without the needs of a solver, it can be used in practice when good solutions are sufficient and computation resources are limited.

5.4.4 Summary of Findings

We have summarised our main findings in experiments from comparisons of the job scheduling methods as follows:

1. The modified CP model with the data preprocessing algorithm was faster than the initial MIP and CP models, and the modified MIP model with the preprocessing.

2. The heuristic method OGSA was significantly faster than any MIP and CP models, and it can find good or near-optimal solutions in many instances.

5.5 Demonstration of Scalability and Optimality

This section demonstrates the scalability and optimality of our FW-DDSM including the effectiveness of our probability-based scheduling method. We tested two versions of our FW-DDSM: the version with the modified CP model and the preprocessing algorithm as the job scheduling method (FW-DDSM-CP), and the version with the OGSA as the job scheduling method (FW-DDSM-OGSA), because we were interested in investigating the trade-off between an optimal approach and a heuristic approach. Moreover, we evaluated the scalability and optimality of our FW-DDSM using problem instances with and without batteries, because we were interested in studying the impacts of using batteries on the optimality and the scalability of FW-DDSM. Note that when solving the instances without batteries, we simply deactivated the battery scheduling module (see Section 4.3.2 of Chapter 4 for explanations) in the household subproblem during the FW-DDSM iterations.

5.5.1 Problem Instance and Parameter Setting

For this set of experiments, we defined a problem instance as a number of households (e.g. 1000 households) and a pricing table rescaled based on the preferred TDP of these households. We created one set of instances without batteries and another with.

5.5.2 Convergence Condition

The condition for our FW-DDSM to converge to the optimal solutions is when the objective value calculated by the pricing master problem reduced within 0.01 in any two consecutive iterations.

Instances without batteries For the set of instances without batteries, we set the electricity cost weight at $\lambda^c = 1$, the inconvenience value weight at $\lambda^u = 5$, the EST of each job to be 0, and the LFT to be the last time interval of the day (144). In total, we generated five sets of problem instances for 50, 100, 500, 1000, 1500, 2000, 4000, 6000, 8000, and 10000 households, where each household had ten jobs. The pricing table was rescaled according to the preferred TDP of all households in each problem instance.

Instances with batteries For the set of instances with batteries, we set electricity cost weight at $\lambda^c = 1$, however, increased inconvenience value weight to be 10, in order to better illustrate the impacts of batteries on the total inconvenience value. We set the maximum capacity of the battery in each household at 3 kWh, the minimum capacity at 0 kWh and the maximum power rate at 3 kW. For simplicity, we assumed the efficiency to be 1. In total, we generated ten sets of problem instances for 200, 400, 800, 1000, 2000, 4000, 6000, 8000, and 10000 households. For comparison, five of these ten sets had batteries in households and the other five did not. Each problem instance had the same number of households and the same number of jobs per household. The pricing table was rescaled based on the preferred TDP of all households in each problem instance.

5.5.3 Demonstration of Scalability

This experiment demonstrates the scalability of our FW-DDSM for DSPs for multiple households with and without batteries.

Criteria

We evaluated the scalability of our FW-DDSM by showing the number of iterations for the FW-DDSM to converge to the optimal solutions and the run times of the household subproblem and the pricing master problem.

For Instances without Batteries

We applied both the FW-DDSM-CP and FW-DDSM-OGSA on each problem instance.

- *Results*: Table 5.6 presents the aggregate run times and the numbers of iterations, including the maximum, the minimum and the average values, which are calculated from problem instances of all numbers of households. Figure 5.9 shows the average scheduling/pricing time per iteration and the average number of iterations before convergence for each number of households.

Table 5.6: Run times and No. of iterations of the FW-DDSM-CP and FW-DDSM-OGSA

Method	Results	Mean	Min	Max
FW-DDSM-CP	Number of iterations	25.09	20.60	30.00
FW-DDSM-OGSA	Number of iterations	24.67	20.20	29.60
FW-DDSM-CP	Pricing time	0.01s	0.01s	0.01s
FW-DDSM-OGSA	Pricing time	0.01s	0.01s	0.01s
FW-DDSM-CP	Scheduling time	1,758.91s	19.68s	5,321.29s
FW-DDSM-OGSA	Scheduling time	6.63s	0.10s	19.92s
FW-DDSM-CP	Scheduling time/household	0.46s	0.39s	0.53s
FW-DDSM-OGSA	Scheduling time/household	0.00s	0.00s	0.00s

- *Analysis*: The results demonstrate that:

  1. The number of iterations before convergence was on average 25 for any number of households and any version of FW-DDSM.

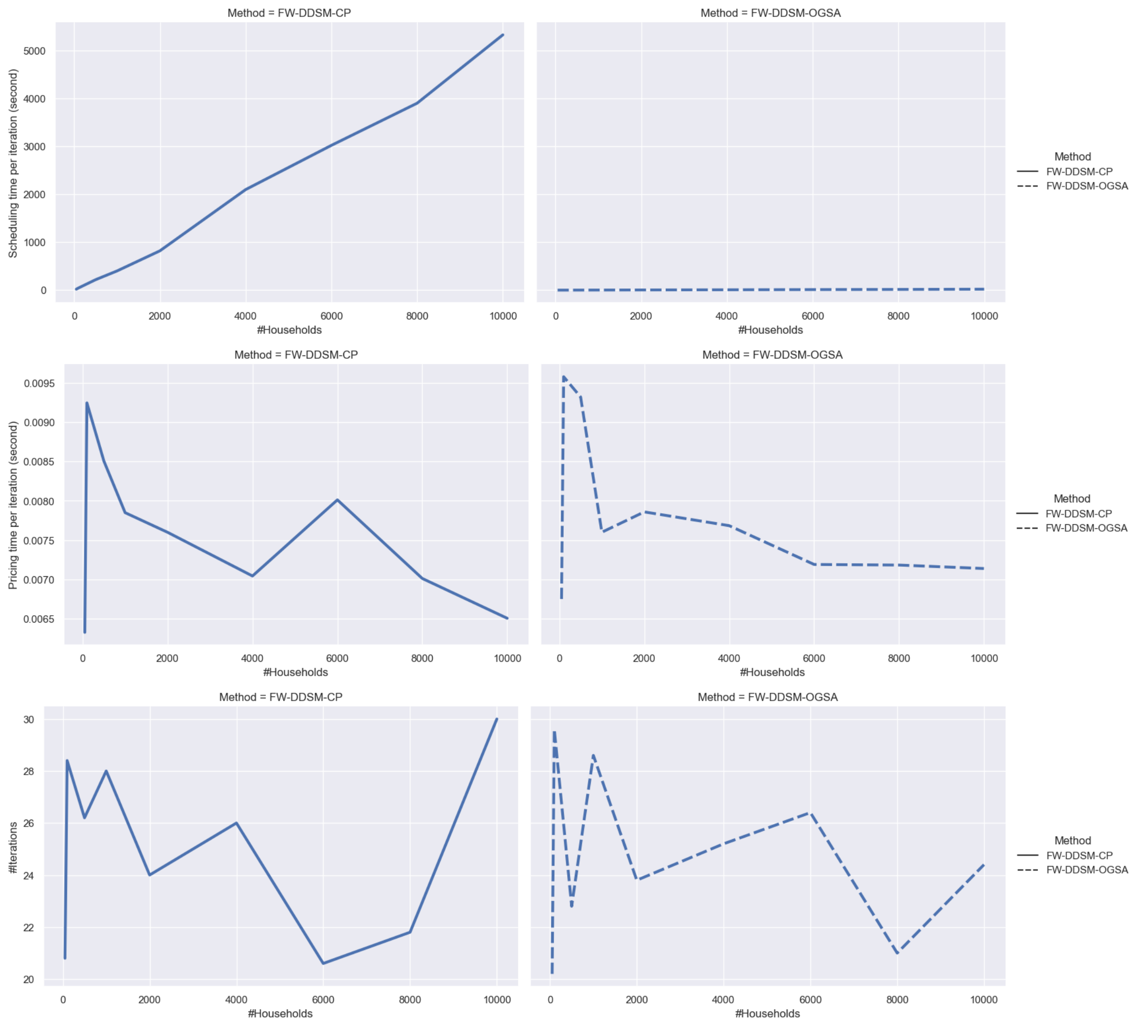


Figure 5.9: Average run times and No. of iterations of FW-DDSM-CP and FW-DDSM-OGSA for varying numbers of households

2. The average scheduling time of all households per iteration increased (almost) linearly with the number of households, which means the number of households had little impact on the scheduling time per household per iteration. The scheduling time per iteration per household of FW-DDSM-CP was on average 0.45 second for any number of household, however, that average scheduling time of FW-DDSM-OGSA was close to zero.
3. The average pricing time per iteration was less than 0.01 second for any number of households and any version of FW-DDSM.

For Instances with Batteries

We applied the FW-DDSM-CP on each problem instance to investigate the impacts of including batteries on the scalability.

- *Results*: Figure 5.10 shows the average number of iterations for convergence and the average time for solving the household subproblem per iteration for each number of households.
- *Analysis*: The results demonstrate that:

  1. The average number of iterations were under 20 for all numbers of households (with or without batteries). Less than 15 iterations were required when there were more than 2000 households.
  2. The average run times per household per iteration were a little higher for instances with batteries, although on average they were less than 0.12 second.

Findings from the Evaluation of Scalability

We have identified the following findings from the evaluation of the scalability of FW-DDSM:

1. *High scalability*: The FW-DDSM is highly scalable. Regardless of the choice of the job scheduling methods, the total scheduling time per household increased almost linearly with the number of households and the number of iterations for convergence was independent of the number of households. Moreover, the inclusion of batteries in households had minimum impacts on the scalability of our FW-DDSM.

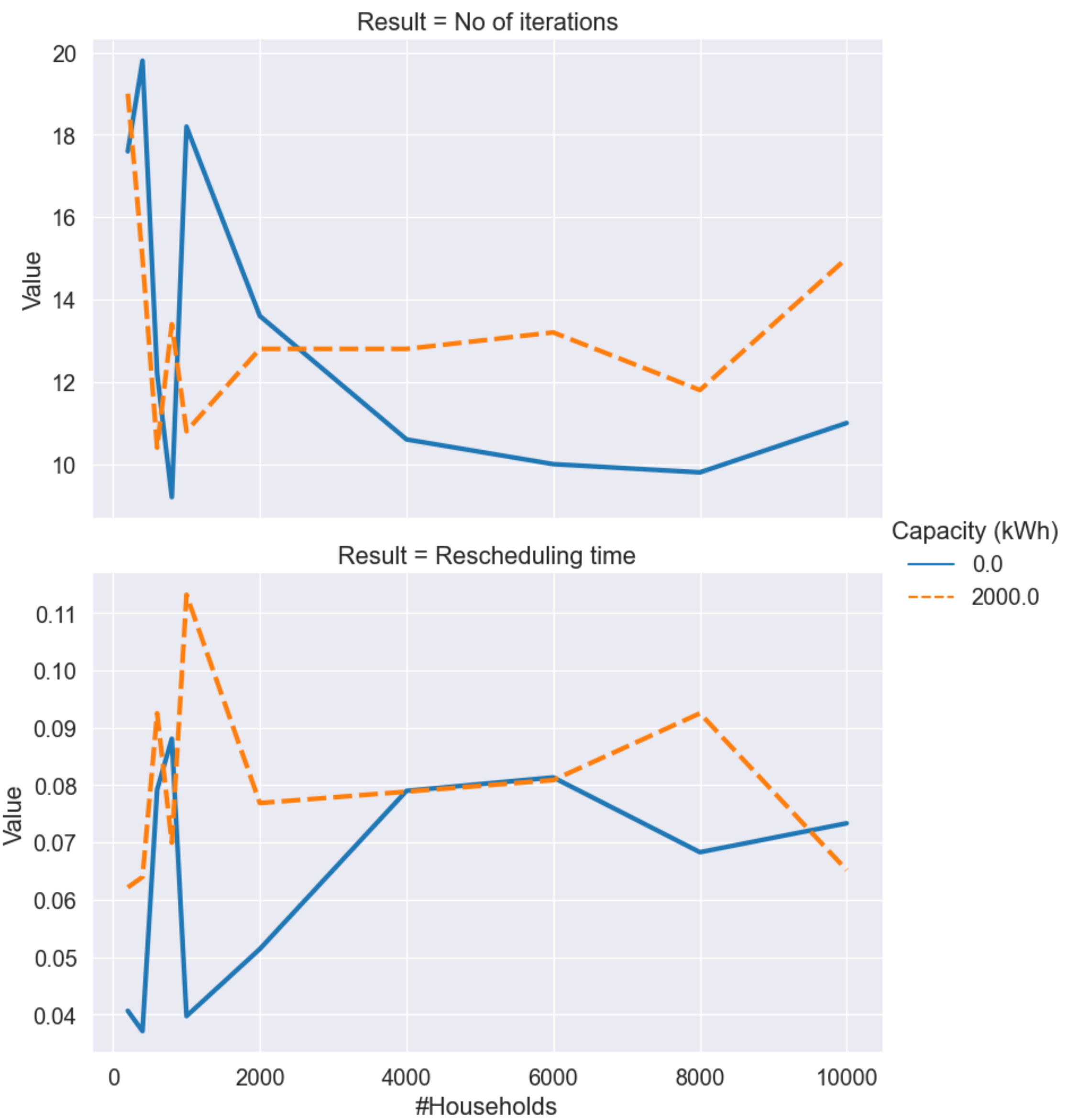


Figure 5.10: Average no. of iterations and scheduling times of the FW-DDSM for varying numbers of households.

2. *Low computation time*: Households can reschedule jobs in parallel in practice, which means the total run time (including pricing time and scheduling time) of FW-DDSM can be less than 10 seconds, making it suitable for real-time application.

Comparison with Existing Works

Our FW-DDSM converges to the optimal solutions in about 20 iterations for less than 10000 households with both appliances and batteries, while existing works achieved their optimal solutions in:

- 100 iterations for 5000 batteries (Ramchurn et al., 2011).
- 30 iterations for 100 users with only appliances (Chavali et al., 2014).
- 60 iterations for 2560 households with only appliances (Mhanna et al., 2016).
- 12 iterations for 4 users with appliances, batteries and small wind/solar generators (He et al., 2019).
- 23 iterations for 35 households with appliances and batteries (Kou et al., 2020).

5.5.4 Demonstration of Optimality

This experiment demonstrates the optimality of our FW-DDSM by showing the peak-to-average ratios (PARs), the maximum demands, the total inconvenience values and the total supply costs of the optimal TDPs (calculated from the FW-DDSM-CP iterations) and the improved TDPs (calculated from the FW-DDSM-OGSA iterations) for all instances.

We then show the effectiveness of our probability-based scheduling (proposed in Section 4.5 of Chapter 4) by comparing the cost reductions (CRs), the maximum demand reductions (DRds) and the PARs of the optimal and improved TDPs with those of the actual TDPs (computed from the probability-based scheduling method after the FW-DDSM-CP or FW-DDSM-OGSA iterations) for each problem instance. Note that for each problem, we have used the probability-based scheduling to generate five actual TDPs, in order to evaluate the average reductions of these actual TDPs.

Criteria

We defined the followings to evaluate the quality of the optimal, improved and the actual TDPs, and the differences in these three types of solutions:

1. CR: the cost reduction of the optimal or improved TDP of all households,
2. CR-D: the distance between the cost reduction of an actual TDP and that of the optimal or improved TDP, which is defined as Equation 5.4.

$$\text{CR-D} = \text{CR of the optimal or improved TDP} - \text{CR of the actual TDP} \tag{5.4}$$

3. DRd: the maximum demand reduction of the optimal or improved TDP,
4. DRd-D: the distance between the maximum demand reduction of an actual TDP and that of the optimal or improved TDP, which is defined as Equation 5.5.

$$\text{DRd-D} = \text{DRd of the optimal or improved TDP} - \text{DRd of the actual TDP} \tag{5.5}$$

5. PAR-D: the distance between the PAR of an actual TDP and that of the optimal or improved TDP, which is defined as Equation 5.6.

$$\text{PAR-D} = \text{PAR of the actual TDP} - \text{PAR of the optimal or improved TDP} \tag{5.6}$$

For Instances without Batteries

We applied both the FW-DDSM-CP and FW-DDSM-OGSA on each problem instance.

- *Results*: Figure 5.11 illustrates the PARs, the cost reductions and the maximum demand reductions of the optimal and improved TDPs. Figure 5.11 and Figure 5.11 demonstrate the preferred, the optimal or improved and the actual TDPs of each problem instance. Table 5.7, 5.8 and 5.9 present the PAR, the PAR distance (PAR-D), the cost reductions (CR), the cost reduction distances (CR-D), the demand reductions (DRd) and the demand reduction distances (DRd-D) of the actual and optimal or improved TDPs.

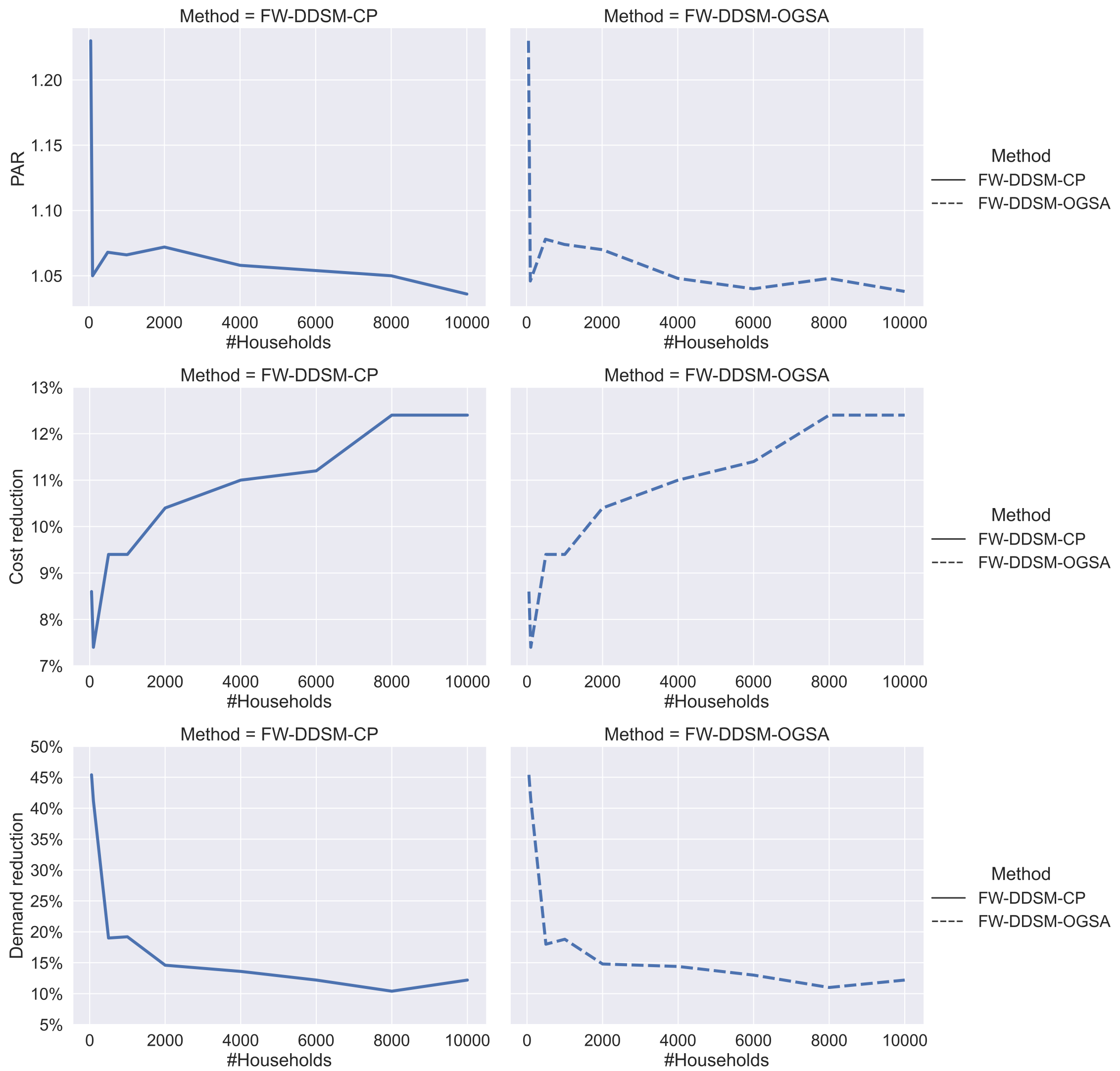


Figure 5.11: Average PARs and cost/demand reductions of the optimal and improved TDPs for varying numbers of households

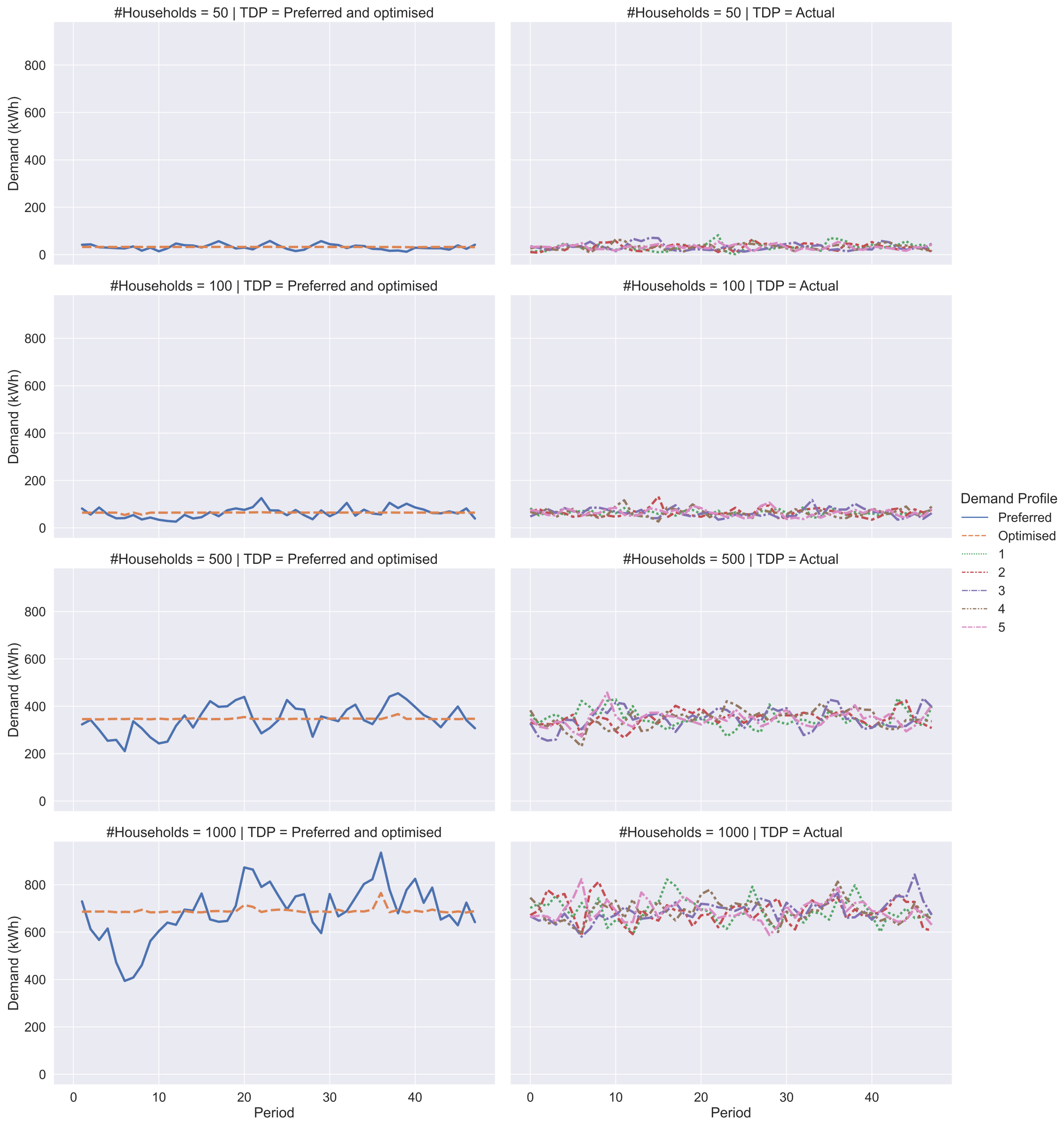

#Households = 50 | TDP = Preferred and optimised
#Households = 50 | TDP = Actual
#Households = 100 | TDP = Preferred and optimised
#Households = 100 | TDP = Actual
#Households = 500 | TDP = Preferred and optimised
#Households = 500 | TDP = Actual
#Households = 1000 | TDP = Preferred and optimised
#Households = 1000 | TDP = Actual
Demand (kWh)
0
200
400
600
800
Period
0
10
20
30
40
Demand Profile
Preferred
Optimised
1
2
3
4
5

Figure 5.11: Preferred, optimal and actual TDPs of FW-DDSM-CP for varying numbers of households

Note: legend 1/2/3/4/5 represents five actual TDPs generated after the FW-DDSM-CP iterations using the probability-based scheduling method

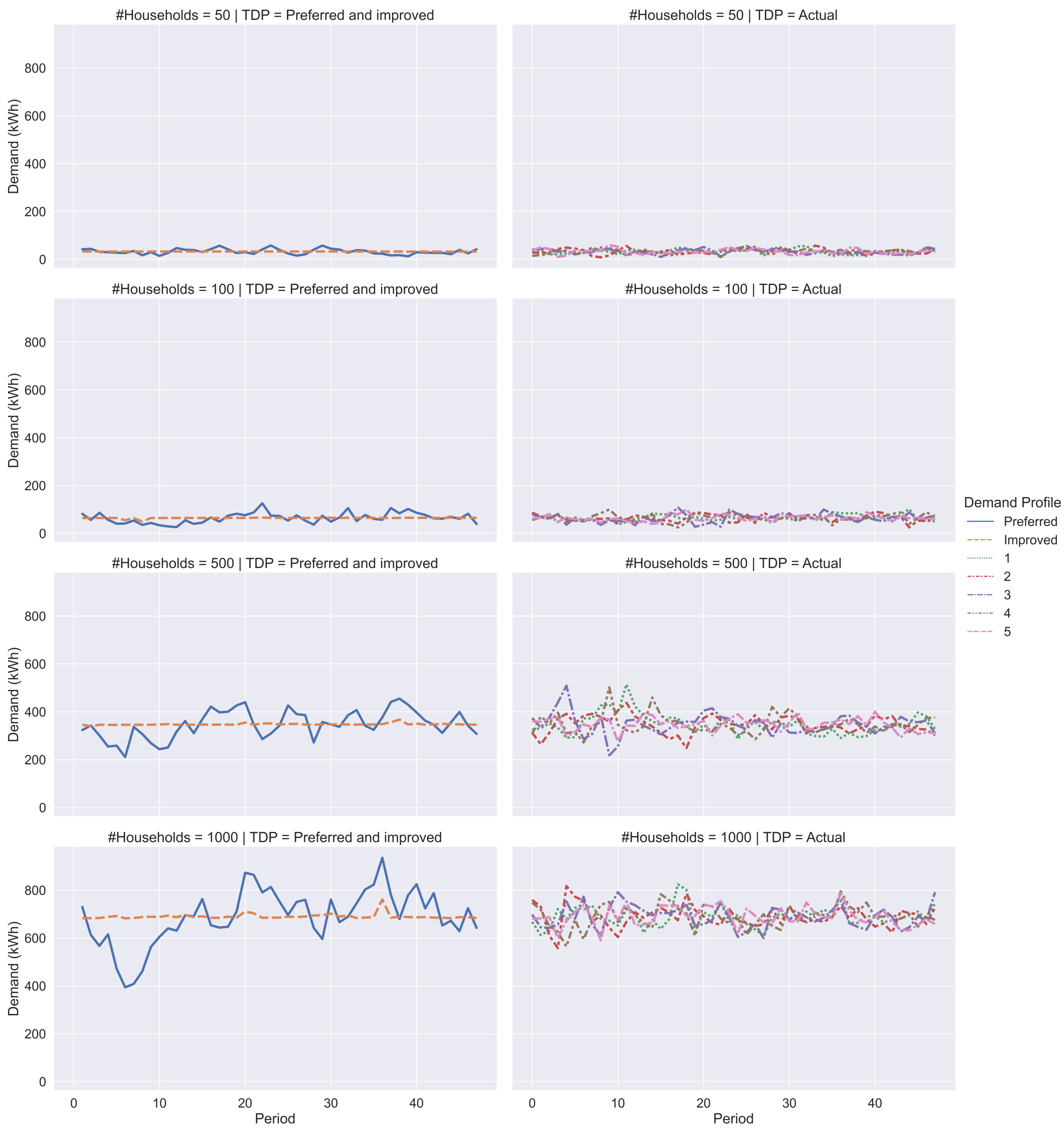

#Households = 50 | TDP = Preferred and improved
#Households = 50 | TDP = Actual
#Households = 100 | TDP = Preferred and improved
#Households = 100 | TDP = Actual
#Households = 500 | TDP = Preferred and improved
#Households = 500 | TDP = Actual
#Households = 1000 | TDP = Preferred and improved
#Households = 1000 | TDP = Actual
Demand (kWh)
Period
Demand Profile
Preferred
Improved
1
2
3
4
5

Figure 5.11: Preferred, improved and actual TDPs of FW-DDSM-OGSA for varying numbers of households

Note: legend 1/2/3/4/5 represents five actual TDPs generated after the FW-DDSM-OGSA iterations using the probability-based scheduling method

Table 5.7: PARs and differences in PARs of the FW-DDSM-CP and FW-DDSM-OGSA solutions for varying numbers of households

#House-holds	Result	TDP	FW-DDSM-CP Min	FW-DDSM-CP Median	FW-DDSM-CP Max	FW-DDSM-OGSA Min	FW-DDSM-OGSA Median	FW-DDSM-OGSA Max
50	PAR	Preferred	1.79	1.79	1.79	1.79	1.79	1.79
50	PAR	Optimal	1.02	1.02	1.02	1.01	1.01	1.01
50	PAR	Actual	1.72	1.99	2.55	1.61	1.72	1.85
50	*PAR-D*	*Actual*	*-0.7*	*-0.97*	*-1.53*	*-0.60*	*-0.71*	*-0.84*
100	PAR	Preferred	1.96	1.96	1.96	1.96	1.96	1.96
100	PAR	Optimal	1.03	1.03	1.03	1.03	1.03	1.03
100	PAR	Actual	1.4	1.82	2.07	1.42	1.57	1.69
100	*PAR-D*	*Actual*	*-0.37*	*-0.79*	*-1.04*	*-0.39*	*-0.54*	*-0.66*
500	PAR	Preferred	1.31	1.31	1.31	1.31	1.31	1.31
500	PAR	Optimal	1.06	1.06	1.06	1.06	1.06	1.06
500	PAR	Actual	1.21	1.25	1.32	1.15	1.44	1.48
500	*PAR-D*	*Actual*	*-0.15*	*-0.19*	*-0.26*	*-0.09*	*-0.38*	*-0.42*
1000	PAR	Preferred	1.36	1.36	1.36	1.36	1.36	1.36
1000	PAR	Optimal	1.11	1.11	1.11	1.1	1.1	1.1
1000	PAR	Actual	1.18	1.19	1.22	1.14	1.15	1.19
1000	*PAR-D*	*Actual*	*-0.07*	*-0.08*	*-0.11*	*-0.04*	*-0.05*	*-0.09*
2000	PAR	Preferred	1.26	1.26	1.26	1.26	1.26	1.26
2000	PAR	Optimal	1.1	1.1	1.1	1.1	1.1	1.1
2000	PAR	Actual	1.1	1.12	1.14	1.1	1.12	1.13
2000	*PAR-D*	*Actual*	*0*	*-0.02*	*-0.04*	*-0*	*-0.02*	*-0.03*
4000	PAR	Preferred	1.18	1.18	1.18	1.18	1.18	1.18
4000	PAR	Optimal	1.03	1.03	1.03	1.03	1.03	1.03
4000	PAR	Actual	1.05	1.06	1.07	1.06	1.07	1.13
4000	*PAR-D*	*Actual*	*-0.02*	*-0.03*	*-0.04*	*-0.03*	*-0.04*	*-0.10*
6000	PAR	Preferred	1.21	1.21	1.21	1.21	1.21	1.21
6000	PAR	Optimal	1.05	1.05	1.05	1.05	1.05	1.05
6000	PAR	Actual	1.05	1.05	1.08	1.06	1.08	1.12
6000	*PAR-D*	*Actual*	*0*	*0*	*-0.03*	*-0.01*	*-0.03*	*-0.07*

Continued on next page

Table 5.7: PARs and differences in PARs of the FW-DDSM-CP and FW-DDSM-OGSA solutions for varying numbers of households

#House-holds	Result	TDP	FW-DDSM-CP			FW-DDSM-OGSA		
			Min	Median	Max	Min	Median	Max
8000	PAR	Preferred	1.21	1.21	1.21	1.21	1.21	1.21
8000	PAR	Optimal	1.08	1.08	1.08	1.09	1.09	1.09
8000	PAR	Actual	1.07	1.09	1.11	1.07	1.08	1.09
8000	*PAR-D*	*Actual*	*-0.01*	*-0.01*	*-0.03*	*0*	*-0.01*	*-0.02*
10000	PAR	Preferred	1.16	1.16	1.16	1.16	1.16	1.16
10000	PAR	Optimal	1.04	1.04	1.04	1.04	1.04	1.04
10000	PAR	Actual	1.03	1.05	1.06	1.04	1.04	1.1
10000	*PAR-D*	*Actual*	*0*	*-0.01*	*-0.02*	*0*	*0*	*-0.06*
	0%							

Table 5.8: Demand reductions and differences in demand reductions of the FW-DDSM-CP and FW-DDSM-OGSA solutions for varying numbers of households

#House-holds	Result	TDP	FW-DDSM-CP			FW-DDSM-OGSA		
			Min	Median	Max	Min	Median	Max
50	DRd	Optimal	43%	43%	43%	43%	43%	43%
50	DRd	Actual	-43%	-12%	4%	-4%	4%	10%
50	*DRd-D*	*Actual*	*86%*	*55%*	*39%*	*47%*	*39%*	*33%*
100	DRd	Optimal	47%	47%	47%	47%	47%	47%
100	DRd	Actual	-6%	7%	28%	14%	20%	28%
100	*DRd-D*	*Actual*	*53%*	*40%*	*19%*	*33%*	*27%*	*19%*
500	DRd	Optimal	19%	19%	19%	19%	19%	19%
500	DRd	Actual	-1%	5%	7%	-13%	-10%	12%
500	*DRd-D*	*Actual*	*20%*	*14%*	*12%*	*32%*	*29%*	*7%*
1000	DRd	Optimal	18%	18%	18%	19%	19%	19%
1000	DRd	Actual	10%	12%	13%	12%	15%	16%
1000	*DRd-D*	*Actual*	*8%*	*6%*	*5%*	*7%*	*4%*	*3%*
2000	DRd	Optimal	13%	13%	13%	13%	13%	13%
2000	DRd	Actual	10%	11%	12%	10%	11%	13%

Continued on next page

Table 5.8: Demand reductions and differences in demand reductions of the FW-DDSM-CP and FW-DDSM-OGSA solutions for varying numbers of households

#House-holds	**Result**	**TDP**	**FW-DDSM-CP**			**FW-DDSM-OGSA**		
			Min	**Median**	**Max**	**Min**	**Median**	**Max**
2000	*DRd-D*	*Actual*	*3%*	*2%*	*1%*	*3%*	*2%*	*0%*
4000	DRd	Optimal	13%	13%	13%	13%	13%	13%
4000	DRd	Actual	9%	11%	11%	5%	10%	11%
4000	*DRd-D*	*Actual*	*4%*	*2%*	*2%*	*8%*	*3%*	*2%*
6000	DRd	Optimal	14%	14%	14%	15%	15%	15%
6000	DRd	Actual	11%	13%	14%	8%	11%	13%
6000	*DRd-D*	*Actual*	*3%*	*1%*	*0%*	*7%*	*4%*	*2%*
8000	DRd	Optimal	10%	10%	10%	10%	10%	10%
8000	DRd	Actual	8%	10%	11%	10%	11%	11%
8000	*DRd-D*	*Actual*	*2%*	*1%*	*0%*	*1%*	*1%*	*0%*
10000	DRd	Optimal	10%	10%	10%	11%	11%	11%
10000	DRd	Actual	9%	10%	11%	6%	10%	11%
10000	*DRd-D*	*Actual*	*1%*	*1%*	*0%*	*5%*	*1%*	*0%*

Table 5.9: Cost reductions and differences in cost reductions of the FW-DDSM-CP and FW-DDSM-OGSA solutions for varying numbers of households

#House-holds	**Result**	**TDP**	**FW-DDSM-CP**			**FW-DDSM-OGSA**		
			Min	**Median**	**Max**	**Min**	**Median**	**Max**
50	CR	Optimal	11%	11%	11%	11%	11%	11%
50	CR	Actual	-453%	-77%	6%	-15%	6%	8%
50	*CR-D*	*Actual*	*464%*	*88%*	*5%*	*26%*	*5%*	*3%*
100	CR	Optimal	6%	6%	6%	6%	6%	6%
100	CR	Actual	-33%	4%	6%	5%	5%	6%
100	*CR-D*	*Actual*	*39%*	*2%*	*0%*	*1%*	*1%*	*0%*
500	CR	Optimal	12%	12%	12%	12%	12%	12%
500	CR	Actual	5%	6%	9%	-44%	-38%	10%
500	*CR-D*	*Actual*	*7%*	*6%*	*3%*	*56%*	*50%*	*2%*
1000	CR	Optimal	7%	7%	7%	7%	7%	7%

Continued on next page

Table 5.9: Cost reductions and differences in cost reductions of the FW-DDSM-CP and FW-DDSM-OGSA solutions for varying numbers of households

#Households	**Result**	**TDP**	**FW-DDSM-CP**			**FW-DDSM-OGSA**		
			Min	**Median**	**Max**	**Min**	**Median**	**Max**
1000	CR	Actual	6%	6%	7%	6%	7%	7%
1000	*CR-D*	*Actual*	*1%*	*1%*	*0%*	*1%*	*0%*	*0%*
2000	CR	Optimal	9%	9%	9%	9%	9%	9%
2000	CR	Actual	7%	8%	8%	8%	8%	8%
2000	*CR-D*	*Actual*	*2%*	*1%*	*1%*	*1%*	*1%*	*1%*
4000	CR	Optimal	14%	14%	14%	14%	14%	14%
4000	CR	Actual	13%	13%	14%	13%	13%	13%
4000	*CR-D*	*Actual*	*1%*	*1%*	*0%*	*1%*	*1%*	*1%*
6000	CR	Optimal	13%	13%	13%	13%	13%	13%
6000	CR	Actual	12%	12%	12%	12%	12%	12%
6000	*CR-D*	*Actual*	*1%*	*1%*	*1%*	*1%*	*1%*	*1%*
8000	CR	Optimal	9%	9%	9%	9%	9%	9%
8000	CR	Actual	9%	9%	9%	9%	9%	9%
8000	*CR-D*	*Actual*	*0%*	*0%*	*0%*	*0%*	*0%*	*0%*
10000	CR	Optimal	13%	13%	13%	13%	13%	13%
10000	CR	Actual	13%	13%	13%	12%	13%	13%
10000	*CR-D*	*Actual*	*0%*	*0%*	*0%*	*1%*	*0%*	*0%*

- *Analysis*: The results demonstrate that:

  1. The optimal or improved PARs were under 1.10 for more than 50 households, which means the optimal or improved TDPs were mostly flat for any number of households when there more than 50 households.
  2. The cost reduction increased sublinearly with the number of households till 8000 households.
  3. The maximum demand reduction reduced with the number of households.
  4. All actual TDPs were better than the preferred TDPs. The distances of the cost reductions were less than 2% for both versions of FW-DDSM.

5. The distances of the maximum demand reductions were less than 5% for FW-DDSM-CP and less than 8% for FW-DDSM-OGSA for more than 1000 households.
6. At least one of the five actual TDPs was better than the preferred TDP for less than 1000 households.

- *Findings*: We have identified the following findings from the results:

1. We have validated our hypothesis (discussed in Section 4.5.3 of Chapter 4) that based on the law of large numbers and the central limit theorem, when a larger number of households and appliances (e.g. more than 1000 households with five shiftable jobs each) were scheduled using a probability distribution calculated from a sequence of step sizes that led to the optimal TDP or an improved TDP, the actual TDP of all households was very close to or even as same as this optimal or improved TDP.
2. In problem instances with more than 50 households, the optimal or improved TDPs were mostly flat regardless of the number of households. The cost reduction increased with the number of households, however, less maximum demand reduction was required to achieve a higher cost reduction when more households were involved.
3. In problem instances with more than 1000 households, the actual TDPs were very close to or as same as the optimal solutions. At least one of five TDPs were very close to the optimal or improved TDP when there were less than 1000 households. In practice, households can generate several actual schedules, report the corresponding demand profiles to the demand response service provider (DRSP) (as they do during the iterations), and let the DRSP to choose the best demand profiles for them.

For Instances with Batteries

We applied the FW-DDSM-CP on each problem instance to investigate the impacts of including batteries on the optimality. Since we have demonstrated the effectiveness of the probability-scheduling method in the results for the instances without batteries, we only evaluated the PARs, the maximum demands, the total inconvenience values and the total supply costs of the optimal TDPs for this set of problem instances.

- *Results*: Figure 5.12 shows the averages of the optimal PAR, maximum demand, total inconvenience value and total supply cost for each number of households.

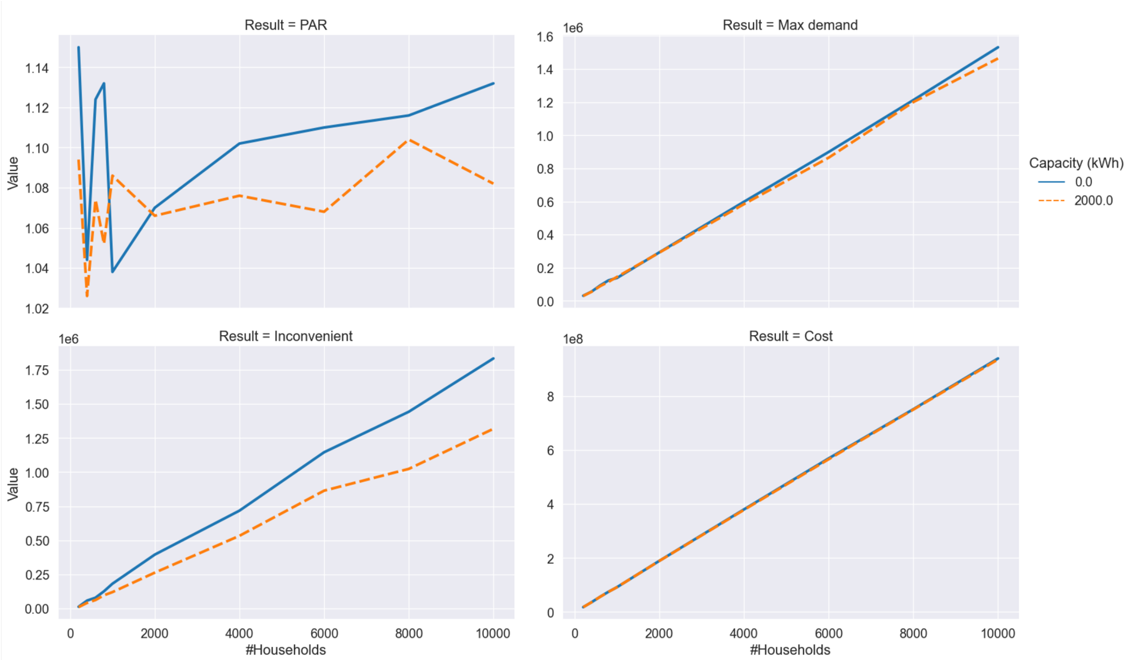


Figure 5.12: PARs, maximum demands, inconvenience values and supply costs of the optimal total demand profiles for varying numbers of households

- *Analysis*: The results show that including batteries further reduced the average of the optimal PARs in most problem instances. Moreover, they further decreased the averages of the maximum demands, the total inconvenience values and the supply costs for all instances. Particularly, the inclusion of batteries reduced the total supply cost values slightly, however, improved the total inconvenience value noticeably.

- *Findings*: The results have informed us that including batteries in households further reduced the total supply cost value and the maximum demand slightly, however, reduced the total inconvenience value noticeably.

5.5.5 Summary of Findings

We have summarised our main findings in experiments from demonstrations of the scalability and the optimality as follows:

1. *Scalability*: The FW-DDSM was highly scalable.

(a) The total run time (including the pricing time and the scheduling time) per iteration increased nearly linearly with the number of households, however, the number of households had minimal impact on the scheduling time per iteration per household.

(b) The number of iterations for convergence was independent of the number of households.

(c) The inclusion of batteries had minimum impacts on the number of iterations.

2. *Optimality*: The probability-based scheduling was very effective at approximating the optimal solutions found by the FW-DDSM iterations.

   (a) The objective values of the actual solutions were indeed very close to or even as same as objective values of the optimal or improved solutions.

   (b) For a larger number (more than 1000) of households, the actual total demand profile of all households computed by the probability-based scheduling method was very close to or even as same as the optimal total demand profile found at the end of the FW-DDSM iterations.

   (c) For a lesser number (less than 500) of households, at least one out of five actual total demand profile was very close to the optimal total demand profile.

   (d) The inclusion of batteries reduced the total supply cost value and the maximum demand slightly, however, reduced the total inconvenience value noticeably.

   (e) Implication: In practice, households can generate several actual schedules and report the corresponding demand profiles to the DRSP (as they do during the iterations) and allow DRSP to choose the best demand profiles for them.

5.6 Investigation of Problem Parameters

This section investigates the impacts of some problem parameters, such as the objective weight, the number of sequential jobs, and the battery efficiency and capacity, on the solutions and the scalability of our FW-DDSM.

5.6.1 Investigation of Inconvenience Value Weight

This experiment evaluated the impacts of increasing the inconvenience value weight on the PAR, the cost and maximum demand reductions, and the scalability of two versions of our FW-DDSM: FW-DDSM-CP and FW-DDSM-OGSA.

- *Problem Instance and Parameter Setting*: We generated nine problem instances of 5000 households. Each problem instance had the same job data, however, the weight of the inconvenience value varied from 1, 5, 10, 50, 1000, 500, 1000, 5000 to 10000. We rescaled the pricing table for these 5000 households. We set the electricity cost weight at $\lambda^c = 1$, the EST of each job to 0 and the LFT to the last time interval (144), in order to minimise the impacts of the scheduling time constraints on the run times and the solutions.

- *Criteria* : We evaluated the impacts using the number of iterations for convergence, the time for solving the household scheduling subproblem per iteration, and the PAR, the cost reduction and the maximum demand reduction of the optimal TDP.

- *Results*: We solved each problem instance using the FW-DDSM-CP and FW-DDSM-OGSA, respectively. Table 5.10 presents the averages of the numbers of iterations for convergence, the scheduling times per iteration, and the maximum demand reductions, the PARs and supply cost reductions of the optimal and improved TDPs of instances for each inconvenience value weight. Figure 5.13 illustrates the average supply cost reductions and maximum demand reductions of the optimal and improved TDPs for the same inconvenience value weights and households.

- *Analysis*: The results show that:

  1. The average number of iterations reduced with the inconvenience value weight.
  2. The average scheduling time went up slightly when the weight went down, however, the range of movement was limited.
  3. The average cost and maximum demand reductions were was the same when the weight was lower than 100, however, reduced gradually when the weight went up from 100.

Table 5.10: Impacts of increasing the inconvenience value weight on the FW-DDSM-CP and FW-DDSM-OGSA solutions for 5000 households

Result	**Weight**	**CP model and data preprocessing**	**OGSA**
Number of iterations	1	28.6	24.0
	5	24.6	24.6
	10	24.6	24.0
	50	21.4	20.4
	100	22.2	18.6
	500	13.6	12.6
	1000	13.0	11.4
	5000	13.2	6.6
	10000	11.4	4.0
Scheduling time per iteration per household (seconds)	1	0.2125	0.0006
	5	0.1843	0.0006
	10	0.1856	0.0006
	50	0.1558	0.0005
	100	0.1523	0.0005
	500	0.1546	0.0004
	1000	0.1564	0.0005
	5000	0.1607	0.0004
	10000	0.1616	0.0004
Maximum demand reduction	1	16%	15%
	5	15%	16%
	10	15%	15%
	50	15%	15%
	100	15%	15%
	500	14%	13%
	1000	13%	12%
	5000	12%	9%
	10000	10%	6%
Optimal or improved PAR	1	1.03	1.04
	5	1.04	1.03
	10	1.04	1.04
	50	1.04	1.04
	100	1.04	1.04
	500	1.05	1.06
	1000	1.04	1.06
	5000	1.08	1.12
	10000	1.10	1.16
Cost reduction	1	10%	10%
	5	10%	10%
	10	10%	10%
	50	10%	10%
	100	10%	10%
	500	10%	10%
	1000	10%	9%
	5000	8%	6%
	10000	7%	4%

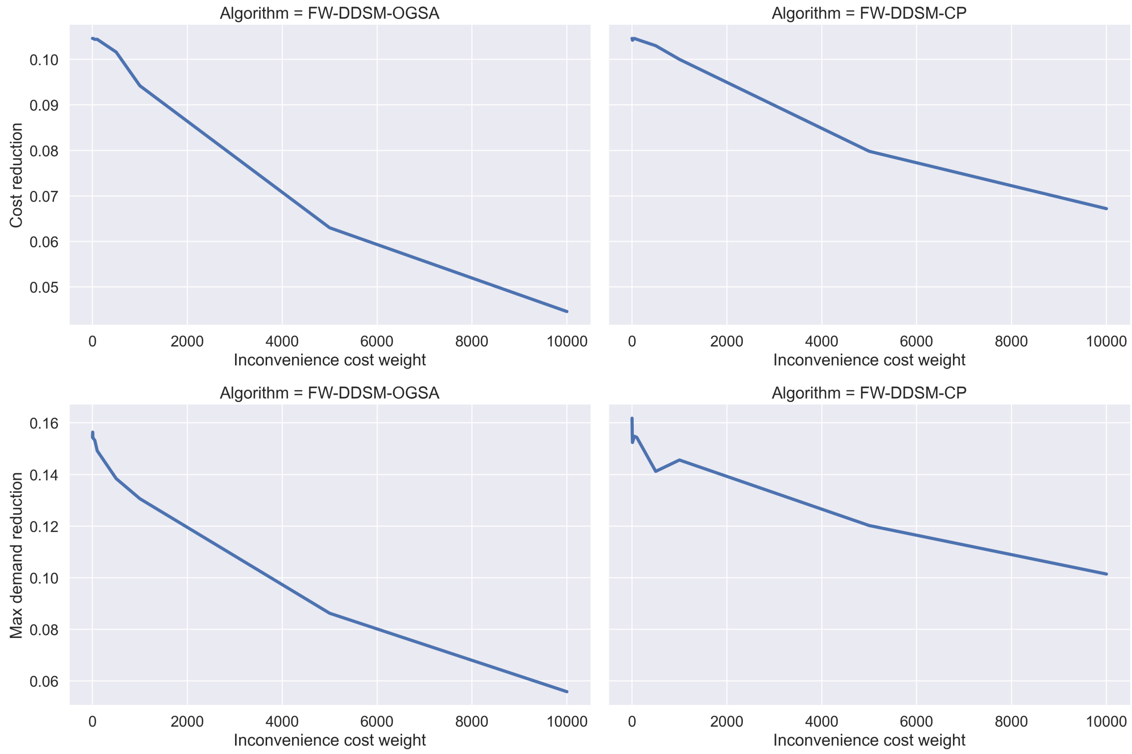


Figure 5.13: Impacts of increasing the inconvenience value weight on the FW-DDSM-CP and FW-DDSM-OGSA solutions for 5000 households

4. The average optimal PAR increased gradually with the weight, however, it was always below 1.1. That means the optimal or improved TDPs were quite flat regardless of the weights.

- *Finding*: We have identified the following findings from the results:

1. *Limited impacts on the cost and maximum demand reductions*: Increasing the inconvenience value weight decreased the cost reduction and the maximum demand reduction slightly, however, the optimal or the improved TDPs were relatively flat regardless.

2. *Limited impacts on scalability*: Increasing the inconvenience value weight increased the scheduling time per iteration slightly, however, deceased the number of iterations before convergence.

5.6.2 Investigation of Sequential Jobs

This experiment evaluated the impacts of increasing the number of sequential jobs per household on the solutions of FW-DDSM-CP and FW-DDSM-OGSA. In particular, we investigated the impacts of the *precedence constraints* and/or the *preceding delay constraints* on the results of FW-DDSM-OGSA as this is a heuristic algorithm that allows constraint violation to achieve a very short computation time.

- *Problem Instance and Parameter Setting*: We generated two sets of problem instances. Each instance had ten households and ten jobs per household. In one set of the instances, the jobs were fully flexible, which means each job had no EST and LFT: it could start from or finish at any scheduling interval. In the other set of instances, the jobs were semi-flexible, which means each job had an EST and a LFT that were randomly selected using a uniform distribution. Within each set, the number of sequential jobs varied from 0, 2, 4, 6 to 8. The pricing table was rescaled for each problem instance. We set the electricity cost weight at $\lambda^c = 1$ and the inconvenience value weight $\lambda^u = 1$.

- *Criteria*: We defined the following criteria to evaluate the differences in the solutions of FW-DDSM-CP and FW-DDSM-OGSA:

    1. PAR difference: the distance between the PAR of the optimal TDP and that of the improved TDP, calculated as the following:

    $$\textsf{PAR difference} = \textsf{PAR of the optimal TDP} - \textsf{PAR of the improved TDP} \quad (5.7)$$

    2. DRd difference: the difference between the maximum demand reduction of the optimal TDP and that of the improved TDP, calculated as the following:

    $$\textsf{DRd difference} = \textsf{DRd of the optimal TDP} - \textsf{DRd of the improved TDP} \quad (5.8)$$

    3. CR difference: the difference between the cost reduction of the optimal TDP (calculated by FW-DDSM-CP) and that of the improved TDP (calculated by FW-DDSM-OGSA), calculated as the following:

$$\text{CR difference} = \text{CR of the optimal TDP} - \text{CR of the improved TDP} \quad (5.9)$$

- *Results*: We solve each problem instance using FW-DDSM-CP and FW-DDSM-OGSA, respectively. Table 5.11 shows the differences in PARs, maximum demand reductions and supply cost reductions using FW-DDSM-CP and FW-DDSM-OGSA.

Table 5.11: Impacts of sequential jobs per household

Sequential jobs	Fully flexible jobs			Semi flexible jobs		
	Min	**Max**	**Median**	**Min**	**Max**	**Median**
PAR difference = PAR of the optimal TDP - PAR of the improved TDP						
0	0	0	0	0	0	0
2	-0.02	0	-0.01	-0.01	0.01	0
4	-0.06	0.01	-0.01	-0.01	0.01	0
6	-0.02	0.02	0	-0.01	0.02	0
8	-0.03	0.02	0	-0.01	0.01	0.01
DRd difference = DRd of the optimal TDP - DRd of the improved TDP						
0	0%	0%	0%	0%	0%	0%
2	0%	2.0%	0%	-1.0%	0%	-1.0%
4	-1.0%	4.0%	0%	0%	1.0%	0%
6	-2.0%	1.0%	-1.0%	-1.0%	0%	0%
8	-2.0%	3.0%	0%	-1.0%	0%	0%
CR difference = CR of the optimal TDP - CR of the improved TDP						
0	0%	0%	0%	0%	0%	0%
2	0%	0%	0%	0%	0%	0%
4	0%	1.0%	0%	0%	0%	0%
6	0%	0%	0%	0%	0%	0%
8	0%	0%	0%	0%	0%	0%

- *Analysis*: The results have demonstrated that:

1. The optimal PAR (the PAR of the optimal TDP) and the improved PAR (the PAR of the improved TDP) were the same when there was no sequential jobs. However, the PAR differences were negative in some instances (some optimal PAR were higher than the improved PARs), which means FW-DDSM-OGSA indeed violated some *precedence constraints* and/or the *preceding delay constraints* and reduced PAR further than FW-DDSM-CP in these instances.

2. The optimal demand reduction and the improved demand reduction were the same when there was no sequential jobs. However, the maximum demand reduction differences were negative in some instances, which again means FW-DDSM-OGSA violated some *precedence constraints* and/or the *preceding delay constraints* and reduced PAR further than FW-DDSM-CP in these instances.

3. The optimal cost reduction and the improved cost reduction were mostly the same for any number of sequential jobs, except in one instances where the cost reduction difference was position. This means that although FW-DDSM-OGSA violated some constraints in some instances and therefore reduced the maximum demand a little further, the extra maximum demand reduction was not sufficient to yield more cost reduction.

- *Findings*: We have validated our hypothesis that FW-DDSM-OGSA is suitable when few coupling constraints (see Definition 2.12 in Chapter 2) are used in households.

5.6.3 Investigation of Battery Capacity

This experiment investigated the impacts of the battery capacity per household on the solution of FW-DDSM. The FW-DDSM-CP was used for this experiment.

- *Problem Instance and Parameter Setting*: Five sets of experiment data were generated. Each set had 2000 households with the same jobs. The EST of each job was randomly selected between the first scheduling interval 0 and its PST, and the LFT was randomly selected between the preferred finish time (PFT) and the last scheduling interval 143. Each household had six jobs and three of them were sequential.

  The maximum battery capacity for each household varied from 2, 4, 6, 8 to 10 kWh. The minimum capacity was 0 kWh. We have assumed that the batteries were all one-hour systems, which means the maximum power rate was always the same as the maximum capacity. The efficiency was assumed to be 1.

  20 problem instances were created in total. The pricing table was rescaled for each problem instance. The electricity cost weight in the objective function was set to 1: $\lambda^c = 1$. The inconvenience value weight was increased to 100: $\lambda^u = 100$, in order to better demonstrate the impact of the battery capacity on the inconvenience value.

- *Criteria*: We evaluated the impacts by comparing the inconvenience value and the objective value of the optimal TDP for each problem instance.

- *Results*: Figure 5.14 shows the averages of the optimal inconvenience and objective values for each data set (or each battery capacity per household).

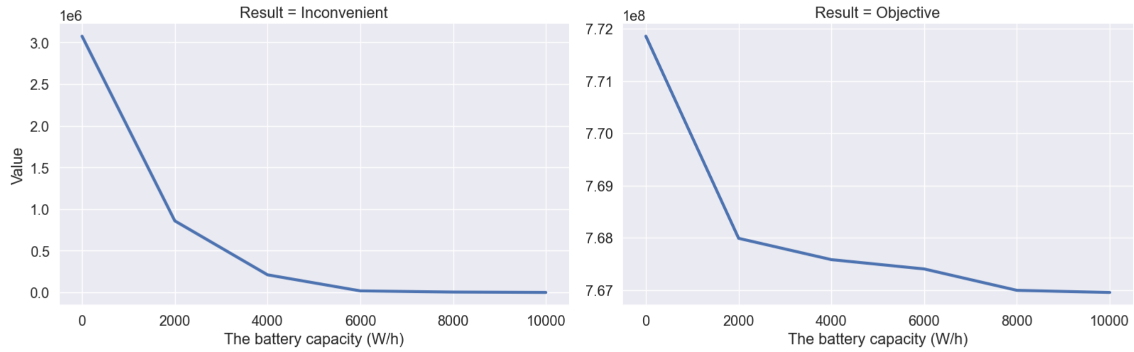


Figure 5.14: Averages of the optimal inconvenience and objective values of the ext-FW-DDSM solutions for increasing the battery capacity.

- *Analysis*: The results show that the optimal inconvenience value and objective values decreased significantly when the battery capacity increased from 0 kWh to 2 kWh, and then decreased slowly when the battery capacity increased from 4kW. Moreover, the inconvenience value was reduced to 0 after the battery capacity reached 6 kWh.

- *Findings*: We have found that increasing the battery capacity reduced the objective and inconvenience values of the optimal TDP. The inconvenience value reached zero when the battery capacity was sufficiently large, and the objective value improved no further after a certain capacity level, such as 8000 kWh in our experiments.

5.6.4 Investigation of Battery Efficiency

This experiment investigated the impacts of reducing the battery efficiency on solutions of FW-DDSM. The FW-DDSM-CP was used for this experiment.

- *Problem Instance and Parameter Setting*: Five sets of experiment data were generated. Each set had five problem instances. Within each set, each problem instance had 2000 households with the same jobs. The EST of each job was randomly selected between the first scheduling interval 0 and its PST, and the LFT was randomly selected between the PFT and the last scheduling interval 143. Each household had six jobs and three of them were sequential.

The battery efficiency varied from 1, 0.9, 0.8, 0.7 to 0.6. The maximum battery capacity for each household was 2 kWh. The minimum capacity was 0 kWh. The maximum power rate was always the same as the maximum capacity.

25 problem instances were created in total. The pricing table was rescaled in each problem instance. The electricity cost weight in the objective function was set to 1: $\lambda^c = 1$. The inconvenience value weight was set to 10: $\lambda^u = 10$.

- *Criteria*: We evaluated the impacts using the inconvenience value and objective values of the optimal solution for each problem instance.

- *Results*: Figure 5.15 shows the averages of the optimal inconvenience and objective values for each data set (or each battery capacity per household).

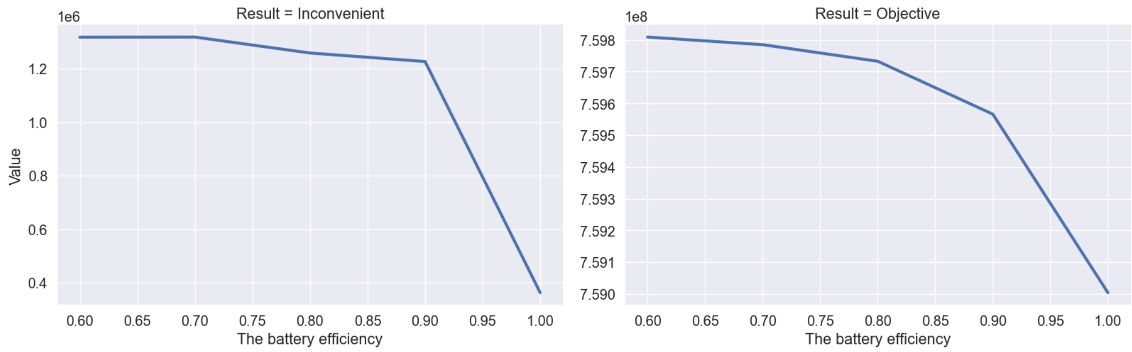


Figure 5.15: Average inconvenience and objective values of the ext-FW-DDSM solutions for various battery efficiencies.

- *Analysis*: The results show that the optimal inconvenience and objective values decreased gradually when the efficiency increased from 0.6 to 0.9, and decreased significantly when the efficiency reached 1.

- *Findings*: We have found that decreasing the battery efficiency increased the inconvenience and objective values, which means finding a practical value for the battery efficiency is important for correctly evaluating the financial benefits of batteries.

5.6.5 Summary of Findings

We have summarised our main findings in experiments from investigations of problem parameters as follows:

1. The inconvenience value weight had limited impacts on the scalability, and the cost and maximum demand reductions.

2. The FW-DDSM-OGSA violated constraints when the number of sequential jobs per household increased.

3. Increasing the battery capacity decreased the inconvenience and objective values. Although, the inconvenience value reached zero and the objective value improved no further when the battery capacity became sufficiently large.

4. The battery efficiency had an important impact on the inconvenience and objective values. This means, using a practical value for the battery efficiency is important for evaluating the benefits of using a battery.

5.7 Summary

This chapter first compares the three types of methods proposed for solving the household subproblem; second demonstrates the optimality and scalability of two versions of our Frank-Wolfe-based distributed demand scheduling method (FW-DDSM): the version with the modified constraint programming (CP) model and the data preprocessing algorithm as the job scheduling method (FW-DDSM-CP), and the version with the Optimistic Greedy Search Algorithm (OGSA) as the job scheduling method (FW-DDSM-OGSA); and third investigates the impacts of some problem parameters, such as the objective weight, the number of sequential jobs, and the battery efficiency and capacity, on the solutions and the scalability of our FW-DDSM.

Firstly, when solving the household subproblem, we have found that the CP models are the fastest especially when used with the data-preprocessing algorithm to compute the optimal solution for a single household. OGSA provided a near-optimal solution, however, it was significantly faster than the CP models and it is easy to implement without the need of a solver. In practice, we propose that we can first use OGSA to find a good solution and then use the CP model with the data preprocessing algorithm to find the optimal solution if needed.

Secondly, when solving DSP-MBs, we have found that the FW-DDSM is highly scalable, as the number of iterations for convergence are independent of the number of households, and the scheduling time per household per iteration is minimally affected by the number of households. Particularly, the objective values of the actual solutions produced by our probability-based scheduling method highly approximate, if not match, the objective values of the optimal solutions calculated by the FW-DDSM. The feasibility of the FW-DDSM may be violated when OGSA is used for scheduling jobs in households, however, this version of FW-DDSM is still suitable for scenarios when households have limited coupling constraints and a low computation cost is the high priority.

Third, the inclusion of batteries in households has limited impacts on the scalability of the FW-DDSM and the use of batteries yields lower total inconvenience values and more supply cost reductions. Increasing the battery capacity reduces the optimal inconvenience and objective values, however these optimal values reduce no further after the battery capacity reaches a certain level. Moreover, decreasing the battery efficiency increases the optimal inconvenience and objective values, which means incorporating a practical battery efficiency in the problem model is important for correctly evaluating the financial benefits of a battery.

Chapter 6

Conclusion

The objective of this thesis is to develop a scheduling algorithm that solves demand scheduling problems (DSPs) especially for a large number of households with batteries under real-time pricing (RTP) in a scalable and efficient manner. This algorithm should ensure that the complex constraints such as the dependencies between appliances are enforced, various types of consumer requirements and preferences are met, peak demand is reduced, and total consumption cost and inconvenience to consumers is minimised. Moreover, the computation time should be minimally affected by the number of households, and minimum manual tuning of parameters and information broadcasting should be required to achieve the best solutions. In order to achieve this goal, this thesis was divided into four parts: literature review, problem model, method design and experimental results. The details of these four parts have been provided in Chapter 2, 3, 4 and 5, respectively.

We have proposed a novel demand scheduling algorithm called the *Frank-Wolfe-based distributed demand scheduling method (FW-DDSM)* to solve *demand scheduling problems for multiple households with batteries (DSP-MBs)*. This is an algorithm that satisfies the requirements we expected for this thesis. The implementation of this algorithm can be found at `https://github.com/dorahee/FW-DDSM`.

6.1 Significance

This thesis has contributed to the state-of-art of demand response (DR) by introducing a novel demand scheduling algorithm that achieves efficiency, flexibility, optimality, feasibility and scalability for DSPs with linear constraints and convex objective functions. This algorithm uses primal decomposition, Frank-Wolfe (FW) method, a data preprocessing algorithm and optimisation models including a constraint programming (CP) model and a linear programming (LP) model.

To the best of our knowledge, this is the first work that applies FW and CP methods to solve DSPs and studies the scalability and efficiency of the scheduling algorithms. The FW algorithm is the key to achieve high efficiency and scalability without the need for manual parameter tuning. CP methods are essential for reducing the computation time for each household while achieving optimality and flexibility.

This algorithm is highly flexible. Unlike existing works that often require consumers to schedule appliances or devices to the next (half an) hour periods. This work allows consumers to choose the granularity of a scheduling time horizon, such as ten minutes or even shorter, offering more flexibility in scheduling without increasing the computation time too much. Consumers can also choose multiple optimisation modules and constraints for their devices based on personal energy needs, such as adding an appliance scheduling module and a battery scheduling module with coupling constraints, without altering the complexity of the algorithm and compromising the optimality of solutions. Moreover, this algorithm does not require manual tuning of parameters to achieve the best solutions, making this algorithm adaptive to any problem instance.

This algorithm is highly scalable. The experimental results have shown that the number of iterations required for our algorithm to find the optimal solution is independent of the number of households. Moreover, unlike existing works that require consumers to schedule devices and broadcast information one by one to achieve the best solutions, this work allows consumers to schedule devices in parallel, further decreasing the total computation time for our algorithm. The results have also shown that with parallel computing, our algorithm can find the optimal solution for 10,000 households with battery energy storage systems (batteries) under 10 seconds. This is a result that has set a new standard in DR scheduling algorithms.

Technically, the main output of this thesis: FW-DDSM has the following benefits:

- The Frank-Wolfe (FW) algorithm has linearised the convex demand scheduling problem and removed the need for manual tuning of the step size to ensure convergence, reducing the problem complexity and making the FW-DDSM adaptive to various problem instances.

- The CP optimisation model with a preprocessing algorithm is demonstrated to be more efficient than a mixed-integer programming (MIP) model for scheduling appliances for each household.

- A battery scheduling module can be easily integrated into each household without compromising the scalability of the entire algorithm.

- The number of input parameters for the master problem and the subproblems are the same for any number of households and any optimisation methods used by each household, ensuring the scalability of this algorithm.

Practically, this FW-DDSM has many benefits for electricity suppliers and consumers, the most significant of which are:

- Lowering costs of meeting peak demand by smoothing total demand profiles.

- Improving the reliability of electricity supply and the utilization rate of existing generation and delivery infrastructure by shifting demand to off-peak times.

- Avoiding load synchronization by preventing consumers from shifting demand to the same times with low prices.

- Allowing consumers to play a more important role in deciding the electricity cost by involving them in DR.

- Protecting consumer privacy by not requiring consumers to expose their appliance details and consumption requirements and preferences to any third parties.

Besides the main output of this research FW-DDSM, this thesis has also produced interesting findings that contribute to the knowledge base of demand scheduling and optimisation. The most significant findings are:

- CP methods with data preprocessing are more efficient than MIP methods for solving demand scheduling problems for a single household (DSP-SHs).

- Using all-purpose optimisation solvers such as Gurobi cannot solve large-scale DSPs in an acceptable time frame.

- Our FW-DDSM has successfully solved large-scale mixed-integer non-linear DR problems with linear constraints efficiently, demonstrating the potential for being transferable to other similar large-scale optimisation problems.

6.2 Application of this Research

A further contribution of this thesis is that the FW-DDSM can be implemented as part of the software used by both utility companies and consumers to jointly manage demand, in order to reduce peak demand and costs. At each household, this software can be implemented on smart devices that connect to other devices such as batteries and appliances. Consumers may set their consumption preferences on this smart device once before the day starts. This software may then automatically collect pricing information from the utility company or demand response service provider (DRSP), decide the best times to use household appliances and batteries based on the consumption preferences, and control appliances and batteries according to the best schedule through the smart device.

At the utility company or DRSP, this software can collect and analyse demand data sent from households, calculate new pricing signals based on the total demand and send these pricing signals back to households. The utility company or DRSP can use the money saved from peak demand reductions to reward participants of DR programs.

6.3 Future Research

This thesis has raised some important questions that can form the basis of future research. Firstly, the FW-DDSM developed in this research computes the best schedule in a day ahead. If no consumption requirements and preferences are changed during the day, the peak demand and cost will reduce as computed by this method. However, there are uncertainties in practice. The consumption preferences or pricing information may change during the day. Future work may consider implementing the FW-DDSM on a rolling horizon and incorporating stochastic algorithms to incorporate uncertainties during the day, making the algorithm more adaptive to changes.

Secondly, this thesis focuses on reducing costs for energy producers and consumers, and ignores constraints and costs for distribution networks. When implementing FW-DDSM on a large-scale in practice, it is important to consider restrictions on distribution networks, such as the maximum capacities of power lines and switches. The future work can combine a distribution network model with the FW-DDSM to produce solutions that are adaptive to limitations of underlying networks.

Thirdly, this thesis has adopted general appliance and battery models that are commonly used in the literature. These models are not specific to any devices in practice. When implementing FW-DDSM in practice, it would be useful to produce more accurate models for devices, in order to more accurately evaluate the benefits and costs of this DR algorithm. Moreover, by incorporating more accurate battery models, this FW-DDSM can be used to estimate how many batteries or how much flexibility in demand response is required from consumers to produce the ideal load profile desired by energy providers.

Fourthly, the consumer take-up of the FW-DDSM is a big issue. It is important to consider questions such as how would users react to the probability scheduling method of FW-DDSM? What appliances are consumers comfortable with losing direct control over? What are the cost and returns of adopting FW-DDSM for a household? Answering these questions are vital for realising the full benefits of the FW-DDSM.

6.4 Summary

To summarise, this thesis has addressed the question of how to schedule a large number of households efficiently under a dynamic pricing scheme to reduce the peak demand, and minimise the electricity cost and inconvenience to consumers while meeting their energy needs and minimising information collected from consumers. The FW-DDSM developed in this thesis has expanded the knowledge of scheduling algorithms for DR and solving algorithms for large-scale optimisation problems, as they relate to this thesis. This thesis has also identified future research topics that can be continued in a research collaboration with industry and communities.

Adapting to ever-changing new challenges in power systems is a long and winding road but advances such as those presented in this thesis have provided more possibilities for success. These new challenges, along with the rapid developments of technologies, have made it an exciting time to work in demand management for power systems.

Appendix A

Power Systems: Now and Then

A.1 Introduction

This chapter provides the background knowledge of power systems, demand response (DR) and dynamic pricing schemes relevant to this thesis. Australian data are used for illustrating the information due to the ease of access, however, the background knowledge described in this chapter is not limited to the Australian electricity systems.

Historically, electricity is produced when it is needed and it must be consumed when it is produced. Power stations and electricity networks had to be built to have the capacity to satisfy electrical demand at any time and any where in the network. It is vital to have a reliable and stable electricity supply at all times for our daily lives, however, there is a trade-off between the reliability and the cost.

Meeting the demand at any time requires extra power stations specially built for peak demand periods. However, these power stations are expensive to operate. Moreover, the peak demand is significantly higher than the average demand and it only occurs for a very short time of a year. It follows that the majority of capacity is idle for a great deal of the time (Van Den Briel et al., 2013; Mishra et al., 2013). This means, the current way to satisfy the peak demand is costly and cause low utilisation of system infrastructure.

Furthermore, current power systems worldwide were built decades ago, meaning that these power stations and associated infrastructure are nearing their optimum working life and required retirement. However, the increasing uses of battery energy storage systems (batteries), electric vehicles (EVs) and electronic devices are introducing more variability in our electrical demands, putting more pressure on ageing components.

We could either keep building new power stations and associated infrastructure to satisfy our growing demands as the traditional approach, which will be costly, or seek alternative solutions to provide reliable electricity supply in a more economic and environment friendly way (Finkel et al., 2017b).

Smart grids are the next-generation power systems that integrate modern information and communication technologies (ICTs) to enable the development of more economic and environment friendly solutions to the above challenges. One of the important goals of smart grid is to realise DR that encourages consumers to better manage their electricity consumption through financial incentives. A main goal of DR is to reduce the peak demand, and make our demands more predictable and economic to satisfy. When consumers can shift demands from peak times to other times of the day and smooth their demand profiles in response to financial incentives, DR can effectively reduce the electricity demand and costs in a more affordable way (Finkel et al., 2017a).

The successful implementation of DR requires the aid of smart grid technologies such as sensing, monitoring, communication and computing technologies. This thesis is interested in investing algorithms that are by computing technologies to facilitate the successful implementation of DR. Before we present our studies on DR algorithms, this chapter introduces the relevant background information on power systems, smart grid technologies and financial incentives for enabling and implementing DR in practice. Specifically, this chapter provides the details on power systems in Section A.2 and DR in Section A.3.

A.2 Power System

Electricity needs to be produced to satisfy our demand at all times of the day at an affordable cost. Achieving these goals requires a network of infrastructure and facilities for generating and delivering the energy, and a mechanism for dispatching power generators at the lowest possible cost. This section introduces the major components in the current power systems, the wholesale market for dispatching power stations, and the challenges faced by current power systems. The information presented in this section is essential for understanding the motivation and benefits of implementing DR, presented in Section A.3.

A.2.1 Component

Traditionally, power systems have three primary physical components: power stations for generating energy, electricity networks for delivering energy, and loads for consuming energy at homes, offices and factories, illustrated in Figure A.1.

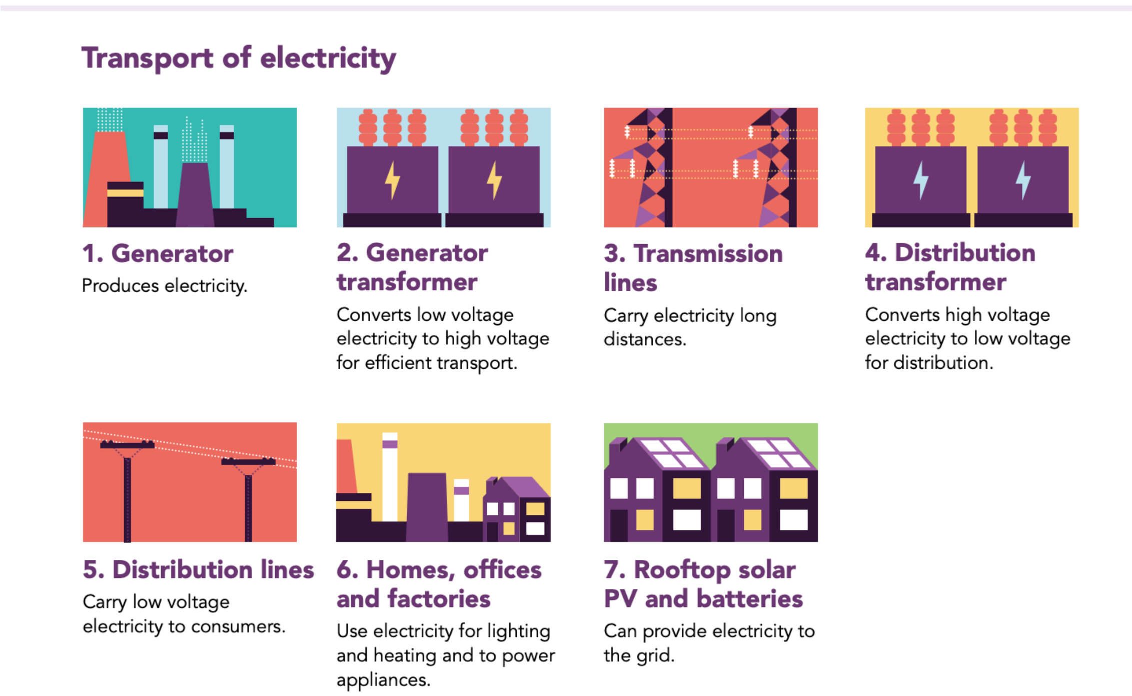


Figure A.1: Transport of Electricity (AEMO, 2020)

Power Stations

Power stations are factories that generate electricity by converting energy in primary sources, such as fossil fuels (coal, natural gas, and oil), nuclear power or renewable energy (e.g. wind and solar), into electrical energy (Biggar and Hesamzadeh, 2014b). Historically, the electricity is supplied in a centralised manner. A few large power stations are built to supply the demand for millions of consumers. Recent years have seen the increasing use of photovoltaics (PVs) panels, batteries and EVs, transitioning electricity generation towards a more distributed delivery model. It is expected that electricity will be produced from large power stations and small generators at the demand side in the near future.

Electricity Networks

Electricity networks are made up of networks of wires and poles that transport and distribute electricity from power stations to customers. These networks are divided into two parts: transmission networks and distribution networks.

Typically power stations are located close to their primary energy sources which are far away from consumption centres (towns or cities) . When electricity is produced from these power stations, it is transmitted through high-voltage transmission networks over long distances to load centres. Then within these load centres, the electricity is distributed to individual consumers through lower-voltage distribution networks. As an example, Figure A.2 shows the image of the national electricity networks in Australia.

Loads

Loads are machines or electronic devices that consume electricity at homes, offices and factories. The usage patterns of these loads follow people's daily activities. As a result, the total demand of loads varies over time. In Australia these days, the total demand reaches a peak in the early morning and evening, and falls when they go to work or sleep. On a very hot day, the total demand can be extremely high when most people use air-conditioners at the same time,as shown in Figure A.3.

A.2.2 Wholesale Electricity Market

Some countries, such as Australia, have introduced a wholesale market for managing the uses of power stations. Within this market, power stations sell electricity to utility companies, however, power stations must compete to provide their services. The relation between a wholesale market and a power system is shown in Figure A.4.

Economic dispatch and in some countries unit commitments are the key operations of a electricity wholesale market (Callaway and Hiskens, 2011; Biggar and Hesamzadeh, 2014a). Unit commitment establishes power stations operating schedules in advance of the operating time (e.g. before a day starts) and takes into account the ramping capabilities, and start-up and shut-down costs of power stations. Economic dispatch is the process of choosing the output levels for power stations that are already started, with the objective of minimising the total cost of meeting demand.

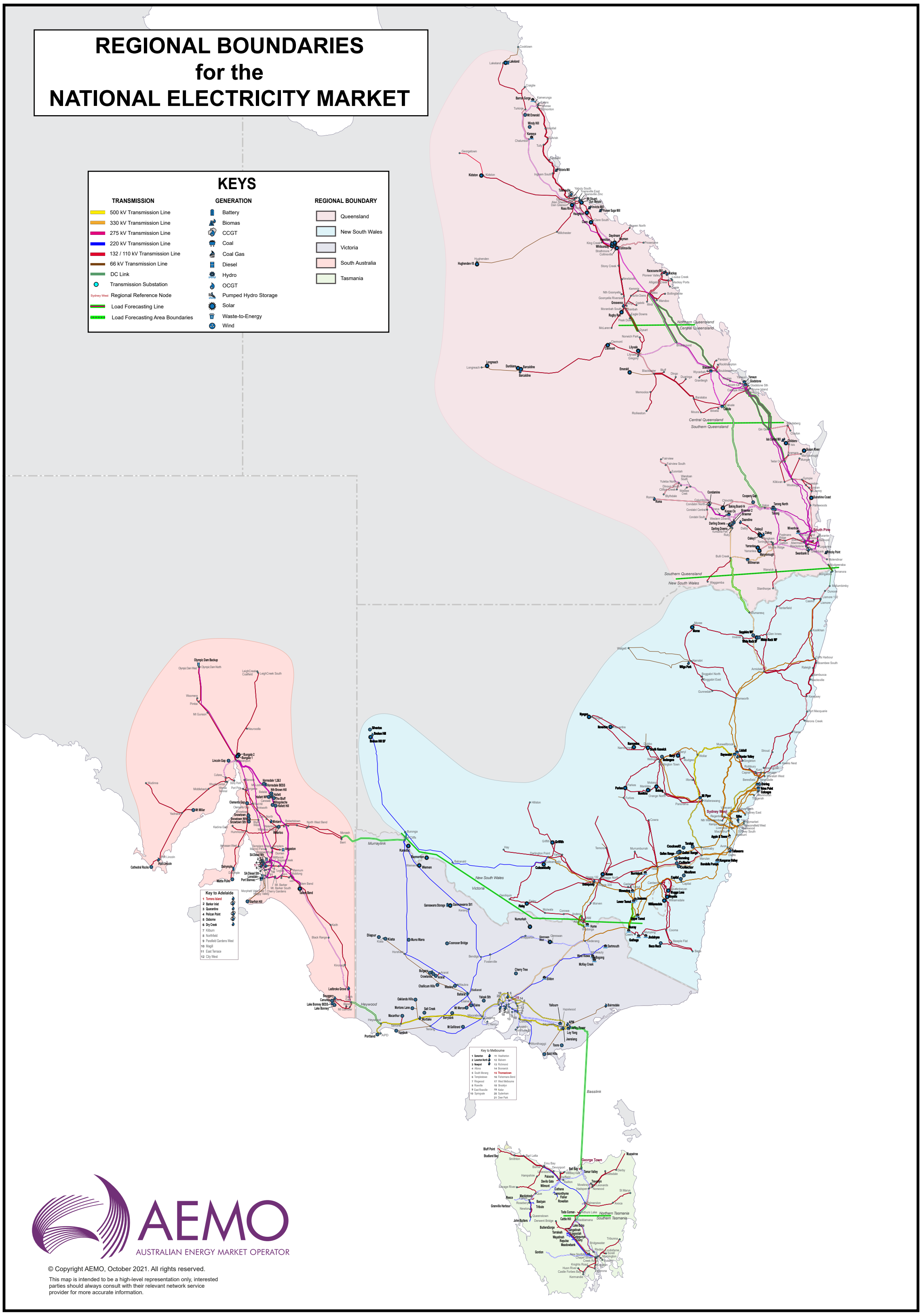


Figure A.2: Electricity Networks of the National Australia Market (AEMO, 2021)

These operations are managed by a third-party organization known as *the market operator*. In Australia, the main steps of these operations include:

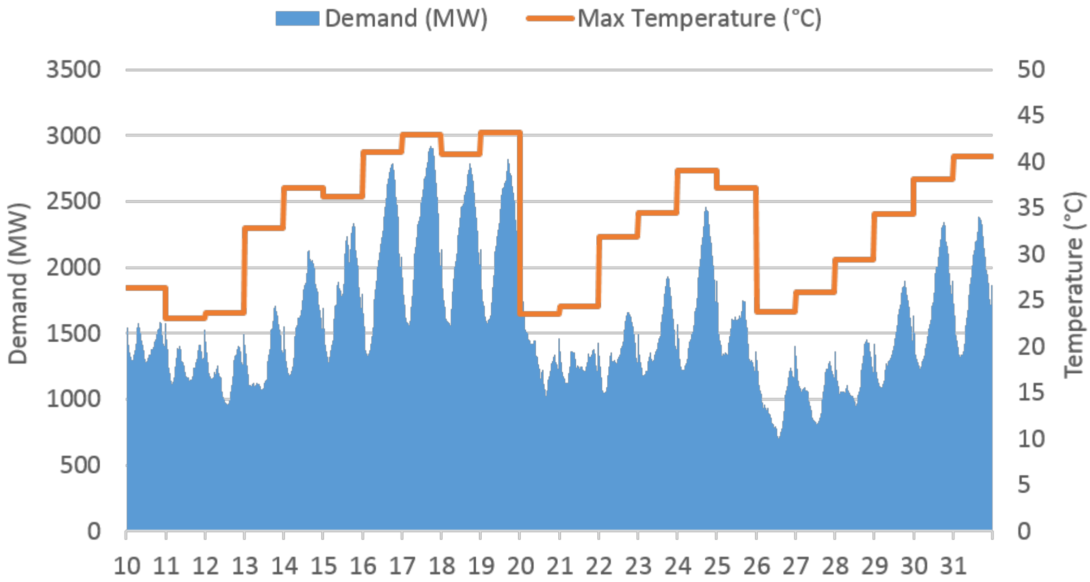


Figure A.3: Total Demand and maximum temperatures for every 30-minute period from 10 to 31 December in 2015 of South Australia, Australia (AEMO, 2017)

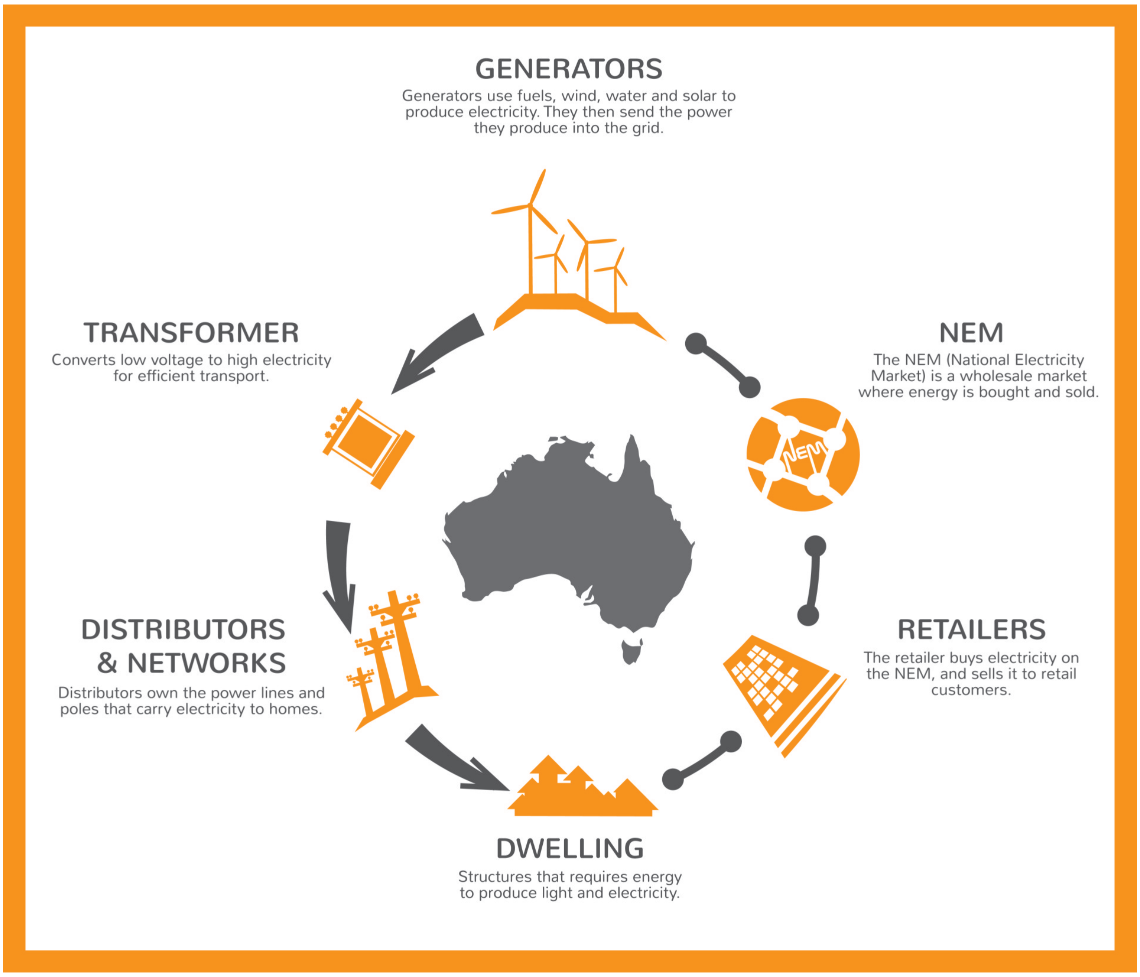


Figure A.4: Electricity wholesale market (GloBird energy, 2021)

- Before a day begins, generators submit an offer of the amount of electricity they can supply at various price levels for every five-minute period of the following day. The market operator schedule those generators in the order of their offer prices from the lowest to the most expensive, which is known as the *merit order*, for every five-minutes period.

- When the day starts, the market operates every five minutes, which means for every five minutes, the market operator dispatch generators according to the pre-defined schedule until the total demand is met in that period, and set a *dispatch price* for that period the price of the most expensive generator dispatched in that period.

- In Australia, the wholesale market traded every 30 minutes before Nov 2021, which means for every 30 minutes, the market operator calculated a *spot price* as the average of all dispatch prices in that period, and paid all dispatched generators with that price.

By dispatching generators in merit order until the demand is met, the market operator can balance the demand and supply at the lowest possible cost. However, on a hot summer day or a cold winter day, the dispatch price fluctuates significantly over time as the demand varies dramatically. Figure A.5 shows the wholesale electricity price of Victoria on 13 April in 2017. All generators are paid at the same spot price during each trading period despite their actual operation costs, which provides a financial incentive for power stations to improve their efficiencies and reduce costs to profit better from the wholesale market.

Bid Stacks

At each dispatch period, the Australian Electricity Market Operator (AEMO) stacks the offers of all generators based on their prices from the lowest to the highest to determine the dispatch order. This stack is called the *bid stack*. While the bid stack always starts with the generators with the cheapest fuels such as wind and coal, and end with those with the most expensive fuels such as diesel or other liquid fuels, the prices offered by the same generators may change over time, depending on the available generators at each time. These changes in prices can be significant when there is a high penetration of variable renewable energy generators (REGs) in the power system.

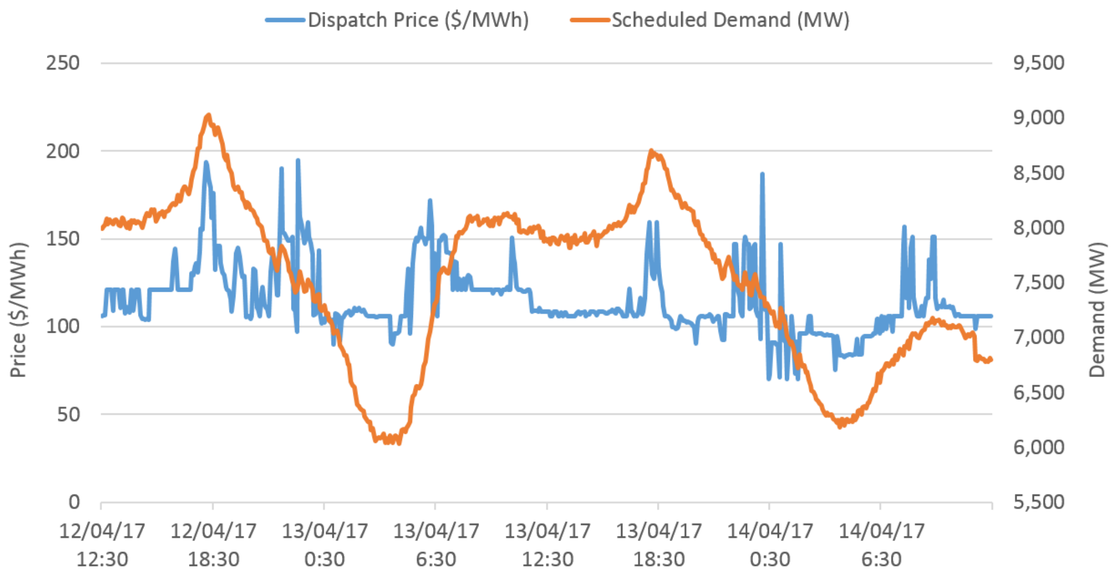


Figure A.5: Wholesale electricity prices of Victoria, Australia on 13 April 2017 (AEMO, 2017)

For example, two bid stacks of South Australia, Australia are provided in Table A.1 and Table A.2. The tables are also illustrated in Figure A.6. These stacks are for two different dispatch periods on two different days. Note that not all generators in the stacks were dispatched but only those were needed. We can observed from the tables that:

- Some generators submit the lowest negative possible bid price allowed by the market regulations for some dispatch periods to ensure they are dispatched at those times. For example, the wind turbines could not be switched off when the wind was blowing so their offered prices were all $-1000/MWh. Moreover, some gas (and often also coal) generators can take hours to start up and cool down and therefore their prices were also $-1000/MWh when the wind resources were abundant.

- The dispatch price increases with the total amount of dispatched generation. However, this price increases slowly when the next dispatched generator uses the same type of fuels, the jumps significantly when switching to another type of fuel. For example, Table A.1 shows that if a liquid fuel generator was required, the price would have jumped from $120.49/MWh to $351.35/MWh.

- The more wind outputs has been dispatched, the less fossil fuel outputs will be required, therefore the lower the dispatch price will be. For example, Table A.1 shows the dispatch price is $79.99/MWh when the total dispatched wind output is

1294 MW, and Table A.2 shows that the dispatch price is $159.99/MWh when the total dispatched wind output is 675 MW.

These observations demonstrate that the design of the bid stacks can indeed help the market operator to ensure the balance between the demand and supply is maintained at the lowest possible cost. Moreover, these observations demonstrate that the cost of electricity generation can indeed be reduced by reducing the peak demand, moving consumption to off-peak times and moving consumption to times when REGs are available.

Table A.1: Bid stacks of South Australia, Australia on 05 Aug 2017 23:30

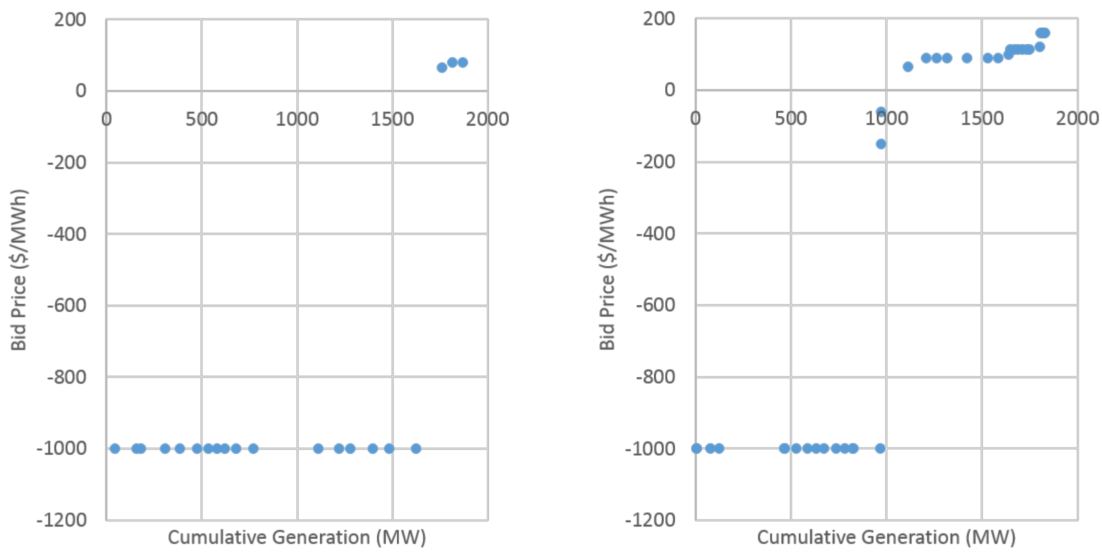
Fuel Type	**Price ($/MWh)**	**Quantity (MW)**	**Dispatched (MW)**	**Cumulative Generation (MW)**
Wind	-1000	159	45	45
Wind	-1000	130	114	159
Wind	-1000	39	21	180
Wind	-1000	144	126	306
Wind	-1000	80	78	384
Wind	-1000	102	89	473
Gas	-1000	60	60	533
Wind	-1000	53	46	579
Wind	-1000	57	41	620
Gas	-1000	60	60	680
Wind	-1000	102	89	769
Gas	-1000	340	340	1109
Wind	-1000	126	110	1219
Wind	-1000	71	62	1281
Wind	-1000	132	116	1397
Wind	-1000	99	87	1484
Gas	-1000	140	140	1624
Gas	65.69	138	138	1762
Gas	79.99	100	55	1817
Gas	79.99	100	54	1871
Gas	120.49	43	0	
Liquid Fuel	351.35	45	0	
Gas	389.99	40	0	
Gas	389.99	40	0	
Gas	578.81	60	0	
Liquid Fuel	592.37	7	0	
Gas	1750.02	15	0	
Liquid Fuel	1750.05	21	0	
Gas	13100.02	46	0	
Gas	13100.02	46	0	
Gas	13100.02	31	0	
Liquid Fuel	13958	44	0	
Gas	13998.99	112	0	
Liquid Fuel	14126	19	0	
Gas	14200	42	0	
Gas	14200	42	0	
Liquid Fuel	14200	42	0	
Gas	14200	124	0	
Gas	14200	24	0	

Table A.2: Bid stacks of South Australia, Australia on 01 Aug 2017 07:30

Fuel Type	Price ($/MWh)	Quantity (MW)	Dispatched (MW)	Cumulative Generation (MW)
Wind	-1000	159	5	5
Wind	-1000	39	1	6
Gas	-1000	74	74	80
Gas	-1000	45	45	125
Gas	-1000	340	340	465
Wind	-1000	57	3	468
Gas	-1000	60	60	528
Gas	-1000	60	60	588
Wind	-1000	43	43	631
Wind	-1000	43	43	674
SA	-1000	60	60	734
SA	-1000	45	45	779
SA	-1000	45	45	824
Wind	-1000	102	4	828
SA	-1000	140	140	968
Wind	-150.01	102	4	972
Wind	-60	130	1	973
Gas	65.69	140	140	1113
Gas	89.99	95	95	1208
Gas	89.99	55	55	1263
Gas	89.99	55	55	1318
Gas	89.99	105	105	1423
Gas	89.99	105	105	1528
Gas	89.99	55	55	1583
Gas	98.93	54	54	1637
Gas	112.99	10	10	1647
Gas	112.99	25	25	1672
Gas	112.99	15	15	1687
Gas	112.99	25	25	1712
Gas	112.99	20	20	1732
Gas	112.99	15	15	1747
Gas	120.49	54	54	1801
Gas	159.99	10	4	1805
Gas	159.99	5	5	1810
Gas	159.99	10	10	1820
Gas	159.99	5	5	1825
Gas	159.99	10	4	1829
Gas	212.99	5	0	
Liquid Fuel	351.35	45	0	
Gas	578.81	60	0	
Gas	588.22	44	0	
Liquid Fuel	592.37	7	0	
Gas	1750.02	15	0	
Liquid Fuel	1750.05	21	0	
Gas	13100.02	46	0	

A.2.3 Challenge and Issue

The current method of balancing the demand and supply has raised two challenges: low utilisation of a power system's capacity, and high generation costs at peak times.



(a) 05 Aug 2017 23:30 (b) 01 Aug 2017 07:30

Figure A.6: Bid stacks of South Australia, Australia

Low Capacity Utilisation

To ensure reliable supply at any time, a power system must plan and build the capacity so as to meet the peak electrical demand at any time (Elgerd, 1982). However, the peak demand is significantly higher than the average demand and it only occurs for a very short time of a year. It follows that the majority of capacity is idle for a great deal of the time (Van Den Briel et al., 2013; Mishra et al., 2013).

As an example, the total demand of Victoria in Australia exceeds 70% of the highest peak demand for only less than 5% of the time in 2011 to 2015. Figure A.7 shows the load duration curves of Victoria state for the 2001-2005, 2006-2010, and 2011-2015 periods, which are created by sorting the total demand of every 30-minute time interval for every five-year period from the largest and the smallest, and normalising the total demand and the time intervals to between 0% to 100%.

High Peak Generation Costs

As discussed in Section A.2.2, the wholesale market operator dispatches power stations based on their offers and the demand every five minutes. Some generators are only dispatched during extreme high demand periods, however, extreme high demand only appears for a small number of hours per year. Therefore, in order to make a profit, these generators must ask for a very high price when they operate. To demonstrate the high cost of these

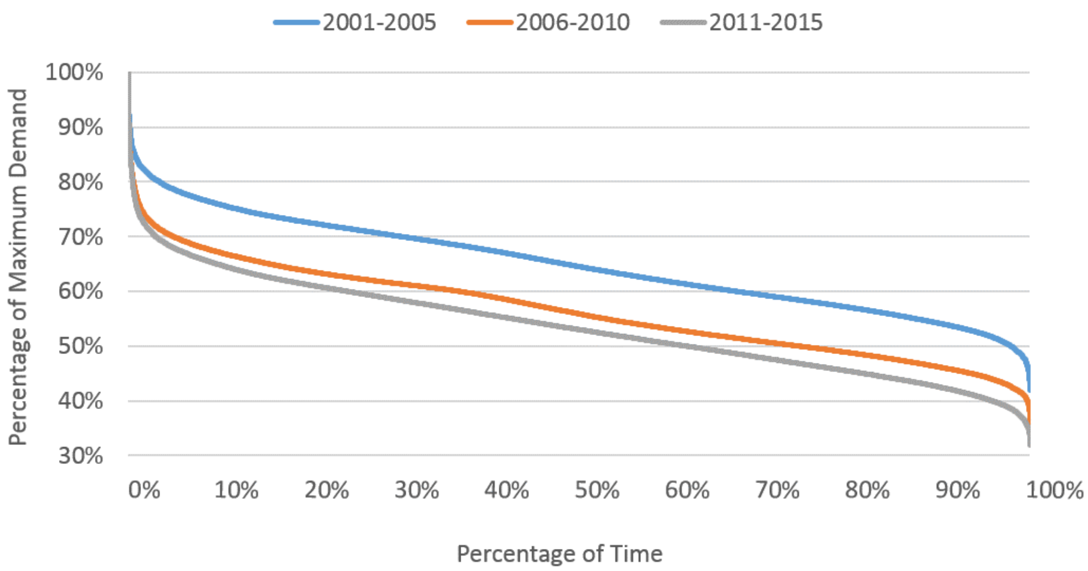


Figure A.7: Load duration curves of Victoria, Australia for 2001 — 2015 (AEMO, 2017)

extreme high demand periods, Figure A.8 shows the actual generation, available generation capacity and the wholesale electricity price in Victoria for every 30-minute period in 2016. Notice that on the hottest day of Victoria in Australia in 2016, the wholesale electricity price in Victoria reached 9137.03 \$/MWh of electricity at 3.30pm, which is 104 times higher than the price of the day before at the same time (87.66 dollars/MWh), and 247 times higher than the one of the day after (36.89 dollars/MWh). That means, the extreme high demand is very expensive to satisfy.

Ageing Components and Increasing Demand Variability

In addition, current power systems in many countries including Australia have been built decades ago. The ageing electrical components have become inefficient and easier to fail, requiring higher repair and restoration costs. Consequently, these components are due for replacements. In Australia, by 2035, 68% of the coal-fired power stations will be over 50 years old, reaching the end of their life (Finkel et al., 2017b) and having to exit the electricity market. However, the increasing uses of batteries, EVs and electronic devices are introducing more variability in our electrical demand. Particularly, the increasing use of heating and cooling devices will increase the peak demand on hot or cold days, putting more pressure on ageing components. These challenges are pushing us to seek alternative solutions to satisfy our demand in a more economic and environment friendly way.

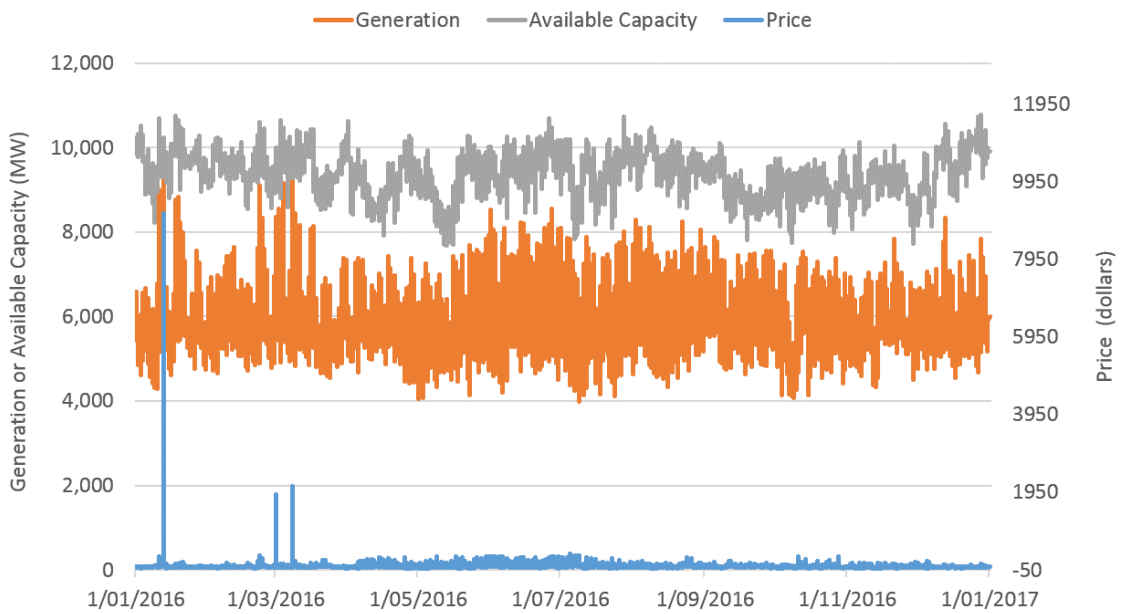


Figure A.8: Generation outputs, available capacities and wholesale prices of Victoria, Australia in 2016

A.2.4 Smart Grid

The challenges in current power systems have introduced pressure for network operators, although they also provide an opportunity for electricity industries to re-think the design of existing power systems, and seek alternative solutions to improve the efficiency and reliability of electricity supply in a more cost-effective manner. These alternatives include transitioning our current power systems into smarter systems that integrate advanced ICTs from generation down to the consumer meters and end-use devices enabling high-speed, two-way communication, sensing, and real-time coordination of key components in power systems (Pratt et al., 2010). Such smarter power systems are known as *smart grids*.

There have been various definitions of smart grids from different institutions (Wakefield and McGranaghan, 2009; Alotaibi et al., 2020). Let us use the definition from the US Department of Energy, which defines a smart grid as “self healing, enables active participation of consumers, operates resiliently against attack and natural disasters, accommodates all generation and storage options, enables introduction of new products, services and markets, optimizes asset utilization and operates efficiently, provides power quality for the digital economy” (U.S. Department of Energy, 2009). The technologies that enable smart grids to achieve these goals and provide better sensing, communicating, monitoring and control include:

- digital sensors for any component of the systems,
- wide-area communications networks, servers and gateways,
- local-area home, commercial building and industrial energy management and control systems,
- consumer information interfaces and decision support tools ,
- utility back-office systems, including billing systems.

Figure A.9 shows these enabling technologies and their interactions in a smart grid. A key feature supported by these enabling technologies is DR, which aims to change the consumption patterns of consumers in ways that reduce electricity costs and improve the reliability and safety of the electricity supply.

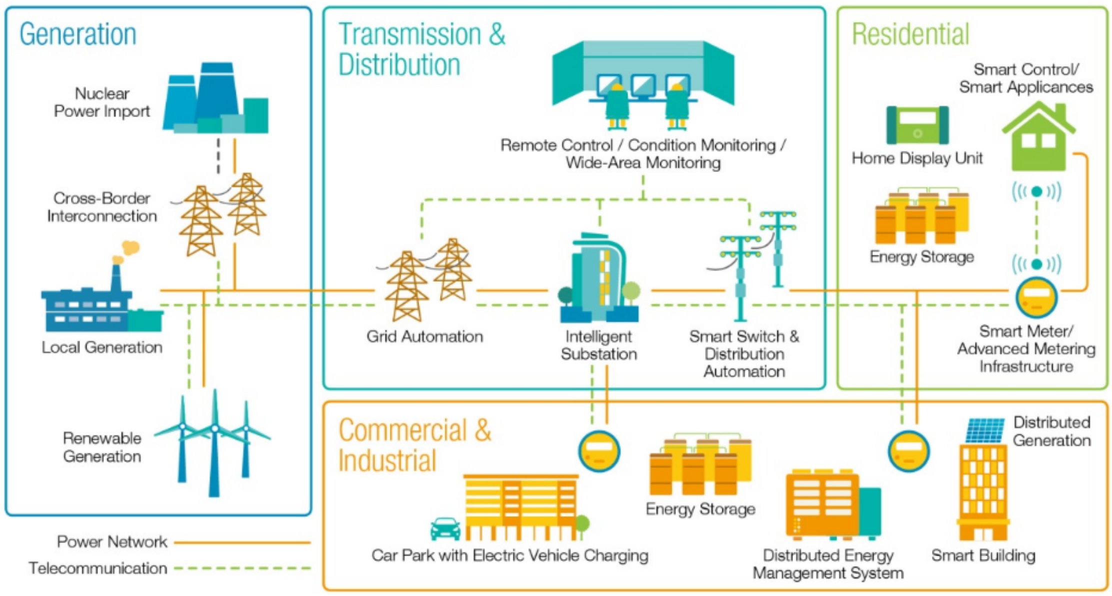


Figure A.9: An overview of the smart grid technology [1]

A.3 Demand Response

DR refers to demand management activities carried out by consumers to actively modify their consumption patterns in response to financial incentives, in order to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized (Albadi and El-Saadany, 2007; Good et al., 2017). The benefits of DR include:

[1]https://www.elprocus.com/overview-smart-grid-technology-operation-application-existing-power-system/

- Reducing consumption under emergency conditions or when the demand nears system capacity to provide reliability at those times.

- Lowering the demand at peak times, which can reduce the need for expensive investment in additional generating capacity, smooth the variability in demand, improve the capacity utilisation, and reduce the overall cost of electricity.

A.3.1 Financial Program

Two types of financial programs that implement DR have been considered in practice to achieve those benefits. These programs are incentive-based or price-based programs.

Incentive-based programs reward consumers for reducing consumption at high demand times or when power systems are under emergency conditions (Albadi and El-Saadany, 2007). These programs are often used on a small number of hours per year (Siano, 2014) to avoid "demand fatigue" where participants decrease their responsiveness or exit from the programs if they are called too frequently. For example, interrupting air-conditioners on the hottest days too frequently can frustrate consumers to the point that they will withdraw from a direct load control program during an emergency condition when their services are needed the most (Callaway and Hiskens, 2011).

Price-based programs offer a dynamic pricing scheme that encourages consumers to reduce the consumption at peak times. Typical pricing schemes include (Albadi and El-Saadany, 2007; Siano, 2014):

- *time-of-use (TOU)* : TOU scheme sets a different rate for different periods of time. For example, the simplest TOU has two rates: a higher rate for the peak period (e.g. 7am – 11pm from Monday to Friday), and a lower rate for the off-peak period (e.g. all the other times). These rates reflect the average costs of electricity at different periods, but they change infrequently in a year.

- *critical peak pricing (CPP)*: CPP is similar to TOU, except the peak rate is raised to several times higher than usual, on days when the demand is expected to be exceptionally high relative to available supply. This critical peak rate is called during contingencies or high wholesale electricity prices for a limited number of days or hours per year (U.S. Department of Energy, 2006).

- *real-time pricing (RTP)*: RTP is a fully dynamic scheme where the rate varies hourly (or more often) to reflect the actual variations in the system's marginal electricity cost in the wholesale market. Participants are informed about the prices on a day-head, hour-ahead or more frequent basis.

These programs allow consumers to reduce their electricity bills by better determining when they use electricity, making the demand more responsive to changes in system conditions and reducing the costs of satisfying the peak demand.

A.3.2 Enabling Technology

Realising these DR programs require technologies that allow utility companies to know the hourly consumption of consumers enrolled in a RTP program for billing purposes, and help consumers to understand their existing consumption patterns to know their capability for demand shifting and reductions. Smart grids provide the enabling technologies required for implementing DR in practice. The technologies can be categorised as follows:

- sensing: metering the electricity at a high frequency
- monitoring and control: providing access to data generated by sensing and computing devices, and controlling devices remotely to switch them on and off or adjust their consumption
- communication: transmitting data in an agreed and standardised way to ensure data security and privacy
- computing: dealing with large amount of data generated by sensing devices in limited time with constrained computation capacity.

Sensing technologies include digital sensors such as smart meters that measure electricity consumption at a higher frequency. Different from existing electricity meters, smart meters can send meter readings to utility companies and receive data from utility companies in real time or near real time through communication networks, enabling two-way communication channels between consumers and utility companies (Pratt et al., 2010). Some smart meters can connect to appliances and control their operations such as switching them on and off, or changing their operation modes. Figure A.10 shows this functionality of a smart meter.

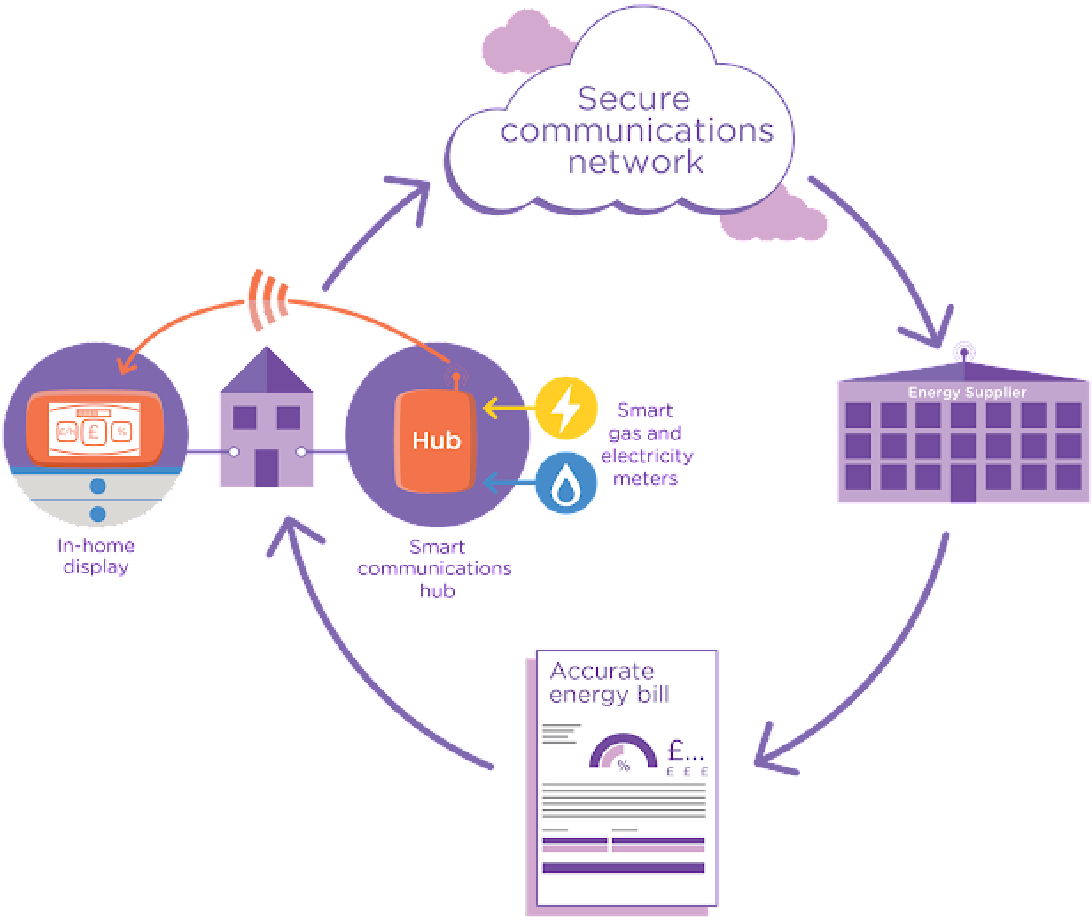


Figure A.10: Flow chart of smart meters' operations [2]

Monitoring technologies include devices that collect data from smart meters and display those data to consumers for decision making. Control technologies may include devices that allow consumers to remotely control their household appliances. For example, a smart plug is a power plug that can be attached to any electrical device, and allows consumers to monitor and control the operation of that device through a wireless network. An energy management system that incorporate these monitoring and control technologies for managing appliances and devices at a household is known as a home energy management system (HEMS). Figure A.11 displays such a system at a household.

Commonly used communication technologies for information exchange between sensing, monitoring and controlling technologies include Home Area Networks (HAN), Field and Neighbourhood Area Networks (FANs/NANs) and Wide Area Networks (WANs) (Siano,

[2]https://www.greencitytimes.com/smart-meters-a-more-efficient-use-of-utilities/
[3]https://www.jcwes.com/blog/2017/6/14/smart-energy-management-system-in-a-domesticated-environment

Figure A.11: A smart home energy management system [3]

2014). A HAN provides the channel for information to transmit between sensors, the smart meter and controllers in a household, so they work together to enable consumers to monitor and control the devices remotely. A NAN connects smart meters and data centres, so consumption readings are delivered to data centres for central storage. A WAN connects the data centres and control centres of a power system, so the system operators have access to all meter reading data. The structure of entire communication networks for demand response is displayed in Figure A.12.

Computing technologies include computing devices, software and algorithms that process and analyse the information and data collected from smart meters, utility companies or another other data sources, such as weather forecasts. These technologies are crucial for achieving the best outcomes of demand response. Consumers can use intelligent computer algorithms such as machine learning algorithms and optimisation algorithms to learn patterns from the data collected by smart meters, calculate the best times to use electrical devices, and control those devices through smart plugs accordingly. Optimisation techniques have been widely used as part of these computer algorithms to schedule devices for

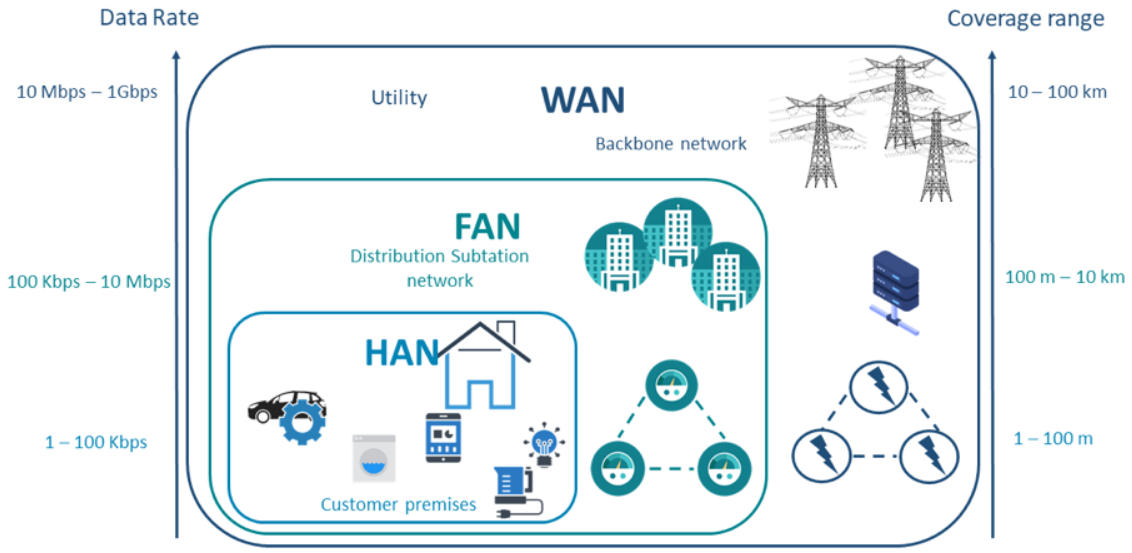


Figure A.12: Communication technologies for smart grids (Ebalance-plus, 2020)

consumers and maximise their benefits (Barbato and Capone, 2014; Deng, Yang, Chow and Chen, 2015; Vardakas et al., 2015). The focus of this thesis is to develop a computer algorithm that utilises optimisation techniques for maximising the benefit of DR for consumers. More details of optimisation techniques used in this thesis will be discussed in Chapter B.

A.4 Summary

The main goal of a power system is to provide reliable electricity supply to consumers at an affordable cost. Reliability can be achieved when the power system can balance the supply and demand at all times. A power system must have sufficient generation capability to meet the extreme demand peaks plus excess capacities. However, those extreme demand peaks occur over a very short time in a year, which results in low utilisation of the system capacity. Furthermore, the ageing power system infrastructure and the growing uses of electronic devices can also introduce problems to the reliability. Additional generation capacities can be used to address this issue, however, this solution will significantly increase the costs of electricity.

These challenges in power systems motivate the electricity industries to re-think the design of power systems, to overcome the challenges in a more economically efficient way.

A *smart grid* is the future electricity grid that uses ICTs to improve the efficiency of electricity generation, transportation, distribution and consumption at a lower cost. A main goal of a smart grid is to provide better visibility and control of the resources at the demand side, to reduce the costs of balancing the supply and demand in real time. This goal can be achieved through DR, which are programs provided by utility companies to motivate consumers to modify their consumption patterns through financial incentives. Various technologies such as smart meters, HEMSs, advanced ICTs are provided by smart grids to help consumers better manage their demand and reduce the costs for both electricity providers and consumers themselves.

Intelligent computer algorithms such as machine learning algorithms and optimisation algorithms can be used by computing technologies for DR, to automate the DR process for consumers and maximise their benefits gained from the DR programs. Optimisation techniques are such algorithms that can find the best way to using electricity for consumers. This thesis focuses on the use of optimisation techniques for achieving the maximum benefits for consumers. The next chapter will provide the background information of optimisation techniques relevant to this thesis.

Appendix B

Optimisation

B.1 Introduction

This chapter presents the optimisation techniques that support the algorithm development in this thesis. Specifically, this chapter presents an overview on four types of optimisation techniques that have been considered in the demand scheduling literature: mixed-integer programming (MIP), constraint programming (CP), continuous optimisation (CO) and problem decomposition. Firstly, we will introduce the methods for formulating an optimisation problem. Secondly, we will demonstrate the methods for solving an optimisation problem using each type of technique. Thirdly, we will present the methods for solving large-scale optimisation problems using decomposition. The methods introduced in this chapter form the foundation for the development of demand scheduling algorithms in the following chapters.

B.2 Optimisation Problem Formulation

Solving a demand scheduling problem (DSP) involves finding the best times to use appliances including battery energy storage systems (batteries) in a way that satisfies consumer requirements and preferences, and minimises costs. A DSP can be formulated as an optimisation problem where the best times to use appliances are *decision variables*, consumer requirements and preferences are *constraints*, and costs are *objectives* of the problem.

In any optimisation problem, *decision variables* represent choices to be made that will directly affect the gains or losses of a consumer. A decision variable is often a numerical value that can be chosen from a range or a set of values. This range or set of values is called the variable domain of that decision. *Constraints* are requirements that decision variables must satisfy to achieve a feasible outcome. The values of decision variables must satisfy all constraints, otherwise the values are infeasible. Constraints are often described as (in)equalities or logic relationships. When all decision variables have chosen values that satisfy all constraints, a feasible solution is found to the optimisation problem. Each solution has a *objective* value that evaluates the "goodness" of that solution. A feasible solution is global optimal when its objective value is better (higher or lower) than the objective value of all other feasible solutions. Let us write:

- decision variables: $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, where $x_n \in \Omega_n$ and Ω_n as the variable domain of x_n
- inequality constraints: $g_p(\mathbf{x}) \leq 0,\ p = 1, ..., P$ where P is the total number of inequality constraints.
- equality constraints: $h_q(\mathbf{x}) = 0$, $q = 1, ..., Q$ where Q is the total number of equality constraints.
- objective function: $f(\mathbf{x})$.

The formal formulation of an optimisation problem can be written as the following:

$$\begin{aligned} \textsf{minimise} \quad & f(\mathbf{x}) \quad (\text{or} \quad F(\mathbf{x})) \\ \textsf{subject to} \quad & g_p(\mathbf{x}) \leq 0,\ p = 1, ..., P \\ & h_q(\mathbf{x}) = 0,\ q = 1, ..., Q \\ & \mathbf{x} \in \mathbf{\Omega} \end{aligned}$$

B.2.1 Multi-objectives

Sometimes, a consumer may want to minimise various types of costs, such as the monetary cost of consuming electricity and the inconvenience or dissatisfaction cost due to moving appliances away from their usual consumption times. These costs can be conflicting, so a balance between these costs needs to be found instead. Finding a solution that optimises multiple conflicting objectives simultaneously is known as multi-objective optimisation. Common ways to find the best balance between conflicting objectives include the *Scalarisation methods* and the *Pareto methods*.

Scalarisation method The Scalarisation methods translate all conflicting objectives into one aggregate objective that contains contributions from all objectives. A common feature required by these method is preferences or weights that indicate the importance of each objective and these weights or preferences need to be decided before the multi-objective optimisation problem (MOOP) is solved. The most commonly used Scalarisation methods is called the *weighted sum (WS)* approach, which assign a weight to each objective and sum up all the weighted objective linearly. As an example, a minimisation goal can be written as the following:

$$\text{miniminse} \quad F(\mathbf{x}) = \sum_{1}^{i} \lambda_i f_i(\mathbf{x}), \tag{B.1}$$

where λ_i is the weight for each objective $f_i(\mathbf{x})$.

Pareto method Unlike Scalarisation methods, Pareto methods keep each objective separately in the optimisation process (De Weck, 2004; Andersson, 2001) . They evaluate each objective value of each solution separately and then find a set of solutions that cannot improve any single criterion without adversely affecting other criteria. This set of solutions is called *Pareto optimal* solutions and they are presented to decision makers to choose the final solution. Unlike Scalarisation methods where the preferences and weights need to be assigned before the optimisation process, Pareto methods assign the preferences after.

Each of these methods have their advantages and drawbacks. Scalarisation methods are easy to understand and to implement therefore commonly used. However, the choices of weights or preferences must be known in advance. Moreover, the weights can have significant impacts on the solutions so their values should be chosen carefully. Although Pareto methods avoid these problems, choosing *Pareto optimal* solutions can be a large computational burden and decision markers can suffer from having to choose from too many solutions (Andersson, 2001). This research has chosen the WS approach under the Scalarisation methods for its simplicity and popularity. More importantly, this approach has been widely used in the existing research of demand management to manage multiple conflicting objectives (Mohsenian-Rad and Leon-Garcia, 2010; Ramchurn et al., 2011; Chen et al., 2011; Li et al., 2011; Yu et al., 2011; Kim and Giannakis, 2013; Rahbari-Asr et al., 2014; Barbato and Capone, 2014; Deng, Yang, Hou, Chow and Chen, 2015; Anvari-Moghaddam et al., 2015; Vardakas et al., 2015; Hossain et al., 2019; Couraud et al., 2020).

B.3 Optimisation Solving Method

The efficient methods for solving optimisation methods depend on the mathematical properties of the problems. This section introduces the mathematical properties commonly considered in an optimisation problem and their solving methods. In addition, we present decomposition methods for solving large-scale optimisation problems.

B.3.1 Mathematical Property

The mathematical properties of an optimisation problem include the properties of decision variables, constraints and the objective function (Conejo et al., 2006). Decision variables can be discrete (integers), continuous (real numbers) or binary (boolean). Constraints can be linear or nonlinear. The objective function can be linear, convex, or nonlinear and nonconvex.

B.3.2 Optimisation Solver and Modelling Language

MIP methods are widely used for solving problems with discrete and continuous variables, linear constraints and linear objective functions. CP methods have been used for solving problems with discrete variables regardless of properties of the objective function. Linear programming (LP) methods are popular for solving problems with continuous variables, linear constraints and linear objective functions.

Some software have implemented these optimisation methods as ready-to-use algorithm packages, known as *solvers*. Users can describe an optimisation problem in a solver-understandable modelling language, feed the model into a solver and retrieve the optimal solution from the solver. Well known optimisation modelling languages include MiniZinc (Stuckey et al., 2018), CVX (Grant and Boyd, 2014), Jualia (Bezanzon et al., 2012) and AMPL (Fourer et al., 2003). Commonly used MIP and LP solvers include Gurobi (Gurobi Optimization, LLC, 2021) and CPLEX (ILOG, Inc, 2006). The state-of-art CP solvers (Lin et al., 2021) include Gecode (Gecode Team, 2017), Chuffed (Chu et al., 2011, n.d.) and OR-Tools (Perron and Furnon, n.d.).

B.3.3 Mixed-integer Programming

Mixed-integer programming (MIP) methods can solve optimisation problems with discrete and continuous decision variables, linear constraints and linear objective functions. We call these problems as MIP problems.

Branch and Bound

A popular MIP method is called Branch and Bound (BB) that finds the best value for each decision variable by splitting the variable domain of each variables into smaller and smaller partitions. This method first relaxes the integrity constraint of the MIP problem. The resulting relaxed problem is a LP problem that can be solved by existing LP methods that have become very mature after decades of research and development. Once the relaxed LP problem is solved, BB will partition the variable domains and again solve a relaxed LP subproblem with only the values in each partition. The optimal solutions to the LP subproblems in all partitions will then be compared, in order to choose the partition that has the best optimal solution. That partition will then be further partitioned. More detailed descriptions of this method are provided as the following:

1. This method relaxes the integer constraints on the variables by treating all integer variables as continuous variables, and solving the resulting relaxed LP problem using LP methods. The primary method for solving a LP problem is *Simplex* introduced by Dantzig (Dantzig, 1990; David G. Luenberger, 2016).

2. If the integer variables in the solution are assigned integer values, then the optimal solution is found. Otherwise, this method will choose an integer variable whose assigned value is continuous in the current solution using some variable selection heuristics and create two subproblems. One subproblem has an additional constraint that the bound of the selected integer variable must be less than or equal to the floor of the assigned value, and another subproblem has a constraint that the bound must be greater than the ceiling of the assigned value instead.

3. If a subproblem has a valid solution, the objective value of this solution will be used as a new bound for the objective value of the original problem. Any other subproblems with a lower or higher objective value will be discarded. This subproblem will then choose another integer variable that has a continuous value in this valid solution

to create two new subproblems. If a subproblem is unsatisfiable, then it will not continue selecting another integer variable and create new subproblems.

4. The process of selecting integer variables and creating subproblems continues until no more subproblems can be created or solved. Then the last found solution is proven to be the optimal solution to the original problem or if no solution can be found, the original problem is declared unsatisfiable.

In practice, the decision variables are often transformed into binary variables by turning each value in the original variable domain into a new decision variable and each new decision variable can only be assigned as zero or one. This transformation is done to improve the efficiency of the partitioning.

B.3.4 Constraint Programming

CP methods are powerful for solving optimisation problems with discrete decision variables (e.g. combinatorial problems). Different from MIP, CP does not consider the linearity of constraints and objective functions. CP provides a general problem-solving paradigm for problems of linear or nonlinear constraints, and linear, convex or nonlinear objective functions. Constraint programming (CP) was first developed for solving constraint satisfaction problems (CSPs), which are concerned with finding a value for each decision variable where constraints specify the values that cannot be used together. The solving methods for CSPs are divided into two broad categories: inference and search. Inference involves propagating constraints to remove values for variables that will never satisfy a constraint. Search involves systematically enumerating all possible values for each variable to find a combination of values that satisfy all constraints. These two categories of techniques are often used together to eliminate useless values sooner and find a feasible solution quicker.

Typically, a backtracking depth-first search algorithm is used with constraint propagation to solve a CSP. The solving process typically involves the following (van Beek, 2006):

1. Constraint propagation is executed to check if any values can be removed from the variable domains. If no values are left after this step, this CSP is declared to be

unsatisfiable. Otherwise, the search algorithm will recursively select variables and assign values to the variables.

2. When a value is assigned to a selected variable, constraint propagation can be executed to remove values of other variables that can never satisfy all constraints when used together with the assigned value to the selected variable.

3. When all variables are assigned, a feasible solution to the CSP is found. If at any point during the search, a variable has no value left to be assigned after propagation, the search algorithm will jump back to the previous selected variable and its assigned value, select a different value for that variable or a different variable to assign value, and continue the search and propagation until a feasible solution is found.

4. If the search algorithm has tried all values for all variables and no feasible solution can be found, this problem is declared to be unsatisfiable.

In some important application areas of CP such as scheduling and planning, in additional to satisfying constraints, an objective function must be optimised (van Beek, 2006). To solve these problems, the common approach is to find a feasible solution by solving a sequence of CSPs. Initially, search and propagation are executed to find a feasible solution to the problem. Then a constraint that requires the objective value of the next solution must be better than the objective value of the current solution is added to the CSP . A new solution is then found for the augmented CSP. This process is repeated until the resulting CSP is unsatisfied. The last solution found by this process is proven to the optimal.

Constraint-based Scheduling

Constraint-based scheduling (CBS) is a discipline that studies how to solve scheduling problems by CP. A scheduling problem is about allocating scarce resources to a given set of activities over time. CBS has grown into the most successful application of CP.

A CBS problem generally includes the following elements (Baptiste et al., 2006):

- activities: an activity has a start time, an end time, a processing time (duration), a release time (the earliest start time), deadline (the latest end time), the latest start time, the earliest end time and an optional due time (preferred start time).

- activity types: an activity can be non-preemptive, which means it can not be interrupted once it has started; or preemptive, which mean it can be interrupted and resumed later; or elastic, which means the resources required by this activity can vary over time but the total consumption of the resources must meet a given amount. Both non-preemptive and preemptive activities consume a fixed amount of resources over time.

- resource types: a resource can be disjunctive, which mean the resource can be used by one activity at one type; or cumulative, which mean the resource can be used by several activities in parallel provided that the resource capacity is not exceeded.

- temporal constraints: a temporal constraint describe the precedence of one activity over another activity.

The decision variable of a CBS problem is usually the start time of each activity. The common objectives includes minimising the total cost of consuming the resources, the number of late activities, the peak or average resource utilisation, the lateness (the difference between the completion time and the due time of each activity) and the penalty for each late job.

Global constrains

A unique and powerful feature in CP is the use of global constraints (van Hoeve and Katriel, 2006). There are two major benefits of using these constraints. One is that using global constraints allows users to describe complex constraints among multiple variables in an easy-to-understand and descriptive way instead of simple inequalities or equalities. For example, a constraint that requires several variables to take different values can be expressed as allldifferent $(x_1, x_2, x_3, ..., x_n)$ in CP instead of $x_1 \neq x_2, \quad x_2 \neq x_3, ..., x_(n-1) \neq x_n$ in using inequalities. Second is that global constraints allow dedicated propagation algorithms to be implemented to eliminate useless values more efficiently, rather than propagating each of inequalities using generic propagation algorithms. In CBS, a commonly used global constraint is cumulative, which is used for ensuring that a feasible schedule for a set of tasks, with durations, durations, resource demands, will not exceed a resource limit at any time.

B.3.5 Continuous Optimisation

Continuous optimisation problems are often solved by an *iterative process* where a sequence of solutions is generated. This process can be described as the following:

1. Choose an initial solution randomly or based on some heuristic.

2. Find a new solution that yields a better objective value than the current solution.

3. Repeat the previous step until a sequence of iterates, such as the gradient of the objective function, reaches (nearly) zero.

Line search methods are commonly used to find a new solution in Step 2. Generally, these methods have the form as the following:

$$\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k * \mathbf{d_k} \tag{B.2}$$

where:

- $\mathbf{x_{k+1}}$: the new solution found after Step 2 in the iteration k,
- $\mathbf{x_k}$: the current solution found before Step 2,
- $\mathbf{d_k}$: a descent direction where $\mathbf{x_k}$ can travel along to reduce the objective value,
- α_k: a *step size* or *step length* which is a scalar that tells how far $\mathbf{x_k}$ will travel along $\mathbf{d_k}$ to become $\mathbf{x_{k+1}}$.

The key steps in this iterative process are the calculation of the descent direction $\mathbf{d_k}$, the step size α_k in the line search methods and the stopping condition in Step 3. These steps determine if the iterative process can converge to the desired solution and the convergence rate when convergence occurs. Each of these steps will be discussed in the following paragraphs:

Step Size Calculation

The simplest way to choose the step size is to pick a fixed value that remains the same at every iteration. However, this naive method can allow the step size to be either too long or too short, leading to *pre-mature convergence* where only a near or close to optimal solution is found or *oscillation* where no solution can be found. Other methods have been developed to choose the step size in a better way.

For example, exact line search methods find the best step size that yields the best amount of decrease in the objective value at each iteration. Generally, these methods require the step size α_k to minimise $f(\mathbf{x_k} + \alpha_k * \mathbf{d_k})$ (assuming the $\mathbf{d_k}$ has already been found) per iteration, turning the calculation of the step size into an additional optimisation problem. These methods are expensive and more commonly used in the early research studies (Robinson, 2020; Hauser, 2007; Vandenberghe and Boyd, 2004).

Inexact line search methods are more commonly used in recent works to address the disadvantages of the exact line search methods. Generally, these methods first guess a useful value for the initial step size and then update the step size at each iteration based on using a chosen scalar and the value of the step size at the previous iteration. A well-known example of such inexact methods is called *backtracking- Armijo* line search method (Robinson, 2020; Hauser, 2007).

Descent Direction Calculation

In the optimisation literature, both the first and the second derivatives of the objective function have been used for creating descent directions to guide the search in the iterative process (Chong and Zak, 2001a,b,c). In the power system literature and the demand response literature, methods that use the first derivatives (gradient) have been widely used and these methods are called *gradient methods*.

Gradient methods work on the basis that 1) the gradient of a given function becomes zero at the optimal solution and 2) given a small displacement at any given point, the given function increase in the direction of the gradient than in any other direction(Chong and Zak, 2001a) . In a minimisation problem, the idea of these methods is to move the current solution along the negative direction of the gradient of the objective function at each iteration to gradually reduce the objective value until the gradient becomes or is very close to zero. The general procedure of these methods can be described as Algorithm 1

Simple Gradient Method The simplest gradient method uses the same fixed value for the step size at each iteration. This method is easy to implement, however, the step size needs to be chosen carefully to avoid being too short and too long.

Algorithm 1 The gradient methods

Set iteration counter $k = 0$ and make an initial guess of the minimum point x_0.
repeat
    Calculate the gradient of the objective function f at x_k: $-\nabla f(x_k)$.
    Calculate the step size α_k using the fixed value methods, the exact line search methods or the inexact line search methods
    Update $x_{k+1} = x_k + \alpha_k * \nabla f(x_k)$.
until Stopping condition is satisfied.

The Method of Steepest Descent A more complex version called the method of steepest descent uses the exact line search methods to calculate a new step size at each iteration instead. This version finds the best step size at each iteration but increases the computational costs significantly.

Neither the simple gradient method nor the method of steepest descent considers constraints when moving the current solution. The new solution may be outside the feasible domain even if the step size is well chosen. This problem can be addressed by other gradient methods such as projected gradient methods and the conditional gradient method.

Projected Gradient Methods The projected gradient methods add a *projection step* after a new solution is found at each iteration. This projection step involves constructing a projection matrix that projects the new solution to a feasible solution that is closest to this new solution and replace this new solution with its closest feasible solution (Nesterov, 2013; Chong and Zak, 2001d).

Conditional Gradient Method/Frank-Wolfe Algorithm The conditional gradient method, more widely known as the Frank-Wolfe algorithm, calculates a feasible descent direction that ensures the new solution is also feasible (Luenberger and Ye, 2016). This feasible direction is calculated at each iteration by first solving the linear approximation of the objective function at the current solution and then computing the difference between the solution of approximation function and the current solution. The procedure can be described as Algorithm 2. Compared to the projected gradient methods, the Frank-Wolfe algorithm has the benefits of ensuring the feasibility of the solution by simply solving a linear approximation function without needing to construct a complex projection matrix, reducing the computational cost significantly. However, the Frank-Wolfe algorithm is limited to solving convex optimisation problems with linear constraints. The projected gradient methods can solve more general problems.

Algorithm 2 The Frank-Wolfe algorithm (the conditional gradient method)

1: Set iteration counter $k = 0$ and make an initial guess of the minimum point x_0.
2: **repeat**
3: Solve the linear approximation of the objective function at point x_k: $F_{x_k}(x) = f(x_k) + \nabla f(x_k)(x - x_k)$. Let us write the solution of this approximation function as y_k.
4: Calculate the descent direction as $d_k := y_k - x_k$.
5: Choose the step size $\alpha_k \in [0, 1]$
6: Update $x_{k+1} = x_k + \alpha_k * d_k$.
7: **until** Stopping condition is satisfied.

Stopping Conditions

Multiple stopping conditions have been used in the literature (Chong and Zak, 2001a). A common stopping condition is when the gradient at x_{k+1} is zero. However, in practice the gradient will rarely be exactly zero. A practical modification is to check if the norm of the gradient is less than a predefined threshold ϵ as the following:

$$\|\nabla f(x_{k+1})\| \leq \epsilon, \tag{B.3}$$

An alternative is to check whether the different between the objective function values of two successive iterations is less than a predefined threshold ϵ as the following:

$$| f(x_{k+1}) - f(x_k) | \leq \epsilon, \tag{B.4}$$

Another alternative is to check if the norm of the difference between two successive solutions is less than a predefined threshold ϵ as the following:

$$| x_{k+1} - x_k | \leq \epsilon. \tag{B.5}$$

B.3.6 Decomposition Method

In practice, many optimisation problems have a complexity that grows more than linearly. Moreover, some of these problems have large numbers of decision variables and constraints. Solving these problems using standard optimisation methods is not possible within an acceptable time frame even on a super computer (Boyd et al., 2015). However, there are times when optimality is important but computational times and resources are limited. Alternatives methods are required to tackle these problems in a more scalable manner.

Fortunately, some of these problems are decomposable. That means, a problem can be decomposed into smaller subproblems that can be solved independently in parallel or sequentially. The solutions to the subproblems can be combined together in such a way as to solve the original problem. The subproblems typically have a lower complexity and therefore can be solved on one or multiple lower-level computers, eliminating the need for a super computer. Moreover, solving subproblems in parallel or even sequentially can save the computational time and resources substantially when the original problem complexity grows more than linearly with the problem size.

Although decomposing a problem can save computational times and resources, decomposing can be difficult when the problem has *complicating constraints* and/or *complicating variables* (Conejo et al., 2006). A complicating variable couples multiple constraints together and a complicating constraint couple multiple variables together, making it difficult to separate the original problem easily.

Complicating constraints These constraints involve multiple decision variables. As an example, let us consider the following, of the form:

$$\begin{aligned} \mathsf{minimise} \quad & f(\mathbf{x}) = f_1(x_1) + f_2(x_2) \\ \mathsf{subject\ to} \quad & x_1 \in \Omega_1, \;\; x_2 \in \Omega_2 \\ & h_1(x_1) + h_2(x_2) \leq 0 \end{aligned} \tag{B.6}$$

The constraint $h_1(x_1)+h2(x_2) \leq 0$ involves both decision variables x_1 and x_2. Without this constraint, the original problem can be separated into subproblems f_1 and f_2. Therefore this constraint is considered a *complicating constraint.*

Complicating Variables These variables are involved in multiple terms of the objective function. Let us consider an example, of the form:

$$\mathsf{minimise} \quad f(\mathbf{x}) = f_1(x_1, y) + f_2(x_2, y) \tag{B.7}$$

The decision variable y presents in the both terms f_1 and f_2 of the objective function. Without this variable, the original problem can be divided into two subproblems f_1 and f_2. Therefore this variable is called a *complicating variable.*

Decomposition methods have been proposed to address the challenges of separating problems with complicating constraints and variables. Typical decomposition methods include the primal decomposition method and the dual decomposition method (Conejo et al., 2006; Boyd et al., 2015).

Primal Decomposition

The primal decomposition method separates the original problem into a master problem and subproblems based on its primal variables (in contrast to dual decomposition that separates based on its dual variables). The primal decomposition method works well when the original problem has few complicating variables and the subproblems are fast to solve (Boyd et al., 2015). The methods for decomposing with complicating variables and with complicating constraints are different but similar. We focus on describing the method with complicating variables as this method is used in Chapter 4.

Decomposing with Complicating Variables First, the primal decomposition method splits the variables into *public variables* and *private variables.* Second, it creates the master problem with the public variables and the subproblems with the private variables. Third, it solves the master problems and subproblems in an iterative manner to find the solution to the original problem.

Let us consider the problem example B.6. y is shared by both f_1 and f_2, therefore it is a *public variable.* In fact, the public variable is the complicating variable. x_1 only exists in f_1, x_2 only exists in f_2, therefore they are *private variables.*

When the public variable y is fixed, f_1 and f_2 become separable, so a *subproblem* can be created for f_1 and f_2, respectively, as the following:

$$x_1 = \underset{x_1 \in \Omega_1}{\arg\min}\, f_1(x_1, y) \tag{B.8}$$

$$x_2 = \underset{x_2 \in \Omega_2}{\arg\min}\, f_2(x_2, y) \tag{B.9}$$

When these subproblems are solved, the private variables x_1 and x_2 become fixed. If we write f_1 with fixed x_1 as $\theta_1(y)$, and f_2 with fixed x_2 as $\theta_2(y)$, the original problem can be written as the following:

$$\textsf{minimise} \quad \theta(y) = \theta_1(y) + \theta_2(y) \tag{B.10}$$

This problem is called the *master problem.* This problem inherits the mathematical properties of the original problem. For example, if the original problem is convex, the master problem is also convex (Boyd et al., 2015).

After the master problem and the subproblems are created, they are solved iteratively to search for the solution to the original problem. In each iteration, the subproblems are solved independently given some values of the public variables and the master problem is solved given the solutions to the subproblems. The subproblems can be solved by standard optimisation methods such as MIP or CP methods. The master problem is often solved by gradient methods. An example of such iterations is described in Algorithm 3. The stopping condition can be same as those in standard gradient methods.

Algorithm 3 Primal Decomposition with Complicating Variables

```
1: Set iteration counter k = 0 and make an initial guess of the public variable y_0.
2: repeat
3:     Given y_k, find the optimal solution x_{i,k} to each subproblem f_i using MIP/CP.
4:         Update each private variable x_i ← x̄_{i,k}
5:     Given x_{i,k}, solve the master problem using a gradient method
6:         Calculate a descent direction d_k of the master problem function θ(y)
7:         Calculate a step size α_k using an inexact line search method
8:         Update the public variable y_{k+1} ← y_k − α_k * d_k
9: until Stopping condition is satisfied.
```

Decomposing with Complicating Constraints The example B.6 contain complicating variables only. When complicating constraints present, the primal decomposition method separates the complicating constraints into each subproblem in addition to the private variables. Then a dual variable is introduced for each separated constraint and this method solves both the private variables and the dual variables in each subproblem. The descent direction in the master problem can be updated using the dual variables from the subproblems instead (Boyd et al., 2015).

Dual Decomposition

The dual decomposition method transforms the original problem into its Lagrangian dual problem and separates problems based on its dual variables. An advantage of using the dual decomposition method is that the dual function is always concave even when the original problem is not convex (Boyd and Vandenberghe, 2004). That means, the dual problem is easier to solve even if the original is nonlinear and very difficult to solve.

Decomposing with Complicating Variables The procedure of decomposing with complicating variables using the dual decomposition method is similar to that using the primal decomposition method, with additional steps of replacing the primal complicating variables with a complicating constraints and introducing dual variables as the new complicating variables. Let us consider the problem example B.6 again.

First, the dual decomposition method creates a local version of the complicating variable y for each potential subproblem: y_1 and y_2 and a consistency constraint $y_1 = y_2$ that requires all local versions have the same value, shown as the following:

$$\begin{aligned} &\textsf{minimise} && f(\mathbf{x}) = f_1(x_1, y_1) + f_2(x_2, y_2) \\ &\textsf{subject to} && y_1 = y_2 \end{aligned} \tag{B.11}$$

Second, this modified original problem is transformed into its Lagrangian dual problem as the following:

$$L(x_1, y_1, x_2, y_2, v) = f_1(x_1, y_1) + f_2(x_2, y_2) + v^T y_1 - v^T y_2. \tag{B.12}$$

where v is the dual variable. Now this dual variable exists in all terms in the objective and therefore it becomes the new public/complicating variable and original variables become the private variables.

Third, similar to the primal decomposition, we decompose this problem based on its public and private variables. The subproblems g_i minimise over the private variables x_i and y_i given a fixed dual variable v as the following:

$$g_1(v) = \inf_{x_1, y_1} (f_1(x_1, y_1) + v^T y_1) \tag{B.13}$$

$$g_2(v) = \inf_{x_2, y_2} (f_1(x_2, y_2) + v^T y_2) \tag{B.14}$$

The master problem maximise over the dual variable v given the fixed private variables as the following:

$$\text{maximise} \quad g(v) = g_1(v) + g_2(v) \tag{B.15}$$

Fourth, again similar to the primal decomposition, the master problem and subproblems can be solved iteratively using standard optimisation methods.

Decomposing with Complicating Constraints Decomposing problems with complicating constraints using the dual decomposition is straightforward. The procedure is the same as decomposing with complicating variables but without the first step.

B.4 Summary

This section has introduced the optimisation techniques that support the algorithm development in this thesis. A DSP is often formulated as an optimisation problem that finds the best times to use appliances in ways to minimise the costs of consumers. An optimisation problem includes three basic elements: decision variables, constraints and objectives. Some optimisation problems include multiple conflicting objectives. The Scalarisation methods and the Pareto methods can be used to find the best balance between these conflicting objectives.

Solving an optimisation problem requires firstly understanding the mathematical properties of a problem and secondly choosing an optimisation technique according to the mathematical property. Mixed-integer programming (MIP) techniques are best for optimisation problems with both discrete decision variables and continuous variables, and convex constraints and objective functions. Linear programming (LP) can be considered as a subset of MIP for solving problems whose decision variables are continuous only, and constraints and objective functions are linear. Constraint programming (CP) is best for optimisation problems with discrete variables, however, constraints and objective functions can be linear, convex or nonlinear and nonconvex.

Some optimisation problems are much larger and difficult to be solved as a whole. Some of these problems have complicating constraints or complicating variables which allow the original problems to be decomposed into smaller and more achievable subproblems. When complicating constraints present in an optimisation problem, primal decomposition method can be used for breaking down large optimisation problems into smaller and more achievable subproblems. When complicating variables present, dual decomposition method can be used instead.

The methods introduced in this chapter for formulating, solving and decomposing an optimisation problem will form the foundation for developing algorithms for solving demand scheduling problems for multiple households with batteries (DSP-MBs) in this thesis. Each of the following chapter will demonstrate how these methods will contribute to the final research outcomes of this thesis.

Appendix C

Additional Data, Code and Figures

Figure C.1: Sample Input Data for the initial MIP/CP Models

```
num_intervals = 144;

prices = array1d(INTERVALS,[141, 141, 141, 141, 141, 141, 141, 141, 141, 141,
  141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141,
   141, 143, 143, 143, 142, 142, 142, 148, 148, 148, 189, 189, 189, 163, 163,
  163, 145, 145, 145, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141,
   141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141, 141,
  141, 141, 141, 141, 141, 141, 142, 142, 142, 141, 141, 141, 146, 146, 146, 158,
   158, 158, 203, 203, 203, 313, 313, 313, 1155, 1155, 1155, 1155, 1155, 1155,
  1446, 1446, 1446, 616, 616, 616, 616, 616, 616, 363, 363, 363, 363, 363, 363,
  313, 313, 313, 427, 427, 427, 221, 221, 221, 158, 158, 158, 148, 148, 148, 158,
   158, 158, 143, 143, 143, 142, 142, 142, 144, 144, 144]);

num_tasks = 10;
preferred_starts = [41, 135, 96, 115, 6, 120, 71, 31, 73, 46];
earliest_starts = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
latest_ends = [143, 143, 143, 143, 143, 143, 143, 143, 143, 143];
durations = [3, 3, 1, 2, 3, 4, 3, 6, 3, 4];
demands = [55, 15, 300, 700, 80, 2400, 3500, 1500, 400, 15]; % in w
care_factors = [1, 0, 9, 2, 1, 4,  3,  2, 1, 7];
inconvenience_weight = 500;
cost_weight = 1;

num_precedences = 2;
predecessors = [1, 7];
successors = [9, 10];
prec_delays = [62, 127];
max_demand = 6275;
```

Algorithm 4 Data Preprocessing Algorithm for the Optimisation Model in Chapter **??**

Input: $\mathbf{t}_{h,i} = \{e_{h,d}, \theta_{h,d}, s^a_{h,d}, s^p_{h,d}, s^e_{h,d}, f^l_{h,d}, \eta_{h,d}, \mathbf{D}^p_{h,d}, \bar{\theta}_{h,d} \mid d \in [1, D_h]\}$, $\mathbf{P} = \{p_t \mid t \in [1, M]\}$, M, λ^u, λ^c

1: $objs \leftarrow []$
2: **for all** $d \in [1, D_h]$ **do**
3:   $objsPerTask \leftarrow []$
4:   **for all** $t \in [1, M]$ **do**
5:    **if** $s^e_{h,d} \leq t \leq f^l_{h,d} - \theta_{h,d} + 1$ **then**
6:     $objsPerTaskPerInterval = |t - s^p_{h,d}| * \eta_{h,d} * \lambda^u + \sum_{t_2=t}^{t+\theta_{h,d}} p_{t_2} * e_{h,d} * \lambda^c$
7:    **else**
8:     $objsPerTaskPerInterval \leftarrow 999999$
9:    append $objsPerTaskPerInterval$ to $objsPerTask$
10:   append $objsPerTask$ to $objs$

Return: $objs$

Algorithm 5 Optimistic Greedy Search Algorithm

for solving a demand scheduling problem for a single household (DSP-SH)

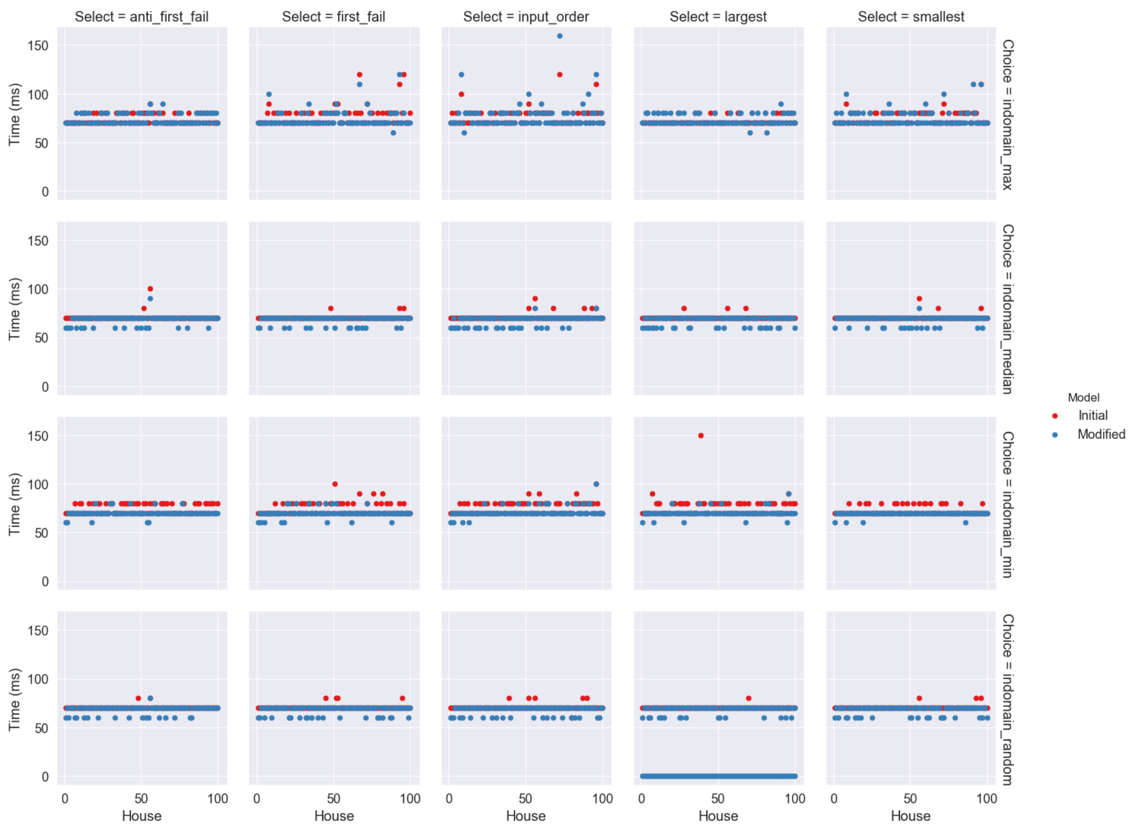
Input: job data and the household demand limit $\bar{E}$.

1: **for all** job $i \in [1, D]$ **do**
2:   find the maximum run cost of this job:
3:   find all time intervals whose run costs are cheaper than the maximum run cost and set those intervals as the feasible start times of this job
4:   **if** job i is a successor of another job **then**
5:    remove feasible start times that are earlier than the actual finish time of the preceding job
6:    remove feasible start times that are later than the actual finish time of the preceding job plus the maximum delay:
7:   **if** job i is a preceding job **then**
8:    **for all** $k \in all\_successors$ **do**
9:     remove feasible start times that are later than the latest finish time (LFT) of the successor
10:   **if** len(feasible start times) > 1 **then**
11:    find the smallest run cost of the remaining feasible start times , select feasible start times whose run costs has the smallest run cost and set those start times as the cheapest feasible start times
12:    choose one cheapest feasible start time randomly
13:    schedule this job at the chosen cheapest feasible start time
14:    calculate the temporary household demand profile
15:    calculate the temporary max demand of this household
16:    **if** temporary max demand $> \bar{E}$ and len(feasible start times)> 0 **then**
17:     archive this temporary start time and the associating max demand
18:     remove this temporary start time from the feasible start times
19:     go to Step 11
20:    **if** len(feasible start time) $== 0$ **then**
21:     choose the archived start time whose associating max demand is the lowest
22:     schedule this job at this archived start time
23:   **else** schedule this job at the only feasible start time
24:   update the actual household demand profile

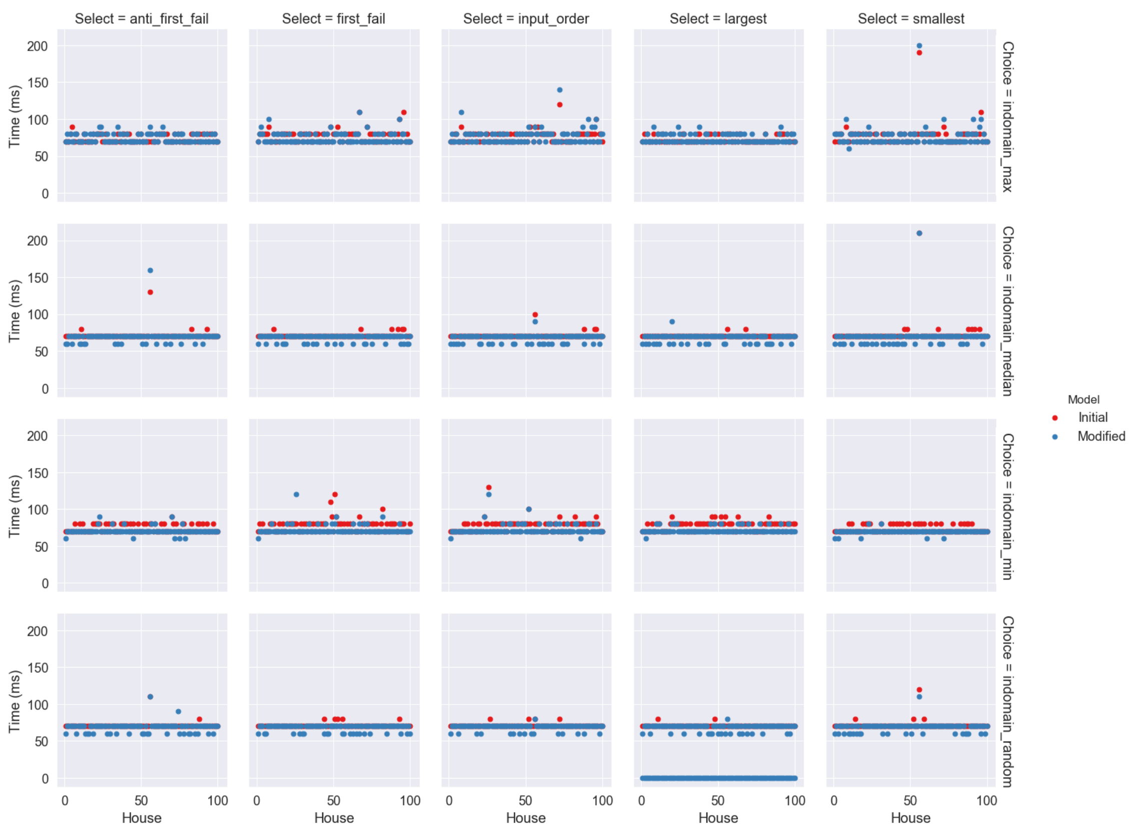
Return: scheduled/actual start times of jobs

Figure C.2: Sample Input Data for the modified MIP and CP Models

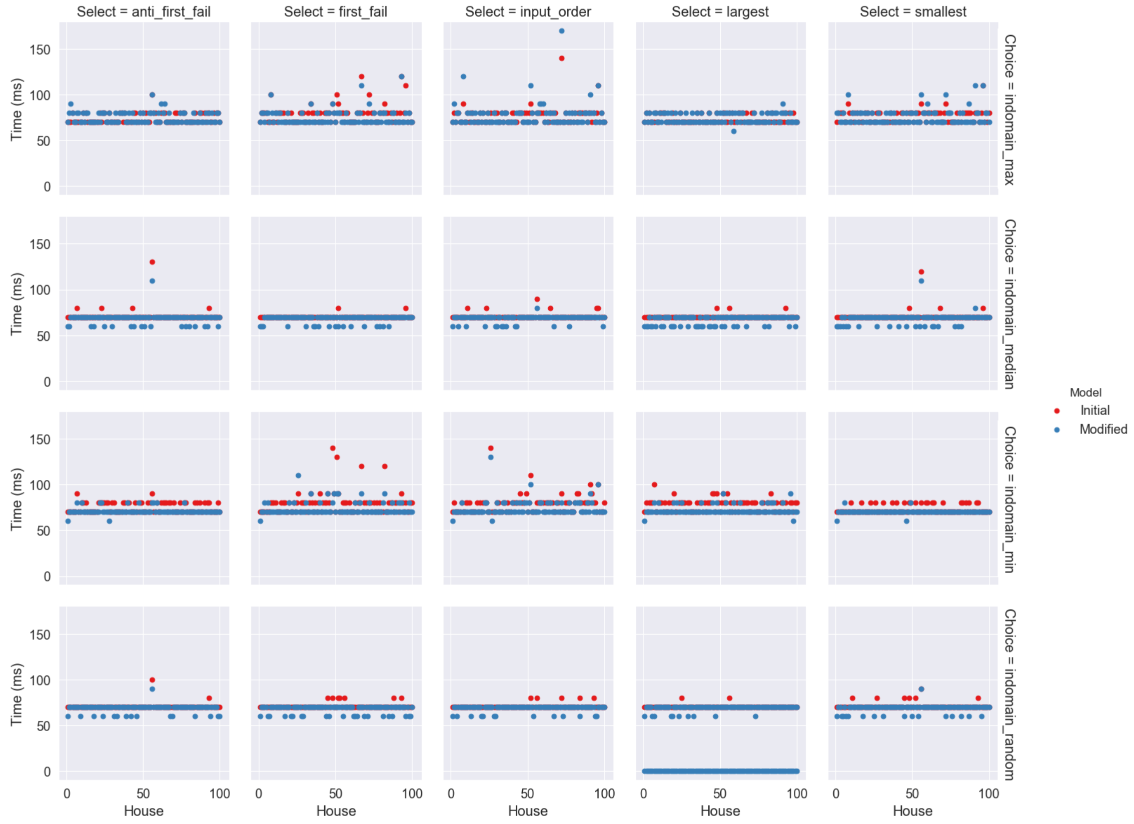
```
num_intervals=144;

num_tasks=10;
durations=[3, 3, 1, 2, 3, 4, 3, 6, 3, 4];
demands=[55, 15, 300, 700, 80, 2400, 3500, 1500, 400, 15];

num_precedences=2;
predecessors=[1, 7];
successors=[9, 10];
prec_delays=[62, 127];
max_demand=6275;

run_costs =
[|23306, 23305, 23304, 23303, 23302, 23301, 23300, 23299, 23298,
  23297, 23296, 23295, 23294, 23293, 23292, 23291, 23290, 23289,
  23288, 23287, 23286, 23285, 23284, 23283, 23282, 23391, 23500,
  23609, 23553, 23497, 23441, 23770, 24099, 24428, 26682, 28936,
  31190, 29759, 28328, 26897, 25906, 24915, 23926, 23707, 23488,
  23269, 23270, 23271, 23272, 23273, 23274, 23275, 23276, 23277,
  23278, 23279, 23280, 23281, 23282, 23283, 23284, 23285, 23286,
  23287, 23288, 23289, 23290, 23291, 23292, 23293, 23294, 23295,
  23296, 23297, 23298, 23299, 23355, 23411, 23467, 23413, 23359,
  23305, 23581, 23857, 24133, 24794, 25455, 26116, 28592, 31068,
  33544, 39595, 45646, 51697, 98008, 144319, 190630, 190631, 190632,
  190633, 206639, 222645, 238651, 193002, 147353, 101704, 101705,
  101706, 101707, 87793, 73879, 59965, 59966, 59967, 59968, 57219,
  54470, 51721, 57992, 64263, 70534, 59205, 47876, 36547, 33083,
  29619, 26155, 25606, 25057, 24508, 25059, 25610, 26161, 25337,
  24513, 23689, 23635, 23581, 23527, 23638, 23749, 23860, 99999999,
  99999999|6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345,
  6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345,
  6345, 6345, 6345, 6345, 6345, 6375, 6405, 6435, 6420, 6405, 6390,
  6480, 6570, 6660, 7275, 7890, 8505, 8115, 7725, 7335, 7065, 6795,
  6525, 6465, 6405, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345,
  6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345,
  6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345, 6345,
  6345, 6360, 6375, 6390, 6375, 6360, 6345, 6420, 6495, 6570, 6750,
  6930, 7110, 7785, 8460, 9135, 10785, 12435, 14085, 26715, 39345,
  51975, 51975, 51975, 51975, 56340, 60705, 65070, 52620, 40170,
  27720, 27720, 27720, 27720, 23925, 20130, 16335, 16335, 16335,
  16335, 15585, 14835, 14085, 15795, 17505, 19215, 16125, 13035,
  9945, 9000, 8055, 7110, 6960, 6810, 6660, 6810, 6960, 7110, 6885,
  6660, 6435, 6420, 6405, 6390, 6420, 6450, 6480, 99999999,
  99999999|......|......|......|......|......|......|......||];
```



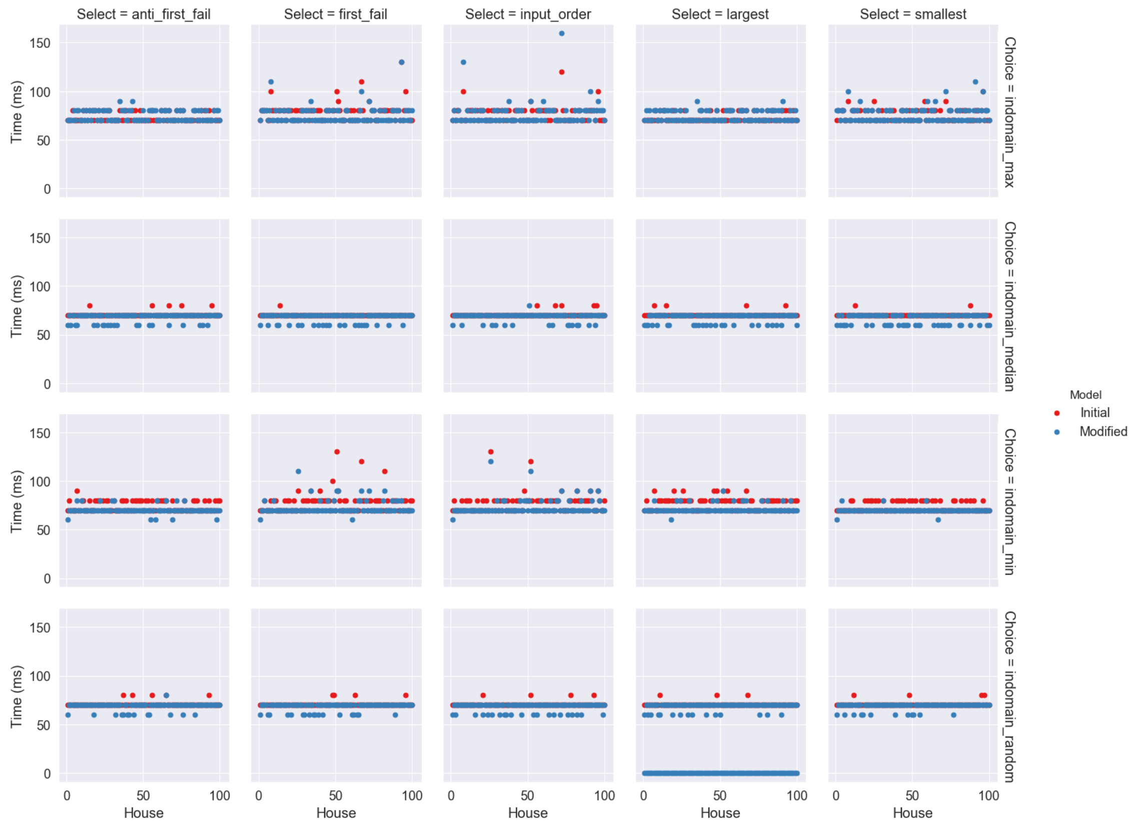
(a) Feb. average price profile



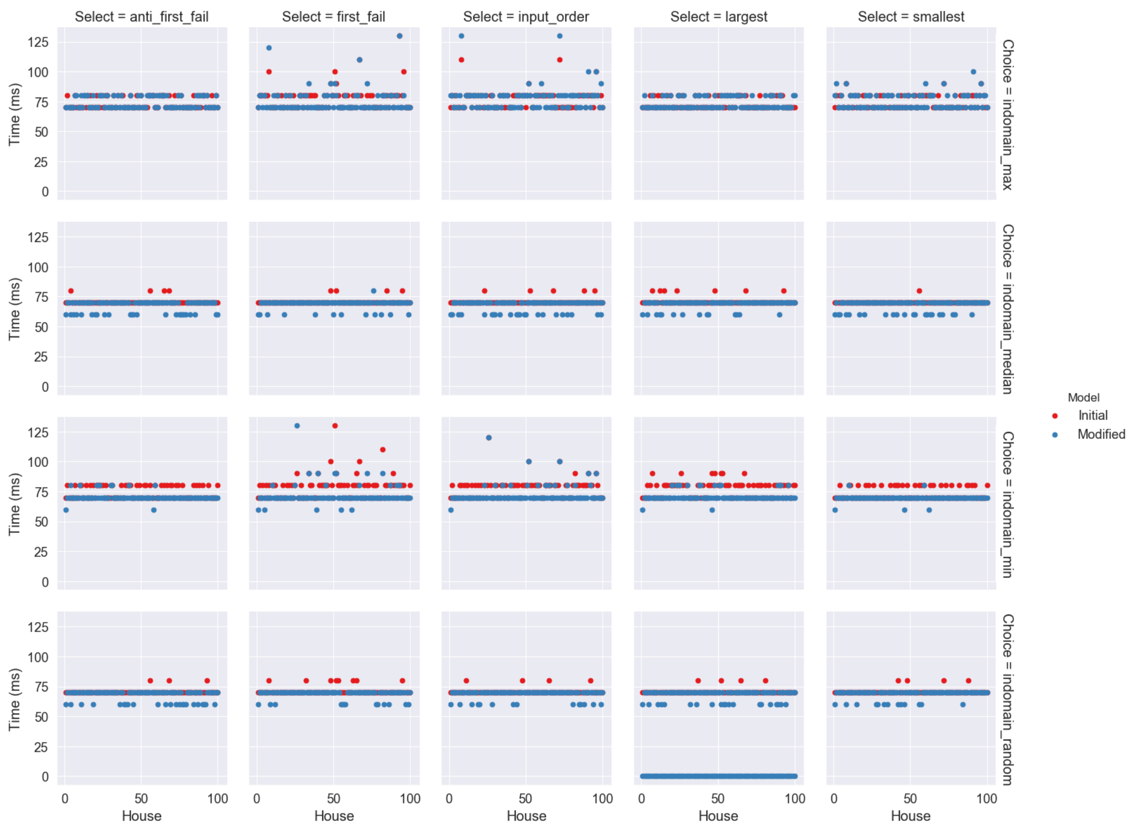
(b) Mar.average price profile



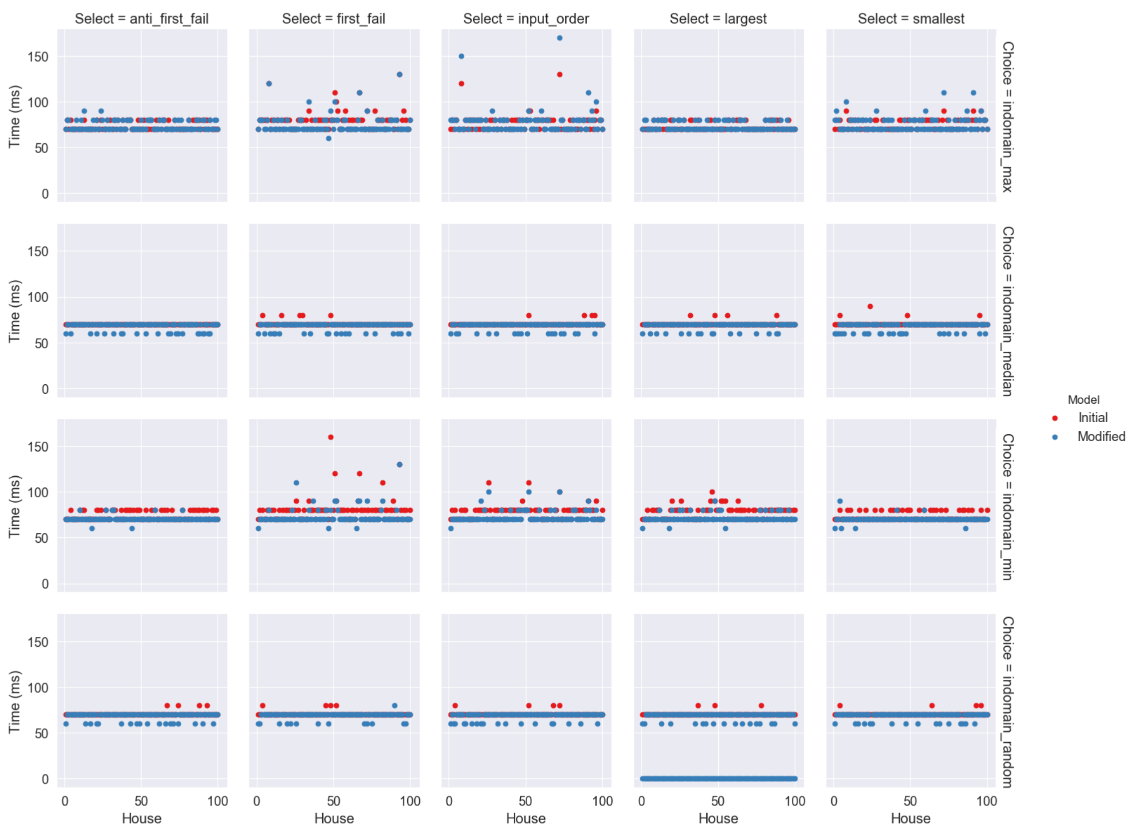
(c) Apr.average price profile



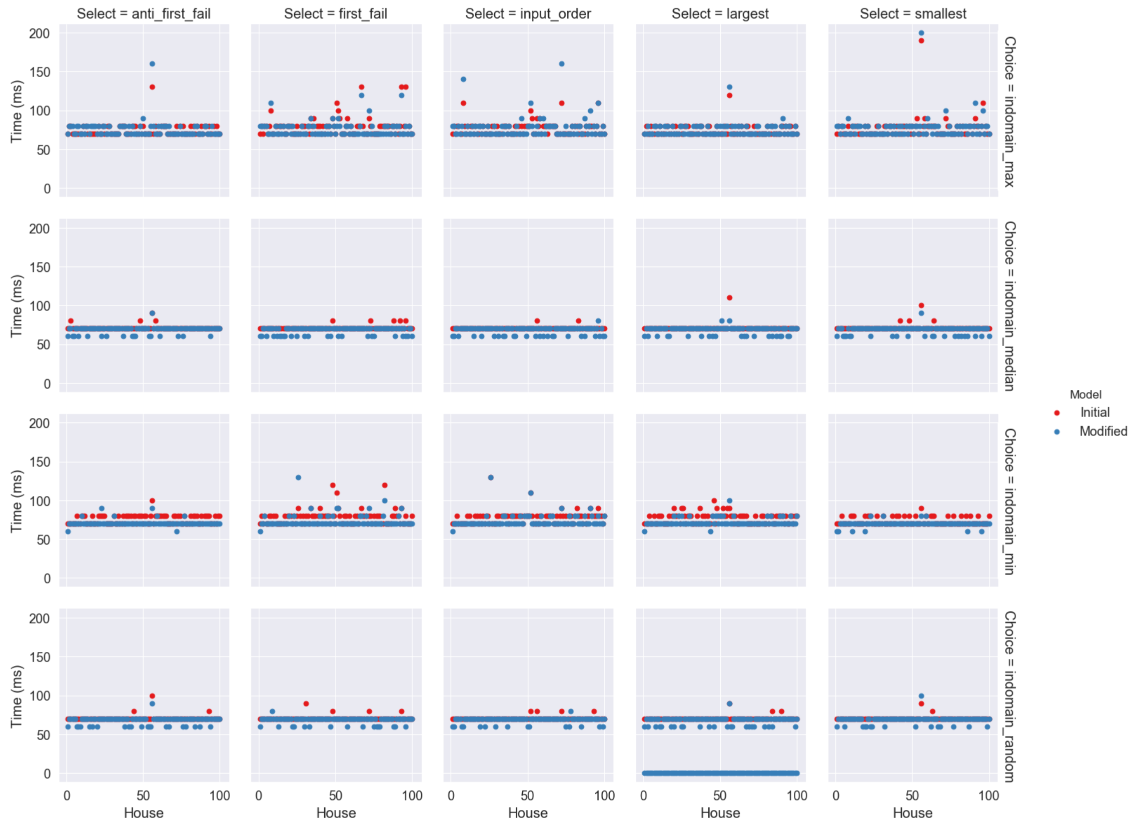
(d) May.average price profile



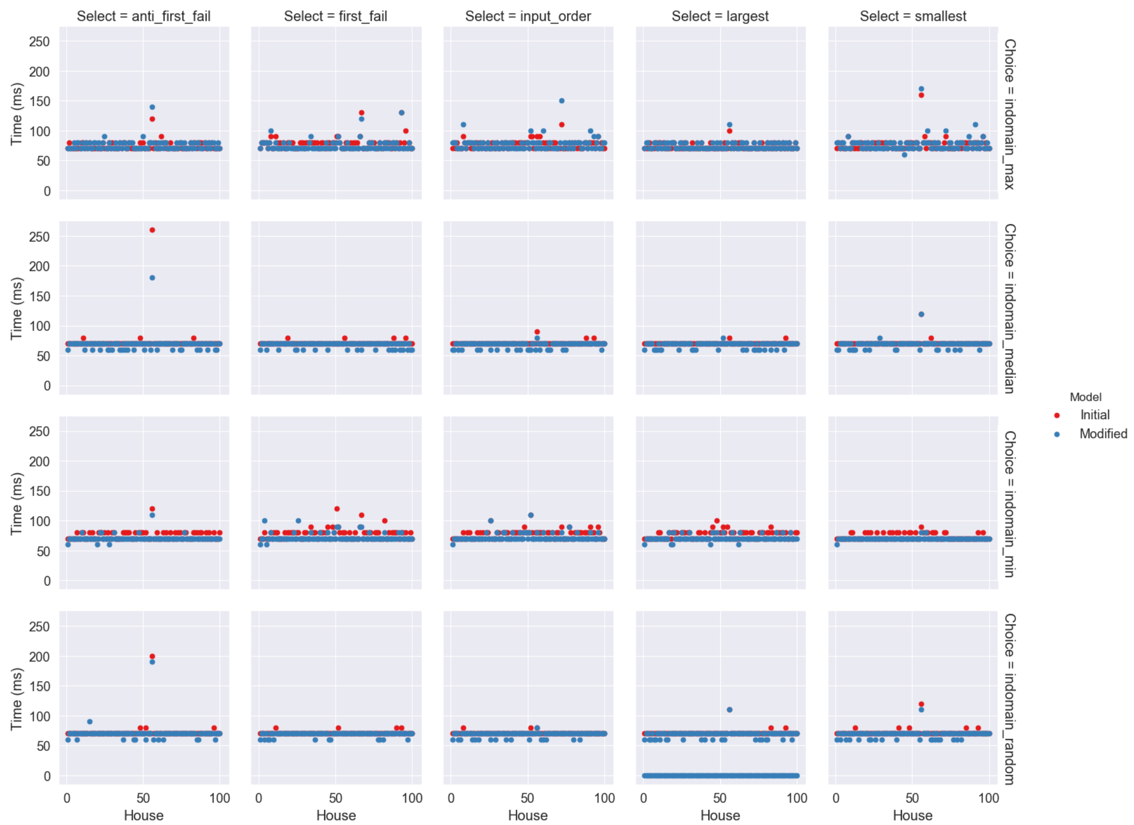
(e) Jun.average price profile



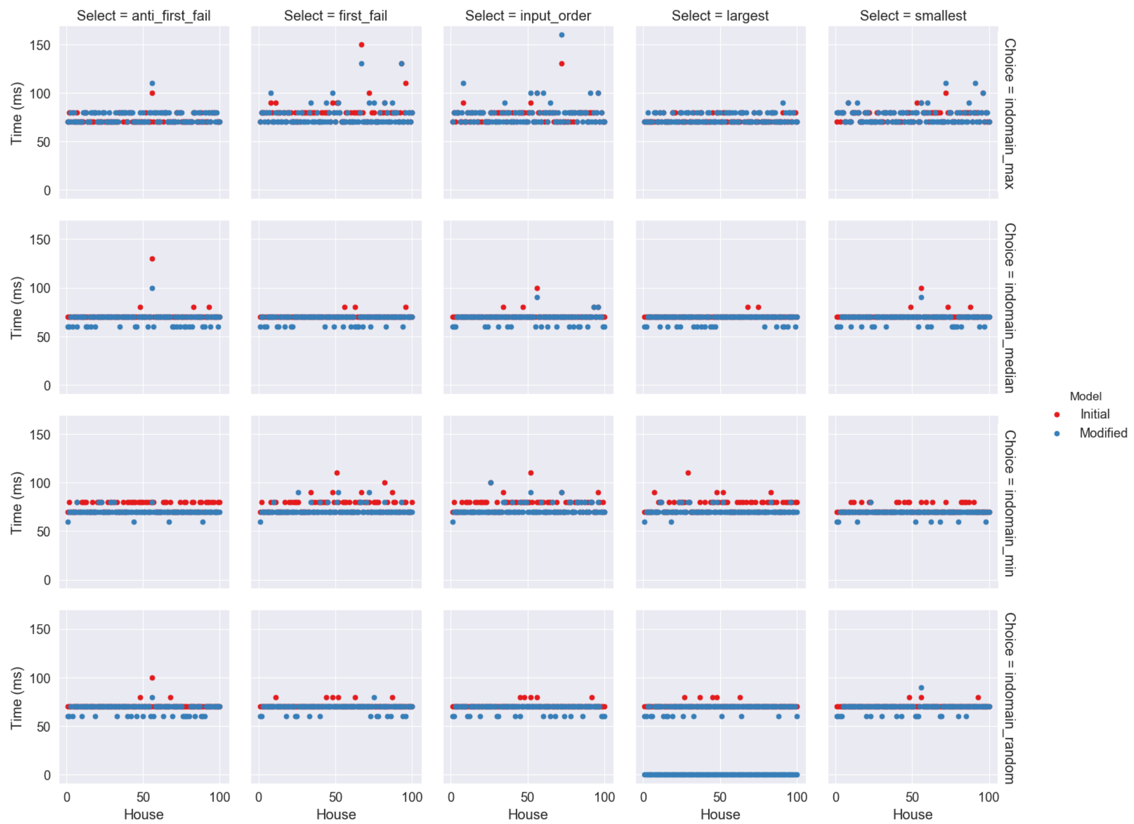
(f) Aug. average price profile



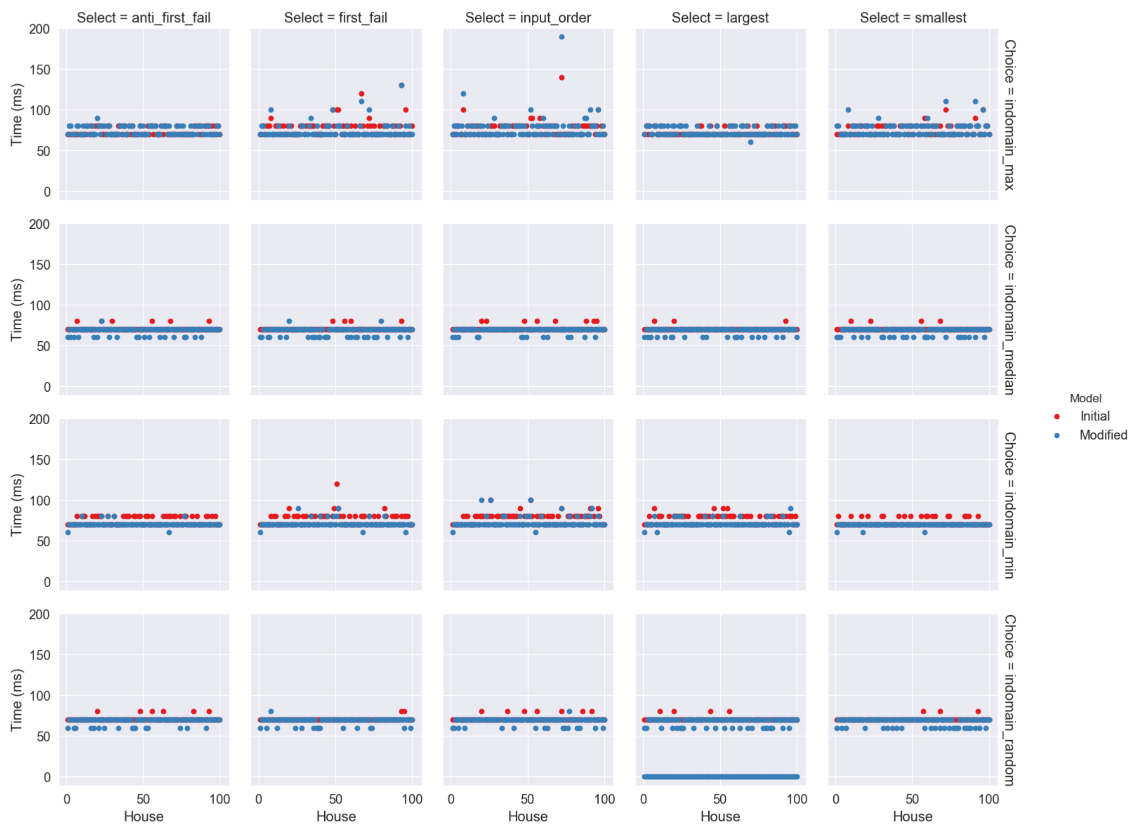
(g) Sep.average price profile



(h) Oct.average price profile



(i) Nov.average price profile



(j) Dec. average price profile

Figure C.-1: Run times of the CP models for various search strategies and different price profiles

References

Abdulla, K., de Hoog, J., Muenzel, V., Suits, F., Steer, K., Wirth, A. and Halgamuge, S. (2018). Optimal operation of energy storage systems considering forecasts and battery degradation, *IEEE Transactions on Smart Grid* **9**(3): 2086–2096.

Adika, C. O. and Wang, L. (2012). Autonomous appliance scheduling based on time of use probabilities and load clustering, *IPEC, 2012 Conference on Power Energy*, pp. 42–47.

Adika, C. O. and Wang, L. (2014). Smart charging and appliance scheduling approaches to demand side management, *International Journal of Electrical Power & Energy Systems* **57**: 232 – 240.
URL: *http://www.sciencedirect.com/science/article/pii/S0142061513005206*

AEMO (2017). Electricity price and demand data, *Technical report*, Australia Energy Market Operator. Accessed: April 2017.

AEMO (2020). Fact sheet: The national electricity market, *Technical report*, Australia Energy Market Operator. Accessed on 12 Nov 2021.
URL: *https://aemo.com.au/-/media/files/electricity/nem/national-electricity-market-fact-sheet.pdf*

AEMO (2021). 2021 NEM Regional Boundaries Map, *Technical report*, Australia Energy Market Operator.
URL: *https://www.aemo.com.au/aemo/apps/visualisations/map.html*

Agnetis, A., de Pascale, G., Detti, P. and Vicino, A. (2013). Load scheduling for household energy consumption optimization, *IEEE Transactions on Smart Grid* **4**(4): 2364–2373.

Ahmed, M. S., Mohamed, A., Khatib, T., Shareef, H., Homod, R. Z. and Ali, J. A. (2017). Real time optimal schedule controller for home energy management system using new

binary backtracking search algorithm, *Energy and Buildings* **138**: 215 – 227.
URL: *http://www.sciencedirect.com/science/article/pii/S0378778816313913*

Albadi, M. H. and El-Saadany, E. F. (2007). Demand response in electricity markets: An overview.

Alotaibi, I., Abido, M. A., Khalid, M. and Savkin, A. V. (2020). A comprehensive review of recent advances in smart grids: A sustainable future with renewable energy resources, *Energies* **13**(23).
URL: *https://www.mdpi.com/1996-1073/13/23/6269*

Andersson, J. (2001). *Multiobjective Optimization inEngineering Design: Applications to Fluid Power Systems*, PhD thesis, Division of Fluid and Mechanical Engineering Systems, Department of Mechanical Engineering, Linköpings universitet, SE-581 83 Linköping, Sweden.

Anvari-Moghaddam, A., Monsef, H. and Rahimi-Kian, A. (2015). Optimal smart home energy management considering energy saving and a comfortable lifestyle, *IEEE Transactions on Smart Grid* **6**(1): 324–332.

Atzeni, I., nez, L. G. O., Scutari, G., Palomar, D. P. and Fonollosa, J. R. (2012). Day-ahead bidding strategies for demand-side expected cost minimization, *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pp. 91–96.

Atzeni, I., nez, L. G., Scutari, G., Palomar, D. P. and Fonollosa, J. R. (2013). Demand-side management via distributed energy generation and storage optimization, *IEEE Transactions on Smart Grid* **4**(2): 866–876.

Ausgrid (n.d.). Appliance usage guide, http://www.ausgrid.com.au/Common/Customer-Services/Homes/Energy-efficiency/Energy-efficiency-at-home-tips/Energy-usage-calculators/ /media/Files/Customer

Azuatalam, D., Paridari, K., Ma, Y., Förstl, M., Chapman, A. C. and Verbič, G. (2019). Energy management of small-scale pv-battery systems: A systematic review considering practical implementation, computational requirements, quality of input data and

battery degradation, *Renewable and Sustainable Energy Reviews* **112**: 555–570.
URL: *https://www.sciencedirect.com/science/article/pii/S1364032119303983*

Balijepalli, V. S. K. M., Pradhan, V., Khaparde, S. A. and Shereef, R. M. (2011). Review of demand response under smart grid paradigm, *ISGT2011-India*, pp. 236–243.

Baptiste, P., Laborie, P., Pape, C. L. and Nuijten, W. (2006). Chapter 22 - constraint-based scheduling and planning, *in* F. Rossi, P. van Beek and T. Walsh (eds), *Handbook of Constraint Programming*, Vol. 2 of *Foundations of Artificial Intelligence*, Elsevier, pp. 761–799.
URL: *https://www.sciencedirect.com/science/article/pii/S157465260680026X*

Barbato, A. and Capone, A. (2014). Optimization models and methods for demand-side management of residential users: A survey, *Energies* **7**(9): 5787–5824.
URL: *http://www.mdpi.com/1996-1073/7/9/5787*

Barbato, A., Capone, A., Carello, G., Delfanti, M., Merlo, M. and Zaminga, A. (2011). House energy demand optimization in single and multi-user scenarios, *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pp. 345–350.

Battery University (2018). Bu-409: Charging lithium-ion, Online.
URL: *https://batteryuniversity.com/learn/article/charging_lithium_ion_batteries*

Battery University (2019). Bu-403: Charging lead acid, Online.
URL: *https://batteryuniversity.com/learn/article/charging_the_lead_acid_battery*

Bayram, I. S. and Ustun, T. S. (2017). A survey on behind the meter energy management systems in smart grid, *Renewable and Sustainable Energy Reviews* **72**: 1208 – 1232.
URL: *http://www.sciencedirect.com/science/article/pii/S1364032116306852*

Beard, K. W. (2019a). *Linden's Handbook of Batteries, Fifth Edition*, 5th edition. edn.

Beard, K. W. (2019b). *Linden's Handbook of Batteries, Fifth Edition*, McGraw-Hill Education: New York, Chicago, San Francisco, Athens, London, Madrid, Mexico City, Milan, New Delhi, Singapore, Sydney, Toronto, chapter COMPARISON OF PERFORMANCE CHARACTERISTICS FOR SECONDARY BATTERY SYSTEMS, Chapter.

URL: *https://www.accessengineeringlibrary.com/content/book/9781260115925/toc-chapter/chapter10/section/section10*

Bezanzon, J., Karpinski, S., Shah, V. and Edelman, A. (2012). Julia: A fast dynamic language for technical computing, *Lang.NEXT*.
URL: *http://julialang.org/images/lang.next.pdf*

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

Biggar, D. R. and Hesamzadeh, M. R. (2014a). *Electricity Industry Market Structure and Competition*, John Wiley & Sons Ltd, pp. 73–90.
URL: *http://dx.doi.org/10.1002/9781118775745.ch03*

Biggar, D. R. and Hesamzadeh, M. R. (2014b). *Introduction to Electric Power Systems*, John Wiley & Sons Ltd, pp. 33–72.
URL: *http://dx.doi.org/10.1002/9781118775745.ch02*

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* **3**(1): 1–122.
URL: *https://doi.org/10.1561/2200000016*

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*, Cambridge University Press, chapter Duality, pp. 215–288.

Boyd, S., Xiao, L., Mutapcic, A. and Mattingley, J. (2015). Notes on decomposition methods. Assessed: 2021-10-10.
URL: *https://web.stanford.edu/class/ee364b/lectures/decomposition_notes.pdf*

Bussieck, M. R. and Meeraus, A. (2004). General algebraic modeling system (gams), pp. 137–157.
URL: *https://doi.org/10.1007/978-1-4613-0215-5_8*

Callaway, D. S. and Hiskens, I. A. (2011). Achieving controllability of electric loads, *Proceedings of the IEEE* **99**(1): 184–199.

Caron, S. and Kesidis, G. (2010). Incentive-based energy consumption scheduling algorithms for the smart grid, *2010 First IEEE International Conference on Smart Grid Communications*, pp. 391–396.

Central Limit Theorem (2008). Springer New York, New York, NY, pp. 66–68.
URL: *https://doi.org/10.1007/978-0-387-32833-1_50*

Chaouachi, A., Kamel, R. M., Andoulsi, R. and Nagasaka, K. (2013). Multiobjective intelligent energy management for a microgrid, *IEEE Transactions on Industrial Electronics* **60**(4): 1688–1699.

Chapman, A. C., Rogers, A., Jennings, N. R. and Leslie, D. S. (2011). A unifying framework for iterative approximate best-response algorithms for distributed constraint optimization problems, *The Knowledge Engineering Review* **26**(4): 411–444.

Charnes, A. and Cooper, W. W. (1962). Programming with linear fractional functionals, *Naval Research Logistics Quarterly* **9**(3-4): 181–186.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800090303*

Chavali, P., Yang, P. and Nehorai, A. (2014). A distributed algorithm of appliance scheduling for home energy management system, *IEEE Transactions on Smart Grid* **5**(1): 282–290.

Chen, C., Kishore, S. and Snyder, L. V. (2011). An innovative RTP-based residential power scheduling scheme for smart grids, *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5956–5959.

Chen, L., Li, N., Low, S. H. and Doyle, J. C. (2010). Two market models for demand response in power networks, *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 397–402.

Chong, E. K. P. and Zak, S. H. (2001a). *An Introduction to Optimization, 2nd Edition*, Wiley-Interscience, chapter Gradient Methods, pp. 113–138.

Chong, E. K. P. and Zak, S. H. (2001b). *An Introduction to Optimization, 2nd Edition*, Wiley-Interscience, chapter Newton's Method, pp. 139–165.

Chong, E. K. P. and Zak, S. H. (2001c). *An Introduction to Optimization, 2nd Edition*, Wiley-Interscience, chapter Conjugate Direction Methods, pp. 151–165.

Chong, E. K. P. and Zak, S. H. (2001d). *An Introduction to Optimization, 2nd Edition*, Wiley-Interscience, chapter Algorithms for Constrained Optimization, pp. 439–453.

Chu, G., Stuckey, P. J., De la Banda, M. G. and Mears, C. (2011). Symmetries and lazy clause generation, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, IJCAI'11, AAAI Press, p. 516–521.

Chu, G., Stuckey, P. J., Schutt, A., Ehlers, T., Gange, G. and Francis, K. (n.d.). Chuffed, a lazy clause generation solver.
URL: *https://developers.google.com/optimization/*

Conejo, A. J., Castillo, E., Mínguez, R. and García-Bertrand, R. (2006). *Decomposition Techniques in Mathematical Programming*, Springer-Verlag Berlin Heidelberg, chapter Motivating Examples, pp. 3–66.

Conejo, A. J., Morales, J. M. and Baringo, L. (2010). Real-time demand response model, *IEEE Transactions on Smart Grid* **1**(3): 236–242.

Couraud, B., Norbu, S., Andoni, M., Robu, V., Gharavi, H. and Flynn, D. (2020). Optimal residential battery scheduling with asset lifespan consideration, *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pp. 630–634.

Dantzig, G. B. (1990). *Origins of the Simplex Method*, Association for Computing Machinery, New York, NY, USA, p. 141–151.
URL: *https://doi.org/10.1145/87252.88081*

David G. Luenberger, Y. Y. (2016). *Linear and Nonlinear Programming*, International Series in Operations Research & Management Science, Springer International Publishing Switzerland, chapter Primial Methods, pp. 357–396.

De Weck, O. L. (2004). Multiobjective optimization: History and promise, *Invited Keynote Paper, GL2-2, The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, Kanazawa, Japan*, Vol. 2, p. 34.

Deng, R., Yang, Z., Chow, M. Y. and Chen, J. (2015). A survey on demand response in smart grids: Mathematical models and approaches, *IEEE Transactions on Industrial Informatics* **11**(3): 570–582.

Deng, R., Yang, Z., Hou, F., Chow, M. Y. and Chen, J. (2015). Distributed real-time demand response in multiseller –multibuyer smart distribution grid, *IEEE Transactions on Power Systems* **30**(5): 2364–2374.

Department of Education, Skills and Employment (n.d.). National collaborative research infrastructure strategy (ncris), https://www.dese.gov.au/ncris.
URL: *https://www.dese.gov.au/ncris*

e Silva de Oliveira, R. M. and de C. Ribeiro, M. S. F. O. (2015). Comparing mixed & integer programming vs. constraint programming by solving job-shop scheduling problems, *Independent Journal of Management & Production* **6**(1).
URL: *http://www.ijmp.jor.br/index.php/ijmp/article/view/262*

Ebalance-plus (2020). Communication technologies for smart grids. hierarchical network, Online. Accessed on 12 Nov 2021.
URL: *https://www.ebalanceplus.eu/the-smart-grid-hierarchical-network/*

Elgerd, O. I. (1982). *ELECTRIC ENERGY SYSTEMS THEORY: An Introduction*, McGraw-Hill, Inc.

Evans, M. J. and Rosenthal, J. S. (2009). *Probability and Statistics - The Science of Uncertainty, Second Edition*, University of Toronto, chapter Chpater 4: Sampling Distributions and Limits, pp. 199–220.

Fan, Z. (2011). Distributed demand response and user adaptation in smart grids, *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pp. 726–729.

Fderal Energy Regulatory Commission (n.d.). Miso daily report, https://www.ferc.gov/industries-data/electric/electric-power-markets/miso.

Finkel, A., Moses, K., Munro, C., Effeney, T. and O'Kane, M. (2017a). *Independent review into the future security of the national electricity market: blueprint for the future [Finkel review]*, Canberra, Australian Capital Territory: Commonwealth of Australia, 2017, chapter Chapter 6 Rewarding Consumers, pp. 137–156.
URL: *https://www.energy.gov.au/government-priorities/energy-markets/independent-review-future-security-national-electricity-market*

Finkel, A., Moses, K., Munro, C., Effeney, T. and O'Kane, M. (2017b). Independent review into the future security of the nem, *Technical report*, Department of the Environment and Energy.

URL: *https://www.energy.gov.au/sites/default/files/independent-review-future-nem-blueprint-for-the-future-2017.pdf*

Fioretto, F., Yeoh, W. and Pontelli, E. (2017). A multiagent system approach to scheduling devices in smart homes, *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 981–989.

URL: *http://dl.acm.org/citation.cfm?id=3091125.3091265*

Fourer, R., Gay, D. M. and Kernighan, B. (1989). *in* S. W. Wallace (ed.), *Algorithms and Model Formulations in Mathematical Programming*, Springer-Verlag, Berlin, Heidelberg, chapter AMPL: A Mathematical Programming Language, pp. 150–151.

URL: *http://dl.acm.org/citation.cfm?id=107479.107491*

Fourer, R., Gay, D. M. and Kernighan, B. W. (2003). *AMPL: A Mathematical Programing Language*, Cengage Learning; 2nd edition (November 12, 2002).

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming, *Naval research logistics quarterly* **3**(1-2): 95–110.

Gecode Team (2017). Gecode: Generic constraint development environment. Available from `http://www.gecode.org`.

Ghazvini, M. A. F., Soares, J., Abrishambaf, O., Castro, R. and Vale, Z. (2017). Demand response implementation in smart households, *Energy and Buildings* **143**: 129 – 148.

URL: *http://www.sciencedirect.com/science/article/pii/S037877881730823X*

GloBird energy (2021). Understanding the electricity market, Online. Accessed on 12 Nov 2021.

URL: *https://www.globirdenergy.com.au/help-support/how-the-energy-market-works/*

Good, N., Ellis, K. A. and Mancarella, P. (2017). Review and classification of barriers and enablers of demand response in the smart grid, *Renewable and Sustainable Energy*

Reviews **72**: 57 – 72.
URL: *http://www.sciencedirect.com/science/article/pii/S1364032117300436*

Goudarzi, H., Hatami, S. and Pedram, M. (2011). Demand-side load scheduling incentivized by dynamic energy prices, *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pp. 351–356.

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1, `http://cvxr.com/cvx`.

Gurobi Optimization, LLC (2021). Gurobi Optimizer Reference Manual.
URL: *https://www.gurobi.com*

Hatami, S. and Pedram, M. (2010). Minimizing the electricity bill of cooperative users under a quasi-dynamic pricing model, *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 421–426.

Hauser, R. (2007). Line search methods forunconstrained optimisation. Accessed: 2021–10-10.
URL: *https://people.maths.ox.ac.uk/hauser/hauser_lecture2.pdf*

He, M.-f., Zhang, F.-x., Huang, Y., Chen, J., Wang, J. and Wang, R. (2019). A distributed demand side energy management algorithm for smart grid, *Energies* **12**(3).
URL: *https://www.mdpi.com/1996-1073/12/3/426*

Hongbo Zhu, Yan Gao, Y. H. (2018). Real-time pricing for demand response in smart grid based on alternating direction method of multipliers, *Mathematical Problems in Engineering* **2018**(8760575).

Hooker, J. (2019). Optimization models for social justice, *Technical report*, Carnegie Mellon University.
URL: *http://public.tepper.cmu.edu/jnh/equityCPAIORmasterClass2019.pdf*

Hossain, M. A., Pota, H. R., Squartini, S., Zaman, F. and Guerrero, J. M. (2019). Energy scheduling of community microgrid with battery cost using particle swarm optimisation, *Applied Energy* **254**: 113723.
URL: *http://www.sciencedirect.com/science/article/pii/S0306261919314102*

Hussain, H. M., Javaid, N., Iqbal, S., Hasan, Q. U., Aurangzeb, K. and Alhussein, M. (2018). An efficient demand side management system with a new optimized home energy management controller in smart grid, *Energies* **11**(1).
URL: *https://www.mdpi.com/1996-1073/11/1/190*

IEA (2020). Global ev outlook 2020, *Technical report*, Paris.
URL: *https://www.iea.org/reports/global-ev-outlook-2020*

ILOG, Inc (2006). ILOG CPLEX: High-performance software for mathematical programming and optimization. See `http://www.ilog.com/products/cplex/`.

Jiang, Z., Lin, R. and Yang, F. (2019). Incremental electricity consumer behavior learning using smart meter data, *Proceedings of the 2019 4th International Conference on Big Data and Computing*, ICBDC 2019, Association for Computing Machinery, New York, NY, USA, p. 54–59.
URL: *https://doi.org/10.1145/3335484.3335517*

Joe-Wong, C., Sen, S., Ha, S. and Chiang, M. (2012). Optimized day-ahead pricing for smart grids with device-specific scheduling flexibility, *IEEE Journal on Selected Areas in Communications* **30**(6): 1075–1085.

Jovanovic, R., Bousselham, A. and Bayram, I. S. (2016). Residential demand response scheduling with consideration of consumer preferences, *Applied Sciences* **6**(1).
URL: *https://www.mdpi.com/2076-3417/6/1/16*

Kelareva, E., Brand, S., Kilby, P., Thiebaux, S. and Wallace, M. (2012). Cp and mip methods for ship scheduling with time-varying draft, *ICAPS 2012 - Proceedings of the 22nd International Conference on Automated Planning and Scheduling* .

Kim, S. J. and Giannakis, G. B. (2013). Scalable and robust demand response with mixed-integer constraints, *IEEE Transactions on Smart Grid* **4**(4): 2089–2099.

Kou, X., Li, F., Dong, J., Starke, M., Munk, J., Xue, Y., Olama, M. and Zandi, H. (2020). A scalable and distributed algorithm for managing residential demand response programs using alternating direction method of multipliers (admm), *IEEE Transactions on Smart Grid* pp. 1–1.

Kuschel, C., Köstler, H. and Rüde, U. (2015). Multi-energy simulation of a smart grid with optimal local demand and supply management, *Smart Grid and Renewable Energy* **06**(11): 303–315.

Laborie, P. (2018). *An Update on the Comparison of MIP, CP and Hybrid Approaches for Mixed Resource Allocation and Scheduling*, pp. 403–411.

Lee, J.-H., Kim, H.-J., Park, G.-L. and Kang, M.-K. (2012). Energy consumption scheduler for demand response systems in the smart grid, *Journal of Information Science and Engineering* **28**(5): 955–969.

Li, J. and van der Linden, W. J. (2018). A comparison of constraint programming and mixed-integer programming for automated test-form generation, *Journal of Educational Measurement* **55**(4): 435–456.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1111/jedm.12187*

Li, N., Chen, L. and Low, S. H. (2011). Optimal demand response based on utility maximization in power networks, *Power and Energy Society General Meeting, 2011 IEEE*, pp. 1–8.

Li, Y., Ng, B. L., Trayer, M. and Liu, L. (2012). Automated residential demand response: Algorithmic implications of pricing models, *IEEE Transactions on Smart Grid* **3**(4): 1712–1721.

Li, Y. and Trayer, M. (2012). Automated residential demand response: Algorithmic implications of pricing models, *2012 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 626–629.

Li, Y., Yang, Z., Li, G., Zhao, D. and Tian, W. (2019). Optimal scheduling of an isolated microgrid with battery storage considering load and renewable generation uncertainties, *IEEE Transactions on Industrial Electronics* **66**(2): 1565–1575.

Lin, S., Meng, N. and Li, W. (2021). Generating Efficient Solvers from Constraint Models, *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2021, Association for Computing Machinery, New York, NY, USA, p. 956–967.
URL: *https://doi.org/10.1145/3468264.3468566*

Lofberg, J. (2004). Yalmip : a toolbox for modeling and optimization in matlab, *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, pp. 284–289.

Logenthiran, T., Srinivasan, D. and Shun, T. Z. (2012). Demand side management in smart grid using heuristic optimization, *IEEE Transactions on Smart Grid* **3**(3): 1244–1252.

Longe, O. M., Ouahada, K., Rimer, S., Harutyunyan, A. N. and Ferreira, H. C. (2017). Distributed demand side management with battery storage for smart home energy scheduling, *Sustainability* **9**(1).
URL: *https://www.mdpi.com/2071-1050/9/1/120*

Luenberger, D. G. and Ye, Y. (2016). *Primal Methods*, Springer International Publishing, Cham, pp. 357–396.
URL: *https://doi.org/10.1007/978-3-319-18842-3_12*

Ma, K., Yao, T., Yang, J. and Guan, X. (2016). Residential power scheduling for demand response in smart grid, *International Journal of Electrical Power & Energy Systems* **78**: 320 – 325.
URL: *http://www.sciencedirect.com/science/article/pii/S014206151500530X*

Maharjan, S., Zhu, Q., Zhang, Y., Gjessing, S. and Basar, T. (2013). Dependable demand response management in the smart grid: A stackelberg game approach, *IEEE Transactions on Smart Grid* **4**(1): 120–132.

Maleck, C., Nieke, G., Bock, K., Pabst, D. and Stehli, M. (2018). A comparison of an cp and mip approach for scheduling jobs in production areas with time constraints and uncertainties, *2018 Winter Simulation Conference (WSC)*, pp. 3526–3537.

Manzoor, A., Javaid, N., Ullah, I., Abdul, W., Almogren, A. and Alamri, A. (2017). An intelligent hybrid heuristic scheme for smart metering based demand side management in smart homes, *Energies* **10**(9).
URL: *https://www.mdpi.com/1996-1073/10/9/1258*

Marzband, M., Alavi, H., Ghazimirsaeid, S. S., Uppal, H. and Fernando, T. (2017). Optimal energy management system based on stochastic approach for a home microgrid with integrated responsive load demand and energy storage, *Sustainable Cities and Society*

28: 256 – 264.
URL: *http://www.sciencedirect.com/science/article/pii/S2210670716304449*

Mhanna, S., Chapman, A. C. and Verbic, G. (2016). A fast distributed algorithm for large-scale demand response aggregation, *CoRR* **abs/1603.00149**.
URL: *http://arxiv.org/abs/1603.00149*

Mishra, A., Irwin, D., Shenoy, P. and Zhu, T. (2013). Scaling distributed energy storage for grid peak reduction, *Proceedings of the Fourth International Conference on Future Energy Systems*, e-Energy '13, Association for Computing Machinery, New York, NY, USA, p. 3–14.
URL: *https://doi.org/10.1145/2487166.2487168*

MITei (2011). *The Future of the Electric Grid: An Interdisciplinary MIT Study*, Massachusetts Institute of Technology, chapter Enhancing the Transmission Network and System Operations, pp. 31–52.

Mohsenian-Rad, A. H. and Leon-Garcia, A. (2010). Optimal residential load control with price prediction in real-time electricity pricing environments, *IEEE Transactions on Smart Grid* **1**(2): 120–133.

Mohsenian-Rad, A. H., Wong, V. W. S., Jatskevich, J., Schober, R. and Leon-Garcia, A. (2010). Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid, *IEEE Transactions on Smart Grid* **1**(3): 320–331.

Muenzel, V., de Hoog, J., Brazil, M., Vishwanath, A. and Kalyanaraman, S. (2015). A multi-factor battery cycle life prediction methodology for optimal battery management, *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*, e-Energy '15, Association for Computing Machinery, New York, NY, USA, p. 57–66.
URL: *https://doi.org/10.1145/2768510.2768532*

Muralitharan, K., Sakthivel, R. and Shi, Y. (2016). Multiobjective optimization technique for demand side management with load balancing approach in smart grid, *Neurocomputing* **177**: 110 – 119.
URL: *http://www.sciencedirect.com/science/article/pii/S0925231215017117*

Nakamura, K., Sakaue, S. and Yasuda, N. (2020). Practical frank–wolfe method with decision diagrams for computing wardrop equilibrium of combinatorial congestion games, *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(02): 2200–2209.
URL: *https://ojs.aaai.org/index.php/AAAI/article/view/5596*

Nan, S., Zhou, M. and Li, G. (2018). Optimal residential community demand response scheduling in smart grid, *Applied Energy* **210**: 1280 – 1289.
URL: *http://www.sciencedirect.com/science/article/pii/S030626191730819X*

Nesterov, Y. (2013). *Introductory Lectures on Convex Optimization*, Springer, chapter Smooth Convex Optimization, pp. 51–110.

Nethercote, N., Stuckey, P. J., Becket, R., Brand, S., Duck, G. J. and Tack, G. (2007). Minizinc: Towards a standard cp modelling language, *in* C. Bessière (ed.), *Principles and Practice of Constraint Programming – CP 2007*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 529–543.

Nguyen, H. K., Song, J. B. and Han, Z. (2012). Demand side management to reduce peak-to-average ratio using game theory in smart grid, *2012 Proceedings IEEE INFOCOM Workshops*, pp. 91–96.

Nicholson, M. (2012). *The Power Makers' Challenge: And the Need for Fission Energy*, Springer-Verlag London, London, chapter The Balancing Act, pp. 19–22.
URL: *http://link.springer.com/content/pdf/bbm:978-1-4471-2813-7/1.pdf*

Ogwumike, C., Short, M. and Denai, M. (2015). Near-optimal scheduling of residential smart home appliances using heuristic approach, *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp. 3128–3133.

Pandžić, H. and Bobanac, V. (2019). An accurate charging model of battery energy storage, *IEEE Transactions on Power Systems* **34**(2): 1416–1426.

Pelzer, D., Ciechanowicz, D. and Knoll, A. (2016). Energy arbitrage through smart scheduling of battery energy storage considering battery degradation and electricity price forecasts, *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, pp. 472–477.

Perron, L. and Furnon, V. (n.d.). Or-tools.
URL: *https://developers.google.com/optimization/*

Pilz, M., Al-Fagih, L. and Pfluegel, E. (2017). Energy storage scheduling with an advanced battery model: A game–theoretic approach, *Inventions* **2**(4).
URL: *https://www.mdpi.com/2411-5134/2/4/30*

Pooranian, Z., Abawajy, J. H., P, V. and Conti, M. (2018). Scheduling distributed energy resource operation and daily power consumption for a smart building to optimize economic and environmental parameters, *Energies* **11**(6).
URL: *https://www.mdpi.com/1996-1073/11/6/1348*

Pratt, R., Balducci, P., Gerkensmeyer, C. and Katipamula, S. (2010). The smart grid: An estimation of the energy and CO_2 benefits, *Technical report*, Pacific Northwest National Laboratory.

Rahbari-Asr, N., Ojha, U., Zhang, Z. and Chow, M. Y. (2014). Incremental welfare consensus algorithm for cooperative distributed generation/demand response in smart grid, *IEEE Transactions on Smart Grid* **5**(6): 2836–2845.

Ramadass, P., Haran, B., Gomadam, P. M., White, R. and Popov, B. N. (2004). Development of first principles capacity fade model for li-ion cells, *Journal of The Electrochemical Society* **151**(2): A196.
URL: *https://doi.org/10.1149/1.1634273*

Ramchurn, S. D., Vytelingum, P., Rogers, A. and Jennings, N. (2011). Agent-based control for decentralised demand side management in the smart grid, *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '11, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 5–12.
URL: *http://dl.acm.org/citation.cfm?id=2030470.2030472*

Rasheed, M. B., Javaid, N., Ahmad, A., Jamil, M., Khan, Z. A., Qasim, U. and Alrajeh, N. (2016). Energy optimization in smart homes using customer preference and dynamic pricing, *Energies* **9**(8).
URL: *https://www.mdpi.com/1996-1073/9/8/593*

Ren, D., Li, H. and Ji, Y. (2011). Home energy management system for the residential load control based on the price prediction, *2011 IEEE Online Conference on Green Communications*, pp. 1–6.

Robinson, D. P. (2020). Line-search methods for smooth unconstrained optimization. Accessed: 2021–10-10.

URL: *https://www.ams.jhu.edu/ abasu9/AMS_553-761/lecture05_handout2.pdf*

Safoutin, M., Cherry, J., McDonald, J. and Lee, S. (2015). Effect of current and soc on round-trip energy efficiency of a lithium-iron phosphate (lifepo4) battery pack, *SAE 2015 World Congress and Exhibition*, SAE International.

URL: *https://doi.org/10.4271/2015-01-1186*

Samadi, P., Mohsenian-Rad, A. H., Schober, R., Wong, V. W. S. and Jatskevich, J. (2010). Optimal real-time pricing algorithm based on utility maximization for smart grid, *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 415–420.

Scott, P. (2016). *Distributed Coordination and Optimisation ofNetwork-Aware Electricity Prosumers*, phdthesis, The Australian National University.

URL: *http://users.cecs.anu.edu.au/ pscott/extras/thesis/phdthesis.pdf*

Sharda, S., Singh, M. and Sharma, K. (2021). A complete consumer behaviour learning model for real-time demand response implementation in smart grid, *Applied Intelligence* .

Sheffi, Y. (1985). *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice-Hall.

Sheikhi, A., Rayati, M., Bahrami, S. and Ranjbar, A. M. (2015). Integrated demand side management game in smart energy hubs, *IEEE Transactions on Smart Grid* **6**(2): 675–683.

Shi, W., Xie, X., Chu, C. C. and Gadh, R. (2014). A distributed optimal energy management strategy for microgrids, *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 200–205.

Shi, W., Xie, X., Chu, C. C. and Gadh, R. (2015). Distributed optimal energy management in microgrids, *IEEE Transactions on Smart Grid* **6**(3): 1137–1146.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press, p. 484.

Siano, P. (2014). Demand response and smart grids — a survey, *Renewable and Sustainable Energy Reviews* **30**: 461 – 478.
URL: *http://www.sciencedirect.com/science/article/pii/S1364032113007211*

Siebert, L. C., Sbicca, A., Aoki, A. R. and Lambert-Torres, G. (2017). A behavioral economics approach to residential electricity consumption, *Energies* **10**(6).
URL: *https://www.mdpi.com/1996-1073/10/6/768*

Song, L., Xiao, Y. and van der Schaar, M. (2014). Demand side management in smart grids using a repeated game framework, *IEEE Journal on Selected Areas in Communications* **32**(7): 1412–1424.

Sou, K. C., Weimer, J., Sandberg, H. and Johansson, K. H. (2011). Scheduling smart home appliances using mixed integer linear programming, *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 5144–5149.

Sperstad, I. B. and Korpås, M. (2019). Energy storage scheduling in distribution systems considering wind and photovoltaic generation uncertainties, *Energies* **12**(7).
URL: *https://www.mdpi.com/1996-1073/12/7/1231*

Stuckey, P. J., Marriott, K. and Tack, G. (2018). *The MiniZinc Handbook 2.2.3*, Monash University and Data61 CSIRO.
URL: *https://www.minizinc.org/doc-2.2.3/en/fzn-spec.html*

Sui, Y. and Song, S. (2020). A multi-agent reinforcement learning framework for lithium-ion battery scheduling problems, *Energies* **13**(8).
URL: *https://www.mdpi.com/1996-1073/13/8/1982*

Swalehe, H. and Marungsri, B. (2018). Intelligent algorithm for optimal load management in smart home appliance scheduling in distribution system, *2018 International Electrical Engineering Congress (iEECON)*, pp. 1–4.

Tsui, K. M. and Chan, S. C. (2012). Demand response optimization for smart home scheduling under real-time pricing, *IEEE Transactions on Smart Grid* **3**(4): 1812–1821.

U.S. Department of Energy (2006). Benefits of demand response in electricity markets and recommendations for achieving them, Report to the United States Congress. Available online: http://eetd.lbl.gov.

U.S. Department of Energy (2009). The smart grid: An introduction.
URL: *https://energy.gov/oe/downloads/smart-grid-introduction*

U.S. Department of Energy (2013). Commercial and residential hourly load profiles for all tmy3 locations in the unites states, *Technical report*, U.S. Department of Energy, Washington, DC, USA.

Vagropoulos, S. I. and Bakirtzis, A. G. (2013). Optimal bidding strategy for electric vehicle aggregators in electricity markets, *IEEE Transactions on Power Systems* **28**(4): 4031–4041.

van Beek, P. (2006). Chapter 4 - backtracking search algorithms, *in* F. Rossi, P. van Beek and T. Walsh (eds), *Handbook of Constraint Programming*, Vol. 2 of *Foundations of Artificial Intelligence*, Elsevier, pp. 85–134.
URL: *https://www.sciencedirect.com/science/article/pii/S1574652606800088*

Van Den Briel, M., Scott, P. and Thiébaux, S. (2013). Randomized load control: A simple distributed approach for scheduling smart appliances, *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, AAAI Press, pp. 2915–2922.
URL: *http://dl.acm.org/citation.cfm?id=2540128.2540548*

van Hoeve, W.-J. and Katriel, I. (2006). Chapter 6 - global constraints, *in* F. Rossi, P. van Beek and T. Walsh (eds), *Handbook of Constraint Programming*, Vol. 2 of *Foundations of Artificial Intelligence*, Elsevier, pp. 169–208.
URL: *https://www.sciencedirect.com/science/article/pii/S1574652606800106*

Vandenberghe, L. and Boyd, S. P. (2004). *Convex Optimization*, Cambridge University Press, chapter Unconstrained minimization, pp. 457–520.
URL: *https://web.stanford.edu/ boyd/cvxbook/*

Vardakas, J. S., Zorba, N. and Verikoukis, C. V. (2015). A survey on demand response programs in smart grids: Pricing methods and optimization algorithms, *IEEE Communications Surveys Tutorials* **17**(1): 152–178.

Varghese, A. C., V., P., Kumar, G. and Khaparde, S. A. (2018). Smart grid consumer behavioral model using machine learning, *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pp. 734–739.

Veit, A., Xu, Y., Zheng, R., Chakraborty, N. and Sycara, K. (2014). Demand side energy management via multiagent coordination in consumer cooperatives, *J. Artif. Int. Res.* **50**(1): 885–922.
URL: *http://dl.acm.org/citation.cfm?id=2693068.2693091*

Voice, T. D., Vytelingum, P., Ramchurn, S. D., Rogers, A. and Jennings, N. R. (2011). Decentralised control of micro-storage in the smart grid, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, AAAI Press, pp. 1421–1427.
URL: *http://dl.acm.org/citation.cfm?id=2900423.2900648*

Vytelingum, P., Voice, T. D., Ramchurn, S. D., Rogers, A. and Jennings, N. R. (2010). Agent-based micro-storage management for the smart grid, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 39–46.
URL: *http://dl.acm.org/citation.cfm?id=1838206.1838212*

Wakefield, M. and McGranaghan, M. (2009). Achieving smart grid interoperability through collaboration, *CIRED 2009 - 20th International Conference and Exhibition on Electricity Distribution - Part 1*, pp. 1–4.

Wang, Y., Mao, S. and Nelms, R. M. (2014). Distributed online algorithm for optimal real-time energy distribution in the smart grid, *IEEE Internet of Things Journal* **1**(1): 70–80.

Wankmüller, F., Thimmapuram, P. R., Gallagher, K. G. and Botterud, A. (2017). Impact of battery degradation on energy arbitrage revenue of grid-level energy storage, *Journal of Energy Storage* **10**: 56–66.
URL: *https://www.sciencedirect.com/science/article/pii/S2352152X16303231*

Worthmann, K., Kellett, C. M., Braun, P., Grüne, L. and Weller, S. R. (2015). Distributed and decentralized control of residential energy systems incorporating battery storage, *IEEE Transactions on Smart Grid* **6**(4): 1914–1923.

Yang, Z., Long, K., You, P. and Chow, M.-Y. (2015). Joint scheduling of large-scale appliances and batteries via distributed mixed optimization, *2015 IEEE Power Energy Society General Meeting*, pp. 1–1.

Yu, R., Yang, W. and Rahardja, S. (2011). Optimal real-time price based on a statistical demand elasticity model of electricity, *2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, pp. 90–95.

Zhang, W., Xu, Y., Liu, W., Zang, C. and Yu, H. (2015). Distributed online optimal energy management for smart grids, *IEEE Transactions on Industrial Informatics* **11**(3): 717–727.

Zhang, Y., Gatsis, N. and Giannakis, G. B. (2013). Robust energy management for microgrids with high-penetration renewables, *IEEE Transactions on Sustainable Energy* **4**(4): 944–953.

Zhao, Z., Lee, W. C., Shin, Y. and Song, K. (2013). An optimal power scheduling method for demand response in home energy management system, *IEEE Transactions on Smart Grid* **4**(3): 1391–1400.

Zhou, Y., Ci, S., Lin, N., Li, H. and Yang, Y. (2018). Distributed energy management of p2p energy sharing in energy internet based on cloud energy storage, *Proceedings of the Ninth International Conference on Future Energy Systems*, e-Energy '18, Association for Computing Machinery, New York, NY, USA, p. 173–177.

URL: *https://doi.org/10.1145/3208903.3208906*