Solution Methods for Optimal Energy System Planning with Renewable and Non-Renewable Generation

by

Semini Wijekoon

B.Eng Electrical and Computer Systems (Hons)

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2020

Caulfield School of Information Technology

Supervisors:

Assoc. Prof. Ariel Liebman

Dr. Simon Dunstall

Dr. Aldeida Aleti

To my parents,
for their constant support and encouragement

Solution Methods for Optimal Energy System Planning with Renewable and Non-Renewable Generation

Declaration

This thesis 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.

Semini Wijekoon
May 28, 2020

Acknowledgments

This thesis could not have been completed without the assistance and patience of many individuals. First of all, I would like to express my gratitude and appreciation to my supervisors for guiding me throughout this PhD journey. Their constant support in numerous capacities and constructive feedback helped immensely towards the completion of this thesis.

I gratefully thank Dr. Simon Dunstall for his invaluable ideas, constant support and enthusiasm, which inspired me to research in the optimisation domain. I would like to thank Assoc. Prof. Ariel Liebman for giving me the opportunity to study the PhD course. His support on various occasions made the duration of the PhD a smooth journey. I am also grateful to Dr. Aldeida Aleti for her attention to details which helped to improve the quality of this research project.

I would like to extend my gratitude to my panel members: Prof. Mark Wallace, Dr. Pierre Le Bodic and Dr. Ross Gawler, for their valuable feedback that helped me to shape this thesis. I am also grateful to Dr. Guido Tack and David Hemmi for providing me with feedback on the technical aspects of this thesis.

My sincere thanks go to Julie Holden for her valuable suggestions regarding the writing of this thesis. Her insights and prompt feedback was extremely helpful to improve the quality of the thesis. I would also like to thank the Graduate Research Team in the Faculty of Information Technology for managing the administrative matters and making the various compulsory processes simple and straightforward.

I am grateful to my colleagues Peter Lusis and Seyedali Meghdadi, for sharing their expertise and providing me the opportunity to have insightful discussions. My sincere thanks also go to my friends Sandith Kelaniyage and Adheesha Bandara, for being by my side throughout this journey and making my life as a PhD student more enjoyable.

And, I am extremely grateful to my parents and my sister for loving and supporting me unconditionally, without whom I will not be here today.

Semini Wijekoon

Monash University
May 2020

Solution Methods for Optimal Energy System Planning with Renewable and Non-Renewable Generation

Semini Wijekoon
`semini.wijekoon@monash.edu`
Monash University, 2020

Supervisor: Assoc. Prof. Ariel Liebman
`ariel.liebman@monash.edu`
Associate Supervisor: Dr. Simon Dunstall
`simon.dunstall@data61.csiro.au`
Associate Supervisor: Dr. Aldeida Aleti
`aldeida.aleti@monash.edu`

Abstract

In the electric power industry, energy system planning addresses the need for new capacities in generation, transmission and storage facilities. This optimisation problem has been solved sufficiently well in the past for conventional systems with dominating fossil fuel-based generation. However, the integration of renewable energy-based generation has exposed new challenges in constructing optimal long-term investment planning and short-term operational planning problems. Due to the significant electricity short-term supply variability inherent to renewable energy-based generators, operational flexibility must also be taken into account in addition to consideration of nominal generation and transmission capacities, if we are to ensure supply-demand balance. This entails consideration of a detailed operational model such as unit commitment at high resolution within a long-term optimal investment planning approach. While such details reduces overall system costs, they are also computationally demanding. Therefore, this thesis develops efficient solution methods to enable operational flexibility and incorporate unit commitment model in the planning context.

More specifically, the thesis proposes a novel decomposition framework based on an existing scenario decomposition approach to decompose the complex energy system planning with unit commitment problem and conducts an extensive analysis to identify the key factors that affect the performance. Based on the analysis, the thesis proposes two extensions: grouping and branching, to increase computational efficiency and improve the performance. In order to tackle the issue of high resolution, the thesis proposes an adaptive resolution approach named Sliding Window with Backtracking (SWBT). The thesis also quantifies the impact of incorporating unit commitment in the planning context in comparison to conventional operational models. And finally, it presents multiple test cases that were upgraded to include investment candidate options and technical parameters required by the unit commitment model.

Contents

List of Tables

List of Figures

List of Abbreviations

AC	Alternating Current
BD	Bender's Decomposition
DC	Direct Current
ED	Economic Dispatch
ESP	Energy System Planning
ESS	Energy Storage Systems
GEP	Generation Expansion Planning
GTEP	Generation and Transmission Expansion Planning
GTEP-UC	Generation and Transmission Expansion Planning with Unit Commitment
LB	Lower Bound
LP	Linear Programming
MIP	Mixed Integer Programming
SD	Scenario Decomposition
SR	Stable Roommate
SWBT	Sliding-window with Back-tracking
TEP	Transmission Expansion Planning
UB	Upper Bound
UC	Unit Commitment

Chapter 1

Introduction

1.1 Overview

Energy system planning determines the most cost-effective technology mix that should constitute the future generation mix. Although this problem has been solved sufficiently well in the past with conventional fossil fuel-based generation, integration of renewable energy-based generators (e.g. wind and solar) has exposed new challenges in the problem domain. Unlike traditional thermal generators, renewable energy-based generation is associated with inherent variability and uncertainty. Hence, to cope with these changes and deliver the desired reliability at an affordable cost, the electricity system must be operationally flexible. Operational strategies that utilise existing grid components (e.g. flexible generators, energy storage systems) are capable of delivering the desired flexibility effectively, but high shares of renewable energy-based generators are likely to require investments in flexibility provisions.

One of the major challenges in this respect is determining the required operational flexibility within the planning context. Accurate representation of operational flexibility requires modelling of detailed operational conditions at high resolution in chronological order. In addition, a large number of integer variables are required to capture the status of individual generators in each operating condition. Hence, an energy system planning problem with a detailed operational model and 20-30 year planning horizon often results in a large-scale mixed inter programming (MIP) problem that is computationally intractable even for small systems. As a result, in traditional planning models, flexibility issues are generally avoided to make the problem computationally feasible. Since ignoring flexibility issues could lead to a future generation mix that is economically inefficient, this thesis aims to develop solution methods to incorporate operational flexibility in energy system planning problems.

1.2 Energy System Planning

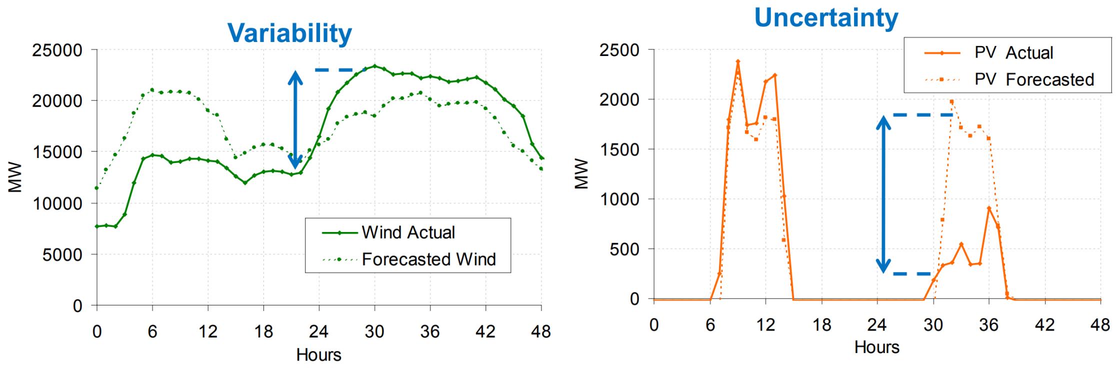
Power systems often demand new investments in generation and transmission facilities to ensure continuous, reliable operation. In the past, these investments were motivated by the necessity to expand capacity with growing demand and ageing facilities. But starting earlier this century, under the simultaneous influences of an imperative to reduce carbon emissions and cost reductions in renewable energy-based technologies, restructuring the electricity grid to achieve renewable energy and emission reduction targets became one of the major drivers for new investments. However, building new electricity provisions is a capital intensive process that requires complex financial arrangements and long payback periods [1]. Furthermore, these assets last for multiple decades spanning over 20-50 years or even longer. As a consequence, to minimise the risk of financial losses and ensure cost-effectiveness throughout their life span, rigorous planning is essential.

Energy System planning formalises this goal through careful modelling and consideration of future scenarios to determine the most cost-effective set of investments that will serve the future system requirements. A typical energy system planning problem considers a long horizon of 20-50 years and makes investment decisions in steps of 1-5 years across several stages to determine suitable time periods for new investments in generation and transmission facilities [1]. Thus, an energy system planning problem is often formulated as a large-scale optimisation problem with a large number of generation and transmission units, multiple periods and other characteristics.

1.3 Variability vs Uncertainty

Variability and uncertainty are two closely related concepts that have gained significant attention in recent decades with the transition towards renewable energy-based generation. Unlike traditional thermal generators, renewable energy-based generators strongly depend on meteorological conditions such as wind speed and solar radiation. Hence, their maximum available generation is subject to rapid fluctuations and hard to predict [2]. These two characteristics: variability and uncertainty inherent to renewable energy-based generators are depicted in Fig. 1.1a and 1.1b respectively [2]. Fig. 1.1a illustrates the rapid fluctuations in wind generation that occur within a short period of time. Fig. 1.1b illustrates the uncertainty in solar generation which shows a significant difference between the predicted and the actual solar power output.

To comply with these variations and uncertainty and ensure instantaneous generation-load balance, other resources such as thermal generators and/or demand must be altered [2]. For example, if the power production from solar units is lower than the expected value, or the generation from a wind unit decreases its output rapidly, the deficiency in generation can be mitigated through dispatchable thermal generators that are capable of increasing their generation output at the same rate and/or by altering the demand itself. Thus, by ensuring that the electricity system is operationally flexible, both variability and short-term uncertainty can be mitigated to ensure supply-demand balance at all times.



(a) Variability of a wind plant

(b) Uncertainty of a solar plant

Figure 1.1: Illustration of variability and uncertainty

1.4 Operational Flexibility and Role of Unit Commitment

Operational flexibility can be defined as the power systems' ability to withstand uncertainty and variability in generation output and electricity demand while maintaining desired reliability at an affordable cost [3]. In current practice, flexibility is provided through multiple operational strategies, which are carried out mostly from the generation aspect and fewer from the demand aspect [4].

1. Operational strategies from the generation side includes flexible units, storage systems, reserve capacity, transmission network, renewable curtailment and electric vehicles.
2. Operational strategies from the demand side includes load shedding and demand management.

On the generation side, flexible or fast dispatchable generators such as hydroelectric and natural gas-fired units are designed to ramp up and down quickly with short on and off periods [5]. They are capable of responding quickly to variations in renewable generation and demand, to maintain short-term supply-demand balance. As a result, fast dispatchable generators often act as the reserved capacity to ensure that the electricity grid has an adequate number of generating units scheduled, to respond and supply generation in sudden events such as an increase in predicted demand or reduction in renewable energy-based generation [6].

In contrast, renewable energy-based generators can be controlled only through curtailment, i.e. by deliberately reducing the generation to lower levels [5]. When renewable generation is too volatile for thermal generators to follow, the outputs of renewable units are reduced to lower levels than the maximum available generation. This inefficiency can be mitigated through the installation of energy storage systems such as battery systems and pumped hydro generators. By storing the excess generation produced by renewable sources during off-peak times, storage systems can minimise the energy wastage. Transmission lines provide flexibility by transporting electricity from neighbouring zones. Strategic deployment of transmission lines provides additional flexibility without installing extra generators and storage units [7]. Such connections also allow the system to benefit from diversity in geographically dispersed renewable energy-based generators. Another potential flexibility provider is electric vehicles. Although electric vehicles are considered as growing demand, their ability to discharge the stored energy when required (given some incentives to the participants) has been recognised as a flexibility service that could be provided by the vehicle to the grid (vehicle-to-grid services) in the future [8].

On the demand side, the most common approach to flatten the peak-load is load shedding through direct intervention. Although heavy penalties apply for such actions, in many cases load shedding is inevitable to safeguard the operation of the grid. Alternatively, demand-side management that focuses on modifying consumers' demand can be employed. By lowering the price during off-peak time and/or raising the price during peak periods, customers can be incentivised to change their daily consumption patterns [9]. Also, participants can be called upon to modify their power consumption when there are security issues (e.g. failures) or for economic reasons (e.g. utility cost exceeds specific pre-defined limit), where they are financially rewarded [5].

Although these operational strategies are capable of delivering desired flexibility effectively, high shares of renewable energy-based generators such as wind and solar are likely to require new investments in flexibility provisions. However, determining the required operational flexibility increases the computational complexity of the problem. Since modelling flexibility entails a detailed operational model and chronological demand and renewable generation profiles with temporal dependencies, the planning problem becomes computationally intensive even for small systems. As a consequence, operational flexibility was ignored in traditional planning problems, and economic dispatch (ED) that has limited ability to determine the required operational flexibility was embedded. Economic dispatch only considers instantaneous operational states and a brief set of generator constraints to determine the least-cost output of the generation fleet, so its ability to address time-dependent variability is limited.

With large shares of renewable energy-based generation being integrated and recent studies reporting that neglecting operational flexibility could lead to inefficient systems, sophisticated operational models such as unit commitment started gaining attention in the planning context since early this decade [10,11]. Unit commitment (UC) is typically used for day-ahead scheduling and intra-day operations to determine the most cost-effective set of generators that could serve the demand [12, 13]. It takes time-dependent variability in demand and renewable generation output into account and captures the impacts of generator's technical limitations such as ramping limits that characterise the flexibility of the system. Because unit commitment is capable of representing operational flexibility accurately, it also qualifies the benefits that storage facilities and demand management could provide to the system [14,15]. As a result, unit commitment plays a significant role in representing operational flexibility in planning and operational simulations and ensuring supply-demand balance.

1.5 Motivation

Incorporating unit commitment in planning problems is computationally challenging due to the large size of the problem. To accurately represent operational flexibility, the short-term operational conditions must be captured at high resolution in chronological order. Variations in renewable generation, especially wind occurs at multiple timescales from seconds to minutes to hours. Hence, a high resolution is essential to capture quick ramping events and determine the required flexibility. As a result, the computational time of a planning problem with unit commitment model explodes quickly even for small systems. For example, if one-hour resolution is considered for a period of one year, 8760 hours must be considered, where the set of decisions variables (generators and transmission lines) is replicated for each hour. Also, it is customary to make investment decisions in multiple periods to determine when to construct in the 20-50 year planning horizon. Such formulation enlarges the problem further, as operational conditions of other periods also must be taken into account. As a result, solving energy system planning problem with unit commitment directly at full resolution (8760 hours) is computationally challenging.

In practice, large-scale optimisation problems are solved via two approaches: heuristics and exact methods based on decomposition techniques. Heuristic methods generally utilise meta-heuristics to determine the best solution. They do not guarantee the globally optimal solution, rather provide a locally optimal solution. On the other hand, exact methods based on decomposition techniques are capable of providing the optimal solution, but rely on specific problem structures. For a planning problem with unit commitment, application of conventional decomposition is limited. Unlike economic dispatch, unit commitment has a large number of binary variables (to determine investment decisions and on-off status of generators) and inter-temporal constraints (to represent generator's ability to change its output e.g. ramp up-down and minimum up-down constraints). Because the problem comprises integer decisions in both investment and operational levels, and operational conditions are linked via inter-temporal constraints, its complicated structure does not comply easily with traditional decomposition techniques. As a consequence, incorporating unit commitment in the energy system planning context still remain a computational challenge. Especially, in the application of exact methods to guarantee the optimal solution.

1.6 Research Questions

Therefore, this thesis aims to answer the following overarching question.

How can we incorporate limitations in operational flexibility, such as unit commitment, in energy system planning problems, where we seek optimal or near-optimal solutions?

To answer the overarching question, the following sub-questions were formed.

RQ1 How can we address the high temporal resolution in unit commitment model efficiently to reduce the computational burden of the overall problem? *How accurate is the solution? How much computational time can we reduce? What are the limitations?*

RQ2 How can we decompose the large-scale MIP problem with integer decisions in both investment and operational levels and inter-temporal constraints, to solve the problem within a reasonable period of time? *What are the limitations? Can we extend the framework for multi-period decision making?*

RQ3 Can we make the decomposition framework computationally more efficient? *What extensions do we require? What are the limitations?*

1.7 Thesis Structure

The organisation of the thesis to answer the research questions as follows.

- Chapter 2 provides a comprehensive review on current practices in modelling and solving energy system planning problems. In particular, the chapter reviews existing techniques utilised to incorporate unit commitment in the planning context.
- Chapter 3 provides the detailed model for the unit commitment problem and investigates its impact on planning problems compared to the traditionally used economic dispatch model. The analysis aims to quantify the quality of the investment solutions obtained from approximated operational models with respect to the exact unit commitment.
- Chapter 4 proposes an adaptive resolution approach to answer RQ1 and aims to mitigate the computational burden resulting from high temporal resolution in the unit commitment model.
- Chapter 5 proposes a decomposition framework based on a scenario decomposition approach for both single and multi-period energy system planning problems with unit commitment. The chapter answers RQ2, which addresses the need for an efficient decomposition framework.
- Chapter 6 proposes two extensions: grouping and branching to answer RQ3, which aims to improve the performance of the proposed decomposition framework.
- Finally, Chapter 7 concludes the thesis by detailing the key contributions and discussing potential future work.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a comprehensive review on current practices in modelling and solving energy system planning problems. The chapter first provides a brief overview of the energy system planning problem and introduces the two key aspects: generation expansion planning (GEP) and transmission expansion planning (TEP). Then, the chapter discusses the solving techniques that are utilised to solve large scale energy system planning problems. In particular, the chapter discusses two methods: heuristics based on simulations, and decomposition techniques based on mathematical programming. Lastly, the chapter reviews the tools and techniques that have been employed to incorporate unit commitment in the planning context and discusses their limitations and the requirement for new approaches.

2.2 Energy System Planning as an Optimisation Problem

In the electric power industry, energy system planning is the problem of determining the most cost-effective set of investments that is necessary to satisfy future demand and other system requirements. This problem was first discussed in the late 1960s [16] and was primarily formed around determining the required capacity in generation facilities. Early energy system planning problems were formulated as deterministic or probabilistic production cost models [17,18], and were solved using graphical methods such as screening curves [19]. Screening methods examine the generation cost models of different candidate options (technologies), and then match the overall least-cost model with the load duration curve [1] graphically to determine the least cost generation mix [21]. By minimising the capital and operational costs of generators implicitly, these methods provided the optimal capacity mix required to serve the increasing electricity demand [22].

However, load duration curves utilised in screening methods disregard the time dependencies in operations and spatial correlations between demand and generation zones. As

---

[1] Load duration curve is the curve between load and time, in which load or net-load curves are estimated by ordering load (demand) from the greatest to the lowest considering a small number of criteria (e.g. base-load, peak demand) [20]

a result, the method not only lacks the ability to consider various operating conditions and dynamics in demand and non-dispatchable units (e.g. wind and solar), but also the ability to account for electricity transmission network. Nevertheless, the screening curve method was sufficient at the time due to the fairly slow dynamics involved in historic loads patterns and power was generated from controllable fossil-fuel based generators. But, with rising complexity in current power systems and recent developments in computer hardware and software, optimisation models based on mathematical programming techniques have become the standard for energy system planning problems.

In the early 1970's, Bessière [23] summarised the main features of the optimisation model by defining the objective function and constraints. A deterministic linear programming (LP) model was then introduced by Anderson [24], which was later extended to mixed integer linear programming (MILP) problems to account for transmission grid physics and ratings [25] and availability of power plants (require binary decisions variables) [26]. Since then, planning problems are usually formulated as MIP optimisation problems and have evolved to consider multiple aspects (e.g. environmental) and many operational conditions (e.g. unit failures) to determine the most cost-effective set of generators and transmission lines that satisfies reliability standards and other criteria (e.g. renewable energy targets).

Nevertheless, in practice, generation and transmission aspects of the electricity planning problem are carried out separately to reduce the computational intensity. First, the generation expansion planning (GEP) is executed to determine the most cost-effective generation mix subject to system requirements [5], and then transmission expansion planning (TEP) is executed to find the network augmentations for the agreed generation mix [27].

2.2.1 Generation Expansion Planning (GEP)

Generation expansion planning (GEP) addresses the adequacy in current generation capacities and determines whether it is necessary to expand existing generating units and/or build entirely new facilities [1]. Essentially, GEP determines the size of the generating unit, technology, location and when to construct in the planning horizon [5]. A typical GEP problem is formulated with one of the two types of objectives: centralised or market oriented. While centralised objectives such as minimising total cost or maximising social-welfare is generally used by regulators and policy makers to design electricity grid expansion and derive new policies (e.g. renewable energy targets, carbon tax, subsidy regimes), market oriented goals such as maximising expected revenue is employed by competitive private investors to make informed decisions regarding their investments [28].

Apart from the investment objective, a GEP problem considers multiple operational conditions to capture system states and determine capacity requirements accurately. These conditions are generally represented by embedding economic dispatch (ED) with hourly resolution as the operational model [29, 30]. Economic dispatch is an optimisation model that considers instantaneous operational states to determine the least-cost output of the generation fleet that satisfies system demand and other operational requirements [31]. Unlike production cost models based on load duration curves [17, 18], with ED, network constraints can be incorporated easily through an additional set of constraints. In addition, ED allows extra operational states to be integrated as another set of variables and constraints. Thus, for many long-term planning models and operational simulations ED serve as the base operational model [29, 30].

Economic dispatch is often combined with stochastic programming and robust optimisation techniques to address the short-term uncertainties in demand and renewable sources. As ED allows multiple operational states to be integrated easily, uncertainties in input parameters are represented through different operational conditions (e.g. low demand high renewable scenario, high renewable low demand scenario). In stochastic programming, a weighted objective that represents the expected cost or revenue across all the uncertain scenarios is optimised, in which the weights represent the probabilities of operational conditions [30, 32]. In contrast, robust optimisation does not rely on any probability distributions. Instead, it optimises over an uncertainty set to determine the worst possible scenario and its optimal investment solution [29].

Although ED is capable of handling short-term uncertainties through stochastic programming and robust optimisation techniques, its ability to address variability and operational flexibility is limited. Economic dispatch ignores temporal dependencies and only considers instantaneous operational states. Thus, it does not account for any time-dependent variability, and also excludes limitations in thermal generators like ramping and minimum up-down constraints that characterise the operational flexibility of a system. While the effect of omitting these constraints is negligible in a system that is dominated by thermal generators, such elimination could lead to unreliable systems with increased penetrations of renewable energy-based generation.

Therefore, recent GEP problems utilise various forms of ED model to capture the required operation flexibility. To address operational flexibility through fast dispatchable units, ED models include ramping requirements as an uncertainty set [33]. For demand management, additional constraints are added to coordinate the demand at different locations [34]. In addition, iterative investment and operational simulations are utilised to consider detailed operational models with ramping requirements and demand management [35, 36]. Similar procedures are also followed to address the functionality of energy storage systems (ESS), where constraints that represent charging and discharging limits are added to the conventional ED model [37, 38]. These constraints also allow GEP to be combined with the expansion of storage systems (SEP) to reduce the investments required in peak/flexible generators by placing storage systems strategically [39, 40].

While extended ED models are capable of capturing operational flexibility approximately, they have limited ability to address temporal dependencies and technical limitations in thermal generators. Thus, more sophisticated operational models such as unit commitment are utilised to ensure supply-demand balance at any given instance [3, 6, 10, 41, 42]. Unit commitment (UC) is typically used for day-ahead scheduling and takes time-dependent variability in demand and renewable generation output into account through chronological profiles with hourly resolution. It is capable of capturing generator's technical limitations such as maximum capacity and ramping limits that characterise the flexibility of the generator and system accurately [3,10], and also emphasises the benefits storage facility and demand management could provide to the system [14,15]. Furthermore, UC allows GEP to be combined with aspects like spinning reserve [2], which plays a significant role in the provision of operational flexibility [6].

2.2.2 Transmission Expansion Planning (TEP)

Transmission expansion planning (TEP) addresses bottlenecks in the electricity network and determines whether to reinforce existing lines or build completely new connections changing the network topology [44]. The cost of transmission expansion is relatively low compared to the cost of generation facilities, however its impact on electricity provision is significant. In a deregulated environment, the need for transmission expansion arises mainly due to the price differences in different nodes [45](apart from the need to connect remotely located renewable energy-based generators). Since transmission congestion prevents flow of cheap electricity from generation areas to demand nodes [1], increasing transmission capacity could provide a non-discriminatory and competitive environment for the stakeholders. Therefore, a typical TEP problem is formulated from a centralised perspective with objectives such as minimisation of market risk [46], nodal prices [47], load curtailment cost [48], transmission congestion cost [49] and maximisation of social welfare [50,51].

For an accurate representation of the electricity network, TEP must consider the physics of electricity flow through AC [52,53], or DC [54] power flow equations. AC power flow models the electricity flow accurately by including both active and reactive power, voltage stability and power losses. However, solving AC power flow is computationally expensive due to the non-linear and non-convex terms in the formulation [55]. DC power flow approximation is more attractive, as it effectively removes the non-linear terms in the AC power flow equations through expert knowledge, and only accounts for active power flow. With some adjustments for line losses, DC model produces a reasonably accurate approximation to AC model under normal operating conditions [56]. As a consequence, capacity expansion planning problems generally use DC approximation [29], while network operation and control problems utilise AC models due to its great deal of accuracy [57]. Nevertheless, continuous efforts are currently being made to integrate AC model with planning problems to provide robust and feasible solutions with less computational effort and to extend its application to large systems (e.g. via linearisation [55,58] and convex relaxation [53,59]).

In addition to the thermal limits enforced by AC or DC power flow equations, a typical TEP problem often includes additional features such as line losses [58], security constraints [60] and uncertainty [61] to model the reliability and stability of the solution. Network losses account for power losses in the transmission network, and security constraints impose

---

[2] Spinning reserve is a reliability standard practised by system operators to ensure that the electricity grid has an adequate number of generating units scheduled, to respond instantly and restore stability in the event loss of a heavily loaded generating unit [43]. Since ”spinning” indicates that the units are already “on-line” and ready to serve, unit commitment plays a significant role in determining which units should be committed to serve as reserve capacity.

limits to avoid overloading lines even during a transmission line failure [43]. Furthermore, with increasing shares of excess renewable energy-based generation, recent TEP models are integrated with storage expansion (SEP) to reduce the investments in transmission capacities by storing the surplus and providing it later or vice versa [7, 62].

2.2.3 Generation and Transmission Expansion Planning (GTEP)

With the necessity to integrate renewable energy-based generators, combined generation and transmission expansion planning (GTEP) started appearing to provide a more holistic investment solution that is beneficial for the entire system [1, 63]. Since the proportion of prospective renewable zones are located distant from the existing transmission network and this technological transition causes a relocation in generation capacity, consideration of GEP along with network augmentations ensures that the infrastructure reinforcements are cost-effective. Such synergy is only beneficial for an entity that aims to improve the operation of the electricity grid as a whole. Thus, the GTEP problem is generally carried out by a central planner such as independent system operator, and formulated with the goal of either minimising total cost or maximising social welfare [64, 65].

Given the long-term nature of electricity assets, it is a common practice to carry out GTEP problems considering a planning horizon of 20-50 years and make investment decisions at the beginning of the horizon (single-period formulation) or in steps of 1-5 years (multi-period formulation). Single-period formulation could force investments at the beginning of the horizon that are not required until the end of the planning duration leading to overcapacity. Hence, for more realistic and accurate solutions, multi-period formulations that make investment decisions in steps of 1-5 years are commonly used. If formulated as a dynamic model, such formulation also allows uncertainty in long-term trends (e.g. changes in aggregated demand, capital investment cost and fuel prices [66]) to be considered for a more robust investment solution [39]. As consideration of a long planning horizon often can be computationally expensive, planning models with a representative year and annualised capital cost are also utilised to make investment decisions [67]. Such models do not specify when to construct new units in the horizon. But, they indicate which technologies must be included in the future generation mix.

GTEP problem often takes a range of aspects into account. Some of these aspects can be listed as environmental (e.g. carbon emissions [68]), regulatory (e.g. renewable energy targets [69]), technical/operational (e.g. storage [70], security [71], reliability [72], flexibility [10]), social and as well as other complementary sectors (e.g. natural gas [73,74]) [5]. Incorporation of all these aspects increases the complexity of the model and the solving process. Hence, GTEP is extensively used to study the impact of specific features either individually [10, 75] or in combination [76, 77]. In addition, multi-objective approaches are also utilised to solve GTEP problems, particularly problems with conflicting attributes such as cost minimisation and carbon emission reduction [78]. Nevertheless, a typical GTEP problem often results in a complex high-dimensional optimisation problem, because of the long planning horizon, an enormous number of model parameters and discrete variables [79].

2.3 Solving ESP as a Large-Scale Optimisation Problem

Solving capacity expansion planning problems have been a difficult and challenging task since the beginning of power system planning, due to the large size of the electricity network and the excessive number of operating conditions it must consider to ensure a cost-effective reliable solution. Direct solving of such a large-scale optimisation model often require a substantial amount of computational memory and processing power. Besides, the significant number of integer variables and non-linearity in the model further complicates the solving process. Hence, these problems are generally solved iteratively to reduce the computational burden and achieve tractability [80]. Over the past few decades mainly two types of iterative approaches have been employed to solve large-scale energy system planning models; heuristic methods based on simulations, and decomposition techniques based on mathematical programming [80].

2.3.1 Heuristic Methods

Heuristic methods utilise rules based on expertise or meta-heuristics to determine the optimal solution. Given an initial solution, the method simulates, and then employs the generated response to derive the next trial solution. Fig. 2.1 [81] illustrates this mechanism where meta-heuristic optimiser derive the trial investment solution to be assessed by the simulation (operational) model. The simulation model then generates the response to be utilised by the meta-heuristic optimiser for the next trial solution. Hence, heuristic methods are applicable to wide class of problems including non-linear and non-convex problems, as they do not depend on any problem structure [79]. Some commonly applied meta-heuristics in large-scale energy system planning problems are expert systems, artificial neural networks, simulated annealing, particle swarm optimisation and evolutionary algorithms [82–85]. Nevertheless, heuristic methods generally have slow convergence rates and do not guarantee the global optimal solution. Instead, they provide feasible solutions that are locally optimal.

2.3.2 Decomposition Techniques

Decomposition techniques divide the optimisation problem into a set of sub-problems and solve them iteratively to obtain the best solution for the overall problem [80]. Unlike heuristic methods, decomposition techniques are guaranteed to converge to the optimal solution within a finite time [80]. However, their application heavily relies on the problem structure, thus limited to only certain types of problems [79]. Nevertheless, with recent

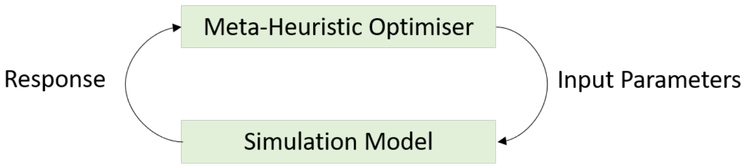


Figure 2.1: Schematic representation of heuristic methods

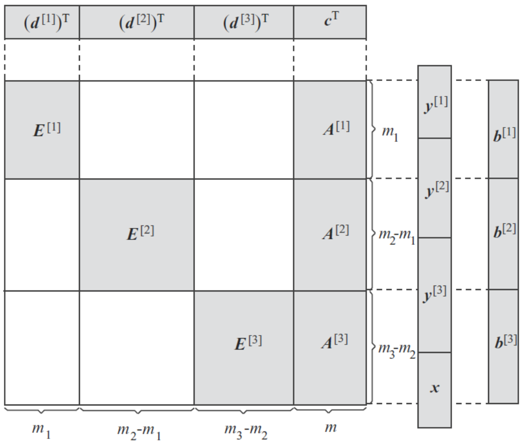


Figure 2.2: Complicating variables A

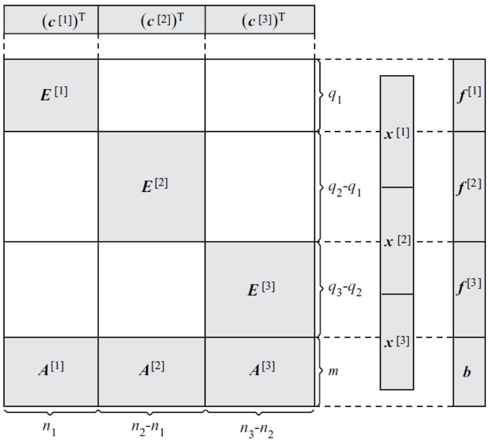


Figure 2.3: Complicating constraints A

advancements in computer hardware and off-the-shelf MIP solvers (e.g. Cplex, Gurobi etc.), decomposition techniques are applied widely to attain tractability in large-scale optimisation problems [86].

Large-scale optimisation problems often posses a block structure that allows them to be divided into multiple sub-problems and solved separately to find the overall optimal solution. In general, these blocks contain information pertaining to other blocks through binding variables and constraints. They are often referred to as complicating variables and constraints, and must be tackled if one desire to decompose the problem. Figures 2.2 and 2.3 [86] illustrates these two concepts, where Fig. 2.2 depicts the complicating variable set A and Fig. 2.3 presents the complicating constraints set A. In Fig. 2.2, if complicating variable set A is fixed, rest of the problem (set E) reduces to a block structure enabling multiple sub-problems. Similarly, if complicating constraints in Fig. 2.3 is removed, the problem (set E) reduces to a block structure forming multiple sub-problems [86].

Decomposition methods manipulate these complicating variables and constraints and exploit the block structure to break the large optimisation problem while bringing information from one sub-problem to another. Some conventional decomposition techniques that have been applied in power system optimisation context are Bender's, Dantzig-Wolfe and Lagrangian where the key differences lie in the method used to pass information across sub-problems.

Bender's Decomposition

Bender's decomposition (BD) was originally proposed by J.F. Benders in 1962 [87] to tackle the issue of complicating variables [86]. By extracting the complicating variables and their associated constraints into a separate problem, Benders decomposition divide the problem into two sub-problems: master and slave. While the master problem consists of complicating variables, the rest of the problem in which the complicating variables are fixed becomes the slave problem. In some instances, the remaining problem is further decomposed to form multiple slave problems. Then the master and slave problems are solved iteratively generating bounds and exchanging information until the optimality gap

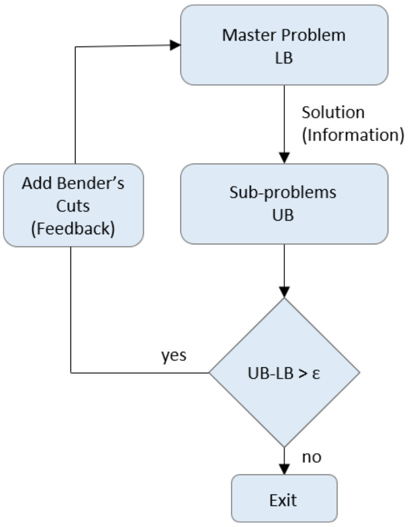


Figure 2.4: Schematic representation of Bender's decomposition

is less than a pre-defined value. The schematic diagram for the Bender's decomposition is presented in Fig. 2.4. The master problem generates a lower bound and a solution to be evaluated by the slave problems. Through evaluation, slave sub-problems generate an upper bound, and provide feedback to the master problem by adding a constraint (row). This constraint is commonly termed as "Bender's cut" and is derived from the dual information obtained from the slave problem. In Bender's decomposition, for the slave problems to extract dual information, the sub-problem must be an LP problem. Thus, with MIP formulations, integer variables are generally extracted as the master problem to ensure a fully LP sub-problem.

In power system context, Bender's decomposition was first applied by J.A. Bloom in 1983 to solve a generation capacity expansion planning problem [88]. Since then, an enormous number of studies have utilised Bender's decomposition to attain tractability in both deterministic and uncertain generation and transmission expansion planning models [89–92]. In MIP generation expansion planning with economic dispatch problems, integer investment decisions are generally extracted as the master problem. Thus, the master problem often becomes the investment planning problem while the slave sub-problems become the operational economic dispatch formulation. Since economic dispatch has no inter-temporal constraints, different operating conditions are further divided into multiple sub-problems [29, 93].

Dantzig-Wolfe Decomposition

Dantzig-Wolfe decomposition [94] decomposes the problems with complicating constraints. By relaxing the complicating constraints in the slave-problems and accounting for them in the master problem through a weighted objective, Dantzig-Wolfe decomposition solves the problem iteratively until the optimality gap is less than a pre-defined value. The schematic diagram for the Dantzig-Wolfe decomposition is depicted in Fig. 2.5. The LP master problem with the weighted objective is solved to obtain upper bound and dual values, which are used to derive cost coefficients for the slave sub-problems. The

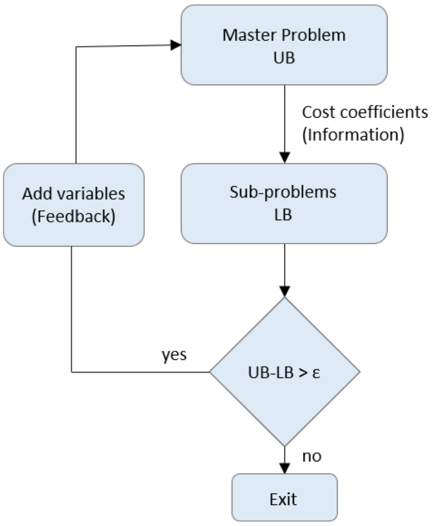


Figure 2.5: Schematic representation of Dantzig-Wolfe decomposition

sub-problems are then solved to generate the lower bound, and the outcomes are utilised to update the weighted objective in the master problem [86]. In particular, a variable (column) is added to the master objective function, where the coefficient represents the solutions generated by the sub-problems in the previous iteration. Thus, the master objective function gets updated in each iteration to represent the original problem more accurately [86]. In the energy system planning context, Dantzig-Wolfe decomposition [94] was employed by Sanghvi and Shavel [95] to solve a generation expansion planning with demand-side investments. In addition, Singh *et al.* [96] utilised the method to solve multistage, stochastic generation expansion planning models.

Lagrangian Decomposition

Lagrangian decomposition or Lagrangian relaxation also handles complicating constraints in the optimisation problem [86]. The method separates the problem by relaxing the complicating constraints and consider them in the objective function through penalty factors. The penalty factors, also known as Lagrangian multipliers penalise the violation of the relaxed constraints and get updated in each iteration according to solutions of the sub-problems [97]. Thus, unlike Dantzig-Wolfe decomposition which adds variables in each iteration, Lagrangian decomposition updates the Lagrangian multipliers to represent the problem better. In practice, several methods are utilised to update the Lagrangian multipliers. Some common approaches can be listed as subgradient method, cutting plane method, bundle method and so on [86]. The difficulty with Lagrangian decomposition is that the computational performance of the method is dependent on the initial estimate of the Lagrangian multipliers and the method that is used to update them [97]. Nevertheless, Lagrangian relaxation has been utilised in power systems context to solve unit commitment problems [97–100], and generation and transmission expansion planning problems [101–103].

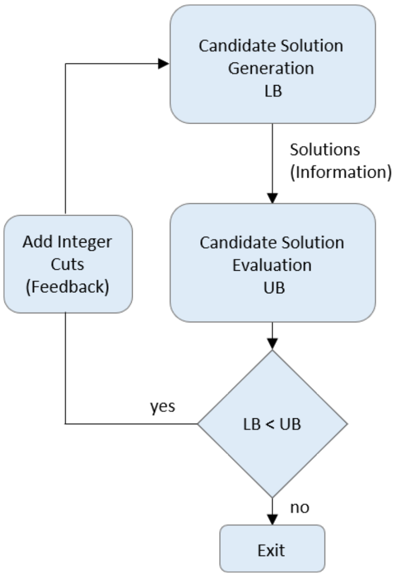

Candidate Solution Generation LB
Solutions (Information)
Candidate Solution Evaluation UB
Add Integer Cuts (Feedback)
yes
LB < UB
no
Exit

Figure 2.6: Schematic representation of scenario decomposition

Scenario Decomposition

Scenario decomposition (SD) was proposed by S. Ahmed in 2013 [104] for two-stage stochastic programs that involve binary first stage decisions. In two stage stochastic programs, the goal is to find first stage decisions that satisfy uncertainties in input parameters represented through various scenarios. Such a problem is often formulated as individual scenarios by making copies of first-stage decisions and forcing constraints to ensure consistency across different scenarios. Scenario decomposition is derived from this scenario formulation, where the scenario sub-problems are solved separately to generate a set of solutions. These solutions are then evaluated against all the scenarios to check feasibility, and eliminated from the search space via constraints to enforce consistency.

The schematic diagram for scenario decomposition is presented in Fig. 2.6, which is consisted of two phases: (1) Candidate solution generation; and (2) Candidate solution evaluation. In the candidate solution generation phase, the scenario sub-problems are solved separately to generate the set of first stage decisions and a lower bound. These binary decisions are then explored in the candidate solution evaluation phase as candidate solutions to the overall problem. Once evaluated, the best bound (lowest) is selected as the upper bound and the best-known solution or the "incumbent". The explored solutions are then eliminated from the feasible search space using "integer cuts" to raise the lower bound and close the optimality gap [104].

Scenario decomposition approach can be viewed as an evaluate-and-cut scheme with a master algorithm that coordinates the information across scenario sub-problems rather than a master problem. Unlike other decomposition techniques, this method does not depend on dual information, thus makes it ideal for a wide class of problems including those which comprise integer variables in both investment and operational levels and non-linearity. In addition, components of the algorithm have very little dependency on each other enabling them to be solved simultaneously [105].

In power system context, scenario decomposition has not been applied to capacity expansion planning problems yet. However, recent literature has employed scenario decomposition to optimise maintenance and operations schedules of generators with unexpected failures [106] and deployment of mobile emergency generator as a resilient response to natural disasters [107].

2.4 Solving Energy System Planning with UC

Unit Commitment in planning context started gaining attention with the requirement of representing operational flexibility accurately to ensure adequacy in flexibility measures. Unlike traditional planning problems with simple economic dispatch models, consideration of unit commitment constitutes an additional layer of complexity. Since short-term operational conditions need to be captured at high resolution (at least one hour) to represent operational flexibility sufficiently, the number variables in the planning problem explodes quickly even for small systems. Besides, a significant proportion of these variables are discrete, binary to be exact, so they often result in enormous branch and bound trees increasing the computational burden. In addition, the necessity to represent the operational conditions chronologically further complicates the problem, because it enforces a large number of inter-temporal constraints increasing the inter-dependencies between variables. As a consequence, recent literature has utilised various tools and techniques including both heuristics and decomposition methods to facilitate energy system planning with unit commitment.

Integration of unit commitment with the energy system planing was fist proposed by Palmintier and Webster [10] as a way of incorporating operational flexibility to generation expansion planning problems. In particular, to successfully incorporate the full resolution of one year (8760 hours) while preserving tractability of the problem, they proposed an unit clustering approach that bundles generators based on the technical parameters (e.g. capacity, heat rate, ramp rate, etc.) [10]. In the method, generators with similar characteristics are grouped together to replace the binary commitment decisions of individual generators with an integer commitment decision that represent the entire cluster [10]. As a result, the method is capable of reducing the number of integer variables and the size of the problem by a large factor. Although the reduced problem size makes the approach computationally attractive for large-scale applications, it suffers from serious drawbacks. Since clustering limits the ability to determine operational costs of individual generators, it induces approximation errors in overall operational costs causing sub-optimality in investment solutions. Hence, if one desire to improve the applicability of the proposed approach, clustering must be carried out in a way that minimises the overall approximation error. However, according to Hua *et al.* [6], finding an optimal clustering strategy that minimises the error itself is a difficult combinatorial optimisation problem. On the other hand, clustering has limited benefits with consideration of the transmission network. If similar units are not co-located clustering will return many clusters making the approach ineffective. Nevertheless, due to the computationally attractive nature, unit clustering approach has been employed by multiple studies to investigate various aspects of operational flexibility [108–110].

In contrast, to reduce the computational burden and overcome the issues related to clustered unit commitment, Ma *et al.* [3] considered an integrated generation expansion planning and unit commitment (GEP-UC) model with a selected set of representative weeks. The study chose four representative weeks from each season to capture the seasonal effect and another week to represent extreme conditions. In addition, an 'offline' index was proposed to estimate the technical ability of the generation mix and provide required flexibility [3]. Similar approach was followed by a number of other studies with various model features to reduce the computational intensity in their studies [11, 41, 111, 112]. Pereira *et al.* [111] chose a representative day (24 hours) from each season to make the GEP model with unit commitment, hydro and wind units tractable. With the extensions of transmission expansion planning, renewable curtailment and pumped hydro, Sharan *et al.* [11] selected representative days to solve the problem within a reasonable time. On the other hand, Bruninx *et al.* [112] employed representative days to examine the impact of Germany's nuclear phase-out on Europe's power generation mix. While these studies have focused on single-period planning problems, Koltsaklis *et al.* [41] utilised representative days from each month to attain tractability in their comprehensive multi-period GEP-UC model. The model includes detailed operational limitations including start-up type, minimum up-down time, ramping limits and system reserve requirements with energy policy issues such as renewables penetration limits and carbon emissions to determine the optimal capacity additions, electricity market-clearing prices, and daily operational planning of the studied power system [41]. In contrast, Jin *et al.* [113] employed scenario reduction techniques to truncate the set of samples and reduce the size of the problem, in their stochastic GEP-UC that addresses both variability and uncertainty in wind-energy based generation. On the other hand, Shortt *et al.* [114] utilised time series with high resolution (30 minutes) but shortened horizon (1 month) to quantify the impact of electric vehicles on capacity expansion planning decisions, relative fuel costs and electric vehicle penetrations. Although representative days, reduced number of scenarios and shortened horizons make the planning problem with unit commitment computationally tractable, they could lead to highly inaccurate solutions with either over-investment or under-investment, due to imprecise representation of renewable energy-based generation and electricity demand.

An alternative approach that has been utilised to reduce the computational intensity without decreasing the spatial or temporal resolution is through approximated or relaxed model. Palmintier [115] employed a linear relaxation of unit commitment model to identify key constraints that capture operational flexibility. In the approach, to make the GEP-UC problem fully linear and tractable, the binary commitment decisions were relaxed to be continuous. In addition, the linear programming (LP) relaxation was compared against the unit clustering method, according to which LP relaxation provides larger errors compared to the unit clustering approach. To reduce the error arising from LP relaxation, representative days and clustered unit commitment, Hua *et al.* [6] proposed a promising approach based on convex relaxation of unit commitment. Similar to LP relaxation, the method relaxes the binary commitment decisions to be continuous however, extra inequalities are added to define the convex hull [6]. In fact, the embedded operational model in the planning problem is the Lagrangian dual of the unit commitment problem. The proposed approach is also compared against the unit clustering approach, where the authors show that the convex relaxation approach outperforms the unit clustering approach in terms of accuracy [6]. Nevertheless, both linearly and convexly relaxed unit commitment models are still approximations. And, they do not quantify the quality of resulting investment solution with respect to the solution produced by the exact unit commitment model, as solving exact unit commitment in planning context is still computationally intractable.

The commercial software PLEXOS manages computational complexity by allowing their users to relax unit commitment model through linear programming relaxation. If further reductions are required, PLEXOS enable their users to employ approximated chronological curves with load blocks that are coarser in temporal resolution [116]. While coarse resolutions limit the ability to capture rapid variations in wind output, linear programming relaxation incur significant approximation errors. Nonetheless, by utilising capabilities of PLEXOS and other commercial software such as TIMES, Welsch *et al.* [117] developed a detailed model via soft-linking to incorporate operational flexibility. In particular, they examined the effects of linking a long-term energy system model (TIMES) with a unit commitment and dispatch model (PLEXOS) for the Irish system. In the TIMES model, day, night and peak times of a single characteristic day were modelled for each of the four seasons over the period 2005–2020 using 12 time slices. PLEXOS, on the other hand, was set up as a chronological hourly unit commitment model. Soft-linking was implemented by feeding the yearly (peak) demand and power plant capacity mixes produced by TIMES model into PLEXOS. PLEXOS then assess the overall operational reliability and technical abilities of the produced capacity mix.

In contrast to previously discussed methods that are based on mathematical relaxations, Hargreaves *et al.* [118] outlined a novel approximated model that considers economic trade-offs and addresses cost implications of adding flexibility resources to a system. They adapted a production simulation methodology that weighs the cost of reliability and sub-hourly flexibility violations, against the cost of available operational flexibility solutions, to develop a stochastic production simulation model, known as Renewable Energy Flexibility (REFLEX). Essentially, the model tracks system load distribution, dispatch, generation, outage and ramping conditions using historical data and a security-constrained unit commitment model. Instead of simulating the entire year, the proposed method utilises a set of randomly generated scenarios (representative days) while preserving seasonal and meteorological correlations. This model is commercialised under the name REFLEX and has been implemented on commercial production simulation platforms including ProMaxLT and PLEXOS to improve scalability and run times in large systems, and as well as to incorporate more advanced modelling options.

Another long-term energy mix optimisation model called “emix” was proposed by Wierzbowski *et al.* [119] to incorporate flexibility in a simplified manner. The proposed model was derived from the conventional GTEP with UC and takes daily requirements of power system operation (primary, secondary and tertiary reserve) and reaction to increasing renewable penetration into account. Unlike the previously discussed unit clustering approach, this model allows representing individual operation of each generating asset in a computationally effective manner. However, both “emix” [119] and REFLEX [118] are based on simplified unit commitment simulations, and they do not provide any bound or quality of the solution with respect to the exact unit commitment solution. So, they can be deemed only as heuristics.

Furthermore, a novel approach based on perturbation was proposed by Belderbos and Delarue [120] to account for operational flexibility in power system planning. The method obtains two initial solutions; one from a classical screening curve model, and another from a model using mixed integer linear programming (MILP), and then perturb the initial solutions while evaluating them using an operational UC model to validate and improve the solution. In particular, they utilise MILP planning model with a set of representative days to obtain the initial solution and then achieve the perturbation by first removing units in the estimated set of generators to check any cost reductions and then adding units to improve the solution [120]. While the method has been able to reduce the complexity of the problem by a large factor compared to other models that serve the same purpose, the approach may not converge to a global optimum. Since the initial solutions were generated by approximated methods like screening curve and MILP with representative days, and the perturbations are carried out until no further reductions in total costs, the perturbation algorithm is likely to converge to a local optimum.

Another technique that is commonly applied to large-scale optimisation problems, and can be applied to solve GTEP-UC problems within a reasonable period of time is decomposition. Kamalinia *et al.* [121] proposed a classic Bender’s decomposition approach to tackle the computational burden in a stochastic GEP-UC model. However, for simplicity, the model excluded the start-up costs and minimum up and downtime constraints that characterise the flexibility of thermal units. Since sub-problems must be linear in Bender’s decomposition to obtain dual information, the method extracted all the integer variables such as unit commitment states to the master problem. To reduce the size of the resulting master problem, only a reduced set of fast response gas turbines was considered. In addition, the number of scenarios that were employed to model uncertainty was also reduced through the application of scenario reduction techniques.

While the classic Bender’s formulation generates only one cut per iteration aggregating dual information obtained from all the sub-problems, in the event of multiple sub-problems, a cut can be generated and added from each sub-problem. Such Bender’s formulation is referred to as multi-cut Bender’s, and generally outperforms the single-cut Bender’s, because it strengthens the master problem more quickly and prevents the loss of information due to aggregation [122]. Recently, a multi-cut Bender’s decomposition was implemented by Schwele *et al.* [123] to solve a network-constrained stochastic GEP-UC model. They utilised two-stage stochastic programming to consider uncertainty in wind power production and followed a structure similar to Kamalinia *et al.* [121] to decompose the problem. The long-term investment and short-term unit commitment decisions were solved in the master problem as first stage decisions, while short-term dispatch decisions that accommodate wind deviation were solved in the sub-problems as second stage recourse decisions. Each scenario was considered as a separate sub-problem, where a scenario represents a day (24 hours). Similar to Kamalinia *et al.* [121], the approach sufferers from a huge master problem, thus minimum up-down constraints were ignored to reduce the complexity.

To overcome the issue of massive master problem with Bender's decomposition, Angela *et al.* [124] proposed an approach based on Dantzig–Wolfe decomposition and column generation to solve computationally intensive multi-period GEP-UC problem. As Dantzig-Wolfe decomposition allows sub-problems to be mixed-integer (MIP) in contrast to Benders decomposition, unit commitment is incorporated directly as the sub-problem. However, to obtain the dual information required for cost coefficients, the master problem is formulated as a linear model relaxing the integrality in investment decisions. If the master problem returns fractional values for investment decisions at the end of the algorithm convergence, a branch and price procedure is followed to obtain integer solutions. In addition, for simplicity, network constraints were excluded and the year was represented using 13 weeks (one from each month and an extreme scenario), where K-Mediods clustering technique was employed to determine the representative weeks. They also utilised other concepts and methods from existing literature such as unit clustering approach [10], tight formulations of the unit commitment problem and classical screening curve method (for the initial point of column generation approach) to reduce the computational burden further. In addition, they demonstrated that the proposed approach based on Dantzig–Wolfe decomposition outperforms direct application of commercial solvers, and significantly reduces both computational time and memory usage.

In this regard, the key methodologies applied to incorporate the unit commitment model in energy system planning problems with reasonable fidelity and solving time can be summarised as follow.

Table 2.1: Key methodologies for capacity expansion planning with unit commitment

Reduction in input data	**Approximated operational models**	**Decomposition techniques**
Temporal Scenario reduction techniques [113] Shortened planning horizon [114] Representative days [3, 11, 41, 111, 112, 125] Coarser resolution [116, 125] Clustering techniques [124, 126]	**Mathematical** Tight formulations of UC [124] LP relaxation [115, 116] Convex relaxation [6]	**Mathematical** Bender's [121, 123] Dantzig-Wolfe [124]
Spatial Aggregate similar units [10, 108–110]	**Based on expertise** REFLEX [118] emix [119]	**Other** Soft-linking PLEXOS and Times [117]

2.5 Summary

Energy system planning problems are primarily two types: generation expansion planning (GEP) and transmission expansion planning (TEP). While GEP addresses the adequacy in power generation, TEP addresses the necessary expansions in the power transmission network. These problems are generally formulated with the objective of either total cost minimisation or profit maximisation considering a number of operating conditions. Economic dispatch is the most commonly used operational model. But, with the integration of renewable energy-based generation, the sophisticated unit commitment model has started to gain attention in the planning problem. However, solving planning problems with unit commitment is computationally expensive because of the large size and details in the operational model.

In general two types of methods are used to solve large scale optimisation problems: heuristics based on simulations and decomposition methods based on mathematical programming. While heuristic methods do not guarantee the global optimal solution, application of decomposition techniques relies on specific problem structure. In particular, for planning with unit commitment, Bender's decomposition is limited in terms of the details that can be considered because of the huge master problem. And, Dantzig-Wolfe decomposition is integrated with other existing techniques (e.g. unit clustering, representative days, screening curve methods) limiting the ability to understand the sole effect of the decomposition approach.

When comparing other key methodologies that have been utilised to incorporate unit commitment in planning problems, representative days provide highly inaccurate solutions because of the imprecise representation of renewable energy-based generation and demand states leading to either over-investment or under-investment. Coarse temporal resolutions are also undesirable as rapid variations in wind output may not be captured sufficiently well leading to imbalances in supply and demand. And, unit clustering approach limits the ability to consider individual status of generators and the transmission network. On the other hand, simplification of unit commitment model via convex/linear programming relaxation and other novel techniques can be deemed only as heuristics, as they are still approximations and do not quantify the quality of the investment solution. In this regard, it is apparent that a complete approach to incorporate unit commitment in planning problems at full resolution is still limited. Therefore, in this thesis, we aim to address this gap and develop solution methods to facilitate unit commitment in the planning context while mitigating the computational challenges.

Chapter 3

Unit Commitment, Formulation and Impact

3.1 Introduction

Although unit commitment plays a significant role in capturing operational flexibility, exact unit commitment at full resolution is often avoided in planning problems due to computational limitations. Instead, conventional planning problems utilise economic dispatch and its variants, which either ignore flexibility constraints completely or consider them approximately. However, the quality of the resulting solution with respect to the solution from the exact unit commitment is often not addressed, because a method that is capable of solving planning problems with exact unit commitment at full resolution is still unavailable.

Therefore, by employing the decomposition framework proposed in Chapter 5, this chapter aims to investigate the impact of incorporating exact unit commitment in the planning context with respect to the traditionally used economic dispatch model and its variants. This chapter first introduces the unit commitment model by providing the objective function and constraints. The subsequent sections of the chapter provide the details of other operational models: economic dispatch and economic dispatch with ramping, and solutions methods utilised to solve the large-scale optimisation problem. Through experimental analysis, the chapter then discusses the impact of incorporating unit commitment in planning problems, and finally concludes emphasising the need for the unit commitment model. This investigation with exact unit commitment at full resolution and economic dispatch constitutes the key contribution of this chapter.

3.2 Unit Commitment (UC)

In power systems operation, the role of unit commitment is to schedule the most cost-effective set of generators that meets the system requirements. For an independent system operator (ISO), these requirements include satisfying the electricity demand with minimum cost or maximum welfare subject to system constraints. Generator companies (GENCOs) on the other hand have minimum desire to satisfy demand as their main focus is to maximise revenue subject to generator limitations. Hence, a typical UC problem is formulated from two perspectives. ISOs use UC in a centralised environment with total cost minimisation to clear day ahead markets and conduct intra-day operations, while GENCOs use UC in a self-scheduling environment with profit maximisation to derive bidding profiles [12, 13].

The need for scheduling is mainly driven by the technical limitations in dispatchable generators. A scheduled or “committed” generator implies that the unit is “turned on”, synchronised to the electricity grid and is available to deliver power. Once committed these generators are required to maintain a minimum generation to “remain on”, which can be fairly expensive and economically inefficient, if the total generation in the system is already exceeding the demand. Switching off (de-commit) units too frequently is also disadvantageous due to techno-economic issues such as re-starting cost, ramping limits, wear and tear etc. As a consequence, the unit commitment problem was formulated to control the on-off states of generators and determine the optimal generator schedule.

This section presents the MIP formulation for UC problem based on the three-binary model. Essentially, the three-binary model indicates that three binary variables are used to capture the three states of the generator, i.e, commitment, start-up and shut down. Although more compact formulations based on one-binary variable (for commitment decision only) are available [127], multiple studies have reported that they hinder the ability to generate strong valid inequalities for constraints that involve start-up and shut down status (e.g start-up cost, ramping, minimum up downtimes) [12, 128]. Therefore, the UC formulation based on three-binaries can be presented as follows [129].

Objective Function

The objective of the UC problem Eq. 3.1 is formulated from the centralised perspective as minimisation of total operation cost, i.e. the sum of production cost, commitment cost, start-up cost, shut-down cost and penalty cost for unsatisfied demand.

$$\min \sum_{t\in\mathcal{T}} \left(\sum_{g\in\mathcal{G}} (C^{var*}_{t,g}(p_{t,g}) + C^c_g u_{t,g} + C^u_g v_{t,g} + C^d_g w_{t,g}) + C^p q_t \right) \tag{3.1}$$

where: t, g are the indices for time periods and generators respectively while $\mathcal{T}, \mathcal{G}$ are the corresponding sets; $C^{var*}(p_{t,g})$ represents the variable cost of generation which is a function of production $p_{t,g}$; C^c_g, C^u_g, C^d_g are commitment, start-up and shut-down costs; C^p is the penalty cost for unserved demand; u, v, w are the binary commitment, start-up and shut-down variables and finally p, q denotes continuous variables for power generation and unserved demand.

The production cost function $C^{var*}(p_{t,g})$ accounts for the cost of power generation and depends on various factors such as generator heat rate, the generator dispatch level and fuel prices. As shown in Fig. 3.1, the function is quadratic in nature. Thus, it is often modelled as a quadratic function or as a set of piece-wise linear functions at the expense of extra constraints. In planning problems, one linear function is also utilised for simplicity [6, 92].

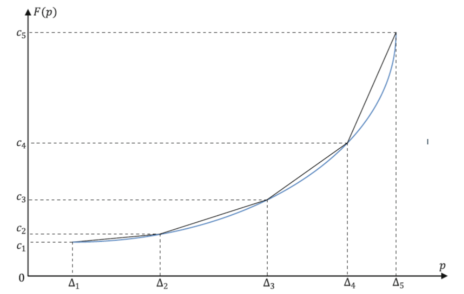


Figure 3.1: Production cost function

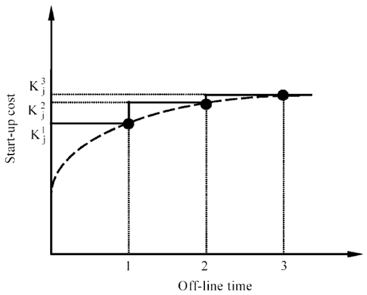


Figure 3.2: Start-up cost function

Start-up cost accounts for the energy that is not reflected in the power generation p but must be expended to bring the unit on-line. As shown in Fig. 3.2 [127] this cost varies depending on the duration that the unit has been "off-line". If the unit has been switched off for a longer period "cold-start" cost is considered, which is much expensive than the "warm-start" cost that applies when the unit was turned off recently and still relatively close to the operational temperatures [43].

Commitment cost or no-load cost refers to the fixed cost that incurs for being just "on-line" even at zero generation (no-load). While the current formulation does not allow the generation to be zero if "on-line" (Eq. 3.7), this is allowed and necessary when the UC formulation is extended to consider spinning reserves.

Penalty cost for unserved demand accounts for loss loads and ensures reliability in the system. In operational simulations, reliability is captured either through a global constraint that includes unserved demand in all time periods or as part of the objective function. When formulated as a constraint, an index such as LOLP (loss of load probability), EENS (expected energy not served) or LOLE (loss of load expectation) is used to constrain unserved demand in the overall system. Alternatively, a penalty cost can be added to the objective function to minimise the loss load. In this formulation, the latter option is utilised, where unserved demand is heavily penalised in the objective function. Such formulation permits time-based decomposition easily, as time periods are not bound by a global constraint.

Constraints

The constraints in unit commitment represents the system requirements and generator limitations. As the list can be exhaustively long with different systems imposing specific rules, the most common set of constraints are presented here.

$$u_{g,t} - u_{g,t-1} = v_{g,t} - w_{g,t} \qquad \forall g, t \tag{3.2}$$

Eq. 3.2 captures the start-up and shut-down status of the generator using two consecutive commitment decisions.[1] These states are used by start-up shut down costs, minimum generation, ramping limits and minimum up down limits to generate strong inequalities.

---

[1] Although three binaries are utilised in this model, due to the linearity in this constraint, it is only necessary to enforce integrality on two of the three decision variables.

$$\sum_{i=t-T_g^U+1}^{t} v_{g,i} \leqslant u_{g,t} \qquad \forall g,t \tag{3.3}$$

$$\sum_{i=t-T_g^D+1}^{t} w_{g,i} \leqslant 1 - u_{g,t} \qquad \forall g,t \tag{3.4}$$

Frequent switching on and off of thermal generators leads to issues such as unnecessary start-up costs, wear and tear or simply impossible due to technical limitations. Hence, on-off times must be controlled forcing the generators to be “on” for a certain period of time if committed and “off” if de-committed. This is governed by Eq. 3.3 and 3.4 where T^U and T^D denote minimum up and downtimes respectively.

$$p_{g,t} - p_{g,t-1} \leqslant R_g^U u_{g,t-1} + S_g^U v_{g,t} \qquad \forall g,t \tag{3.5}$$

$$p_{g,t-1} - p_{g,t} \leqslant R_g^D u_{g,t} + S_g^D w_{g,t} \qquad \forall g,t \tag{3.6}$$

While the generators can adjust their output (ramping) to follow the load profile, at any two consecutive time periods the difference must be within a certain limit. This limit depends on whether the generator is changing its output while remaining on (ramping limits), the generator is starting up (start-up limits) or preparing to shut-down (shut-down limits). Eq. 3.5 and 3.6 impose these ramping up and down constraints where, $R_g^{u/d}$, $S_g^{u/d}$ are the ramp-up, ramp-down limits and start-up shut-down limits respectively.

$$P_g^{min} u_{g,t} \leqslant p_{g,t} \leqslant P_g^{max} u_{g,t} \qquad \forall g,t \tag{3.7}$$

At a given time t, a committed generator must operate below its maximum generation limit and above its minimum generation. Eq. 3.7 bounds the power generation p to be within these limits or zero otherwise.

$$\sum_{g \in \mathcal{G}} p_{g,t} + q_t = D_t \qquad \forall t \tag{3.8}$$

Finally, Eq. 3.8 represents the instantaneous supply-demand balance, where D indicates nodal demand and q denotes unserved demand. For reliability purposes, unserved demand q is penalised in the objective function Eq. 3.1.

3.3 Impact of Unit Commitment in Planning Problems

As previously mentioned, the majority of the operational models embedded in planning problems either disregard operational flexibility completely, or estimate it through approximated operational models. However, these approximations do not specify the optimality or quality of the solution with respect to the exact unit commitment solution, because a complete approach that could deal with the dimensionality of the problem is unavailable.

To analyse the impact of operational flexibility in the planning context and quantify the bound gap caused by approximated flexibility constraints, this section aims to solve the planning problem with multiple operational models that contain different levels of details regarding the operational flexibility. In particular, this section considers economic dispatch (ED) that is traditionally embedded in planning problems, then economic dispatch with ramping (ED+R) that considers operational flexibility partially with ramping constraints and finally exact unit commitment with binary commitment decisions (UC) that represents the required operational flexibility accurately. In addition, decomposition techniques are utilised to decompose the problem into smaller sub-problems and reduce the computational complexity. Since ED and ED+R are fully linear, planning models with ED and ED+R are decomposed using Bender's Decomposition. As exact unit commitment involves binary commitment decisions, planning problem with exact UC is decomposed using scenario decomposition framework. This approach in detail can be provided as follow.

3.3.1 Problem Formulation

The mathematical formulations for the planning problem with three operational models can be provided as follow.

Energy System Planning

Given a set of candidate options, future demand and cost trends, and operational conditions such as demand and renewable generation profiles, energy system planning problems co-optimise the investment and operational cost to determine the most cost-effective investment solution. While planning problems are generally formulated to consider a long planning horizon, in this formulation, a representative year with an annualised capital cost is considered. Such formulation does not specify when to invest in the planning horizon but indicates which investments and technologies should be part of the future generation mix. Thus, planning problem co-optimises the annualised investment cost (C^I) and operational cost (C_{op}) represented by either ED, ED+R or UC for a period of one year at hourly resolution (8760 hrs).

$$\begin{aligned} &\min f = \sum_{i \in \mathcal{I}} C_i^I x_i + \sum_{t \in \mathcal{T}} C_t^{op} \\ &st. \\ &\quad \text{Operational Constraints} \end{aligned} \tag{3.9}$$

where $\mathcal{I}, \mathcal{T}$ are the set of candidate units and time periods, and x is the binary investment decision. For existing units, this variable is pre-defined to be 1. Moreover, the investment cost was annualised by using Eq. 3.10, where, C^T is the total capital cost, γ is the discount rate and LS is the lifespan of the unit in years.

$$C^I = \frac{\gamma C^T}{1-(1+\gamma)^{LS}} \tag{3.10}$$

Unit Commitment (UC)

The unit commitment formulation with investment decisions can be updated as follow.

$$\min \sum_{t\in\mathcal{T}}(\sum_{g\in\mathcal{G}}(C^{var*}_{t,g}(p_{t,g}) + C^c_g u_{t,g} + C^u_g v_{t,g} + C^d_g w_{t,g}) + C^p q_t)$$

$$\begin{aligned} &s.t. \\ &\quad u_{t,g} \leqslant x_g \qquad\qquad \forall g,t \qquad (3.11) \\ &\quad Eq.\ 3.2 - Eq.\ 3.8 \end{aligned}$$

where, Eq. 3.11 limits the commitment to installed thermal units.

Economic Dispatch (ED)

Economic dispatch (ED) aims to find the least cost generation mix that meets the demand subject to generator limitations. Unlike UC, ED does not schedule generators. Instead, it simply assumes that all the generators provided are "on-line". Hence, ED avoids the necessity to use binary variables and is typically formed as a linear programming (LP) model.

$$\min f = \sum_{t\in\mathcal{T}}(\sum_{g\in\mathcal{G}} C^{var*}_{t,g}(p_{t,g}) + C^p q_t) \tag{3.12}$$

$$s.t$$

$$\sum_{g\in\mathcal{G}} p_{g,t} + q_t = D_t \qquad \forall t \tag{3.13}$$

$$0 \leqslant p_{g,t} \leqslant P^{max}_g x_g \qquad \forall g \in \mathcal{G}, t \tag{3.14}$$

In ED, the objective Eq. 3.12 minimises the generation and penalty cost, subject to maximum generation limits (Eq. 3.14) and supply-demand balance (Eq. 3.13). Similar to UC, $C^{var*}_{t,g}(p_{t,g})$ can be either quadratic, piece-wise linear or linear. As one may notice, start-up costs, shut-down costs and commitment costs are excluded. In addition, P^{min} is set to zero, as a non-zero P^{min} without a binary commitment decision could force the generator to be "on" continuously leading to unnecessary generation. Furthermore, ED does not account for ramping limits and minimum up-down times, due to the omission of binary commitment decisions.

Economic Dispatch with Ramping (ED+R)

In addition to the basic economic dispatch formulation presented in Eq. 3.12 - Eq. 3.14, economic dispatch with ramping considers operational flexibility approximately through Eq. 3.15 and 3.16. Eq. 3.15 is the ramp-up constraint and Eq. 3.16 is the ramp-down constraint.

$$p_{g,t} - p_{g,t-1} \leqslant R_g^U \qquad \forall g, t \tag{3.15}$$

$$p_{g,t-1} - p_{g,t} \leqslant R_g^D \qquad \forall g, t \tag{3.16}$$

DC Power flow

As generator dispatch models such as economic dispatch (ED) and unit commitment (UC) do not account for any limitations in the electricity network, they are generally coupled with power flow calculations to provide feasible solutions, which could have been infeasible otherwise due to the aggregated demand and power generation. This formulation utilises DC approximation that accounts for active power flow to model the electricity flow.

$$F_l^{min} x_l \leqslant f_{t,l} \leqslant F_l^{max} x_l \qquad \forall l, t \tag{3.17}$$

$$f_{l,t} = B^{MVA} B_l x_l (\theta_{s,t} - \theta_{r,t}) \qquad \forall l, t \tag{3.18}$$

$$-\theta^{max} \leqslant \theta_{n,t} \leqslant \theta^{max} \qquad \forall n, t \tag{3.19}$$

$$\theta_{n,t} = 0 \qquad n : slack \qquad \forall t \tag{3.20}$$

In the formulation, f_l and θ_n represent power flow in line l and phase angle in node n; x_l indicates the existing status of the line; B_l, B^{MVA} refers to susceptance (per unit) and Base MVA (used for per unit conversions); F^{max}, F^{min} $(-F^{max})$ indicate maximum and minimum power flow limits; $\theta^{max}, \theta^{min}$ denote maximum and minimum phase angle limits. With that, Eq. 3.17 accounts for thermal limits (overloading) in transmission lines which is bounded by the existing status; Eq. 3.18 restricts the power flow according to DC approximation; Eq. 3.19 enforces phase angle limits to ensure stability in frequency[2] and finally, Eq. 3.20 sets the phase angle of the reference bus to zero. Since DC approximation contains a non-linear expression in Eq. 3.18, it is often linearised using the big M method as provided in Eq. 3.21 and Eq. 3.22, where M represents a large constant[3].

$$-(1 - x_l)M \leqslant f_{l,t} - B_l(\theta_{s,t} - \theta_{r,t}) \qquad \forall l, t \tag{3.21}$$

$$(1 - x_l)M \geqslant f_{l,t} - B_l(\theta_{s,t} - \theta_{r,t}) \qquad \forall l, t \tag{3.22}$$

---

[2] In some occasions, instead of the phase angles (angle of the voltage), the angle difference between the nodes that connect the active transmission line is bounded by pi/2 to achieve steady-state stability.

[3] A value just above the maximum F^{max}

Renewable units

Generation output from renewable energy-based units are generally considered as either fixed or semi-dispatchable. In the fixed form, the output is represented through net load by deducting it from the nodal demand. In the case of semi-dispatchable generation, a variable is assigned to control the output of each unit according to a maximum available generation that depends on wind speed or solar radiation. This model considers renewable energy-based generators as semi-dispatchable units, where generation curtailment is allowed without a penalty.

$$0 \leqslant p_{r,t} \leqslant P^{max}_{r,t} x_r \qquad \forall r, t \qquad (3.23)$$

where r is the index for renewable unit and P^{max} is the maximum available generation.

Nodal Power Balance

In this regard, power balance constraint can be updated as follows to incorporate transmission lines and renewable energy-based generators, where r and s denote the receiving and sending nodes of transmission lines.

$$\sum_{g \in \mathcal{G}(n)} p_{g,t} + \sum_{r \in \mathcal{R}(n)} p_{r,t} + \sum_{l \in L | r(l) = n} f_{t,l} - \sum_{l \in L | s(l) = n} f_{t,l} + \quad q_{n,t} = \quad D_{t,n} \qquad \forall n, t \qquad (3.24)$$

The summary of operational models with their ability to represent operational flexibility is given below.

Table 3.1: Summary of operational model features

Feature	ED	ED+R	UC
Start-up cost	no	no	yes
Minimum generation	no	no	yes
Ramping limits	no	approx.	yes
Min up down	no	no	yes

3.3.2 Solution Method

As solving planning problems with hourly resolution for a period of one year is computationally expensive, in this section decomposition techniques are utilised to reduce the computational time. More specifically, two decomposition techniques are employed, where Bender's decomposition [87] is utilised to decompose planning models with ED and ED+R and scenario decomposition [104] is utilised to decompose the planning problem with exact unit commitment. Since ED and ED+R are fully linear and investment decisions are the only integer variables, using Bender's decomposition the planning problem can be decomposed easily into master problem (investment) and sub-problems (economic dispatch). In contrast, exact unit commitment contains integer variables in the operational level limiting the ability to apply Bender's decomposition. Hence, for these problems, scenario decomposition technique is utilised to mitigate the computational burden. In addition, to analyse the actual operational cost of solutions realised by ED and ED+ R, a unit commitment simulation was conducted. Details of these methods are provided in subsequent sections.

Bender's Decomposition (BD)

As described in Section 2.3.2, Bender's decomposition iterates between a master and set of sub-problems, and adds a constraint that is based on dual information at the end of each iteration. For the Bender's approach to extract dual information, the sub-problem must be a fully linear problem. As a result, all the integer variables and their corresponding constraints are generally extracted to master problem. In this formulation, investment decisions are the only integer variables, so the investment problem becomes the master problem while ED and ED+R become the slave sub-problems. In addition, ED and ED+R are further divided considering each day as one sub-problem and a Bender's cut is added from each sub-problem resulting a multi-cut Bender's approach. The master, slave problems and Bender's algorithm are provided below.

Master problem - Investment decisions

$$\min f^{mas} = \sum_{i \in \mathcal{I}} C_i^I x_i + \sum_{s \in \mathcal{S}} \alpha_s \tag{3.25}$$

$st.$

$$\alpha_s \geqslant 0 \qquad \forall s \tag{3.26}$$

$$\alpha_s \geqslant f_s^{sub(j)*} + \sum_{i \in I} \lambda_{s,i}^{(j)} (x_i - x_i^{(j)}) \qquad \forall s, m \geqslant 2, j = 1....m-1 \tag{3.27}$$

where, C_i^I is the annualised investment cost, x_i is the investment decision, s is the index for sub-problem, $f_s^{sub(j)*}$ are the sub-problem optimal objective values in previous iterations, λ_i^j denotes the dual values, x_i^j is the investment decision in the previous iteration and m is the current iteration. In addition, Eq. 3.27 represents the Bender's cut added to the master problem based on the solution of the sub-problem in the previous iteration.

Sub-problem - Operational decisions

The sub-problems are either ED or ED+R, where in addition to the constraints described previously, Eq. 3.28 is added to capture the dual prices of the investment decisions, in which $x^{(m)}$ is the investment decision obtained from the master problem in the current iteration m. In addition, ED+R consists inter-temporal constraints linking one sub-problem to another. Hence, to ensure that operational decisions are consistent and aligned with the next day, operational status at the end of the day is passed on to the next day as the initial status. In addition, an overlap duration of one day is considered at the end of each sub-problem, and then operational cost and dual prices for the overlapped duration are discarded.

$$x_i = x_i^{(m)} \quad : \lambda_i \qquad \forall i \in I \tag{3.28}$$

Bender's Decomposition Algorithm

1. Initialise with the iteration count $m = 1$
2. Solve the Master problem defined by Eq. 3.25 - Eq. 3.27, to obtain the investment decision $x^{(m)}$ and the lower bound LB, where $LB = f^{mas*}$. For the first iteration, solve the master problem without Eq. 3.27.
3. Feed the investment decisions and solve the sub-problems $s \in \mathcal{S}$ sequentially to obtain the operational cost f_s^{sub*} and the dual values λ.
4. Calculate the upper bound UB where,
   $UB = \sum_{i \in \mathcal{I}} C_i^I x_i + \sum_{s \in \mathcal{S}} f_s^{sub*}$
5. If $|UB - LB| \leqslant \epsilon$, where ϵ is the Bender's gap, exit with the current solution. Else, update the iteration count to $m = m + 1$, and go to step 2.

Scenario Decomposition (SD)

In scenario decomposition, the planning problem with unit commitment is decomposed into multiple planning sub-problems with a shorter planning horizon such as a day with daily capital costs. Assuming that they are independent, each sub-problem is then solved separately to generate a set of candidate investment solutions, where the summation of individual costs provide the lower bound (LB). Then, the generated candidate investment solutions are evaluated by solving the unit commitment model for the entire planning horizon. This yields a series of upper bounds, and the solution with the lowest total system cost becomes the incumbent solution and upper bound (UB). The algorithm then discards all the evaluated solutions from the search space by means of an integer cut, forcing candidate solution generation phase to generate a more expensive set of solutions in the next iteration. Thus, the algorithm converges when LB exceeds UB proving optimality. This approach is discussed elaborately in Chapter 5.

Unit Commitment Simulation

Ideally, to examine the true operating cost of a given investment solution, the unit commitment model must be solved for the entire planning horizon. Solving unit commitment problem for one year (8760 hours) is computationally tedious, because of the enormous number of binary variables. Therefore, in this analysis commonly applied rolling horizon technique is utilised to conduct the unit commitment simulation [6]. In the rolling horizon approach, daily unit commitment problems are solved sequentially forcing operational status at the end of one day as the initial status for the next day. In addition, an overlap duration is considered at the end of each day to ensure intra-day consistency, and the

Table 3.2: A summary of test cases

Feature	6-Bus	14-Bus	18-Bus	24-Bus
Number of nodes (N)	6	14	18	24
Number of thermal generators (G)*	7	11	13	18
Number of renewable generators (R)*	4	6	8	10
Number of lines (L)*	10	27	31	40

* The provided values include both candidate and existing units

Table 3.3: Model attributes for the planning problem with unit commitment

Model attributes	Nodes	Thermal generators	Renewable generators	Transmission lines
Number of integer variables	0	G(1 + 3*T)	R	L
Number of continuous variables	2*N*T	G*T	R*T	L*T
Number of equality constraints	T(1+N)	G*T	0	0
Number of inequality constraints	4*N*T	7*G*T	2*R*T	4*L*T

where T is the number of time periods, N is the number of nodes, G is the number of thermal generators, R is the number of renewable energy-based generators and L is the number of transmission lines.

operational cost and decisions for the overlap duration are discarded. As commitment decisions have a finite duration that they continue to affect future system behaviour, if the overlap duration is large enough accurate results can be obtained [6,130]. In practice, this overlap duration is determined by the generator specifications, the longest minimum up and downtime to be precise, which is typically 24 hours [6,130].

3.4 Experimental Analysis

To investigate the impact of incorporating operational flexibility in capacity expansion problems, four test cases: 6-bus, 14-bus, 18-bus and 24-bus systems were utilised with hourly chronological load and renewable generation profiles obtained from AEMO (Australian Energy Market Operator). The cost parameters and details of the test cases are provided in Appendix A and a summary of test cases is provided in Table 3.2. In addition, a high renewable scenario, where existing renewable energy-based generators supply around 50% of the total demand was considered for each test case, as variability becomes significant with high shares of renewable energy-based generators. Moreover, for all test cases, a planning period of one year was considered. Furthermore, Table 3.3 provides the attributes of the model for planning problem with unit commitment, which briefly indicates the size of the problem given the system parameters and the number of time periods.

All algorithms and models were implemented and solved using Python 2.7 and Gurobi 8.0 [131] on Monash Cluster utilising eight cores clocking at 2.70GHz with a RAM memory of 16 GB. In addition, Gurobi's default MIP gap 1×10^{-4} and a Bender's gap of 1×10^{-4} were utilised, where the thread count was set to number of cores (only MIP models). In addition, a time limit of 22 hours was assigned to all algorithms for convergence and the dual simplex method was chosen to obtain the dual values. Furthermore, the penalty cost of unserved demand was set to 1000\$/MWh.

Table 3.4: Nominal

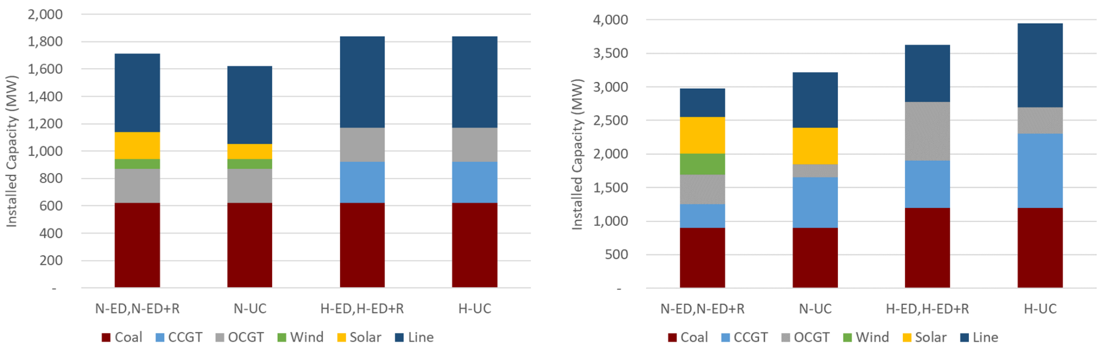
Test Case	Model	Gap%	C^I (M$)	Obj.(M$)	Δ%
6 Bus	ED	0	46.59	153.95	1.624
	ED+R	0	46.59	154.00	1.588
	UC	0	46.59	156.49	NA
14 Bus	ED	0	187.80	637.00	1.199
	ED+R	0	187.80	637.06	1.190
	UC	0	178.22	644.73	NA
18 Bus	ED	0	222.01	613.71	0.400
	ED+R	0	222.01	613.85	0.378
	UC	1.20	222.01	616.17	NA
24 Bus	ED	0	415.30	1608.65	0.530
	ED+R	0	415.30	1608.67	0.528
	UC	2.85	389.54	1617.22	NA

Table 3.5: High renewable

Test Case	Model	Gap%	C^I(M$)	Obj.(M$)	Δ%
6 Bus	ED	0	50.53	103.99	8.193
	ED+R	0	50.53	104.70	7.566
	UC	0	50.53	113.27	NA
14 Bus	ED	0	203.01	369.80	7.016
	ED+R	0	203.01	370.80	6.764
	UC	0.82	203.01	397.70	NA
18 Bus	ED	0	167.50	406.89	5.267
	ED+R	0	167.50	408.24	4.953
	UC	1.24	167.50	429.52	NA
24 Bus	ED	0	449.01	1121.19	7.079
	ED+R	0	449.01	1127.29	6.569
	UC	5.43	469.28	1206.55	NA

Table 3.4 and 3.5 illustrate the obtained results for nominal and high renewable instances, where gap indicates the optimality gap at the end of the maximum time limit, C^I denotes the investment cost in M$, Obj. represents the current best objective in M$ and finally, Δ% is the cost difference with respect to the total cost provided by the exact unit commitment solution. According to Tables 3.4 and 3.5, nominal instances show reasonably small cost differences across different operational models however, high renewable instances show considerably large differences. This performance is expected as ED and ED+R overestimate the generator capabilities causing the power to be generated at a lower cost than the actual, which becomes significant in high renewable instances due to the increased ramping requirements. Besides, with ED and ED+R start-up costs and cost for minimum generation are ignored.

For ED models, these differences can be as high as 8%, and decrease slightly with ED+R when ramping constraints are brought into the model. This reveals that ramping constraints only have a slight impact on the planning problem, while the significant differences are caused by start-up costs, minimum generation and minimum up-down times. Although, larger differences are observed in high renewable instances, interestingly, investment solutions are consistent across different operational models. On the other hand, nominal instances that have provided relatively smaller cost differences exhibit inconsistency in investment solutions, especially at 14-bus and 24-bus instances. Thus, it can



(a) 14 Bus system

(b) 24 Bus system

Figure 3.3: Installed capacity

be observed that apart from the changes in cost, the underlying problem structure also impacts the optimality of the investment solutions when flexibility constraints are avoided or approximated.

These changes in investment solutions are driven by either overestimation of operational flexibility (leads to under-investment) or underestimation of operational cost (leads to over-investment). In 14-bus nominal instance, exact UC eliminated a solar unit that was installed by ED and ED+R. This reduction in installed capacity is visible in Fig. 3.3a which illustrates the installed capacity in 14-bus system for both nominal (N) and high renewable (H) instances. Similar results can be observed in 24-bus nominal instance, where exact UC replaced an OCGT (open cycle gas turbine) unit installed by ED and ED+R with a CCGT (combined cycle gas turbine) unit that has more capacity. Also, UC eliminated a wind unit installed by ED and ED+R and invested in extra transmission line capacity. This increase in installed capacity is depicted in Fig. 3.3b, which illustrates the installed capacities for 24-bus nominal (N) and high renewable instances (H).

In order to evaluate the investment solutions provided by ED and ED+R and to determine their true operational costs, unit commitment simulation was carried out. Through the simulation, it was observed that investment solutions provided by ED and ED+R for these instances yield higher total costs compared to the solutions provided by the exact UC. These results are presented in Table 3.6. Although the percentage cost difference (Δ^{UC}) between true cost of solutions provided by ED and ED+R and true cost of solutions obtained from exact UC are relatively small, these results prove that approximated operational models like ED and ED+R provide sub-optimal solutions.

In addition, for 24 bus high renewable instance, exact unit commitment has provided slightly more expensive investment solution (-0.473% in Table 3.6). This is because of the large optimality gap (5.43%) at the end of the maximum time limit in scenario decomposition algorithm, which indicates that high-quality solutions may have not been discovered yet. Moreover, the consistency in investment solutions across operational models in other instances suggest that their existing capacities are adequate to deal with the drastic ramping requirements and/or underestimation of operational cost is not significant enough to drive a different investment solution. Thus, it is reasonable to deduce that significance and impact of operational flexibility is specific to the characteristics of a given system.

Table 3.6: Unit commitment simulation

Test Case	Model	UC Sim.(M$)	$\Delta^{UC}\%$
14 Bus N	ED,ED+R	644.87	0.021
24 Bus N	ED,ED+R	1620.73	0.217
24 Bus H	ED,ED+R	1200.85	-0.473

When comparing the installed capacity provided in Fig. 3.3, high renewable instances show an increase in installed capacity compared to the nominal instances. Also, high renewable instances avoid investments in renewable energy-based generators to mitigate the existing variability in generation. Instead, they lean more towards stable thermal generators, especially flexible generators like CCGT and OCGT, and also extra transmission line capacity to acquire operational flexibility from neighbouring nodes. These capacities indicate that investments in flexibility provisions become essential as the share of renewable energy-based generation (wind and solar) increases, and it is crucial to consider unit commitment in the planning stage to identify the required operational flexibility accurately and avoid sub-optimal solutions (ED and ED+R provide investment solutions that are more expensive to operate).

3.5 Conclusion

This chapter provides the detailed formulation of the unit commitment problem and investigates the impact of incorporating unit commitment in the planning problem. The chapter considers three operational models with different details of operational flexibility, i.e. traditionally used economic dispatch (ED), economic dispatch with ramping constraints (ED+R) and full unit commitment (UC) problem.

The numerical analysis shows that investments in flexibility provisions such as as flexible generators (e.g. CCGT and OCGT) and transmission line capacities are becoming crucial to ensure supply-demand balance when the percentage of renewable energy-based generators increases in the system. In addition, this study shows that operational models like ED and ED+R could provide sub-optimal investment solutions that are operationally more expensive. Although the cost difference is insignificant, the analysis reveals that the impact of incorporating UC depends on the underlying problem structure and characteristics of the system. As these solutions heavily rely on multiple factors such as network topology, existing capacity, candidates availability and other input parameters, a constrained system with a limited number of options is more likely to be affected by approximated or ignored flexibility constraints than a versatile system with a diverse set of generators and numerous candidate options. Therefore, to mitigate the risk of infeasibility and inefficiency, it is imperative to incorporate operational flexibility in the planning stage through unit commitment constraints.

Chapter 4

Time Resolution in Unit Commitment

4.1 Introduction

One of the most popular and simplest ways to reduce the time complexity in unit commitment is to adjust the temporal resolution. A coarse resolution with fixed length can quickly decrease the required computational time by a large factor, because of the reduced problem size. However, fixing the resolution to a specific time interval exposes the risk of neglecting important time periods with significant variations. Since these time periods characterise the operational flexibility in a system, the omission of them could lead to inaccurate solutions. Therefore in this chapter, we propose an approach to obtain coarse resolutions that are adaptive to the given data sets. The key idea is to reduce the resolution where it can be afforded such that, the size of the problem is reduced efficiently.

The first section of this chapter provides a brief background of current practices and the need for an adaptive time resolution. The next section presents the algorithm utilised to achieve an adaptive resolution and the extensions required in the unit commitment model to facilitate a variable resolution. The proposed algorithm and extensions collectively form the key contribution of this chapter, and it is already published in IEEE Power and Energy Society General Meeting 2018 as “An Efficient Method Based on Adaptive Time Resolution for the Unit Commitment Problem”. The subsequent section evaluates the proposed approach, where multiple criteria were considered including accuracy, computational speed-up, quality of the solution, ability to capture the required operational flexibility when a significant percentage is renewable energy-based generation and improvement in accuracy compared to the fixed length resolution. Finally, the last section concludes the chapter and provides limitations of the proposed approach.

4.2 Motivation and Background

Time resolution in power system models plays a vital role in both operational and planning context as the final decisions depend on the level of granularity considered. Especially with large penetrations of renewable energy-based generators as their generation is highly variable and intermittent [132]. By increasing the temporal resolution, these variations can be captured to provide a more realistic estimation for total generation costs. And, the flexibility of thermal units can be determined precisely including the short cycling and quick ramping events, which would have been invisible with a coarse resolution [133]. However, high resolutions enlarge the size of the problem substantially increasing the computational cost or even making the problem computationally intractable.

In operational simulations, simultaneous consideration of multiple resolutions has become a popular option to account for frequent variations in generation with reduced computational costs [134, 135]. In these studies, a fine resolution (e.g. 5 min, 15 min) is used for more critical tasks such as automatic generation control and contingency analysis and a coarse resolution (e.g. hourly) is employed for other tasks such as day-ahead scheduling and reserve requirements (unit commitment) [134, 135]. In addition, a scheduling horizon with a variable resolution is reported in [136], where early hours of the time horizon are scheduled using a detailed model at high resolution, while remaining hours are scheduled using a reduced model with a coarse resolution.

On the other hand, long-term capacity expansion models settle for a low resolution that captures only seasonal, weekly and daily variations in wind and demand. Since one year (or more) planning horizon with conventional hourly resolution is computationally expensive even for small systems, carefully chosen representative days [3] and/or aggregated time blocks [116] are generally used. For example, seasonal and weekly variations can be expressed using two days in each month representing weekday and weekend, and daily variations can be captured using aggregated times blocks of 4-6 hours instead of all 24 hours. In addition, if temporal dependencies in the system are ignored (e.g. economic dispatch), similar demand and wind scenarios (hours) can be clustered using well-known clustering techniques such as k means [137].

While representative days could produce inaccurate solutions that are far from optimal, clustering techniques lose continuity in data that is essential for the treatment of inter-temporal constraints such as ramping limitations (unless the clustering technique is applied to cluster demand and renewable generation profiles). It limits the ability to consider storage as a flexibility measure since continuity in data is necessary to track energy inventories. In contrast, pre-defining the coarse resolution to a fixed length (e.g. 4-6 hours) exposes the risk of neglecting important time periods with significant variations. Since these periods characterise the flexibility in the solution, in operational problems this could result in under-commitment. Furthermore, low resolutions could lead planning problems to completely different investment solutions as a result of overestimation or underestimation of variable renewable generation [138]. According to [139, 140], models with fixed coarse resolutions tend to invest more in base-load technologies [139], whereas models with fine resolutions tend to select diversified portfolios with renewable-energy based generators and flexible mid-load plants providing cheaper investment solutions [140].

Therefore, a resolution that is adaptive to a given data seems to be an appealing option to reduce the size of the problem while preserving accuracy. Such a method can be used to improve the efficiency in operational problems and tractability in capacity expansion planning problems when high resolution is required. Adaptive resolution can be achieved by aggregating consecutive time periods to form a single longer interval. As demand and/or renewable generation profiles often contain time periods that are operationally similar, they can be grouped into one time period replacing the fixed resolution with a variable resolution that is adaptive to the data set. Thus, an hourly interval-based unit commitment model with 24 hours will reduce to $\mathbf{k}$ intervals with variable lengths, where $\mathbf{k} < 24$ reducing the number of variables required, thus improving computational efficiency.

Alternatively, this method can be viewed as time-series segmentation as it partitions the time series into multiple segments with similar data points. Thus, time series segmenting techniques can be applied to obtain adaptive resolutions, as they share the same goal. The objective of segmenting methods is to split the time series by minimising an arbitrary cost function that expresses the similarity within each segment [141]. For example, the absolute error between original and approximated data point can be defined as the cost measure, and the sum of absolute errors across all segments can be minimised to obtain the desired segmentation.

In the literature, both exact and heuristic techniques have been utilised to minimise the cost functions and achieve segmentation. An exact approach that is based on mixed-integer programming (MIP) is reported recently in [142]. The method aggregates time intervals by minimising the gradients within the grouped intervals, until the pre-specified aggregation level is satisfied. Nevertheless, heuristic greedy approaches that produce good but sub-optimal k-segmentation are more frequently used because of their simplicity and reduced computational complexity [141]. Some commonly applied heuristic techniques are Top-down, Bottom-up, Sliding-window and clustering-based methods [143, 144]. Top-down approach partitions a given time series recursively until some stopping criterion (e.g. user-specified number of segments or error threshold) is met, by scanning the entire time series repeatedly. On the other hand, Bottom-up method starts from the finest possible approximation and scans the entire time series repeatedly to merge segments with the lowest cost until some stopping criterion is met. In contrast to Top-down and Bottom-up methods, Sliding-window approach anchors the left-most data point and increases the segment length to the right in an online manner considering one data point at a time, until the approximation error exceeds a predefined error threshold. Clustering-based methods adapt well-known clustering techniques such as k means to group the time periods while imposing a hard sequentiality constraint [144]. However, for this method, initial boundaries need to be specified by means of another algorithm.

In this chapter, we employ a heuristic algorithm to group the successive demand levels and approximate the demand series, because they are simple, computationally practical and produce solutions with reasonable quality [143]. In particular, we utilise the Sliding-window (SW) [143] algorithm. Unlike Top-down and Bottom-up approaches, Sliding-window does not require scanning of entire time series repeatedly to make segmenting decisions. In addition, the method has the advantage of finding an appropriate resolution, because the partitioning procedure is terminated by means of a predefined error threshold, not by number of segments. However, SW only propagates from left to right and does not attempt to change the partitions on the left side. Therefore, a novel backtracking algorithm (BT) is also introduced to move towards the left and ensure that the approximation error is minimum. This combined segmenting algorithm is named as "Sliding Window with Backtracking (SWBT)", and is elaborated in the next section.

4.3 Adaptive Time Resolution

This section describes the proposed adaptive time resolution based on SWBT algorithm and its application to unit commitment problem in detail. First, the Sliding Window (SW) and Backtracking (BT) procedures are described individually, providing complete algorithms. Then, the application of SWBT to an hourly unit commitment model is presented where the objective function and few constraints are modified to account for the variable time resolution.

4.3.1 Sliding Window with Backtracking (SWBT)

As previously mentioned, the goal of the proposed approach is to reduce the resolution where it can be afforded. This reduction can be achieved by aggregating finer time periods to increase the time interval and averaging the value over the aggregated periods. In SW (Sliding-Window) algorithm, starting from the left most data point, a segment or a time interval is increased until the approximation error incurred by averaging the value exceeds some predefined error threshold α. For the SW, the average of all relative errors (ARE) given in Eq. 4.2 was chosen as the error metric, as ARE allows the predefined error threshold $\boldsymbol{\alpha}$ to be independent of the range. Such a generic error threshold is useful to find a common resolution when dealing with demand/wind profiles in different regions (nodes) that are in different ranges.

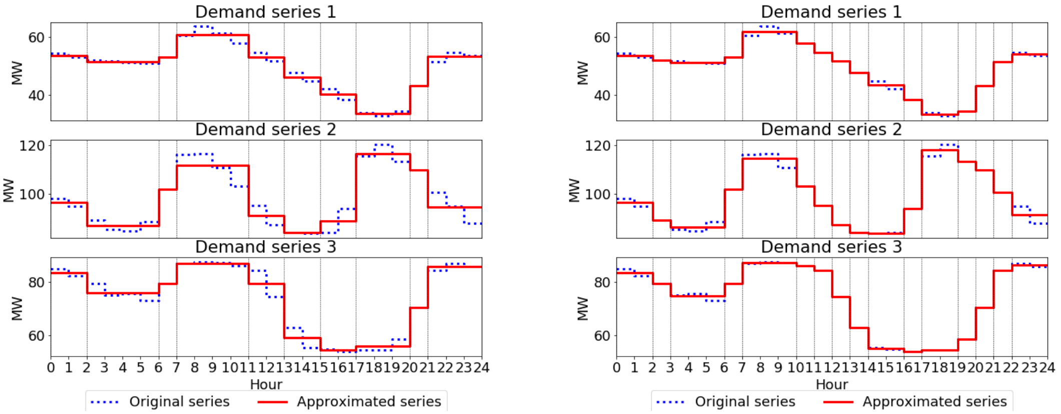
In the SWBT Algorithm 1, the general SW starts with the left-most data point making it the boundary $\mathbf{b_j}$ of a potential segment $\mathbf{j}$. Then the algorithm attempts to approximate (average) the demand to the right, by adding one data point $\mathbf{i}$ at a time. At some point $\mathbf{i}$, ARE for the potential segment $\mathbf{b_j}$ exceeds the threshold $\boldsymbol{\alpha}$, so the subsequence from $\mathbf{b_j}$ to $\mathbf{i-1}$ becomes one segment. Then the data point $\mathbf{i}$ becomes the boundary for the next potential segment $\mathbf{b_{j+1}}$ and the process repeats until the entire time series (T data points) has been partitioned into $\mathbf{k}$ segments [143]. However, in contrast to the general SW, which considers total ARE of the segment to decide whether to create a new segment or not, in SWBT, ARE at each data point in each profile is considered to make the segmenting decision so as to ensure that drastic variations in each profile is captured.

As SW propagates from left to right, the Backtracking (BT) Algorithm 2 reiterates to check whether ARE can be further reduced by moving the boundaries towards left. When a new segment is created updating the next potential boundary $\mathbf{b_{j+1}}$, BT backtrack to the left boundary of the just-completed segment $\mathbf{b_j}$, and checks whether the total approximation error across all profiles can be reduced by moving the boundary to the left $\mathbf{b_j - 1}$. This is done by calculating the change in the sum of relative errors across all profiles ΔRE (Eq. 4.4). If ΔRE is negative the boundary is moved, as this indicates that the error can be reduced. Then, BT checks whether the error can be further reduced by moving $\mathbf{b_j}$ to $\mathbf{b_j - 2}$. This procedure continues until the boundary cannot be moved any further. If $\mathbf{b_j}$ is moved, then BT backtracks to the segment before $\mathbf{b_j}$ which is $\mathbf{b_{j-1}}$ and repeats the same procedure. Thus, the SWBT algorithm continues to move one boundary at a time until the error cannot be reduced any further.

Therefore, SWBT can be summarised as follows.

1. Sliding Window (SW) algorithm finds the number of segments and initial boundaries so that the average relative error across all profiles, at each data point would be less than α
2. Backtracking (BT) algorithm ensures that the total error is minimum for a given number of segments k and boundaries.

Application of SWBT to three demand profiles is illustrated in Fig. 4.1, where the resolution is controlled by the error threshold α. Higher α values allow large approximation errors hence, demand profiles are approximated with lower resolution. In contrast, a lower α values approximate the demand profile with higher resolution. This is depicted in Fig. 4.1a and Fig. 4.1b respectively. In Fig. 4.1a, α was set to 0.05 which approximates 24 periods with 10 periods and in Fig. 4.1b, α was set to 0.02 which approximates the demand profiles with 16 periods. Note that all profiles have the same segmentation.



(a) $\alpha = 0.05$ (b) $\alpha = 0.02$

Figure 4.1: Adaptive time resolution application

$$\mu_m^{D'} = \frac{\sum_{b_k}^{i} d_{m,i}}{L_k + 1} \tag{4.1}$$

$$ARE_l = \frac{1}{M} \sum_{1}^{M} \frac{|d_{m,l} - u_m^{D'}|}{d_{m,l}} \tag{4.2}$$

$$\mu_{m,j}^{D'} = \mu_{m,j}^{D} + \frac{d_{m,q} - \mu_{m,j}^{D}}{L_j + 1} \tag{4.3}$$

$$\Delta RE = \sum_{1}^{M} \frac{|d_{m,q} - u_{m,j}^{D'}|}{d_{m,q}} - \sum_{1}^{M} \frac{|d_{m,q} - u_{m,j-1}^{D}|}{d_{m,q}} \tag{4.4}$$

Algorithm 1 Sliding Window with Backtracking

```
Input : Threshold $\alpha$, Demand Profiles $D_1, .., D_M$ where $D_m = \{d_{m,1}, .., d_{m,P}\}$
Output : No. segments $k$, Centeroids $\mu^D_{m,1}..\mu^D_{m,k}$, Length of segments $L_1, .., L_k$
procedure SW($\alpha, D$)
    % Initialise
    $k = 1, b_k = 1, L_k = 0$
    for $i = 1$ to $T$ do
        for $m = 1$ to $M$ do
            %Calculate new mean $\mu^{D'}_m$ with the data point $i$ (Eq. 4.1)
        for $l = b_k$ to $i$ do
            Calculate Average Relative Error (ARE) (Eq. 4.2)
        if all $ARE_l < \alpha$ then
            %Update the current segment
            $seg = false$
            $L_k = L_k + 1, \quad \mu^D_{m,k} = \mu^{D'}_m$
        else
            %Create a new segment
            $seg = true$
            $k = k + 1, \; b_k = i, \; \mu^D_{m,k} = d_{m,i}, \; L_k = 1$
        if $i = T$ then
            $complete = true, \quad start = k$
        else
            $start = k - 1$
        if ($seg$ or $complete$) and ($start > 1$) then
            BT($start, b$)
```

Algorithm 2 Backtracking

```
procedure BT($start, b$)
    for $j = start$ downto 2 do
        $transfers = false, \quad last = b_j - 1, \quad first = b_{j-1}$
        for $q = last$ down to $first$ do
            for $m = 1$ to $M$ do
                Calculate the new mean $\mu^{D'}_j$ when boundary $b_j$ move to $b_j - 1$ (Eq. 4.3)
            Calculate difference in relative error $\Delta RE$ (Eq. 4.4)
            if $\Delta RE < 0$ then
                $transfers = true$
                % Update boundaries and segments
                $b_j = b_j - 1$
                $L_j = L_j + 1$
                $L_{j-1} = L_{j-1} - 1$
                $\mu^D_{m,j} = \mu^{D'}_{m,j}$
                $\mu^D_{m,j-1} = \mu^D_{m,j-1} - \dfrac{d_{m,q} - \mu^D_{m,j-1}}{L_{j-1}}$
            else    break
        if not $trasfers$ then
            break
```

4.3.2 Extended Unit Commitment Formulation

This section presents the modifications required in the unit commitment model to account for the adaptive time resolution. As the set of time periods $\mathcal{T}$ reduces to set $\mathcal{K}$ after application of SWBT, the aggregated time periods must reflect in the unit commitment model. Since the detailed unit commitment model is provided in Chapter 3, only the modifications are presented in this section.

Objective Function

The objective function Eq. 3.1 in Chapter 3 is updated to account for the cost parameters in the aggregated time periods by including the length of the time period L_t as an additional coefficient.

$$\min \sum_{t\in\mathcal{K}} \sum_{g\in\mathcal{G}} ((C^{var*}_{g,t}(p_{g,t}) + C^{c}_{g} u_{t,g})L_t + C^{u}_{g} v_{g,t} + C^{d}_{g} w_{g,t}) \tag{4.5}$$

Power Balance Constraint

The power balance constraint Eq. 3.24 in Chapter 3 is updated to consider the new approximated demand $\mu^{D}_{n,t}$.

$$\sum_{g\in\mathcal{G}(n)} p_{g,t} + \sum_{r\in\mathcal{R}(n)} p_{r,t} + \sum_{l\in L|r(l)=n} f_{l,t} - \sum_{l\in L|s(l)=n} f_{l,t} = \mu^{D}_{n,t} \tag{4.6}$$

Minimum Up and Down Constraints

In minimum up and down constraints Eq. 3.3 - Eq. 3.4 given in Chapter 3, $i = t - T^{u,d}_{g}$ is replaced with $i = t - C^{u,d}_{g,p}$ to consider the variable time period length. The parameter $C^{u,d}$ is the number of data points to count back from the current point t which is found using Algorithm 3. The intuition behind the algorithm is to count back from the current point t until the sum of the segment lengths does not exceed or become equal to the given minimum up or downtime. While counting back, if the algorithm reaches the beginning of time horizon, it accounts for the initial up and down-times and if the algorithm is unable to count back further it exits. For simplicity, this aspect is not shown in Algorithm 3. It is important to note that, $C^{u,d}$ is calculated for all generators and periods, and Algorithm 3 depends heavily on how minimum up and down constraints are modelled.

Algorithm 3 Count Back

```
Input : Current point $t$, Length of segments $L_1, .., L_k$, Minimum up or down time $T^u_g, T^d_g$
Output : $C^{u,d}$
sum = 0,    $C^{u,d} = 0$
while $(sum + L_{p-(C^{u,d}+1)}) < T^{u,d}_g$ do
    $C^{u,d} = C^{u,d} + 1$
    $sum = sum + L_{p-C^{u,d}}$
```

4.4 Experimental Analysis and Discussion

This section presents the experimental analysis conducted to evaluate the performance of the proposed approach based on SWBT. The algorithm was applied to a thermal unit commitment problem with hourly resolution where renewable sources are ignored without loss of generality. The model was then extended according to Section 4.3.2 to facilitate the variable resolution with valid cost parameters and constraints.

The analysis was carried out by employing two case studies: 6 bus and IEEE 118 system, which can be found at *motor.ece.iit.edu*. The simple six bus system which is consisted of 3 generators, 7 lines, and 3 loads, is utilised to illustrate the changes in UC solution with adaptive time resolution and the 118 system with 54 generators, 186 lines and 91 loads is used to show the performance of the proposed approach with an extensive system. In both cases, a system-wide demand profile was considered for a period of 24 hours, where the demand at each bus was determined by a load distribution factor. As each demand bus usually contains a unique profile, another instance was considered where multiple demand profiles were taken into consideration. These two instances can be elaborated as follows.

- In instance A, system-wide load profiles and distribution factors provided in the test cases were utilized.
- In instance B, multiple load profiles were considered where each demand bus was given a unique profile. These profiles were obtained from electricity market regions and planning sub-regions of Australia which can be found at *aemo.com.au* and were scaled to suit the system. The 3 demand profiles that represent the 3 loads in the 6 bus system is presented in Fig. 4.1. Since it is not practical to consider 91 load profiles, the 118 bus system was divided into 5 zones and 5 different demand profiles were utilised with load distribution coefficients. The 5 zones are. Zone1: buses 1-19 and 117, zone2: buses 20-32 and 113-115, zone3: buses 33-67, zone4: buses 68-102,116 and 118, and zone5: buses 103-112.

These demand profiles were segmented using SWBT algorithm to obtain adaptive time resolutions. As this analysis aims to compare the performance of SWBT on unit

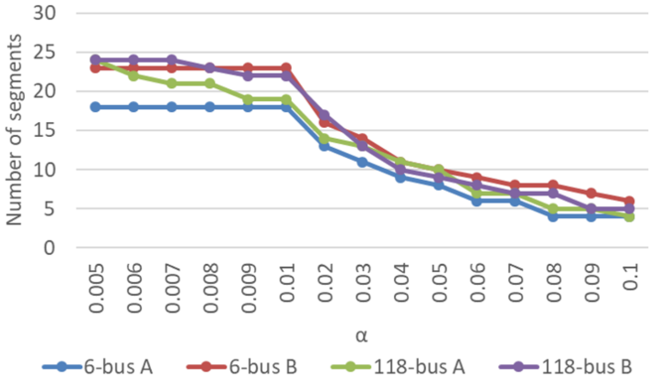


Figure 4.2: Number of segments for different α values

commitment with different adaptive resolutions, the SWBT algorithm was applied multiple times considering a range of α values to obtain a series of adaptive resolutions.

The outcome of SWBT with different α values is illustrated in Fig. 4.2 where the resolution (number of segments) decreases as the α value (error threshold) increases. Using these segmented profiles a series of experiments were carried out to analyse the performance of SWBT according to following criteria.

1. The performance of the proposed approach based on SWBT algorithm
2. The quality of UC solutions with adaptive time resolutions.
3. The ability of SWBT to accurately represent the required operational flexibility when significant penetrations of renewable sources are involved in the system.
4. Improvement in accuracy compared to the fixed coarse resolution.

All the algorithms and models used in these experiments were implemented and solved using Python 2.7 and Gurobi 7.0.2 [131] on a PC with a quad-core processor clocking at 2.8GHz and 8 GB RAM. In addition, the default MIP gap of Gurobi 1×10^{-4} was used except for the IEEE 118 system instance A in which 1×10^{-3} was used due to prolonged computation times. Moreover, for all test cases and instances, the hourly resolution is considered as the base case.

4.4.1 Performance of the SWBT Algorithm

The performance of SWBT algorithm was analysed by solving the unit commitment problem repeatedly for all the adaptive resolutions obtained from different α values. The results are depicted in Fig. 4.3 and Fig. 4.4 in terms of percentage difference in total cost $\%\Delta C$ and gain in computational speed respectively. Eq. 4.7 presents the percentage difference in total cost $\%\Delta C$, where $C_{original}$ represents the total cost (objective value) obtained with the original profile at 1 hour resolution, and $C_{segmented}$ represents the total cost obtained with the segmented profiles. Hence, a positive $\%\Delta C$ value indicates a total cost greater than the actual cost with the original demand series and negative otherwise.

$$\%\Delta C = \frac{C_{segmented} - C_{original}}{C_{original}} \times 100\% \tag{4.7}$$

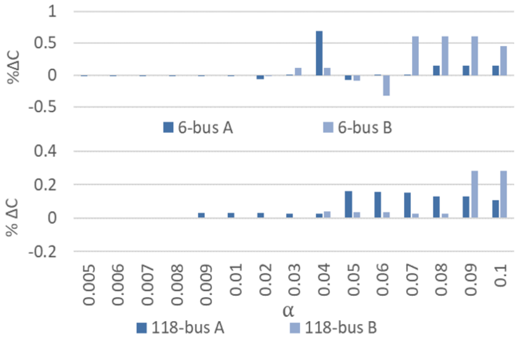


Figure 4.3: Percentage cost differences

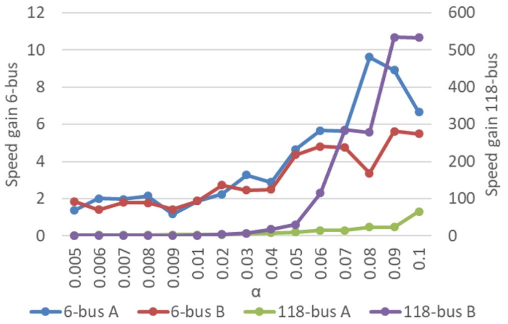
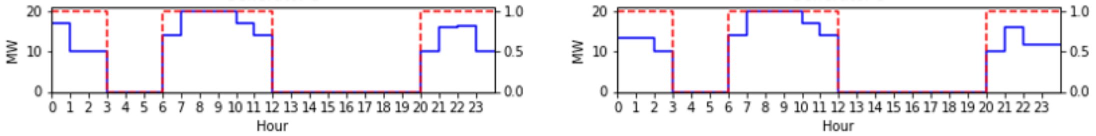
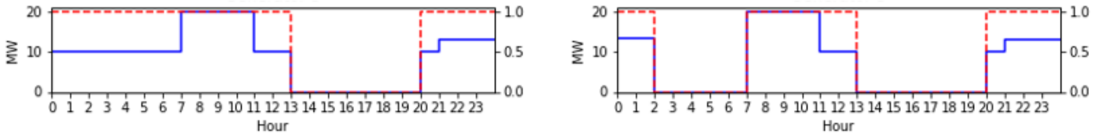


Figure 4.4: Gain in computational speed



(a) With original demand series (b) With segmented demand series at $\alpha = 0.02$

Figure 4.5: UC and dispatch solution for G3 in 6-bus system for Experiment 1-B



(a) Dispatch solution for over estimated demand with $\alpha = 0.08$ (b) Dispatch solution for under estimated demand with $\alpha = 0.06$

Figure 4.6: UC and dispatch solution for G3 in 6-bus system for Experiment 1-B

In Fig. 4.3, it is apparent that all $\%\Delta C$ values are within $\pm 1\%$, and they become negligible for smaller α values. As smaller α values increase the resolution, the commitment solution becomes very similar to the actual solution with the original demand series or even exactly the same. This is presented in Fig. 4.5b, where the UC solution with segmentation at $\alpha = 0.02$ (red dashed line) is the same as the original UC solution shown in Fig. 4.5a, but with a slightly different dispatch solution (blue thick line). Therefore, the very small $\%\Delta C$ is due to the difference in dispatch solutions. Also, 118 bus system shows relatively small $\%\Delta C$ values compared to the 6-bus system, because in large networks, cost difference due to demand approximation is only a small proportion of the total generation costs. Thus smaller differences in 118 system indicate that adaptive resolution is more beneficial for larger systems.

When comparing the reduction in computational time illustrated in Fig. 4.4, it is visible that the 118 bus system has gained significant computational speed-ups compared to the 6 bus system. For 118 bus system instance B, with an α value of 0.1, the unit commitment is 500 times faster with only 0.3% error in total cost. This improvement in computational speed further supports the notion that larger systems could gain significant benefits from adaptive time resolution. The notable difference between computational gains of instance A and B in 118 system is due to the different MIP gaps, which indicates that for smaller MIP gaps the speed-up would be higher.

4.4.2 Quality of Unit Commitment Solutions

The quality of UC solutions was assessed by solving the UC problem again with hourly resolution while imposing the commitment decisions obtained from the adaptive resolution. Forcing commitment decisions could make the problem infeasible. Hence, unserved demand was introduced to the power balance constraint with a high penalty cost of 1000\$/MWh to ensure feasibility.

The results obtained are depicted in Fig. 4.7, in terms of increase in cost as a percentage of the actual cost with original resolution. In Fig. 4.7, a majority of UC solutions shows only a negligible increase in cost indicating that they are of high quality. In addition, IEEE 118 bus system shows a relatively small increase in cost compared to the 6-bus system. This is because, in large networks, increased number of generators and their different properties allow the system to offset approximation errors easily.

However, in some instances, higher α values have produced UC solutions that are extremely expensive to operate. These significant peaks are mostly caused by the penalty cost that is related to unserved demand. Unserved demand occurs when the demand is underestimated with the adaptive resolution. As this makes the UC problem to commit less number of units, the available units to serve the true demand becomes insufficient. This is presented in Fig. 4.6b where time intervals (hours) from 2-3 and 6-7 are under-committed. In other words, at time intervals 2-3 and 6-7 where the generator is committed with the original hourly resolution (refer to Fig. 4.5a), in the adaptive resolution approach, the generator is switched off (refer to Fig. 4.6b).

In contrast, the slight increase in cost is produced when the demand is overestimated with the adaptive resolution. Over-estimated demand commits extra units that are also expensive to operate. As a result, generators are forced to continue power generation at a higher cost due to the minimum generation requirement. This is shown in Fig. 4.6a in which, from time intervals 3-6, the generator is over-committed (switched on unnecessarily) compared to the UC solution with the original resolution (refer to Fig. 4.5a).

$$\%\Delta^I C = \frac{C_{new} - C_{penalty} - C_{segmented}}{C_{segmented}} \times 100\% \tag{4.8}$$

These results are also summarised in Tables 4.1 and 4.2, where k is the number of segments obtained for each α, SPF (speed-up factor) is the gain in computational speed, $\Delta^I C$ is the percentage increase in cost excluding the penalty cost (to avoid accounting for unserved demand twice) and $\%U$ is the unserved demand as a percentage of the total demand. $\Delta^I C$ is shown in Eq. 4.8 where, C_{new} is the new cost obtained for the original demand series with imposed UC decision.

According to Tables 4.1 and 4.2, for a given system the ideal resolution (number of segments k) to operate would be the α value that has provided the highest computational gain (SPF) with zero $\%\Delta^I C$ or $\%U$ (highlighted rows in Tables 4.1 and 4.2). For example,

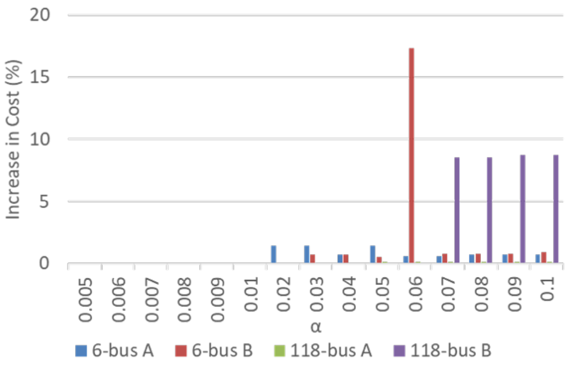


Figure 4.7: Quality of unit commitment solutions

Table 4.1: 6-bus comparison with different α

α	Instance A: 1 Profile				Instance B: 3 Profiles			
	k	SPF	$\%\Delta^I C$	$\%U$	k	SPF	$\%\Delta^I C$	$\%U$
0.1	4	6.656	0.739	0	6	5.489	0.9	0
0.09	4	8.909	0.739	0	7	5.634	0.748	0
0.08	4	9.606	0.739	0	8	3.366	0.748	0
0.07	6	5.638	0.582	0	8	4.743	0.748	0
0.06	6	5.666	0.582	0	9	4.799	0	0.273
0.05	8	4.661	0	0.023	10	4.359	0	0.008
0.04	9	2.876	0.739	0	11	2.493	0.145	0.008
0.03	11	3.269	0	0.023	14	2.453	0.145	0.008
0.02	13	2.234	0	0.023	**16**	**2.720**	**0**	**0**
0.01	**18**	**1.871**	**0**	**0**	23	1.877	0	0
0.009	**18**	**1.179**	**0**	**0**	23	1.419	0	0
0.008	**18**	**2.145**	**0**	**0**	23	1.770	0	0
0.007	**18**	**1.976**	**0**	**0**	23	1.794	0	0
0.006	**18**	**2.011**	**0**	**0**	23	1.409	0	0
0.005	**18**	**1.358**	**0**	**0**	23	1.855	0	0

Table 4.2: 118-bus comparison with different α

α	Instance A: 1 Profile				Instance B: 3 Profiles			
	k	SPF	$\%\Delta^I C$	$\%U$	k	SPF	$\%\Delta^I C$	$\%U$
0.1	4	65.800	0.148	0	5	532.726	0.078	0.12
0.09	5	23.620	0.148	0	5	534.291	0.078	0.12
0.08	5	23.478	0.148	0	7	278.753	0	0.12
0.07	7	14.914	0.145	0	7	285.235	0	0.12
0.06	7	15.344	0.144	0	8	115.374	0.043	0
0.05	10	10.330	0.141	0	9	30.423	0.043	0
0.04	11	8.148	0.028	0	10	17.844	0.043	0
0.03	13	6.147	0.025	0	**13**	**7.229**	**0**	**0**
0.02	14	3.329	0.032	0	17	4.142	0	0
0.01	19	3.068	0.028	0	22	0.867	0	0
0.009	19	3.094	0.028	0	22	0.761	0	0
0.008	**21**	**2.722**	**0**	**0**	23	0.892	0	0
0.007	**21**	**2.724**	**0**	**0**	24	0.974	0	0
0.006	22	2.271	0	0	24	1.008	0	0
0.005	24	1.000	0	0	24	0.967	0	0

the ideal resolution for 118 bus system Instance B is at $\alpha = 0.03$, which has utilised 13 segments to provide accurate results with a computational gain of 7.2. For the same system, a good quality solution with negligible $\%\Delta^I C$ can be obtained 115 times faster using only 8 segments. Similar results can be observed for other instances around the same α values. In 6 bus system instance A, ideal resolution to operate is $\alpha = 0.01$ with 18 segments while good quality solutions can be obtained below $\alpha = 0.05$. This indicates that for a given system, 8-18 segments, i.e. 25-60% aggregation level is the most suitable range for adaptive time resolution to obtain high-quality solutions with significant computational gains. This roughly translates to an α range of $0.01 - 0.05$ depending on the data provided. Although it is possible to obtain good quality solutions at higher α values with significant computational gains, according to Fig. 4.7 these values may also provide highly inaccurate solutions that are extremely expensive to operate. Therefore, large α values are not recommended unless a substantial gain in computational speed is essential at the cost of lower accuracy.

4.4.3 Impact of High Penetrations of Renewable Generators

This section examines the ability of SWBT to accurately represent the required operational flexibility when significant penetrations of renewable sources are involved in the system. A large percentage of renewable resources was included by replacing the original demand profiles with artificial net demand profiles as shown in Fig. 4.9. These net load profiles represent significant penetrations of solar, which require high ramping capabilities during the evening peak because of the increasing demand with decreasing solar output. Thus, they are utilised to investigate the ability of low resolutions to represent the required operational flexibility accurately in a system.

The previous set of experiments were repeated with the new set of net demand profiles, where once again the profiles were segmented considering a range of α values. Fig. 4.8 depicts the new segmented demand profiles for $\alpha = 0.05$, which evidently shows SWBT's ability to determine significant variations through increased resolution at the evening peak and low resolution during other time periods. Consequently, SWBT was able to capture the required operational flexibility accurately, which is demonstrated in Fig. 4.10 and Fig. 4.11 in terms of percentage difference in total cost $\%\Delta C$ and the percentage increase in total cost respectively.

According to Fig. 4.10, for all α values the proposed approach has provided accurate results with small $\%\Delta C$ values. This indicates that SWBT algorithm is capable of providing precise solutions for demand profiles with significant variations even at low resolutions. On the other hand, Fig. 4.11 verifies that obtained UC solutions are of good quality since all the $\%\Delta^I C$ values are negligible. Absence of significant peaks in Fig. 4.11 denotes that increase in costs are mostly due to over-commitment rather than unserved demand, thus further emphasises the high quality of UC solutions. Once again, IEEE 118 system exhibits considerably smaller increases in cost compared to the 6 bus system due to the large number of generators, which enhances the benefit of the proposed approach for large systems.

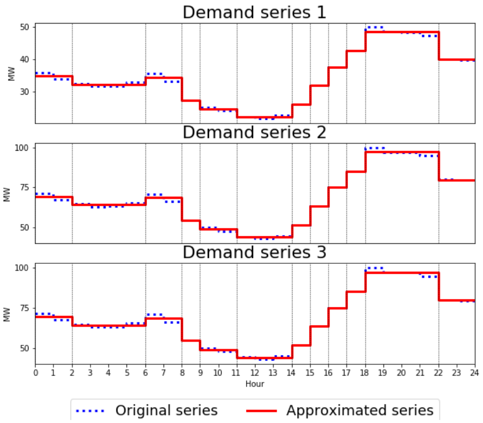
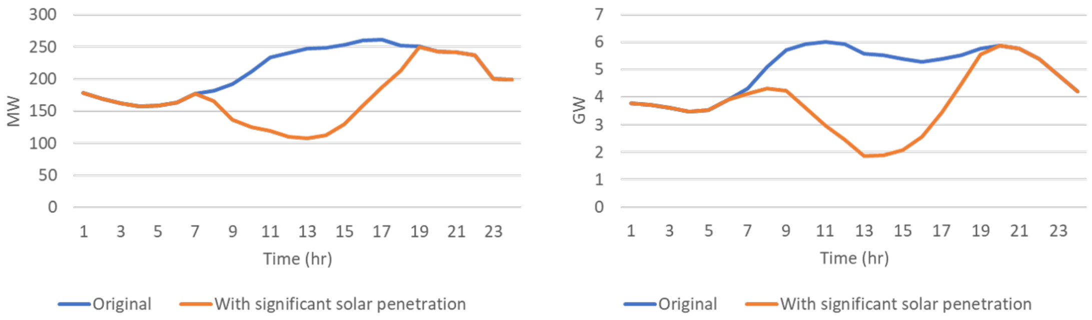


Figure 4.8: 6 Bus system instance A with significant variations at $\alpha = 0.05$



(a) 6 Bus system instance A (b) IEEE 118 Bus system instance A

Figure 4.9: Net load with significant penetrations of rooftop solar

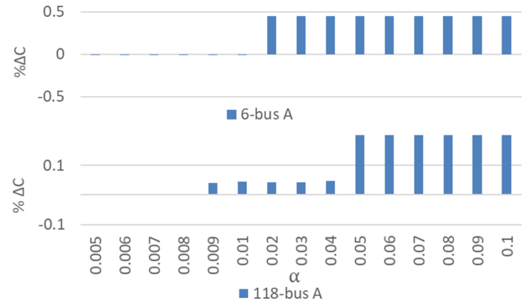


Figure 4.10: Percentage difference in cost

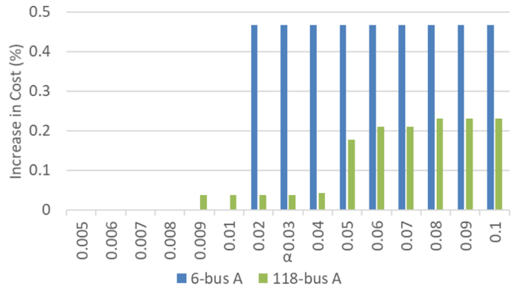
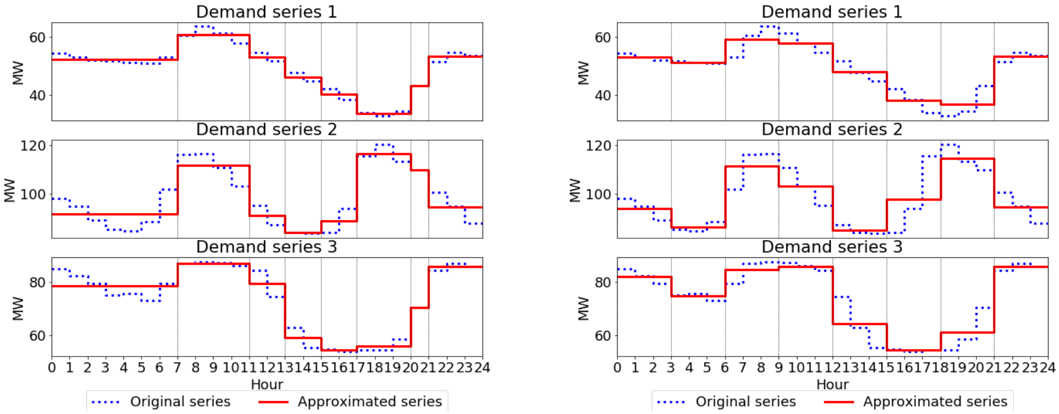


Figure 4.11: Percentage increase in cost

4.4.4 Adaptive Vs Fixed Resolution

This section compares the adaptive resolution approach to the commonly used fixed resolution method. For a fair comparison, both methods were investigated with the same number of segments. In fixed resolution, 2-6 hr time blocks were considered to obtain 12, 8, 6 and 4 segments. The same number of segments were achieved with adaptive resolution by varying the α value. This is illustrated in Fig. 4.12 which depicts the segmentation for both adaptive and fixed approaches. The figure clearly shows the difference in segment lengths for 8 time segments in the two approaches. Similar to other evaluations, the experiments were repeated with fixed and variable resolution demand profiles, and the results obtained are provided in terms of percentage difference in cost and percentage increase in cost in Fig. 4.13 and Fig. 4.14 respectively.

When comparing the percentage difference in total costs (Fig. 4.13), fixed resolution constantly underestimated the total cost in 6 bus system compared to the adaptive resolution. Since underestimations correspond to under-commitment, these solutions yielded extremely high penalty costs when evaluated due to unserved demand (Fig. 4.14). In contrast, for 118 system, fixed resolution estimated the total cost more accurately with less over-commitment. However, the quality of solutions was poor (large percentage increase in cost Fig. 4.14). This is due to the distortion that occurs in demand profiles with fixed time blocks (Fig. 4.12). Distorted demand profiles cause the system to schedule a different commitment profile making the generators available to supply actual demand insufficient in some time periods. As a result, solutions provided by the fixed resolution overall had poor quality compared to the solutions provided by the adaptive time resolution.



(a) Adaptive resolution with $\alpha = 0.07$ (b) Fixed resolution with 3 hr length

Figure 4.12: Adaptive and fixed resolution with 8 segments (66.67% aggregation level)

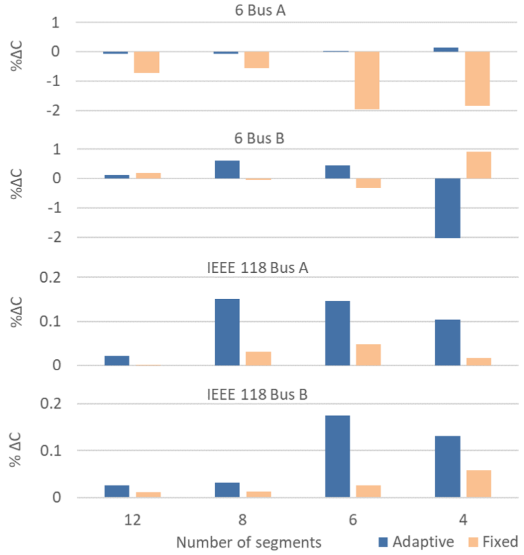


Figure 4.13: Percentage difference in cost

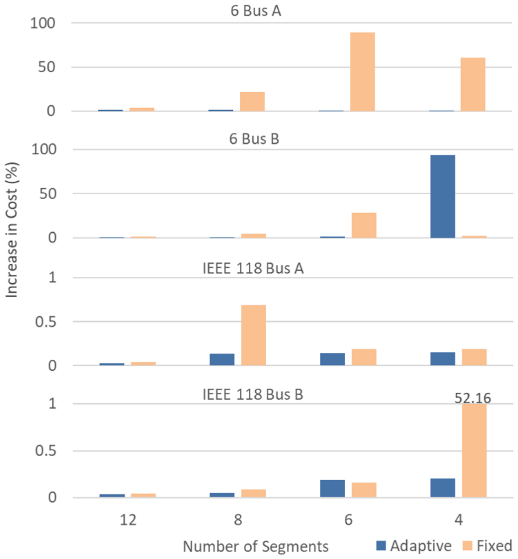


Figure 4.14: Percentage increase in cost

Furthermore, at extremely coarse resolutions both methods showed poor performance producing low-quality solutions. However, adaptive resolution performed comparatively better with over-committed solutions reducing the risk of unserved demand and penalty costs. Unlike the adaptive resolution approach, fixed resolution approach is also limited to a smaller set of aggregation levels. The smallest possible aggregation level (50%) already discards half of the time periods and the next possible aggregation level (66.67%) falls outside the recommended range (25-60%). So, the fixed resolution method has only a limited set of possible aggregation levels compared to the adaptive resolution approach. Therefore, it is evident that the proposed approach based on adaptive resolution (SWBT) improves the accuracy and provides more flexibility compared to the fixed resolution approach.

4.4.5 Application to Operational Simulations and Planning Problems

The proposed approach based on adaptive resolution can be successfully applied to day-ahead unit commitment problems and even unit commitment simulations with longer durations (e.g. one year) to determine the generator schedule and their availability with reduced computational effort. In addition, it can be applied to operational simulations that aim to evaluate and check the feasibility of a given investment solution (e.g. unit commitment simulation with the rolling horizon technique discussed in Section 3.3.2).

In addition to the unit commitment simulations, the proposed adaptive resolution approach based on SWBT can also be applied to find optimal investment solutions with rolling horizons. For example, optimal investment solutions can be found by solving the daily unit commitment problems in a cyclic manner with investment decisions set across all representative periods [120]. Adaptive resolution can be applied to solve these cyclic unit commitment problems more efficiently.

In addition, the SWBT algorithm can be utilised to reduce the computational burden when solving the GTEP-UC formulation directly with commercially available solvers. However, to achieve a resolution that is capable of producing results within a reasonable period of time for a horizon of one year, the problem will require an extremely coarse resolution of 8-12 hr segment lengths. Since long time segments can cause the investment solutions to deviate significantly as a result of the distorted demand and renewable generation profiles, in such cases, it is more appropriate to use SWBT in conjunction with another approach such as representative days or decomposition techniques. As decomposition techniques divide the problem into multiple sub-problems potentially with shorter horizons, SWBT can acquire significant computational speed-ups in the sub-problem level, and thus in the overall problem without compromising the accuracy substantially.

4.5 Conclusion and Limitations

This chapter proposes an adaptive time resolution approach to reduce the computational time required to solve power system operational models. While the most commonly used approach is fixed coarse resolution, this method could produce inaccurate solutions due to the exclusion of important time periods that characterises operational flexibility. Hence, a coarse resolution that is adaptive to a given data set is proposed and used. In the approach, consecutive time periods with similar operational conditions are grouped together to lengthen the time interval and lower the resolution where it can be afforded. A heuristic sliding window (SW) algorithm is utilised to achieve this goal, where time segment lengths are increased until the approximation error exceeds a predefined error threshold. As SW only propagates right and does not attempt to rearrange the segments on the left side, a novel backtracking algorithm (BT) is also used to reduce the error further and approximate the demand series better. This combined algorithm is referred to as the sliding window with backtracking (SWBT). Since the aggregated time periods must be accounted for in the unit commitment model, the objective function and few constraints were also modified to ensure that the cost parameters and system limitations are valid.

Evaluation of the proposed approach showed that adaptive resolution is capable of providing accurate solutions with substantial gains in computational speed, especially in large systems. These solutions are also of high quality as they only require a negligible increase in cost to meet the real demand. Moreover, the analysis identified that the suitable error threshold range to operate SWBT is from 0.01-0.05 which roughly represents an aggregation level of 25-60%. Within this range, the proposed approach based on SWBT can be successfully applied to unit commitment problems to reduce the computational time while providing high-quality solutions.

Furthermore, additional investigations showed that SWBT is capable of producing accurate results even with drastic variations in demand profiles. By increasing the resolution along the periods with significant fluctuations and decreasing the resolution at other periods, SWBT was able to provide good approximations for demand profiles with coarser resolutions. In comparison to the commonly used fixed resolution method, the proposed approach based on SWBT provided better quality solutions. As the profiles are segmented in an adaptive manner, the demand profiles showed minimum distortion thereby improving accuracy. Furthermore, adaptive resolution has the benefit of operating at many aggregation levels in contrast to the fixed resolution method. Thus, SWBT based on adaptive resolution is an improved approach compared to the widely used fixed resolution approach.

While the adaptive resolution method is highly appealing to reduce the computational burden in operational models, sole use of SWBT is mostly suitable for problems with short planning horizons (e.g. day-ahead unit commitment, planning and operational simulations with rolling horizon). When the planning horizon is long, the segment lengths must be increased substantially to reduce the number of segments and solve the problem within a reasonable period of time. Since long time segments cause the demand profiles to distort significantly, the resulting inaccurate operational decisions may produce investment solutions that are far from optimal. Thus, for problems with long planning horizons, it is more appropriate to use SWBT with other approaches that reduce computational complexity, such as decomposition techniques.

Chapter 5

Scenario Decomposition for GTEP-UC

5.1 Introduction

Large optimisation problems often comprise a decomposable structure allowing them to be solved as multiple sub-problems in a distributed manner. Energy system planning problem with unit commitment also consists of such structure, which has been exploited already by Bender's and Dantzig-Wolfe decomposition techniques. However, application of these methods has been less effective, because of the integer variables in both investment and operational levels, and inter-temporal constraints in the problem. Therefore, this chapter proposes a novel decomposition framework based on an existing scenario decomposition approach to decompose the generation and transmission expansion planning problem with unit commitment(GTEP-UC).

The first section of this chapter introduces the GTEP-UC problem and its structure for both single and multi-period formulations. Then, the chapter proposes the scenario decomposition framework and its application to GTEP-UC problem. The next section evaluates the proposed framework discussing its potential and limitations comprehensively, and compares it with the Bender's decomposition. The decomposition framework and the extensive analysis constitute the key contributions of this chapter. Finally, the chapter concludes that the proposed approach is suitable to decompose GTEP-UC problems and provides insights for algorithmic extensions.

5.2 Motivation

Although Bender's and Dantzig-Wolfe decomposition have already been applied to the energy system planning problem with unit commitment [121, 123, 124], their application has been less effective in practical instances. For Bender's and Dantzig-Wolfe decomposition, binary variables in the problem must be part of either master or slave sub-problem to obtain dual information that is essential for bound improvement and algorithm convergence. Because of these requirements, Bender's decomposition often results in a master problem that is too large for efficient solution [121,123], and having to integrate additional approaches like branch and price with Dantzig-Wolfe decomposition [124].

On the other hand, the scenario decomposition approach has the advantage that it has reduced dependency on problem structure and does not require dual information to be passed between sub-problems. Rather, it follows an evaluate and cut scheme to improve the bounds and converge the algorithm. The method also facilitates parallelisation to gain computational speed-ups, as components of the algorithm have very little dependency on each other.

However, scenario decomposition approach was initially proposed for stochastic optimisation problems with independent scenarios/sub-problems. This is in contrast to decomposing a deterministic optimisation problem into sub-problems that are linked by inter-temporal constraints. The adaptation of scenario decomposition to deterministic optimisation problems thus requires innovation. Therefore, this chapter proposes a decomposition framework based on scenario decomposition for the deterministic generation and transmission expansion planning problem with unit commitment.

In addition, GTEP-UC problems are typically formulated with multiple periods to determine when to construct in the planning horizon. These formulations are computationally more challenging because they must account for investment and operational decisions in multiple periods. Benders and Dantzig-Wolfe decomposition techniques have the ability to deal with decision variables in multiple periods through the dedicated master problem (investment decisions) and as an additional set of sub-problems (operational decisions). Scenario decomposition however does not comprise a master problem and the scenario sub-problems are also generally limited to one period. Therefore, this chapter aims to describe the application of scenario decomposition to multi-period problems.

5.3 Problem Formulation and Structure

Energy system planning problem assumes that a set of candidate investments including thermal generators, transmission lines, large scale wind and solar farms are provided. In addition, it assumes that for each candidate option, size, technology and location are already defined such that only the investment decision is left to be decided. Hence, binary decision variables are utilised to determine the “invest” and “do not invest” decisions. The single-period model that considers a representative year with annualised capital cost can be provided as follows.

$$\min \sum_{i\in\mathcal{I}} C^{I}_{i} x_{k,i} + \sum_{t\in\mathcal{T}} \left(\sum_{g\in\mathcal{G}} (C^{var*}_{t,g}(p_{k,t,g}) + C^{c}_{g} u_{k,t,g} + C^{u}_{g} v_{k,t,g} + C^{d}_{g} w_{k,t,g}) + \sum_{n\in\mathcal{N}} C^{p} q_{k,t,n} \right) \tag{5.1}$$

s.t.

$$u_{k,g,t} \leqslant x_{k,g} \qquad \forall k,g,t \qquad (5.2)$$

$$u_{k,g,t} - u_{k,g,t-1} = v_{k,g,t} - w_{k,g,t} \qquad \forall k,g,t \qquad (5.3)$$

$$\sum_{i=t-T_g^U+1}^{t} v_{k,g,i} \leqslant u_{k,g,t} \qquad \forall k,g,t \qquad (5.4)$$

$$\sum_{i=t-T_g^D+1}^{t} w_{k,g,i} \leqslant 1 - u_{k,g,t} \qquad \forall k,g,t \qquad (5.5)$$

$$P_g^{min} u_{k,g,t} \leqslant p_{k,g,t} \leqslant P_g^{max} u_{k,g,t} \qquad \forall k,g,t \qquad (5.6)$$

$$p_{k,g,t} - p_{k,g,t-1} \leqslant R_g^U u_{k,g,t-1} + S_g^U v_{k,g,t} \qquad \forall k,g,t \qquad (5.7)$$

$$p_{k,g,t-1} - p_{k,g,t} \leqslant R_g^D u_{k,g,t} + S_g^D w_{k,g,t} \qquad \forall k,g,t \qquad (5.8)$$

$$F_l^{min} x_{k,l} \leqslant f_{k,t,l} \leqslant F_l^{max} x_{k,l} \qquad \forall k,l,t \qquad (5.9)$$

$$-(1 - x_{k,l})M \leqslant f_{k,l,t} - B_l(\theta_{k,s,t} - \theta_{k,r,t}) \qquad \forall k,l,t \qquad (5.10)$$

$$(1 - x_{k,l})M \geqslant f_{k,l,t} - B_l(\theta_{k,s,t} - \theta_{k,r,t}) \qquad \forall k,l,t \qquad (5.11)$$

$$-\theta^{max} \leqslant \theta_{k,n,t} \leqslant \theta^{max} \qquad \forall k,n,t \qquad (5.12)$$

$$\theta_{k,n,t} = 0 \qquad n : slack, \forall k,t \qquad (5.13)$$

$$0 \leqslant y_{k,r,t} \leqslant Y_{k,r,t}^{max} x_{k,r} \qquad \forall k,r,t \qquad (5.14)$$

$$0 \leqslant q_{k,n,t} \leqslant \Delta D_{k,n,t} \qquad \forall k,n,t \qquad (5.15)$$

$$\sum_{g \in \mathcal{G}(n)} p_{k,g,t} + \sum_{l \in L|r(l)=n} f_{k,t,l} - \sum_{l \in L|s(l)=n} f_{k,t,l} + \sum_{r \in \mathcal{R}(n)} y_{k,r,t} \qquad (5.16)$$

$$+ q_{k,n,t} = D_{k,t,n} \qquad \forall k,n,t \qquad (5.17)$$

In the formulation above, i, k, t, n, g, l, r are indices for candidate unit, period ($k = 1$ in single-period formulation), time, node (bus), thermal generator, transmission line and renewable generator; $\mathcal{I}, \mathcal{K}, \mathcal{T}, \mathcal{N}, \mathcal{G}, \mathcal{L}, \mathcal{R}$ are the corresponding sets; $r(l), s(l)$ denote receiving and sending bus of line l; C^I indicates the annualised investment cost of candidate units; $C^{var*}(p_{t,g})$ represents the variable cost of generation; C_g^c, C_g^u, C_g^d are commitment, start-up and shut-down cost; C^p is the penalty cost for unserved demand (i.e. demand not met); $P_g^{max/min}, T_g^{u/d}, R_g^{u/d}, S_g^{u/d}$ are maximum and minimum generation limits, minimum up and down times, ramp up and down limits and start-up and shut-down limits; $F_l^{max/min}, B_l$ are maximum and minimum power flow limits and susceptance; θ^{max} denotes maximum phase angle; Y^{max} is the maximum available renewable output; $D, \Delta D$ are the nodal demand and nodal net demand; x, u, v, w are binary decision variables for investment, commitment, start up and shut down status; and p, y, f, θ, q are continuous variables for thermal generation, renewable generation, power flow phase angle and unserved demand.

The objective function 5.1 minimises the annualised investment cost, hourly power generation cost, commitment cost, start-up and shut-down cost and penalty cost of unserved demand for a period of one year. In addition, for the power generation model $C^{var*}(p_{t,g})$, linear cost is used. Constraint 5.2 limits the commitment to existing and invested generators, where as 5.3 manages the start-up and shut-down events. In addition, constraints 5.4

and 5.5 controls the minimum up and downtimes. Constraint 5.6 bounds the power generation between minimum and maximum generation limits, while constraints 5.7 and 5.8 set ramp-up, ramp-down, start-up and shut-down limits for all generators in all periods. Constraints 5.9 - 5.11 restrict the power flow through existing and candidate transmission lines following a DC model. The "Big M" method is used to linearise the non-linear DC equation, in which M represents a large constant. The linearisation also decouples the phase angles mathematically at nodes r and s, and enables the power flow $f_{k,l,t}$ over line l between r and s to correctly take the value zero, when the line l is not constructed (i.e., when $x_{k,l} = 0$). Constraint 5.12 bounds the phase angles and 5.13 sets the phase angle of reference (slack) bus to zero. Constraint 5.14 limits the renewable output where dispatchable renewables are considered with no penalty for curtailment. Constraint 5.15 restricts the unserved demand and finally, constraint 5.17 enforces the nodal power balance.

In addition, the multi-period formulation that considers a long planning horizon and makes investment decisions in multiple periods can be provided as follows. The planning horizon is divided into multiple periods with a constant step size (d), and for each period, a representative year is considered with an annualised capital cost, which is assumed to be at the beginning of the period. Consideration of a representative year reduces the size of the problem, and annualised capital cost ensures that expensive long-lived assets are not discriminated. Thus, the total cost for the representative year is multiplied by a factor α to account for the duration of the entire period. This factor α that is calculated using Eq. 5.18 also accounts for the discount rate and converts the future values to present values using the net present value (NPV) concept.

$$\alpha_k = \sum_{j=1}^{d} (1+\gamma)^{(k-1)d+j} \qquad k = 1..|\mathcal{K}| \tag{5.18}$$

$$\min \sum_{k \in \mathcal{K}} \alpha_k \Big[\sum_{i \in \mathcal{I}} C_i^I x_{k,i} + \sum_{t \in \mathcal{T}} C_t^{op} \Big] \tag{5.19}$$

$$s.t$$

$$x_{k-1,i} \leqslant x_{k,i} \qquad \forall k, i \tag{5.20}$$

$$Eq.\ 5.2 - Eq.\ 5.17$$

In this set of equations, k is the period; α is the multiplication factor; γ is the discount rate, d is the duration of the period in years and $x_{k,i}$ is the investment decision in period k. The objective function Eq. 5.19 minimises the discounted investment and operational costs. The building constraint Eq. 5.20 ensures that all the existing and invested assets are available for subsequent stages. Without loss of generality, retirements are ignored in this formulation.

In this regard, GTEP-UC consists a block structure which can be exploited to apply a decomposition technique. Fig. 5.1 illustrate this structure where investment decisions are considered as the complicating variables and inter-temporal constraints are considered as the complicating constraints. Since investment decisions have an impact on every

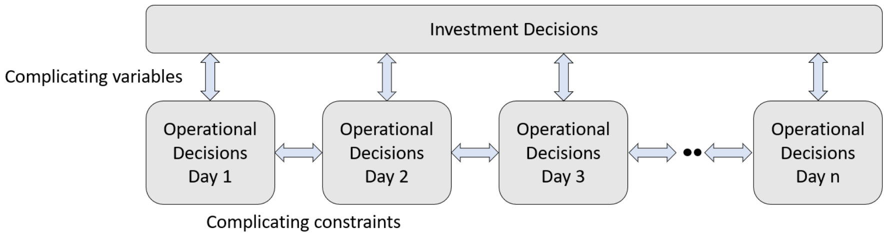

Investment Decisions
Complicating variables
Operational Decisions Day 1
Operational Decisions Day 2
Operational Decisions Day 3
Operational Decisions Day n
Complicating constraints

Figure 5.1: GTEP-UC problem structure

operational decision and fixing them in the operational level separate the problem into two parts: investment and operational, investment decisions can be considered as complicating variables. Similarly, if the inter-temporal constraints such as ramping limits and minimum up-down times at the boundaries of days or weeks are removed, the days and weeks become distinct allowing the operational problem to be divided into multiple sub-problems. Thus, the inter-temporal constraints can be considered as complicating constraints.

5.4 Scenario Decomposition Application

Scenario decomposition [104] was initially proposed for stochastic problems, where each scenario is an independent realisation of some stochastic processes. In the stochastic paradigm, the goal of scenario decomposition is to find a solution that is feasible for all the scenarios, even though only one scenario is expected to be realised. Thus, each scenario is given a weight that reflects the estimated probability of occurrence, and is solved as a separate sub-problem.

In contrast, GTEP-UC is deterministic, and aims to find an investment solution that is feasible throughout the planning horizon. Scenario decomposition is utilised to exploit the time decomposable structure of the GTEP-UC problem, where each sub-problem is a planning problem with a shorter horizon (e.g. a day or a week). These sub-problems are illustrated in Fig. 5.2, where all sub-problems are comprised of both investment and operational decisions. Because of the deterministic nature, the notion of "scenario" in GTEP-UC context refers to a daily or weekly sub-problem that is thought to be likely to happen. So, for the application of scenario decomposition to be successful in GTEP-UC context, the inter-temporal constraints that bounds daily and weekly scenarios must be tackled. This section aims to elaborate on the modifications that are essential to facilitate scenario decomposition in GTEP-UC context, and presents the decomposition framework for both single-period and multi-period GTEP-UC problems.

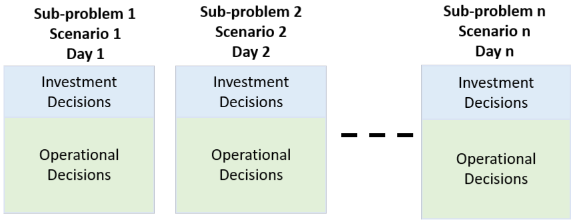


Figure 5.2: Time-based GTEP-UC decomposition with scenario decomposition

5.4.1 Single-Period GTEP-UC

As described in [104], the scenario decomposition method consists two main phases: candidate solution generation and candidate solution evaluation. In the candidate solution generation phase, independent scenarios are solved as separate sub-problems to generate potential solutions. In the candidate solution evaluation phase, these solutions are evaluated and eliminated to find the optimal solution.

Similarly, the proposed decomposition framework presented in Algorithm 4 consists two phases: candidate solution generation and candidate solution evaluation. In the candidate solution generation phase, the sub-problems are formulated as planning problems with shorter horizons by bringing complicating yearly investment decisions into each sub-problem. Since scenario sub-problems are (deterministically) certain to happen, they are formulated with unit weights, and the annualised capital cost is discounted[1] to account for the shortened horizon. The daily or weekly sub-problems are then solved separately to find candidate investment solutions (line 7). As optimal investment solutions of individual sub-problems provide the best solutions for themselves, the summation of sub-problem objective values provide a lower bound LB to the overall problem (line 8).

The complicating inter-temporal constraints are tackled by completely removing them near the boundary of each sub-problem (at the beginning and end of the shortened horizon). This removal eliminates the intra-day and intra-week dependencies between sub-problems and makes each sub-problem independent allowing them to be solved separately. As a result, initial conditions for a particular day are no longer passed from the previous day and the sub-problems are forced to initialise from an arbitrary state. While this arbitrary state can be either "switched off" or "switched on" with some generation for existing generators, for candidate thermal generators the initial state must be "switched off". Thus, if a particular generator is invested in every scenario, this arbitrary "commitment" state could accumulate start-up costs unnecessarily making the lower bound invalid. In addition, such initial state could also lead to over-commitment, as the invested generator is forced to be "on" for the required duration of minimum up time in every scenario. But in reality, the particular unit may have already served the required minimum up time during the preceding hours, days or weeks, and shutting down can be freely permitted.

Also, the arbitrary initial "generation" states could over-constrain the system flexibility and lead to unnecessary generation, because of the ramp up-down and startup-shutdown limitations. As a consequence, they could incur further additional costs. Therefore, in the sub-problem formulation, the following modifications are made to ensure that the lower bound is always valid. Note that, these modifications make the lower bound value even lower, as decisions in the first time period are allowed to take any value at zero cost.

---

[1] For example, if each sub-problem is a day, the daily capital cost can be found by dividing the annualised capital cost by 365.

Algorithm 4 Scenario Decomposition

```
Initialise: $UB^U = \infty, LB = -\infty, iter = 0, Pool = empty$
while $LB < UB^U$ do
    $iter+ = 1$
    % Candidate solutions generation
    $LB = 0 \qquad solutions = empty$
    for $s$ in $Sub\text{-}problems$ do
        $(obj, sol) = \mathrm{Solve}(s)$
        $LB+ = obj$
        $solutions = \mathrm{Include}(sol, solutions)$
    % Candidate Solution Evaluation
    for $sol$ in $solutions$ do
        $\zeta^L = 0$
        for $s$ in $Sub\text{-}problems$ do
            $s = \mathrm{Fix}(s, sol)$
            $obj = \mathrm{Solve}(s)$
            $\zeta^L+ = obj$
        if $\zeta^L < UB^U$ then
            $Pool = \mathrm{Add}(sol, Pool)$
            $\zeta^U = \mathrm{RollingHorizon}(sol)$
            if $\zeta^U < UB^U$ then
                $UB^U = \zeta^U$
    % Add integer cuts
    $Sub\text{-}problems = \mathrm{IntegerCuts}(solutions, Sub\text{-}problems)$
```

1. To avoid accumulating start-up cost in every scenario, start-up cost of thermal generators are disregarded in the first time period.
2. To avoid over-constraining the system flexibility, ramping constraints for the first time period are ignored allowing the thermal units to settle down to the required generation without any additional cost.
3. To allow realistic flexible start-up and shut-down status, the commitment decisions for the first time period are removed from the minimum up and down constraints in each sub-problem.

When choosing the sub-problem size, it is imperative to select a size that maximises the efficiency of the algorithm. If the chosen problem size is smaller than ideal, exclusion of many inter-temporal constraints and start-up costs will weaken the lower bound delaying the convergence. If the problem size is too large, the time required to solve sub-problems will be significant, leading to the algorithm being less efficient. Hence, a planning horizon of one day with daily capital cost is chosen for the experiments reported in this chapter. Since 24 hours is the longest minimum up and downtime for a standard thermal generator (e.g. coal) in a system, one day is reasonable to model the functionality of thermal units with less discrepancy.

In the candidate solution evaluation phase, the previously relaxed complicating variables and constraints are taken into account by fixing the generated candidate investment solutions and evaluating them individually. The investment solution with the lowest total system cost becomes the upper bound UB and the incumbent solution. Note that, every investment solution is considered as a feasible solution to the full problem, because unserved demand is penalised rather than being considered an infeasibility in the formulation. The algorithm then prohibits revisiting each of the evaluated solutions using the integer cut given in Eq. 5.21 (line 23), which eliminates exactly one solution at a time. The integer cuts force different solutions in the next iteration that are more expensive for the sub-problem that generated them, but globally more optimal. Therefore, they raise the sub-problem lower bounds, and consequentially the global lower bound LB converging the algorithm when global LB exceeds global UB.

$$\sum_{j:\hat{x}_j=1}(1-x_j)+\sum_{j:\hat{x}_j=0}x_j \geqslant 1 \qquad \forall \hat{x}\in\mathcal{S} \tag{5.21}$$

Ideally, to evaluate the solutions and generate an upper bound UB for any given set of investment decisions, unit commitment (UC) model must be solved for the entire planing horizon. This is a computationally-intensive process and rather impractical even with the investment variables taking fixed values. In our framework, we therefore check convergence by calculating an upper bound on the global upper bound UB_U, instead of calculating the actual global upper bound UB. By modifying the algorithm to converge when global lower bound LB exceeds UB_U (line 2), the framework ensures that the global optimal solution has been already realised at some stage in the solution process.

Specifically, for any investment solution i, let ζ_i^U be an upper bound and ζ_i^L be a lower bound on the optimal total cost ζ_i^* of the GTEP-UC problem with the investment solution i. The lower bound on total systems cost ζ_i^L is generated by fixing the investment solutions in the modified sub-problems (line 14) and resolving them. Since individual sub-problems produce the lowest cost for themselves, the sum of the resulting objective values provides a lower bound on the exact total cost of investment solutions (line 16). The upper bound ζ_i^U is obtained by utilising the rolling horizon technique (line 19) discussed in Section 3.3.2. Since rolling horizon force operational decisions of the previous day to next day, the summation of objective values provides an upper bound on the total system costs.

Recall that the upper bound on the current-best solution UB^U is the minimum value of upper bounds across the set of all candidate option solutions $\mathcal{I}$. Hence, any solution i with $\zeta_i^L \leq UB^U$ (line 17) becomes a potential optimal solution, as it could satisfy the condition $\zeta_i^L \leq \zeta_i^* \leq UB \leq UB^U$, where UB is the current global upper bound, for which the true optimum value is unknown (as also holds for ζ_i^*). Thus, a solution i with $\zeta_i^L \leq UB^U$ is added to a solution pool $\mathcal{K}$ (line 18), and is evaluated through rolling horizon approach to determine ζ_i^U (line 19). Then, if $\zeta_i^U \leq UB^U$ (line 20), the UB^U is also updated to the newly found ζ_i^U (line 21).

Once the global lower bound LB exceeds UB^U, then the globally optimal solution has been found and will be one of the solutions in pool $\mathcal{K}$. However, which of the solutions in $\mathcal{K}$ is the global optimal is not immediately known. Heuristically, the best solution can be considered as the solution with the lowest ζ^U value, or alternatively, a unit commitment model for the entire planning horizon can be calculated for the few potential solutions.

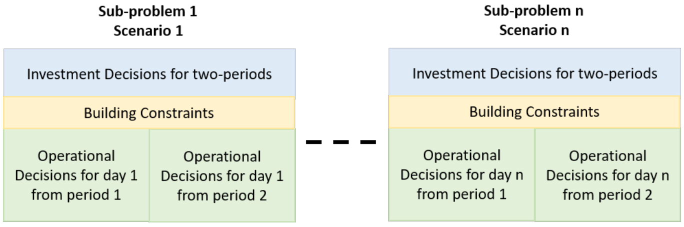

Sub-problem 1
Scenario 1
Investment Decisions for two-periods
Building Constraints
Operational Decisions for day 1 from period 1
Operational Decisions for day 1 from period 2
Sub-problem n
Scenario n
Investment Decisions for two-periods
Building Constraints
Operational Decisions for day n from period 1
Operational Decisions for day n from period 2

Figure 5.3: Scenario decomposition for two-period GTEP-UC problem

5.4.2 Multi-Period GTEP-UC

When generating candidate solutions with two or more sequential investment periods, the resulting investment solutions are comprised of decisions for each of the periods. Hence, to apply the scenario decomposition approach, each sub-problem must contain the investment decision variables from every investment period. Constraints must act on these variables in combination if they are to take sensible values. So, we propose to combine daily scenarios from every investment period into each of the sub-problems. In other words, we propose to generate initial sub-problems in a way that each sub-problem consists at least one scenario from each period. This unification allows constraints that couple different periods to be added into the sub-problems, as they now contain investment decisions and daily operations from all periods. Nevertheless, each daily scenario is independent of others in the sub-problem except for the common dependence on the investment decision variables and the investment coupling constraints that involve these variables.

In this chapter, the SD algorithm is applied to a two-period formulation. The two-period SD algorithm therefore commences with pairs of scenarios rather one. To determine the initial pairs and form the set of sub-problems, we utilised corresponding days from each period. For example, day 1 from each period is combined into one sub-problem and the building constraint is added across all of them. Sub-problem formulation with two periods are depicted in Fig. 5.3. Once sub-problems with pairs of scenarios are formulated, the candidate solution generation phase is executed similar to single-period application. To ensure a valid lower bound, each sub-problem (both scenarios) is modified by removing the initial conditions (refer to the previous section). As all the sub-problems choose the best two-period solution for themselves, the lower bound is found by summing the objective values of sub-problems.

In the candidate evaluation phase, the evaluation is carried out in a similar fashion to the single-period problem. To generate ub^L, each investment solution is evaluated by fixing the investment decisions in the sub-problems and resolving them. And, to generate ub^U rolling horizon approach is employed. In rolling horizon approach, no overlap duration is considered between periods, because only the first year of each period is taken into consideration. Once evaluated, investment solutions with $ub^L \leq UB_U$ are added to a solution pool $\mathcal{K}$, and UB_U is updated accordingly. Similarly to the approach taken to the single-period problem, the evaluated solutions are excluded from the search space by means of integer cuts, which include decisions from all periods.

5.5 Bender's Decomposition for Single-Period GTEP-UC

This section presents the application of Bender's decomposition to the GTEP-UC problem for comparison with the scenario decomposition approach. The method is applied similar to Kamalinia *et al.* [121] and Schwele *et al.* [123], in which the Bender's decomposition was utilised to decompose a stochastic GTEP-UC problem. In this chapter, Bender's decomposition is applied to a deterministic GTEP-UC problem where, inter-temporal constraints are tackled using both classic and multi-cut Bender's. In addition, minimum up-down constraints and start-up costs are taken into consideration unlike the previous applications [121, 123]. Bender's decomposition operates as a master problem, and one or more slave sub-problems iterating between the two until the stopping criterion is met. Therefore, this section presents the details of the master problem, slave sub-problems and the Bender's algorithm.

Master problem

In Bender's decomposition, to obtain dual information from the sub-problems, the sub-problems must be an LP problem. As a consequence, all integer variables must be present in the master problem [2]. In the GTEP-UC formulation, the integer variables or rather binary variables are the yearly investment decisions and hourly commitment, start-up and shut-down decisions for a period of one year. Thus, these binary decisions with their corresponding constraints constitute the master problem. Formulation for the master problem can be given as follows.

The objective function 5.22 represents the summation of annualised capital cost, commitment, start-up and shut-down costs and operational cost α for slave sub-problem s, and constraints 5.23 - 5.26 present the commitment limits for thermal generators, start-up shut-down limits, minimum up and minimum downtime limits. In addition, constraint 5.27 bounds the operational costs α to be non-negative. And finally, constraint 5.28 represents the Bender's cut which get added in each iteration j until current iteration m, where $f_s^{sub(j)*}$ are the sub-problem objective function values in previous iterations and $\lambda, \mu, \nu, \omega$ are the dual values for investment, commitment, start-up and shut-down decisions obtained from the slave sub-problems. When the number of sub-problems is more than one, the approach results in a multi-cut Bender's formulation, where a cut is added from each sub-problem.

$$\min f^{mas} = \sum_{i\in\mathcal{I}} C_i^I x_i + \sum_{s\in\mathcal{S}} \left(\sum_{t\in\mathcal{T}} \sum_{g\in\mathcal{G}} + C_g^c u_{s,t,g} + C_g^u v_{s,t,g} + C_g^d w_{s,t,g} \right) + \alpha_s \tag{5.22}$$

---

[2] Unless we relax the problem in some way that enables integer variables to be replaced by continuous variables in the sub-problems

$$s.t.$$

$$u_{s,g,t} \leqslant x_g \qquad \forall s,g,t \qquad (5.23)$$

$$u_{s,g,t} - u_{s,g,t-1} = v_{s,g,t} - w_{s,g,t} \qquad \forall s,g,t \qquad (5.24)$$

$$\sum_{i=t-T_g^U+1}^{t} v_{s,g,i} \leqslant u_{s,g,t} \qquad \forall s,g,t \qquad (5.25)$$

$$\sum_{i=t-T_g^D+1}^{t} w_{s,g,i} \leqslant 1 - u_{s,g,t} \qquad \forall s,g,t \qquad (5.26)$$

$$\alpha_s \geqslant 0 \qquad \forall s \qquad (5.27)$$

$$\begin{aligned}\alpha_s \geqslant f_s^{sub(j)*} &+ \sum_{i \in I} \lambda_{s,i}^{(j)}(x_i - x_i^{(j)}) \\ &+ \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} \mu_{s,g,t}^{(j)}(u_{s,g,t} - u_{s,g,t}^{(j)}) \\ &+ \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} \nu_{s,g,t}^{(j)}(v_{s,g,t} - v_{s,g,t}^{(j)}) \\ &+ \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} \omega_{s,g,t}^{(j)}(w_{s,g,t} - w_{s,g,t}^{(j)}) \qquad \forall s, m \geqslant 2, j = 1....m-1 \qquad (5.28)\end{aligned}$$

Slave Sub-problem

Since both investment and commitment decisions are taken to the master problem, the slave sub-problem is essentially an economic dispatch model with ramping constraints, where the dispatch decisions are bounded by the unit commitment decisions.

$$\min f_s^{sub} \sum_{t \in \mathcal{T}} \left(\sum_{g \in \mathcal{G}} (C_g^{var*}(p_{s,t,g}) \right) + \sum_{n \in \mathcal{N}} C^p q_{s,t,n}) \qquad (5.29)$$

$$s.t.$$

$$Eq.\ 5.6 - Eq.\ 5.17$$

$$x_i = x_i^{(m)} \quad : \lambda_i \qquad \forall i \in I \qquad (5.30)$$

$$u_{s,g,t} = u_{s,g,t}^{(m)} \quad : \mu_{s,g,t} \qquad \forall g,t \qquad (5.31)$$

$$v_{s,g,t} = v_{s,g,t}^{(m)} \quad : \nu_{s,g,t} \qquad \forall g,t \qquad (5.32)$$

$$w_{s,g,t} = w_{s,g,t}^{(m)} \quad : \omega_{s,g,t} \qquad \forall g,t \qquad (5.33)$$

The objective function 5.29 presents the cost of generation and penalty for unserved demand, where $C_g^{var*}(p_g)$ is assumed to be linear. In addition, constraints 5.6 - 5.17 represent the operational constraints: generation limits, DC power flow, renewable generation constraints and nodal power balance. Finally, constraints 5.30 - 5.33 fix the decisions obtained from the master problem, in which $\lambda, \mu, \nu, \omega$ represent the associated dual prices.

This slave sub-problem can be formed either as one big LP problem (classic) or multiple sub-problems (multi-cut). If formulated as multiple sub-problems, similar to Section 3.3.2, an overlap duration can be considered to account for inter-temporal ramping constraints and satisfy boundary conditions. Then the operational cost and dual prices for the overlap duration can be discarded when computing f_s^{sub}. This overlap will eliminate the need for feasibility cuts in the Bender's formulation [3].

Therefore, Bender's decomposition is formulated as both classic and multi-cut Bender's to investigate its performance with scenario decomposition. While the classic Bender's considers one sub-problem for the entire duration, multi-cut Bender's considers several sub-problems with an overall duration of 48 hours (24 hrs sub-problems with 24 hrs overlap). These variations are achieved by controlling the set of time periods $\mathcal{T}$ and the number of sub-problems in the above formulation.

Bender's Decomposition Algorithm

1. Initialise the Benders algorithm with iteration count set to $m = 1$
2. Solve the Master problem defined by Eq. 5.22 - 5.28, to obtain the investment decision $x^{(m)}$ and the lower bound LB where $LB = f^{mas*}$. For the first iteration solve the master problem without the Bender's cut (Eq. 5.28).
3. Fix the investment decisions, and solve the sub-problems $s \in \mathcal{S}$ to obtain the operational cost f_s^{sub*} and the dual values $\lambda, \mu, \nu, \omega$. For multi-cut Bender's solve the sub-problems sequentially.
4. Calculate the upper bound UB where,
   $UB = f^{mas*} - \sum_{s \in \mathcal{S}} \alpha_s + \sum_{s \in \mathcal{S}} f_s^{sub*}$
5. If $|UB - LB| \leqslant \epsilon$ where ϵ denotes the Bender's gap, exit with the current solution, else update the iteration count to $m = m + 1$ and go to step 2.

5.6 Experimental Analysis and Discussion

This section provides details of the experimental analysis conducted to evaluate the performance of the proposed scenario decomposition framework in both single-period and multi-period setting.

5.6.1 Experimental Setup

To analyse the performance of scenario decomposition framework, four test cases: 6 bus, 14 bus, 18 bus and 24 bus, with different numbers of nodes were utilised. The details of the test cases are provided in Appendix A, and a summary of test cases is provided in Table 3.2. In addition, a summary of model attributes is provided in Table 3.3. For the single-period formulation, a representative year was considered with hourly time resolution. For the multi-period formulation, a 20 year horizon was considered with two 10 year periods and hourly time resolution. In each period, it was assumed that the representative year with hourly resolution is at the beginning of the duration. The total cost for the 10 year duration was taken into account through the co-efficient α.

To analyse the performance with different problem sizes, each test case was represented with 24 days (two days from each month), 84 days (one week from each month) and

---

[3] Feasibility cuts are not required with the current formulation, as unserved demand is considered as a penalty rather than as an infeasibility

Table 5.1: Performance of scenario decomposition with single-period

Test Case	Instance	Opt. gap %	Res. gap %	Num. sol.
6 Bus	24	0	0.1225	2
	84	0	0.0122	1
	365	0	0.3012	2
14 Bus	24	0	0.1171	3
	84	0	0.0416	1
	365	2.195	0.2059	4
18 Bus	24	4.430	0.0045	1
	84	5.972	0.0029	1
	365	8.038	0.0013	1
24 Bus	24	6.880	0.0794	3
	84	7.566	0.0243	1
	365	10.211	0.1013	2

364 days (to allow groups of 4 scenarios, discussed in next chapter). To estimate the operational cost for a year of 365 day duration, a linear scaling factor was applied. Thus, altogether 4×3 test instances were considered to analyse the performance of the algorithm. For all instances, the penalty cost was set to 1000 \$/MWh of unserved energy, and an annual monetary discount rate of 10% was considered. For the multi-period formulation a load growth of 15% was uniformly applied to the demands in the second period.

All algorithms and models were implemented using Python 2.7 and the optimisation problems were solved using Gurobi 8.0 [131] on a computing node that utilises eight cores clocking at 2.70 GHz with 16 GB RAM. In addition, Gurobi's default MIP gap 1×10^{-4} was used, and the *number of thread count* parameter was set to equal the number of cores (in MIP models). For each test case, the maximum time for the entire algorithm (the summation of all solving times provided by Gurobi) was limited to 22 hours. For Bender's decomposition, an additional time limit of 10 hours was set to the master problem.

5.6.2 Performance of Scenario Decomposition

Table 5.1 and Table 5.2 summarise the computational results obtained with single-period and two-period formulations. In these tables, the columns indicate (respectively): the test case identifier; the number of days in the problem instance; optimality gap between lower bound (LB) and upper of the upper bound (UB_U), as a percentage, at the end of the maximum time limit; the residual gap between the lower of the upper bound (UB_L) and the upper of the upper bound (UB_U), as a percentage; and, finally, the number of potential solutions in the pool at the end of the computation period.

With reference to Table 5.1, the smaller test cases with fewer days in the instance have managed to converge within the given time limit, but the larger test cases have optimality gaps at the end of the maximum time limit. The optimality gap for the 24 bus 365 day instance is the largest of such gaps. This is expected given the size of the problem and the limited computation time allocated to solve it. On the other hand, the residual gaps obtained from all instances were interestingly small leading to only small number of solutions in the pool at the end of the solving time period. This suggests

Table 5.2: Performance of scenario decomposition with two-period

Test Case	Instance	Opt. gap %	Res. gap %	Num. sol.
6 Bus	24	1.990	0.1347	4
	84	5.811	0.0217	1
	364	8.754	0.2592	3
14 Bus	24	1.282	0.1051	3
	84	2.058	0.0342	1
	364	2.609	0.1676	4
18 Bus	24	5.650	0.0033	1
	84	6.171	0.0021	1
	364	7.952	0.0007	1
24 Bus	24	7.757	0.0802	14
	84	7.496	0.0226	5
	364	9.734	0.1315	1

that the global optimal solution can be determined after executing only a few full unit commitment simulations.

The value of the residual gap strongly depends on the characteristics of the problem such as the demand at the beginning of the sub-problem horizon and the minimum up-down times. For example, if the time-based decomposition is carried out when the demand is at its peak, the residual gap is likely to be significantly higher, as units are likely to start up and the cost with that decisions is ignored (recall that in the SD algorithm, start-up costs for the first period in the sub-problem horizon are neglected to ensure a valid lower bound). Since the days were split when the demand is reasonably flat in this set of experiments, the small cost difference is mainly due to the discrepancies in minimum up-down times and ramping limits. Nevertheless, instances with 365 days show larger residual gaps with more potential solutions in the pool. As each day is represented as a sub-problem, when the number of days increases, the presence of numerous sub-problems amplifies the negative effect of the relaxations and causes UB_L to be substantially lower than UB_U. Therefore, when the number of sub-problems is large, it is likely that more potential optimal solutions will be in the pool, and in turn will require more (computationally expensive) full unit commitment simulations to be carried out.

Table 5.2 shows that none of the two-period formulations have converged. This is in part due to the large number of sub-problems. While smaller instances like 6 bus 24 day have exited with small optimality gaps (1.99%), larger instances like 24 bus 364 days show significant gaps (9.7%) at the end of the maximum time limit. In addition, two-period results show an increase in the number of potential solutions but the residual gaps do not exhibit substantial differences between single-period and two-period cases. Considering the multi-period case and the 24 bus 24 day instance, there are 14 potential solutions, and because the algorithm is still far away from reaching optimality (7.757%), it is likely that there are more potential solutions that have not been discovered yet. This larger number of close-to-optimal solutions is mainly because of the increased number of investment decisions in the problem with the two periods. Therefore, multi-period formulations will require a lot more full unit commitment simulations to identify the optimal solution, compared to the single-period instances.

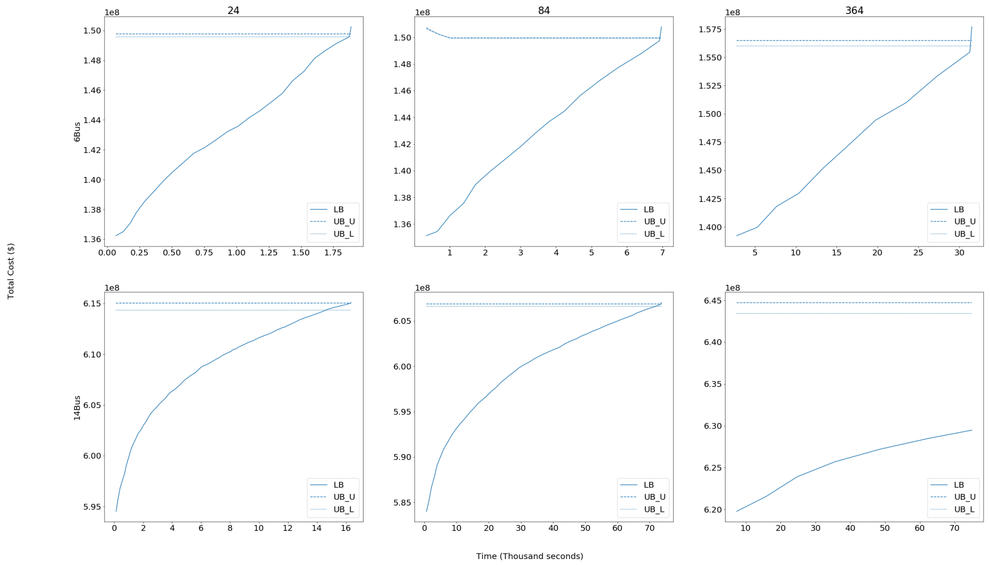


Figure 5.4: Performance of scenario decomposition

To analyse the progression of the scenario decomposition algorithm, the change in bounds were examined with respect to time. This change is illustrated in Figure 5.4 for single-period formulations, in which the bound calculations LB, UB_L and UB_U are depicted along with the time it was calculated. As shown in Fig. 5.4, unlike other decomposition algorithms, the scenario decomposition framework has found high-quality solutions providing a tight upper-bound in the first few iterations. In fact, except for the 6 bus 84 day instance, the best-known solution was found in the first iteration itself. This is beneficial when solving large-scale optimisation problems, as good quality solutions can be realised without having to wait for a great deal of wall-clock time to pass. Nevertheless, the algorithm relies on the lower bound to prove optimality and finally converge. Since the initial lower bound is weak and also improves slowly, the convergence is often delayed.

Recall that the lower bound is improved by eliminating solutions from the search space via integer cuts. As each integer cut prohibits only one investment solution at a time, to remove a particular solution from the search space, first, it must be visited (evaluated). Naturally, sub-problems produce solutions that are optimal only to themselves. So, for the algorithm to improve the global lower bound and finally converge, the SD algorithm must visit many or all of the solutions that lie in between the individually preferred solution of a sub-problem and the globally-optimal solution to eliminate them. If there are many solutions in between the preferred solutions and the optimal solution, due to the weak nature of the integer cuts, many cuts and iterations are required to eliminate these solutions and close the optimality gap. With two-period formulations, there will be even more solutions to visit, because of the increased number of investment decisions.

Fig. 5.5 aims to illustrate this issue by presenting the solutions generated by a subset of daily scenarios from the 6-bus 24-day instance. Fig. 5.5 depicts the solutions generated by sub-problems 18-23 and their progression from the preferred solution to the globally

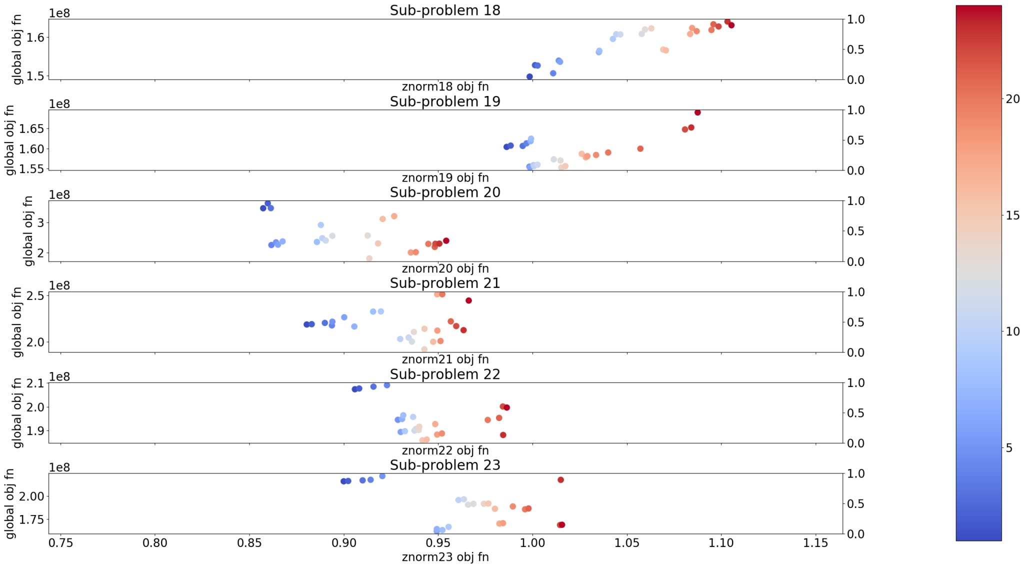
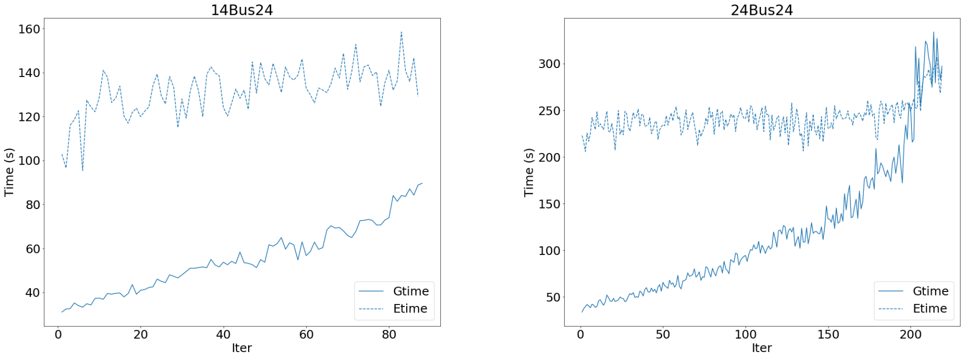


Figure 5.5: Progression of solutions in individual sub-problems

best-known solution. The solutions are depicted with its total cost (global objective) on the y-axis and normalised objective values for that sub-problem on the x-axis. The normalisation was carried out by dividing the sub-problem objective function value with its objective function value in the globally best-known solution. This normalisation is provided in Eq. 5.34, where $\mathcal{S}$ denotes the set of sub-problems and $\mathcal{I}$ denotes the set of iterations. Hence, for all the sub-problems, the objective function value for the globally best-known solution lies at 1.0 and all the solutions which has a value less 1.0 are *pre-optimal* solutions while the rest are *post-optimal*. The data-point colour indicates in which iteration the solution was generated.

$$z^{norm}_{s,i} = \frac{z^{obj}_{s,i}}{z^{*obj}_{s}} \qquad \forall s \in \mathcal{S}, i \in \mathcal{I} \tag{5.34}$$

With reference to Fig. 5.5, the first solution provided by sub-problems 18 and 19 are close to the best-known solution for the instance as a whole, where as first solutions provided by sub-problems 20 to 23 have sub-problem objective function values that are quite distant from those for the globally best solution. It can also be observed that the early-found solutions for the majority of the sub-problems lie well before the best known solution, leading to a weak initial lower bound. Sub-problems eliminate their next-best preferred solutions sequentially as the algorithm iteratively progresses towards the globally optimal solution. Sub-problems 20, 21 and 22 have explored many of their pre-optimal solutions before convergence, and sub-problems 18, 19 and 23 have explored all its pre-optimal solutions. The algorithm will converge when a sufficient number of sub-problems have explored all their pre-optimal solutions, because this is necessary in order for the lower bound to be raised to a sufficiently high value. If there are many solutions between the preferred solutions for the sub-problems and the global solution applied to



(a) 14 Bus System

(b) 24 Bus system

Figure 5.6: Phase performance

each sub-problem, many integer cuts will be required to remove these solutions and close the optimality gap.

The addition of the integer cuts as the algorithm progresses acts to increase the computational difficulty in two ways. First, adding the integer cuts increases the time that is required to solve individual sub-problems. This is demonstrated in Fig. 5.6a and Fig. 5.6b, which illustrate the time taken by the candidate solution generation phase (Gtime) and evaluation phase (Etime) at each iteration for 14 bus 24 day and 24 bus 24 day instances. As depicted, the time taken by the candidate solution generation phase shows an increasing trend with the growing number of iterations. With 24 bus 24 day instance (Fig. 5.6b), the time taken by candidate solution generation phase exceeds the time taken by evaluation phase when the algorithm has reached a large number of iterations. Since the number of integer cuts in the sub-problem increments at the end of each iteration, the time required to solve individual sub-problem in candidate solution generation phase also increases. Second, adding an integer cut is a computationally intensive task in itself. Each cut is added only after a computationally expensive evaluation step, hence to generate many cuts a large number of expensive evaluations must have been carried out. According to Fig. 5.6a and 5.6b, it is clearly visible that the time spent on the evaluation phase is fairly constant, but this time is significant compared to the candidate solution generation phase. Since each generated solution is evaluated by solving the set of sub-problems repeatedly, a very substantial period of time is spent evaluating the solutions prior to adding the cuts of the sub-problems.

An important factor contributing to the weak initial lower bound and its slow improvement is the diversity between scenarios. If scenarios are similar to each other and prefer the same set of investments solutions, the summation of individual scenario optimal objective values will provide a stronger lower bound. However, in reality, daily scenarios have different peak demands and availability in renewable energy-based generation, so they tend to produce different investment solutions with a different capacity mix. As a result, when the scenarios are completely different to one another, the resulting lower

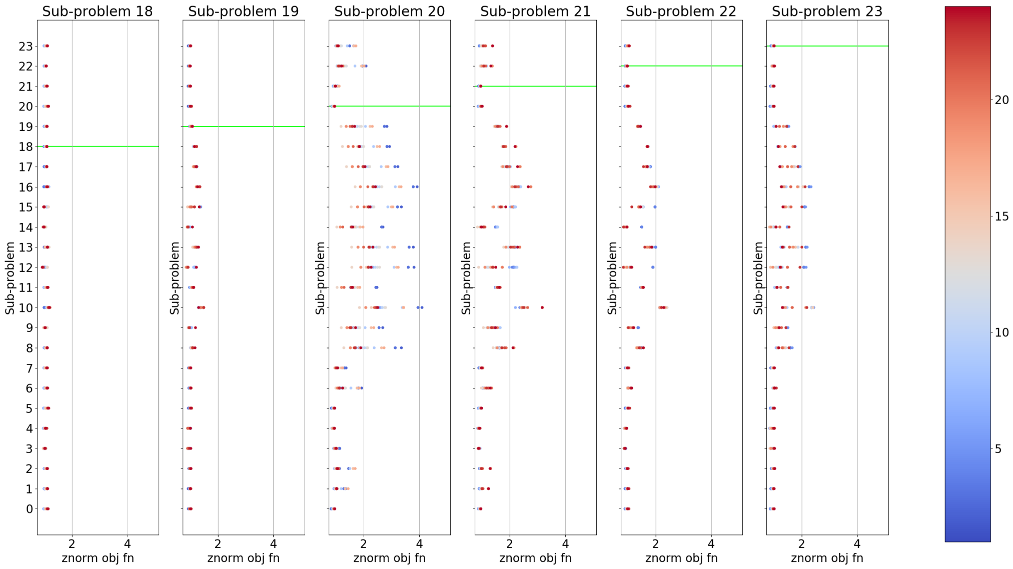


Figure 5.7: Solution performance on other sub-problems

bound becomes very weak due to the contrasting nature in the investment solutions. Besides, the difference in investment solutions also causes integer cuts to have only a limited impact on the other scenarios, because solutions that are cut-off by a particular scenario may not be preferred by (i.e., are not relevant to) the other scenarios. Thus, for a large portion of the process, scenarios continue to explore around their own regions until all or many investment solutions combinations around their preferred solutions are exhausted. As a result, cuts generated by diverse scenarios only provide limited assistance towards improvements in the overall lower bound and the algorithm's convergence.

The diversity in scenarios is illustrated in Fig. 5.7 by showing the performance of solutions generated by a particular sub-problem on other sub-problems. Fig. 5.7 illustrates the solutions generated by sub-problems 18 to 23 and their objective function values when applied to the other sub-problems. The y-axis denotes the sub-problem number, and the x-axis denotes the normalised sub-problem objective function value for the sub-problem in the plot title. A substantial number of solutions generated by sub-problems 20–23 have no merit for the other scenarios; they give rise to very poor objective function values for these other sub-problems, especially during the earlier iterations of the algorithm. This indicates that these scenarios are very different to the other scenarios, and they produce very different investment solutions: as a result, the penalty cost for unmet demand drives the normalised objective values of other scenarios extremely high. In contrast, sub-problems 18 and 19 have provided good quality solutions for the full global instance throughout the algorithm's progression — almost all of their generated solutions have a normalised global objective function value close to 1.0. These results indicate that scenarios 18 and 19 are what could be called "principal", "non-extreme" or "best compromise" scenarios, that give rise to investments that are broadly similar to those in the globally-optimal solution.

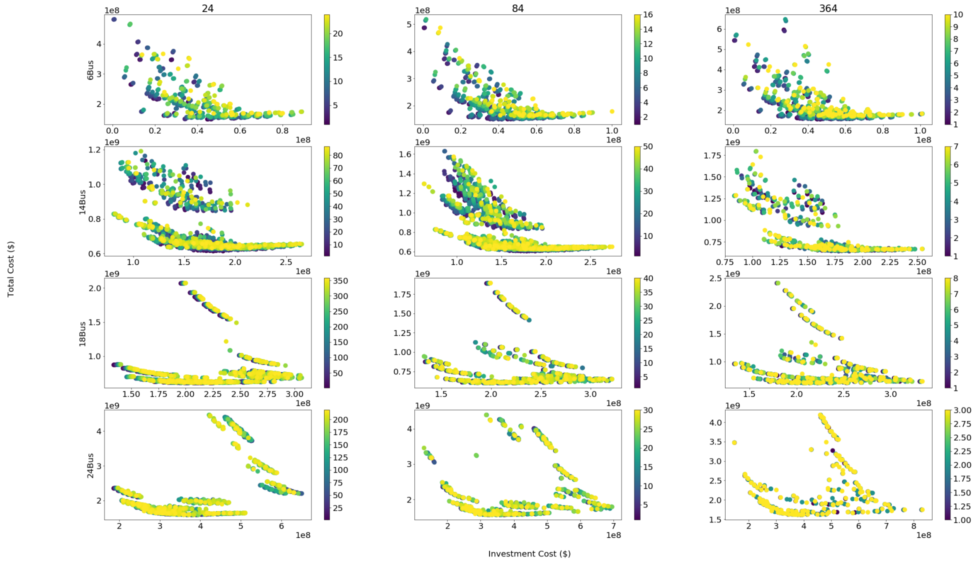


Figure 5.8: Explored solution space

These results are also in consistent with the previous discussion related to Fig. 5.5, where it was noted that sub-problems 20–23 have initial preferred solutions that are quite distant in objective function terms from the globally preferred solutions, while sub-problems 18 and 19 are very near-optimal from the beginning of the algorithm. The supply and demand patterns of sub-problems 20–23 are different to those of the other sub-problems and in particular the “best compromise” scenarios like those of sub-problems 18 and 19. Because of this, the sub-problems 20–23 need many cuts to be added before their remaining next-best solutions are near the global optimum (by sub-problem objective function value). As a result, we conclude that diverse scenarios delay the algorithm convergence. This is a finding of major importance when considering algorithm enhancements in the subsequent chapter of this thesis.

Diverse scenarios also cause many poor quality solutions to be evaluated in the process of reaching globally best-known objective. While diverse sub-problems themselves produce poor quality solutions and must evaluate all or many surrounding poor quality solutions to reach globally best-known objective, they also cause the minority who produced good quality solution and preferred solutions around globally best known objective to explore solutions that are both individually and globally worse, due to the iterative nature in the algorithm. Fig. 5.5 illustrate this phenomenon, where both pre-optimal solutions generated by sub-problem 20–23 in early iterations and post-optimal solutions generated by sub-problem 18–19 in late iterations shows extremely high total cost values (global objective function value). Fig. 5.8 depicts the outcome of exploring a large solution space, for each of the different test cases and instances, in terms of the investment cost of solutions and the total costs (UB_L) of solutions. These costs are calculated during the evaluation step. The data-point colour indicates the iteration in which the solution was generated. Darker-coloured data points relate to earlier iterations.

Table 5.3: Performance of Bender's decomposition

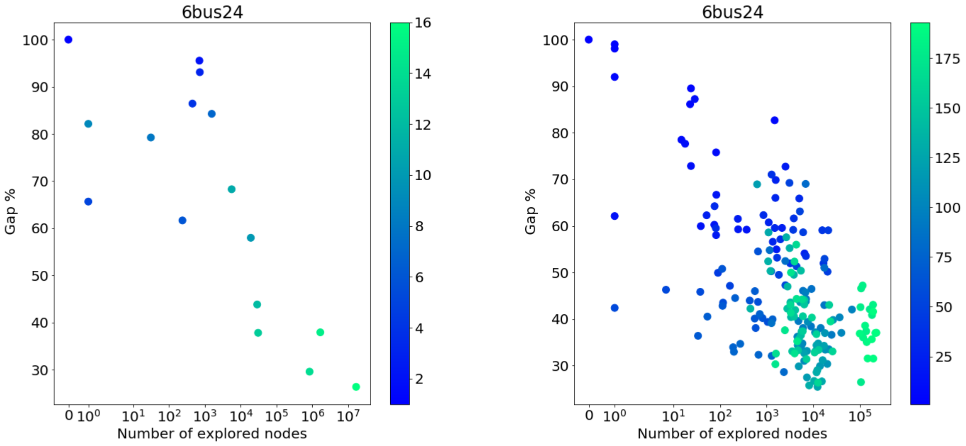
Test Case	Instance	Num Int Vars	Single Sub-problem Gap	Single Sub-problem Reason	Multiple Sub-problems Gap	Multiple Sub-problems Reason
6 Bus	24	12,124	25.4	time out	26.4	out of mem.
	84	42,364	33.6	time out	55.0	out of mem.
	364	183,484	49.9	time out	82.8	out of mem.
14 Bus	24	19,063	71.3	time out	62.8	out of mem.
	84	66,583	63.9	time out	86.2	master time out
	364	288,343	84.8	out of mem.	90.5	out of mem.
18 Bus	24	22,529	3.1	time out	4.6	out of mem.
	84	78,689	5.2	time out	13.8	out of mem.
	364	340,769	24.8	out of mem.	31.0	out of mem.
24 Bus	24	31,190	6.7	time out	17.4	out of mem.
	84	108,950	10.8	time out	37.5	out of mem.
	364	471,830	72.9	out of mem.	85.6	out of mem.

As shown in Fig. 5.8, a wide range of total costs and investment costs has been explored by the algorithm. This is the case for every instance, and indicates that the algorithm has done a large sweep of the solution space. At the beginning of the process (darker-colour data points), there are many evaluated solutions that are far beyond optimality. These extreme solutions are a result of extreme sub-problem pre-optimal solutions, and typically represent either too few or too many energy system investments for the problem as a whole (i.e., extremely high total costs due to a large amount of unmet demand or to an over-investment in network assets). Evaluation of such extreme solutions is a waste of computational effort, and a major factor delaying convergence. Nevertheless, the figure also illustrates an optimal frontier in each test case instance, and that very good solutions have tended to be found early in the solution process. This offers strong potential for the design of heuristics; if the algorithm faces converging issues and has explored numerous solutions, this frontier could be exploited to determine the best solution heuristically.

5.6.3 Performance of Bender's Decomposition

Table 5.3 summarises the obtained results with Bender's decomposition for both single and multi sub-problem approaches, where the optimality gap at the end of the algorithm and the reason for algorithm termination is provided. As shown in the Table 5.3, Bender's decomposition terminates with significantly larger optimality gaps compared to scenario decomposition approach. While instances with 24 days have provided comparatively smaller optimality gaps, larger instance like 364 days have terminated with optimality gaps as large as 90.5%. The main reason for such a huge optimality gap is the early exit caused by insufficient memory. Because the master problem consists a substantial number of integer variables (shown in Table 5.3), the problem has become practically intractable.

When comparing the two approaches, almost all the instances with multiple sub-problems including smaller instances like 6 bus 24 days terminate due to insufficient memory. Although the master problems in these instances are small, the set of cuts that get added at the end of each iteration have increased the memory requirement. In



(a) Multiple sub-problems

(b) Single sub-problem

Figure 5.9: Number of nodes explored with Bender's decomposition

particular, multiple cuts have increased the number of branch and bound nodes[4] that need to be explored to solve the master problem. So, after a certain number of iterations, the branch and bound tree explodes terminating the algorithm. Fig. 5.9a illustrates this issue, where the Bender's optimality gap and the number of nodes explored at each iteration are shown for the 6 bus 24 days instance with multiple sub-problems (multiple cuts). The colour ramp indicates the iteration at which the optimality was obtained. According to the figure, for the optimality gap to reduce below 30%, around 1 million nodes need to be explored. For larger instances, this number could be extremely large.

In contrast, Bender's approach with single sub-problem (single-cut) is more efficient with memory. As shown in Fig. 5.9b, the same instance with a single sub-problem has achieved similar optimality gap by exploring a substantially fewer number of nodes. However, this performance is achieved at the cost of a higher number of iterations (shown in the colour ramp). While the Bender's approach with multiple sub-problems reached 26.4% gap in 16 iterations, the single sub-problem approach has taken more than 100 iterations to reach the same optimality gap. This differences in performance with single and multiple sub-problems are also visible in Fig. 5.10, which illustrates the optimality gap at the end of each iteration with respect to time. From Fig. 5.10, it is evident that multi-cut Bender's reduces the gap drastically with fewer iterations (data points), while single-cut Bender's progresses slowly towards optimality.

When the time duration is considered, many smaller instances give rise to similar algorithm performance for both approaches. Although multi-cut Bender's reduces the gap drastically with few iterations, it explores a large number of nodes in an iteration taking a significant period of time. In contrast, single-cut Bender's achieves the same gap via many

---

[4] Mixed-integer problems are generally solved using the branch and bound method, where the feasible solution space is partitioned into a set of sub-spaces by branching on the variable, i.e, restricting the range of the variable [145]. Then, the linear programming relaxation of the problem is solved with the restriction, where each linear problem corresponds to a node in the enumeration tree. The branch and bound method attempts to prune this tree systematically [145].

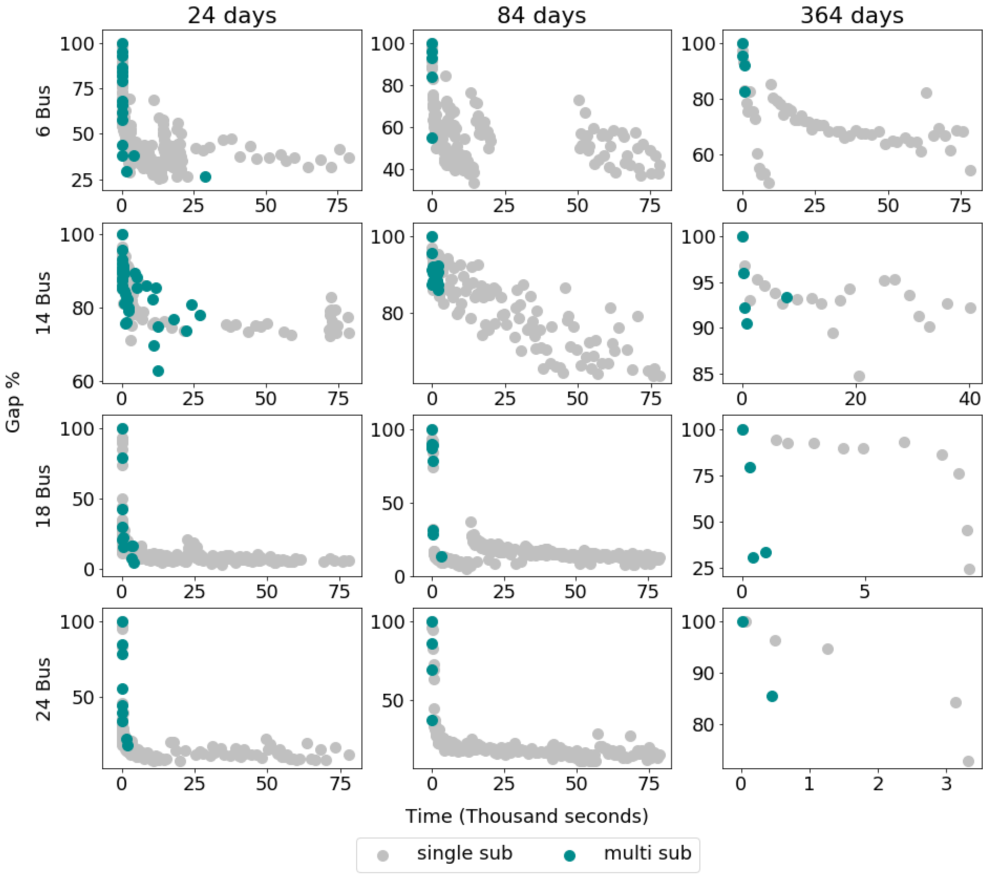


Figure 5.10: Performance of Bender's decomposition

iterations by exploring a smaller set of nodes at a time, thus taking substantially a shorter period of time per iteration. For larger instances like 364 days however, the performance differs significantly. Convergence rate of single-cut Bender's is rather slow compared to the multi-cut Bender's, because solving 364 days at once is computationally more expensive, and only one cut is added per iteration. As a consequence, the majority of the instances with single-cut Bender's exits because of the time limitation in the algorithm, while almost all the instances with multi-cut Bender's terminate because of the memory shortage.

In this regard, it is evident that scenario decomposition approach is computationally more efficient compared to the Bender's decomposition method. Unlike Bender's decomposition, the scenario decomposition approach provides high-quality solutions with tighter bounds in the first few iterations. Components of the SD algorithm also have very little dependency on each other allowing the approach to be parallelised easily.

5.7 Conclusion

This chapter proposes and explores a scenario decomposition framework to decompose the GTEP-UC problem with improved tractability. The proposed framework facilitates binary variables in both investment and operational levels, and inter-temporal constraints resulting from generator limitations. In addition, the developed framework enables investment decisions to be made in multiple periods; particularly two periods, by combining daily scenarios from each period into one sub-problem.

Through numerical analysis, we observed that for both single and two-period formulations, scenario decomposition is capable of providing high-quality solutions with tight upper bounds in the first few iterations. However, the method relies on the weak lower bound that progresses slowly to prove optimality and finally converge. As a consequence, the convergence is often delayed even if the optimal solution is found earlier in the process. From the analysis, we identified “weak integer cuts” and “diversity in scenarios” as the major factors that hinder the rapid improvement of lower bound. As each cut eliminates only one solution, many integer cuts are required to eliminate the solutions between the preferred solution and optimal solution, and close the bound gap. On the other hand, diversity in scenario reduces the performance by increasing the distance between the preferred solution and optimal solution. Diverse scenarios also cause the algorithm to explore a large solution space leading to many unnecessary evaluations. These two key issues must be tackled, if we are to improve the performance of the proposed scenario decomposition approach.

The chapter also investigates the application of classic and multi-cut Bender’s decomposition for the GTEP-UC problem. Experimental analysis demonstrated that the convergence rate of classic Benders is slow because of the single cut that get added at the end of each iteration, and multi-cut Bender’s is memory intensive because of the large number of nodes it needs to explore. As a result, both approaches exited with reasonably large bound gaps at the end of the maximum time limit compared to the scenario decomposition framework. Therefore, we conclude that the scenario decomposition approach is computationally more efficient compared to the Bender’s decomposition in terms of both run-time, and memory requirements, thus can be successfully applied to decompose the GTEP-UC problem. In the next chapter, we aim to discuss a few algorithmic extensions to enhance the performance of the proposed scenario decomposition framework for the GTEP-UC problem.

Chapter 6

Scenario Decomposition Extensions

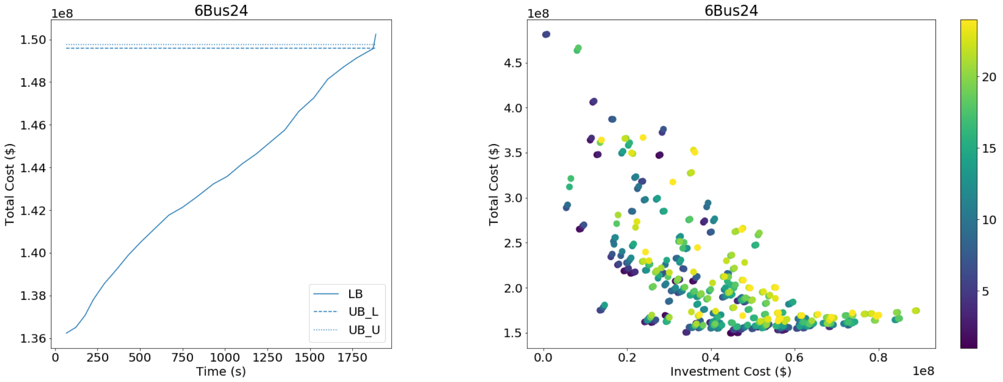
6.1 Introduction

The scenario decomposition algorithm for GTEP-UC has shown to be capable of finding high-quality solutions earlier in the solving process. Its rate of convergence is however hindered by: (i) diversity between scenarios; (ii) integer cuts which are ineffective for many sub-problems and weak cuts which eliminate just a single solution; and (iii) can over time bloat the size of the MIP formulation and the time taken to solve it. Therefore, in this chapter, we propose two extensions to mitigate the issues of diversity and weak integer cuts. We propose a scenario grouping approach to tackle the issue of diversity in scenarios, and a branching approach to explore solutions more quickly through fixing investment decisions and generating stronger cuts.

The chapter first provides a brief recapitulation of the issues related to the scenario decomposition algorithm, their causes and impact on the algorithm performance. Then the subsequent sections provide the methodology for grouping and branching mechanisms. In particular, we discuss two grouping mechanisms: heuristic and stable, and two branching strategies: local and multiple, which are the contributions of this chapter. Then, the subsequent section provides the details of the experimental analysis carried out to evaluate the performance of the proposed extensions on the scenario decomposition framework. And finally, the last section summarises the findings and concludes the chapter.

6.2 Motivation

The performance of scenario decomposition framework is delayed by the slow improvement in lower bound and the large solution space explored by the algorithm. This is depicted in Fig. 6.1a and Fig. 6.1b respectively, which illustrate the performance of 6 bus system with 24 days. As shown in Fig. 6.1a, although the framework is capable of finding high-quality solutions in the first few iterations, it relies on the weak lower bound that progresses slowly to prove optimality. On the other hand, as demonstrated in Fig. 6.1b,



(a) Algorithm convergence

(b) Explored solution space

Figure 6.1: Scenario decomposition performance for 6 Bus 24 days instance

SD algorithm explores numerous solutions that are far from optimal, especially during early iterations (the colour ramp indicates the iteration which generated the solutions).

As previously mentioned, one of the main reasons for weak lower bound and large solution space is diversity in scenarios. Scenarios with diverse properties generate conflicting investment solutions weakening the lower bound. Besides, evaluation of these solutions often leads to an exploration of a larger than necessary number of solutions prior to algorithm convergence, because of the irrelevance of some sub-problems' solutions to the other sub-problems and the global problem. Also, diversity between sub-problems causes the sub-problems to explore their own regions for a large portion of the process. Thus, a grouping approach that bundles scenarios with dissimilar properties into each of the sub-problems could overcome the issue of diversity by forcing the sub-problems to compromise their investment solutions.

Apart from diversity in scenarios, another factor that has contributed towards slow improvement in lower bound and unnecessary evaluations is weak integer cuts. Since the algorithm does not involve any dual information, the improvement in lower bound is afforded only through integer cuts that eliminate only one solution at a time. Hence, if there are many solutions between the preferred solutions of individual sub-problems and the optimal solution, it would be more efficient if we could avoid visiting all of them, or visit them at minimum computational expense. Branching would explore these solutions more quickly through fixing the values of some binary investment variables, and generate stronger cuts that rule out larger numbers of solutions.

6.3 Grouping

Grouping multiple scenarios into a single sub-problem is one of the most effective approaches used by many studies [146–151] to strengthen the lower bound and improve solution quality. Since grouping enforces subset of consistency constraints back in the sub-problem tightening the relaxation [146, 149], the resulting solutions improve the lower bound and solution quality at the expense of increased sub-problem solving times. Even though random grouping is the most common approach [146–148], multiple studies have shown that grouping utilising the information obtained from the algorithm can improve the solution quality and computing efficiency significantly, compared to the single-scenario sub-problems or randomly grouped multiple-scenario sub-problems [149–151].

In current literature, a number of methodologies [146–151] have utilised information from algorithm to form scenario groups in a systematic manner, where either generic [149, 151] or problem-specific metrics [150] have been employed to measure similarity or dissimilarity between scenarios. This includes heuristics such as k-means clustering and greedy approaches [150, 151], and exact methods that focus on solving an optimisation problem (e.g maximise the bound improvement [149]) to find the optimal grouping strategy [149,151]. While optimal grouping techniques have shown to provide stronger bounds compared to the heuristic methods [151], depending on the group size, solving an optimisation problem is a computationally expensive process [151].

For the scenario decomposition approach to GTEP-UC, we use grouping to address the issue of diversity between daily scenarios. In particular, we create sub-problems within which dissimilar scenarios are grouped together. The intuition behind combining scenarios with dissimilar properties is to make the resulting solution a comprise between the two contrasting sub-problems. As dissimilar grouping implicitly makes the groups somewhat similar to each other, it minimises the generation of outlying candidate solutions and also enhances the effectiveness of the integer cuts improving the lower bound and convergence rates.

Intuitively, to form groups of dissimilar scenarios, low demand and/or high renewable scenarios (produce inadequate investments) must be grouped with high demand and/or low renewable scenarios (produce surplus of investments). This requires multi-dimensional analysis of time series, as each sub-problem consists of multiple contiguous demand and renewable generation profiles (for each node and generator). In addition, high/low demand/generation is specific to a certain time interval, hence to classify an entire sub-problem as high/low demand/renewable scenario additional classification steps are required.

Therefore, to carry out grouping while mitigating these issues, we propose to measure dissimilarity in more abstract level and ignore the details of low/high demand/renewable scenarios. Following the lead of others' work reported in the literature, we could achieve this abstraction by utilising either generic metrics such as distance between two solutions or problem-specific metrics such as penalty cost. Since dissimilar scenarios are likely to produce completely different solutions, by measuring the distance between two solutions similarity or dissimilarity between two scenarios can be determined explicitly. On the other hand, the penalty cost relates to unserved demand. So for evaluation of any given scenario's solution, the alternative scenario with the highest penalty cost is likely to be the one with the most contrasting features to the scenario that generated the solution. Hence, by identifying the scenario with the highest penalty cost, an opposite scenario of the scenario that generated the solution can be found implicitly.

Once the dissimilarity between scenarios is measured, the most dissimilar scenarios can be grouped together to form groups of dissimilar scenarios. However, this kind of grouping is not straightforward, as the relationships between scenarios are rather complicated and multi-way. Hence, the application of heuristic methods such as k-means clustering is limited. On the other hand, current methods for optimal grouping comprise a complicated structure (e.g. mixed integer bi-level) and require dual variables and additional algorithms (e.g. branch and cut) to determine the optimal groups. Therefore, we propose to form groups progressively by ranking scenarios and forming groups of two (pairs) at a time. This is carried out by ordering scenarios from most dissimilar to least dissimilar for each candidate solution and paring the most dissimilar scenario with the scenario that generated the solution. However, a particular scenario could be the most dissimilar for many scenarios or a certain scenario could have multiple dissimilar scenarios which are also dissimilar to each other. Therefore, to carry out grouping in a more systematic manner, we propose and explore two grouping approaches: a heuristic and a stable grouping approach.

Once the pairs of dissimilar scenarios are determined using either heuristic or stable grouping approach, the scenarios are grouped into one sub-problem through a common set of investment decisions to ensure that the investment solution is a compromise between multiple scenarios. Since grouping allows non-contiguous daily scenarios to be grouped into one sub-problem, the same modified daily sub-problem (with no start-up costs, and inter-temporal constraints for the first period) utilised in the candidate solution generation phase (Section 5.4) is employed to form sub-problems with multiple days. And, the sub-problem objective is updated to the correct weight of investment and operational cost i.e., the daily capital costs are multiplied by the number of days in the sub-problem, to ensure that the lower bound is valid with multiple days.

6.3.1 Heuristic Grouping

Heuristic grouping follows a greedy approach pairing daily scenarios with the most dissimilar alternative scenario. If a scenario is already paired with another scenario, then the next most dissimilar scenario is used to form the pairs. Hence, it is ideal if the measurement of dissimilarity between two scenarios are different to another as it allows scenarios to be ordered with a minimum number of ties. Since solution distance is likely to produce many ties once ordered, for heuristic grouping the problem-specific "penalty cost" is utilised as the metric to measure dissimilarity. This method is provided in Algorithm 5 and can be described as follows.

Heuristic grouping (Algorithm 5) initiates at the end of the first iteration, with a set of arrays that provides penalty cost of each scenario for each candidate solution as the input (line 1). Then for each candidate solution, the scenarios (days) are sorted in the order from highest penalty cost to lowest (line 9), and the scenario with the highest penalty cost is paired with the scenario that generated the solution following a greedy approach (line 14). If a scenario has already been included in a group, then the scenario with the next highest penalty cost is used (line 17) restricting the scenarios to be part of only one group. Thus,

Algorithm 5 Heuristic Grouping

```
Input : $solutions$, $arrays$
Output : $pairs$
$pairs = empty$, $used\_sub\text{-}problems = empty$
for $sol$ in $solutions$ do
    $s = \text{get}(sol)$ % Get the scenario that generated the solution
    if $s$ not in $used\_sub\text{-}problems$ then
        $used\_sub\text{-}problems = \text{include}(s)$
        $array = \text{get}(arrays, sol)$ % Get the penalty cost array for that solution
        $sorted\_array = \text{sort}(array)$
        % Make pairs
        $index = 0$
        while True do
            if $sorted\_array[index]$ not in $used\_sub\text{-}problems$ then
                $pairs = \text{add}(s, sorted\_array[index])$
                $used\_sub\text{-}problems = \text{include}(sorted\_array[index])$
                exit
            else $index+ = 1$
```

at the end of the first iteration, the number of sub-problems reduces by half as each sub-problem now consists a pair of scenarios. A similar procedure is followed at the end of the second iteration for further grouping, where pairs of scenarios are combined into groups of four scenarios. Since grouping any further could make the scenario decomposition approach less efficient because of the increased sub-problem solving times, the grouping process terminates at the end of the second iteration after the construction of groups with four scenarios. Thus, any candidate solution generation phase after the second iteration will utilise sub-problems of four scenarios to generate solutions. However, for candidate evaluation phase, single scenario sub-problems are used to avoid escalation in evaluation time.

When there are two or more periods in the GTEP-UC formulation, the penalty cost per sub-problem for each candidate solution can be obtained by averaging the penalty cost produced by scenarios in different periods. For example, in the two-period formulation, the penalty cost produced by sub-problem x can be found by averaging the penalty cost produced by day x in both periods for any given solution. Recall that in the two-period formulation, initial sub-problems are formed by combining corresponding scenarios across different periods (refer to Section 5.4.2). So, if sub-problem x has the worst penalty cost for the investment solution produced by sub-problem y, they can be grouped together. This difference in grouping for single and two-period formulations is illustrated in Fig. 6.2. With two periods, the SD algorithm creates sub-problems of four daily scenarios at the end of first iteration itself. Hence, at the end of the second iteration, periods are switched according to the pairs suggested by heuristic grouping instead of bundling them into one sub-problem. Such action will provide additional diversity within the sub-problem without increasing the size. This procedure is illustrated in Fig. 6.3, where days x, y in sub-problem n and days u, v in sub-problem m are switched between periods.

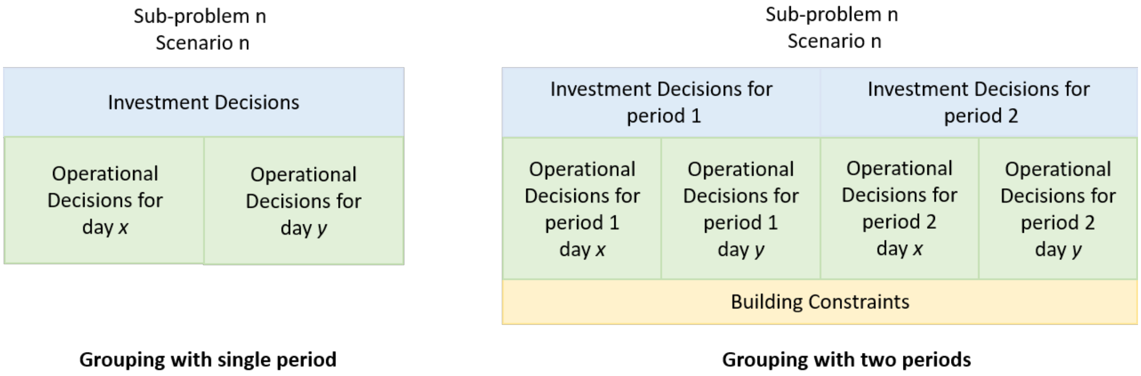

Sub-problem n
Scenario n
Investment Decisions
Operational Decisions for day x
Operational Decisions for day y
Grouping with single period
Sub-problem n
Scenario n
Investment Decisions for period 1
Investment Decisions for period 2
Operational Decisions for period 1 day x
Operational Decisions for period 1 day y
Operational Decisions for period 2 day x
Operational Decisions for period 2 day y
Building Constraints
Grouping with two periods

Figure 6.2: Grouping with single and two-period GTEP-UC problems

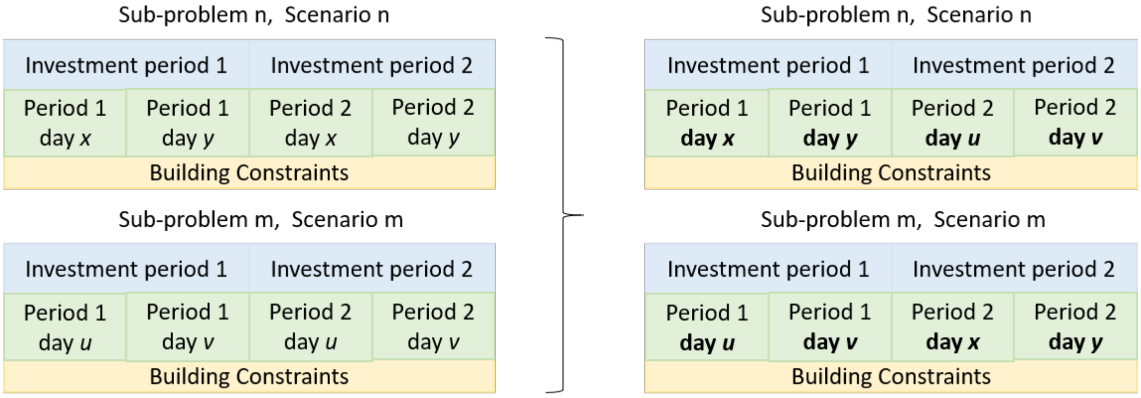


Figure 6.3: Grouping switching with two-period GTEP-UC problems

6.3.2 Stable Grouping

In the stable grouping approach, the grouping problem is formulated as a stable room-mate (SR) problem, to keep the grouping problem simple and less computationally intensive. Stable room-mate (SR) problem is the problem of finding a stable matching i.e., stable room-mate pairs, from an even-sized set of participants with a list of preferences. A matching is defined as stable if no two non-room-mate participants in the set prefer each other more than their current room-mate under the matching. In technical terms, this can be expressed as follows. Given an even-sized set of cardinality n with a list of preferences for each element that ranks $n-1$ others in order, the goal of SR problem is to find $n/2$ pairs, such that no two elements in the set which are not room-mates prefer each other more than their current room-mate [152].

SR problem is often viewed as a variant of the well known stable marriage (SM) problem introduced by Gale and Shapley to assign applicants to colleges [153]. The goal of SM problem is to find stable matchings from two disjoint sets, commonly referred to as “men” and “women”. The SM problem initiates with individuals in each set having ranked n members of the opposite sex in order of preference, and a stable matching is found when no couple prefer each other more than their actual partners. For SM problem, there is at least one stable matching for every problem instance. Thus, an algorithm was designed by Gale and Shapley to produce one such solution [153]. This algorithm was extended by Irvin to find stable room-mate pairs in SR problem [152]. However, in stable room-mate problem, a stable matching may not be possible for some problem instances, thus the algorithm was designed to find a stable matching if exists or indicate otherwise. This algorithm is commonly known as Irvin's algorithm for SR problem, and is widely used to solve SR problems [154–156].

In scenario decomposition context, each sub-problem becomes a participant with a list of preferences, where the preference is based on the dissimilarity. Hence, for each sub-problem the alternative sub-problem that differs the most is ranked with the highest preference. Previously, with heuristic grouping, problem-specific "penalty cost" was chosen as the performance metric to quantify the dissimilarity in an abstract level. For the SR problem, we utilise "solution distance" between candidate solutions to rank the sub-problems, as "solution distance" ensures that at least one stable matching exists for every problem instance (discussed later in this section). And, unlike the "penalty cost" that requires computationally expensive evaluations, "solution distance" utilises the less intense candidate solution generation stage to obtain necessary information for grouping. It also enables the approach to be generalised to other problem domains with similar structure (co-optimisation of investment and operations) because of its problem independent nature.

Thus, the application of SR problem for grouping is comprised of two main steps.

1. Creation of preference list
2. Application of Irvin's algorithm [152]

When creating the preference list, a simple distance function Eq. 6.1 is utilised to calculate the distance between two solutions, where x^a and x^b are the investment solutions obtained from two distinct sub-problems and $\mathcal{N}$ is the number of investment decisions. Note that for multi-period formulations, x^a and x^b include investment decisions from all the periods. Then, by calculating the distance between solutions provided by each sub-problem, a distance matrix is created where the number of rows and columns are equal to the number of sub-problems. This will result in a symmetric distance matrix with zeros in the main diagonal and each element denoting the distance from one solution to another. To distinguish between sub-problems with exactly the same solution (distance = 0) to itself, the diagonal is replaced with -1. Then based on the distance matrix, a preference matrix of same size is created, where in each row, sub-problems are prioritised from the largest distance to the smallest. As a result, rows in the preference matrix will provide lists of preferences for the sub-problems. An illustrative example for this particular step can be provided as follow.

$$\sum_{i \in \mathcal{N}} \mid x_i^a - x_i^b \mid \tag{6.1}$$

Example 1 *Consider the following set of sub-problems and their respective solutions.*

$\mathcal{S} = \{s_1 = [1,1,1,0],\ s_2 = [1,1,0,0],\ s_3 = [0,0,0,0],\ s_4 = [1,1,0,0]\}$

The distance matrix D can be given as follows, where the element $d_{i,j}$ represents the distance between the solutions obtained from sub-problems s_i and s_j. Note that, when $i = j$, the diagonal is -1 instead of 0. Once the distance matrix is created, by ranking the *sub-problems from highest to lowest distance, preference matrix P is obtained, where row i in matrix P demonstrates the preference list for sub-problem s_i.*

$$D = \begin{bmatrix} -1 & 1 & 3 & 1 \\ 1 & -1 & 2 & 0 \\ 3 & 2 & -1 & 2 \\ 1 & 0 & 2 & -1 \end{bmatrix} \qquad P = \begin{bmatrix} s_3 & s_2 & s_4 & s_1 \\ s_3 & s_1 & s_4 & s_2 \\ s_1 & s_2 & s_4 & s_3 \\ s_3 & s_1 & s_2 & s_4 \end{bmatrix}$$

Once the preference matrix is formulated, Irvin's algorithm for SR problem [152] is utilised to obtain the suitable pairs. Irvin's algorithm for SR problem [152] can be summarised into the following three steps.

1. Step 1: The algorithm aims to find an initial matching by making participants to propose to their highest preference. The recipients accept the proposal if that's their only proposal or they have received a better proposal from a more preferable participant. If any proposal gets rejected, the participant keeps on proposing to other participants in the order of their preference until one of them accept.
2. Step 2: The second step of the algorithm removes impossible pairs by eliminating the participants that they prefer less than their currently accepted one. This is executed by simply removing the participants on the right hand side of the accepted participant in the preference matrix in a symmetrical fashion. For example, if A rejects B, A must be removed from B's preference list. If at any point a preference list for a participant becomes empty, that indicates that no stable matching exists for this particular instance.
3. Step 3: The final step aims to find a stable matching by reducing the preferences lists that contain more than one preference. This is achieved by identifying cycles in the preference sequence and removing any undesirable pairs. This step repeats until no participant has more than two preferences, or one participant runs out of participants to propose to, in which no stable matching exists. Once each participant has only one participant in the preference list, a stable matching has been found.

For this dissimilar grouping problem, if solution distance is utilised as the metric to rank the sub-problems, at least one stable matching exists for every problem instance. This can be explained as follows. Since the distance matrix is symmetric, the preferences of opposite scenarios are ordered in a manner that compliments each other. This is visible in the illustrative Example 1 provided above. For s_1 the sub-problem with the highest preference is s_3. Similarly, for s_3 the scenario with the highest preference is s_1. Therefore, it is guaranteed that s_1 and s_3 belong to one pair. Although s_1 and s_3 come at a higher preference for s_2 and s_4, because s_2 and s_4 come later in s_1 and s_3 preference list, s_2 and s_4 will have to settle with each other. Thus, the pairs become (s_1, s_3) and (s_2, s_4). Note that s_2 and s_4 are exactly similar yet they have been paired together. This is because, in the set of solutions provided above, s_1 can be considered as one of the extreme scenarios with the most number of investments, and s_3 can be considered as the other extreme scenario with no investments. For extreme sub-problems, moderate sub-problems such as s_2 and s_4 becomes the least preference. This property is ideal to omit outlying candidate solutions

Algorithm 6 Stable Grouping

```
1: Input : solutions
2: Output : pairs
3: % Generate the distance matrix
4: $D$ = matrix(|solutions|, |solutions|)
5: for $i$ in range(|solutions|) do
6:     $D_{i,i}$ = -1 % Set the distance of itself to -1
7:     for $j$ in range(i+1,|solutions|) do
8:         $d$ = distance(solutions[i],solutions[j])
9:         $D_{i,j}$ = d
10:        $D_{j,i}$ = d
11: % Obtain the preference matrix
12: $P$ = matrix(|solutions|, |solutions|)
13: for $i$ in range(|solutions|) do
14:     $P[i, : ]$ = sort$D[i, : ]$])
15: pairs = StableRoommate($P$)
```

in the scenario decomposition framework, as the algorithm requires extremely different sub-problems such as s_1 and s_3 to be grouped together, and moderate sub-problems such as s_2 and s_4 to be grouped together.

Algorithm 6 provides the procedure for stable grouping utilising Irvin's algorithm for SR problem [152]. The method initiates with the generated solutions for each sub-problem as the input (line 1). Then, as previously described, it creates the distance matrix in a symmetric fashion (lines 4 - 10), which in turn is used to generate the preference matrix by ordering each row (lines 12 - 14). This preference matrix is provided as an input to the stable room-mate algorithm [152] to obtain pairs of scenarios (line 15), and the resulting pairs are returned as the output to the main SD framework (line 2). The grouping continues until groups of four scenarios are formed, as prolonged sub-problem solution times for larger groups could erode the computation-time advantages of the scenario decomposition approach.

For the two-period formulation, stable grouping is executed only once, so that the groups contain four scenarios and sub-problem solution times remain moderate. In the next iteration, scenarios are exchanged between the groups according to the pairs suggested by the Irvin's algorithm for additional diversity. Because solutions are now forced to compromise across four different scenarios (refer to Fig. 6.3), such an exchange is likely to improve the lower bound of individual sub-problems, and as a result, the global lower bound.

6.4 Branching

Branching techniques are used extensively in combinatorial optimisation problems to find good-quality solutions and improve computational efficiency (e.g. branch and bound [157], local branching [158]). In scenario decomposition context, the application of branching techniques was first explored by Hemmi *et al.* in [148]. The method was introduced as "diving" rather than as a form of branching, with the objective of raising lower bound and improving convergence rates.

The goal of "diving" algorithm [148] is to build a partial solution extending one (binary variable) decision at a time, until the solution is complete (a new best-found solution candidate), or the lower bound (sum of sub-problem lower bounds) given the partial solution exceeds the current global upper bound. In the latter case, it is proven that no full solution stemming from the partial solution (that has not already been cut from the sub-problems) can be optimal. Thus, a cut derived from the partial solution, i.e., a "partial cut", can be added to all the sub-problems so as to remove the entire set of solutions containing that combination of decision variable values. An illustrative example for "partial cut" and "diving" algorithm [148] follows.

Example 2 *Consider the following set of sub-problems and their candidate solutions obtained at the end of the candidate solution generation phase.*

$\mathcal{S} = \{s_1 = [1,1,0,0,1],\ s_2 = [1,1,0,0,0],\ s_3 = [1,0,1,0,0],\ s_4 = [1,1,1,0,1]\}$

The "diving" method [148] is initiated in a separate algorithm by fixing the most prevalent set of investment decision values chosen from the current set of candidate solutions. These values include the decisions that are common in all the solutions, and the next most prevalent decision. In this example, x_1, x_2 *and* x_4 *are set to 1, 1 and 0 respectively, i.e., the combination* $[1,1,x_3,0,x_5]$*. Note that,* $x_1 = 1$ *and* $x_4 = 0$ *are the decisions that are common in all solutions, where as* $x_2 = 1$ *is the next most prevalent decision. Once the decisions are fixed, suppose that following solutions are obtained in the next diving iteration, where the lower bound is still below the global upper bound. Note that this lower bound is conditional on the partial solution, and estimates the optimal value of a full solution stemming from the partial solution.*

$\mathcal{S} = \{s_1 = [1,1,0,0,1],\ s_2 = [1,1,0,0,0],\ s_3 = [1,1,0,0,0],\ s_4 = [1,1,1,0,1]\}$

Then, the most prevalent decision value is chosen again and is fixed for the next diving iteration. In this example, $x_3 = 0$ *is the next most prevalent decision, as it occurs in all* s_1, s_2 *and* s_3*, hence the combination* $[1,1,0,0,x_5]$ *is fixed for the next diving iteration. Note that* s_1, s_2 *and* s_4 *have not changed their solutions, and* s_3 *has now generated the same solution as* s_2*. This repetition arises because the solutions are not evaluated nor eliminated during the diving process. The solutions are constrained only by the cuts added to the sub-problems before diving commenced and the partial solution that is fixed by the diving process, but there are no new cuts added while diving is occurring.*

After this second iteration of diving, the lower bound exceeds the global upper bound. The diving algorithm has proved that no full solution can be derived from the partial solution $x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = 0$*, that is better than the current best solution. Therefore, the partial solution can be added as a partial cut Eq. 6.2 to the scenario decomposition framework. Such a partial cut is stronger than the full integer cut as it removes all the solutions with the combination* $[1,1,0,0,x_5]$*, which are* $[1,1,0,0,0]$ *and* $[1,1,0,0,1]$ *in this example.*

$$(1 - x_1) + (1 - x_2) + x_3 + x_4 \geqslant 1 \tag{6.2}$$

Clearly, if the partial solution's combination of decision variables and their values is longer, then fewer solutions are eliminated by the partial cut.

The diving algorithm [148] has shown to reduce the number of iterations required by the SD algorithm to converge. It also proves useful for small instances for the GTEP-UC problem. However, "diving" [148] can be wasteful of computational effort when the partial solutions are arrived at after several cycles of partial solution extension. Unless the full solutions from sub-problem solves are progressively added as cuts (which would then require computational effort to be expended on an additional evaluation step per solution explored), no information is gained from the early cycles in the diving process other than insight into the next decision variable to fix. Each cycle requires the sub-problems to be solved in full (using the partial solution). The total time taken for the sub-problem solves may be comparable to the total time saved by eliminating solutions via a partial cut. Especially, in cases where the cut has many variables included and/or a good proportion of the eliminated solutions are rather unfavourable and would not be explored by the main algorithm before convergence anyway. Also for GTEP-UC, diving process often proved that the partial cuts comprise many fixed decisions (i.e., "long cuts"), and so eliminate only a relatively small set of decisions. Therefore, in this section, we propose more efficient branching mechanisms for the scenario decomposition framework proposed in Chapter 5.

Grouping brings about diversity (in electricity supply and demand) within sub-problem instances, and similarity between sub-problem instances, and proves to be very helpful. Nevertheless, branching that is achieved through fixing decision variable values in the sub-problems can assist in two ways. First, by fixing decision variable values, the sub-problems have fewer columns and can be solved more rapidly. As long as the partial solution has a high probability of being part of a close-to-optimal solution, this brings efficiency to the overall algorithm. Second, as is the case with diving, if and when the (branching) lower bound exceeds the current (global) upper bound, then the partial solution can be cut from all sub-problems. The main differences between branching (as defined here) and diving (as in [148]) are that: (i) partial solution extension does not occur after each iteration in branching; and (ii) during branching all other steps of the iterative scenario decomposition algorithm continue to be utilised so that the information uncovered by the sub-problem solves is exploited more fully.

Therefore, the main purpose of branching is to explore and eliminate solutions strategically with minimum computational effort, by hard fixing subset of investment decisions and forcing all the sub-problems to search a particular region at the same time. This procedure is provided in Algorithm 7 and Algorithm 8, where Algorithm 7 provides the steps for branching mechanism, while Algorithm 8 shows how branching is executed within the scenario decomposition framework.

The branching mechanism initialises by fixing the set of investment decisions (line 4 Algorithm 7) selected by the SD algorithm (line 10 Algorithm 8). These investment decisions are selected based on the outcomes of the previous iterations. The procedure for selecting these decisions are discussed later in this section. With the fixed investment decisions, the branching algorithm (Algorithm 7) continues to iterate between candidate

Algorithm 7 Branching

```
Input: $fix\_decisions$, $Sub\text{-}problems$, $UB_U$, $Pool$
Output: $UB_U$, $Pool$
$LB_B = -\infty$
$Sub\text{-}problems$ = Fix($Sub\text{-}problems$, $fix\_decisions$)
while $LB_B < UB_U$ do
    Candidate solutions generation
    Candidate solution evaluation
    Add integer cuts
$Sub\text{-}problems$ = Free($Sub\text{-}problems$, $fix\_decisions$)
```

Algorithm 8 Scenario Decomposition with Branching

```
Initialise: $UB_U = \infty, LB = -\infty, iter = 0, Pool = empty$
while $LB < UB_U$ do
    $iter += 1$
    Candidate solutions generation
    Candidate solution evaluation
    Add integer cuts
    % Initialise Branching
    if branch then
        % Select the decisions to be fixed
        $fix\_decisions$ = Select($solutions$)
        Branching($fix\_decisions$)
        % Add Partial Cuts
        $Sub\text{-}problems$ = PartialCuts($fix\_decisions$, $Sub\text{-}problems$)
```

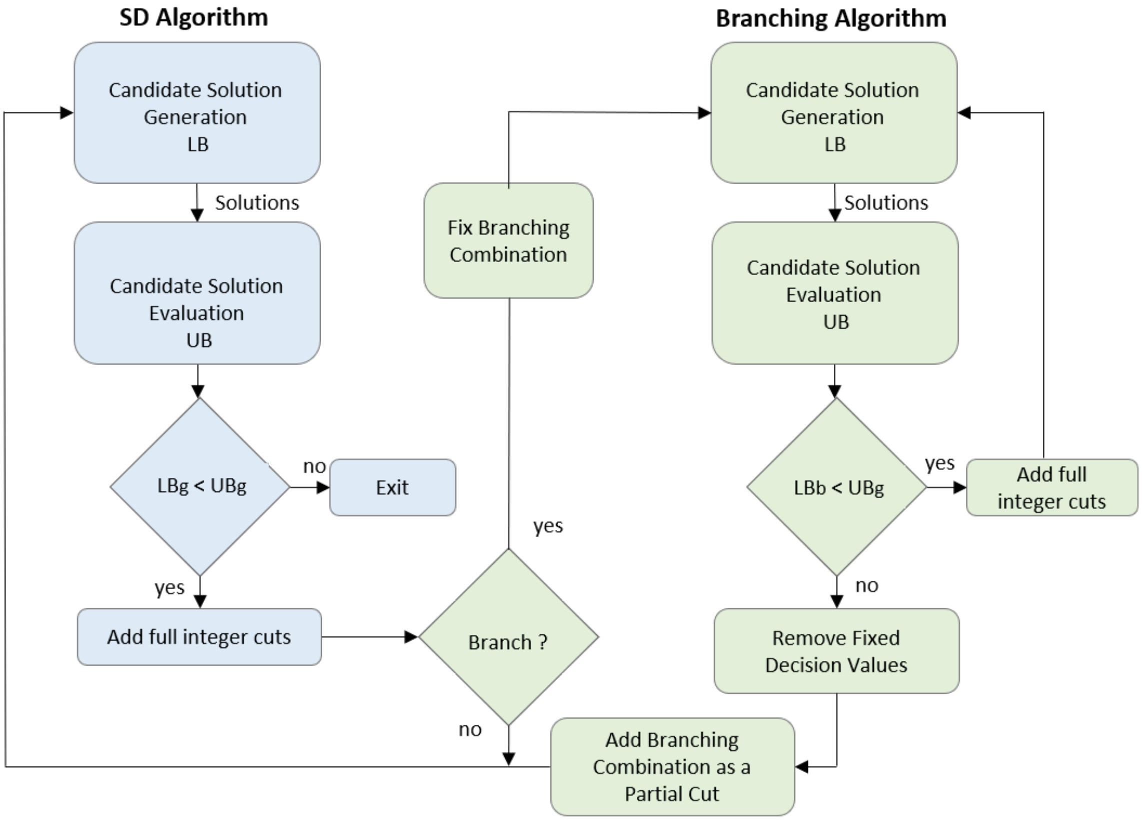

SD Algorithm
Branching Algorithm
Candidate Solution Generation LB
Solutions
Candidate Solution Evaluation UB
LBg < UBg
no
Exit
yes
Add full integer cuts
Branch ?
yes
no
Fix Branching Combination
Candidate Solution Generation LB
Solutions
Candidate Solution Evaluation UB
LBb < UBg
yes
Add full integer cuts
no
Remove Fixed Decision Values
Add Branching Combination as a Partial Cut

Figure 6.4: Scenario decomposition with branching

solution generation phase and evaluation phase until the branching lower bound LB_B exceeds the global upper bound. Note that, unlike the "diving" approach in [148], full integer cuts are added at each branching iteration after evaluation (line 8 Algorithm 7). Once the branching algorithm (Algorithm 7) terminates, the fixed combination is added as a partial cut to the SD algorithm (line 23 Algorithm 8) to eliminate the respective solutions, and the decisions are freed from the fixed values (line 9 Algorithm 7). By repeating this procedure with different sets of fixed decisions over several iterations of SD algorithm (Algorithm 8), multiple partial cuts are added to remove multiple sets of solutions. The schematic diagram for the branching mechanism along with the main SD framework is provided in Fig. 6.4.

The most challenging aspect of branching is selecting the decisions to fix from the previous iterations of SD algorithm (line 10 Algorithm 8). In order to address this challenge, we introduce the concept of frequent combinations. Essentially, a frequent combination is a partial solution that occurs repeatedly in generated investment solutions. While this partial solution could include both binary 1 and 0 decision values, in this chapter we limit the frequent combination to binary 1 decisions, i.e, combinations of most frequently invested units. This is because the investment solutions tend to be sparse arrays in non-zero values (for the test cases utilised in this thesis). And, shorter the combination stronger the partial cut, so we limit ourselves to frequent combinations of invested units.

Fig. 6.5 aims to depict these frequent combinations in solutions generated by 6 bus 24 day instance (with grouped scenarios). The y-axis represents the solution id, x-axis represents the investment decisions, where G, R and L indicate thermal generators, renewable generators and transmission lines respectively, and filled lines represent the investment (binary 1). The figure clearly shows that generators G5, G6 and lines L7 have been invested in many solutions, thus can be considered as one of the frequent combinations. For the same problem instance, G3, G5, G6 and L7 and G5, G6, R2 and L7 can also be considered as frequent combinations, because many solutions contain at least one of them. By identifying these frequent combinations, they can be explored progressively. With multiple frequent combinations, the order of execution is crucial, as exploration of certain combinations prior to others may improve the computational efficiency. For example, combination G5,G6,L7 is a subset of both G3,G5,G6,L7 and G5,G6,R2,L7. Hence, by first eliminating solutions with combinations G3,G5,G6,L7 and G5,G6,R2,L7, the combination G5, G6, L7 can be explored more efficiently.

Therefore, the objective of branching mechanisms is to determine these frequent combinations from the outcomes of the previous iterations with their order of execution and explore them systematically. In this section, we propose two branching mechanisms: local and multiple, where local branching forces sub-problems to explore solutions near the (local to) current best solution, while multiple branching forces sub-problems to explore preferred regions of individual sub-problems.

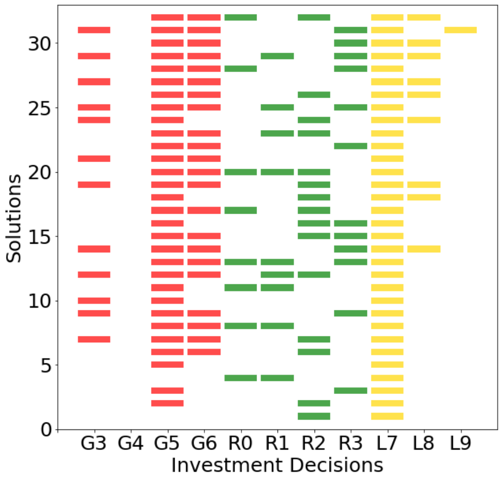


Figure 6.5: Illustration of frequent combinations

6.4.1 Local Branching

Local branching identifies the frequent combinations in near-optimal solutions and forces the sub-problems to explore around them. Since near-optimal solutions are good-quality solutions [158] preferred by many sub-problems, by exploring them quickly (through decision variable fixing) and eliminating them via cuts, the lower bound can be improved significantly. These combinations can be identified by maintaining a solution pool with the best $\boldsymbol{N}$ solutions obtained from the evaluation phase and extracting the decision values that are common among them. Recall that we assume solutions are sparse arrays, so we extract only the decisions with value 1. For example, if a solution pool of size 10 is selected, decision values (binary 1) that are common in all 10 solutions could be selected as the frequent combination. Note that, this solution pool is different to the solution pool discussed in Section 5.4, as it stores all the solutions with the lowest ζ^L whether or not they are potentially optimal solutions[1] .

This procedure is shown in Algorithm 9, which demonstrates the scenario decomposition algorithm with local branching. The pool of best N solutions (where $N = 10$ for the experiments reported in this thesis), is represented through $\mathcal{N}$, and $\mathcal{X}$ denotes the set of investment decision variables. The pool $\mathcal{N}$ is updated by adding the solutions with $\zeta^L < \zeta^L_{mx}$, where ζ^L is the lower bound on the total objective value (refer to Section 5.4) of a solution and ζ^L_{mx} is the solution with the largest objective value in the pool (lines 8-10). The solutions with the largest objective values are removed from the pool simultaneously to maintain a constant N. Then, the frequent combination is obtained by calculating the frequency of investment (binary 1) for each decision, and extracting the decisions with N frequency (lines 14 - 16).

---

[1] ζ^L is the lower bound on the total objective value – refer to Section 5.4

Algorithm 9 Scenario Decomposition with Local Branching

```
1: Initialise: $UB_U = \infty, LB = -\infty, iter = 0, Pool = empty, \mathcal{N} = empty$
2: while $LB < UB_U$ do
3:     $iter+ = 1$
4:     Candidate solutions generation
5:     Candidate solution evaluation
6:     Add integer cuts
7:     % Maintain the pool of best $N$ solutions where $N = 10$
8:     for $sol \in solutions$ do
9:         if $\zeta^L_{sol} < \zeta^L_{mx}$ then
10:            Add the solution to the pool and remove from the pool a solution with $\zeta^L = \zeta^L_{mx}$
11:    % Initialise Branching
12:    if local branch then
13:        % Select the decisions to be fixed
14:        for $x_i \in \mathcal{X}$ do
15:            if $\sum_{n \in \mathcal{N}} x_{i,n} = |\mathcal{N}|$ then
16:                Add $x_i = 1$ to the $fix\_decisions$
17:        Branching($fix\_decisions$)
18:        % Add Partial Cuts
19:        $Sub\text{-}problems$ = PartialCuts($fix\_decisions$, $Sub\text{-}problems$)
```

Once the frequent combination is extracted from the solution pool, by hard fixing the decisions in the combination and calling the branching algorithm (Algorithm 7), the frequent combination can be explored and eliminated. Recall that, the solution pool gets updated continuously as the algorithm proceeds. Thus, determining when to extract the frequent combination and initialise the branching algorithm is crucial. If initialised at the beginning of the scenario decomposition algorithm, the frequent combination will be extremely short as a variety of solutions are likely to be in the solution pool. Although short combination makes the partial cut extremely effective (since many solutions are eliminated with a single cut), a substantial amount of time will also be required to enumerate and validate that the combination cannot be extended to a better solution than the current best solution. Hence, to improve the computational efficiency, the branching mechanism initialises when the frequent combination is reasonably long. This is achieved by pre-determining the iteration that initialises the branching mechanism or alternatively, the algorithm can be designed to initialise branching when the number of decisions in the frequent combination becomes reasonable or stable.

Once the partial cut that represents the frequent combination is added to the SD algorithm (line 19 Algorithm 9), sub-problems are likely to select a different combination and explore around them. If we are to improve the convergence rate of the SD algorithm, these combinations must also be explored and eliminated. The other frequent combinations can be determined by identifying the decision that sub-problems trade once the partial cut is added. Because sub-problems are likely to retain the most important set of decisions in the combination, and trade the least important decision/s, removal of the least frequent decision within the frequent combination gives rise to another frequent combination. Therefore, local branching starts with a reasonably long frequent combination and then it removes a least frequent decision at a time until the global lower bound (computed across all sub-problems, without any restrictions on decisions except for

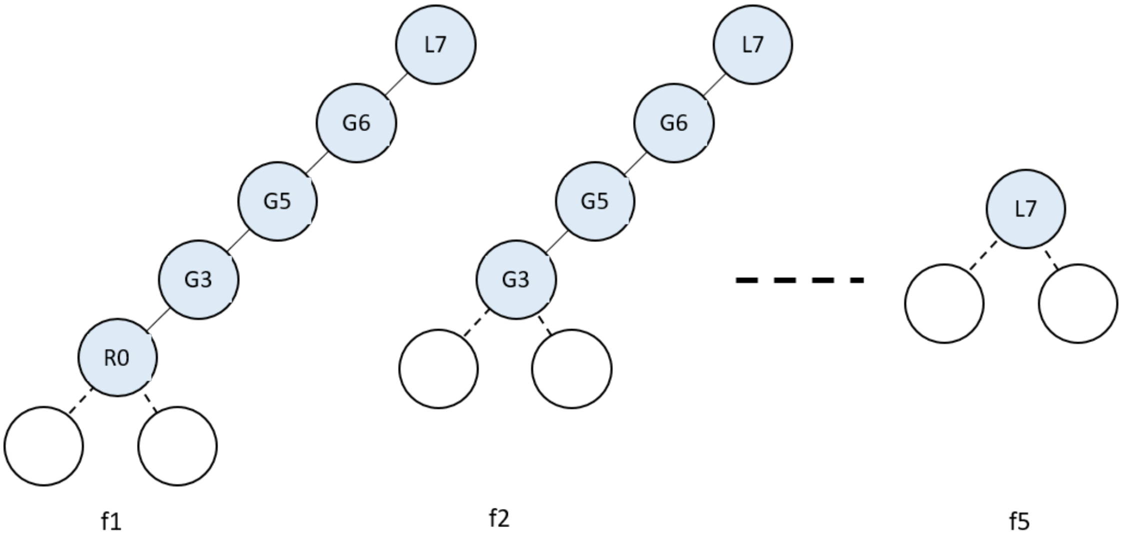


Figure 6.6: Illustration of local branching

solution-eliminating cuts) exceeds the global upper bound. Although only one decision is fixed at the end of the mechanism leaving a large space to be explored, since previous combinations have excluded most of the prominent solutions, the combination with one decision can be explored with much less computational effort.

Example 3 *Consider the following frequent combination* $F = \{G3, G5, G6, R0, L7\}$. *Assume that the decisions are traded, or the least popular decisions within the frequent combination are in the order of* $\{R0, G3, G5, G6, G7\}$
Then, the frequent combinations are,
$F = \{f_1$ *= [G3, G5, G6, R0, L7],* f_2 *= [G3, G5, G6, L7],* f_3 *= [G5, G6, L7],*
f_4 *= [G6, L7],* f_5 *= [L7]*$\}$
These combinations are depicted in Fig. 6.6.

These combinations are explored by calling the branching algorithm (Algorithm 7) multiple times over several iterations of SD algorithm (Algorithm 8). After each execution, the branching algorithm return to the SD algorithm, to add the corresponding partial cut and check algorithm convergence. Returning to the SD algorithm after each call to compute the global lower bound with the newly added partial cut minimises the solution evaluations that are not necessary for algorithm convergence i.e., switching between branching and SD algorithms ensures that the overall procedure has not over-branched.

6.4.2 Multiple Branching

Multiple branching is an alternative to the local branching approach. The aim is to explore solution regions preferred by individual sub-problems. This provides the ability to eliminate solutions around the best-known solution (near-optimal solutions), and also solutions in other regions that are being explored by various sub-problems. In the event where a substantial number of sub-problems are not exploring around near-optimal solutions, multiple branching will ensure that these regions are also explored and eliminated.

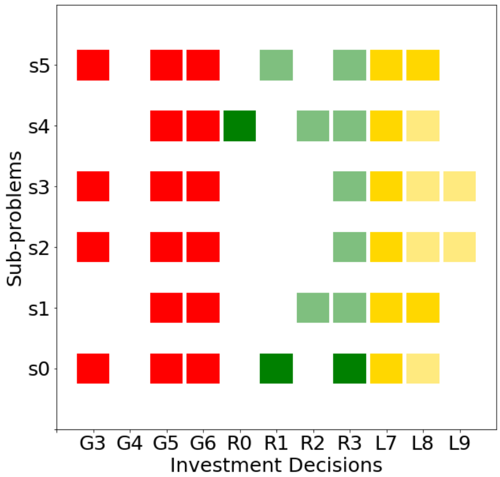


Figure 6.7: Frequent combinations for 6 bus 24 day per sub-problem

The first step of multiple branching is to determine the preferred regions of sub-problems. This is achieved by identifying the frequent combinations in solutions provided by the distinct sub-problems over several SD iterations. Essentially, the approach keeps track of the solutions generated by each sub-problem, and extracts the common decisions between iterations with binary value 1 (assuming that investment solutions are sparse arrays). For example, consider the previously discussed 6 bus 24 day instance with groups of four daily scenarios (refer to Fig. 6.5). For the same instance, generated solutions are demonstrated per sub-problem in Fig. 6.7, where y-axis represents the sub-problem id, x-axis represents the investment decisions, coloured block indicates an investment, and intensity in colour indicates how often that investment is chosen by that sub-problem (darker the colour, the decision is more frequent). The figure clearly shows that for each sub-problem, a certain combination is more prominent. For sub-problem s_0, the most prominent combination is G3, G5, G6, R1, R3 and L7. For sub-problem s_1, it is G5, G6, L7, and L8. For both s_2 and s_3, the most prominent combination is G3, G5, G6 and L7. For sub-problem s_4, it is G5, G6, R0 and L7, whereas for s_5, it is G3, G5, G6, L7 and L8. Thus, by fixing these frequent combinations, the preferred region of each sub-problem can be explored systematically.

This procedure is shown in Algorithm 10, where $\mathcal{S}$ is the set of sub-problems, $\mathbb{S}$ is a solution generated by a sub-problem $s \in \mathcal{S}$, $\mathcal{K}$ is a matrix that counts the frequency of decision variables being binary one for each sub-problem, $\mathcal{F}_s$ is the frequent combination for sub-problem s, and $\mathcal{X}$ is the set of investment decisions. To keep track of the generated solutions per sub-problem, for each sub-problem s, the frequency of decision x_i being one ($k_{s,i}$) is calculated (lines 9 - 12). This frequency is calculated for all SD iterations until the iteration which initiates the branching procedure, so that, all the solutions generated before the branching procedure are taken into account. Then, once the branching procedure is initialised, for each sub-problem s, investment decisions that match a certain threshold N are selected as the frequent combination F_s for that sub-problem (lines 18-20). In the

Algorithm 10 Scenario Decomposition with Multiple Branching

```
Initialise: $UB_U = \infty, LB = -\infty, iter = 0, Pool = empty$
% Matrix that keeps track of solutions per sub-problem
$\mathcal{K}$ = matrix($|\mathcal{S}|, |\mathcal{X}|$)
% Set of frequent combinations
$\mathcal{F}$ where $|\mathcal{F}| = |\mathcal{S}|$
while $LB < UB_U$ do
    $iter += 1$
    % Candidate solutions generation
    for $s \in \mathcal{S}$ do
        $\mathbb{S}$ = solve($s$)
        for $x_i \in \mathcal{X}|\mathbb{S}$ do
            $k_{s,i} += x_i$ where $k_{s,i} \in \mathcal{K}$
    Candidate solution evaluation
    Add integer cuts
    % Initialise Branching
    if multiple branch then
        % Select the frequent combinations of individual sub-problems
        for $s \in \mathcal{S}$ and $i \in \mathcal{X}$ do
            if $k_{s,i} = N$ then
                Add $k_{s,i}$ to $\mathcal{F}_s$
        Branching($fix\_decisions$)
        % Add Partial Cuts
        $Sub\text{-}problems$ = PartialCuts($fix\_decisions$, $Sub\text{-}problems$)
```

current setting, N is set to the iteration that initiates the multiple branching procedure, so that, only the decisions (binary 1) common in all the iterations are selected as the frequent combination.

Once the unique frequent combinations are identified, the next challenge is to execute multiple branching in an order that improves the efficiency of the overall algorithm. Recall that, execution of short combinations prior to long combinations can be computationally inefficient, especially if the short combination is part of the long combination. Thus, the order of execution is determined by the length of the combination, and the branching algorithm proceeds from the longest frequent combination to the shortest. In the event where there are multiple long combinations, the priority is given to the combination preferred by the majority[2]. For better performance, the order of execution can be improved by employing other measuring parameters such as investment cost and total objective value, and by carefully sorting the common subsets of decisions in the frequent combinations. However, in this chapter, the order of execution depends only on the length of the combination.

Example 4 *Consider the 6-bus 24 day instance, where frequent combinations of sub-problems are demonstrated in Fig. 6.7. For each sub-problem, frequent combinations are as follows.*

$\mathcal{F} = \{f_0 =$ *[G3, G5, G6, R1, R3, L7]*, $f_1 =$ *[G5, G6, L7, L8]*,

$f_2 = f_3 =$ *[G3, G5, G6, L7]*, $f_4 =$ *[G5, G6, R0, L7]*, $f_5 =$ *[G3, G5, G6, L7, L8]* $\}$

The order of execution based on length for these combinations can be listed as follows.

---

[2] Given multiple combinations with the same length, prioritising the most prevalent combinations has negligible impact on the algorithm performance (positively or negatively). It is simply used to resolve the ties between priorities

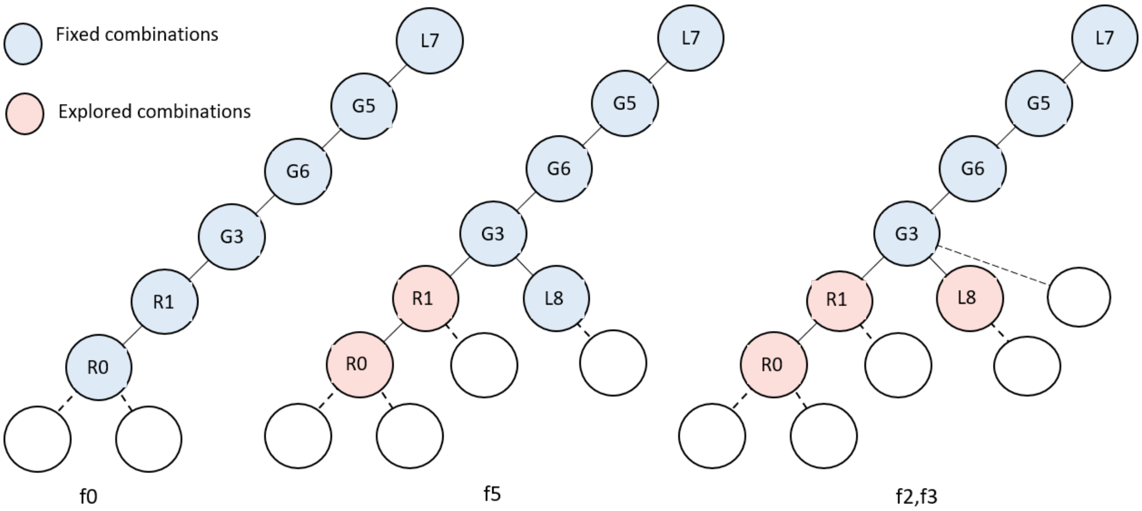


Figure 6.8: Illustration of multiple branching

$\mathcal{F}_{order} = \{f_0, f_5, (f_2, f_3), f_4, f_1\}$

First three combinations f_0, f_5 and (f_2, f_3) are illustrated in Fig. 6.8

Once the priorities of the frequent combinations are determined, they are executed in the respective order by calling the branching algorithm (Algorithm 7) multiple times, where the unique frequent combinations become the decisions to be fixed and explored in each call.

As discussed with local branching, determining when to initialise the branching mechanism is crucial. Initialising the multiple branching mechanism too early in the SD algorithm can make the entire process inefficient. At the beginning of the SD algorithm, the frequent combinations are likely to be long[3]. In addition, the number of unique combinations are likely to be significant[4]. Because long combinations lead to weak partial cuts that remove only fewer solutions, if branching is initiated too early, the procedure (Algorithm 7) is likely to explore many weak combinations that do not contribute significantly towards the improvement of the global lower bound.

Thus, the multiple branching is delayed by pre-defining a later iteration to initialise. Alternatively, the SD algorithm can be continued until the length and/or the number of unique frequent combinations becomes reasonable. The advantage of multiple branching over local branching is that, it does not rely on the expensive evaluation stage (in local branching to obtain the pool of near-optimal solutions, the evaluation step is essential). Hence, if the evaluation step is too expensive to allow multiple iterations, sub-problems can be solved repeatedly without evaluation, solely to extract frequent combinations. However, in such situations, extra care must be taken not to remove these solutions from the solution space without evaluation.

---

[3] Because the full integer cut exclude exactly the generated solution, and the sub-problems are likely to change only few decisions in the next iteration, many decision values will be common between the iterations for each sub-problem leading to long combinations

[4] Longer the combination, more variations

6.5 Experimental Analysis and Discussion

This section provides the details of the experimental analysis conducted to evaluate the performance of the proposed extensions for scenario decomposition in both single-period and multi-period setting.

6.5.1 Experimental Setup

The extensions were analysed using four test cases: 6 bus, 14 bus, 18 bus and 24 bus systems, with three instances in each test case where, 24 days (2 days from each month), 84 days (1 week from each month) and 364 days (to make it divisible by four for grouping) were used to represent the 365 days. To account for an operational cost of 365 days, the operational costs were multiplied by their respective ratios. In addition, a period of one year was considered for single-period formulation, while a horizon of 20 years with two periods (10 years each) was considered for two-period formulation. In addition, the penalty cost was set to 1000 \$/MWh of unserved energy, and an annual monetary discount rate of 10% was considered. For the two-period formulation, a load growth of 15% was uniformly applied to the demands in the second period.

Moreover, the grouping algorithms were executed only in the first three iterations of the SD framework limiting the sub-problems to groups of four scenarios. Both branching algorithms were pre-defined to start at iteration 10, and they are compared against the "diving" algorithm [148], which initialises from iteration 1. All algorithms and models were implemented using Python 2.7 and the optimisation problems were solved using Gurobi 8.0 [131] on a computing node that utilises eight cores clocking at 2.70 GHz with 16 GB RAM. In addition, Gurobi's default MIP gap 1×10^{-4} was used and the *number of thread count* was set to number of cores. For each test case, a maximum time limit of 22hrs was set for the entire algorithm (the summation of all solving times provided by Gurobi).

6.5.2 Impact of Dissimilar Grouping

To investigate the impact of dissimilar grouping, heuristic and stable grouping algorithms were applied to the previously described four test cases and three instances (without diving or branching) considering a single-period formulation. Table 6.1 summarises the obtained results, where the columns illustrate test case, instance, type of the algorithm, best-known objective value (UB_U), the optimality gap at the end of the maximum time limit, the residual gap and the number of potential solutions in the pool respectively. Three types of algorithms were considered, where basic indicates basic scenario decomposition (SD) framework without grouping, heuristic denotes heuristic grouping approach and stable denotes stable room-mate grouping approach. According to the Table 6.1 it is visible that grouping algorithms have managed to prove optimality[5] for some instances (such as 14-Bus 364-day) for which the non-grouping algorithm did not converge. For the instances that were not fully solved to optimality within the time limit, grouping has

---

[5] Prove optimality for one of the solutions in the pool

Table 6.1: Performance of scenario decomposition with grouping

Test case	Instance	Type	Best obj. (M$)	Opt. gap (%)	Res. gap (%)	N. sol.
6 Bus	24	Basic	149.77	0	0.123	2
		Heuristic	149.77	0	0.123	2
		Stable	149.77	0	0.123	2
	84	Basic	149.96	0	0.012	1
		Heuristic	149.96	0	0.012	1
		Stable	149.96	0	0.012	1
	364	Basic	156.49	0	0.301	2
		Heuristic	156.49	0	0.301	2
		Stable	156.49	0	0.301	2
14 Bus	24	Basic	615.05	0	0.117	3
		Heuristic	615.05	0	0.117	3
		Stable	615.05	0	0.117	3
	84	Basic	606.86	0	0.042	1
		Heuristic	606.86	0	0.042	1
		Stable	606.86	0	0.042	1
	364	Basic	644.73	2.20	0.206	4
		Heuristic	644.73	0	0.206	4
		Stable	644.73	0	0.206	4
18 Bus	24	Basic	608.41	4.43	0.005	1
		Heuristic	608.41	0.45	0.005	1
		Stable	608.41	0	0.005	1
	84	Basic	601.52	5.97	0.003	1
		Heuristic	601.52	1.10	0.003	1
		Stable	601.52	0	0.003	1
	364	Basic	616.17	8.04	0.001	1
		Heuristic	616.17	3.06	0.001	1
		Stable	616.17	1.20	0.001	1
24 Bus	24	Basic	1591.90	6.88	0.079	3
		Heuristic	1589.80	2.20	0.065	2
		Stable	1589.80	2.43	0.065	2
	84	Basic	1564.41	7.57	0.024	1
		Heuristic	1564.41	2.07	0.024	1
		Stable	1564.41	1.70	0.024	1
	364	Basic	1617.22	10.21	0.101	2
		Heuristic	1617.22	4.08	0.101	3
		Stable	1617.22	2.85	0.101	3

provided much tighter bounds and better solutions. In particular, for 24 bus 24 day instance, grouping algorithms have found a new best-known objective with fewer potential solutions in the pool. In addition, for 24 bus 364 days instance, grouping has managed to find an additional high-quality solution that was not discovered previously with the basic SD framework.

When comparing the two grouping algorithms, stable grouping outperforms heuristic algorithm in many instances. While heuristic grouping only provides better bounds for 18 Bus 24 and 18 Bus 84 instances, stable grouping has managed to prove optimality. For the instances that did not converge, stable grouping provides tighter bounds compared to the heuristic grouping approach. This performance is expected as stable grouping takes preferences of all groups into account, whereas the heuristic follows a greedy approach. The significant difference in performance also indicates that selecting which scenarios are to be grouped plays a significant role.

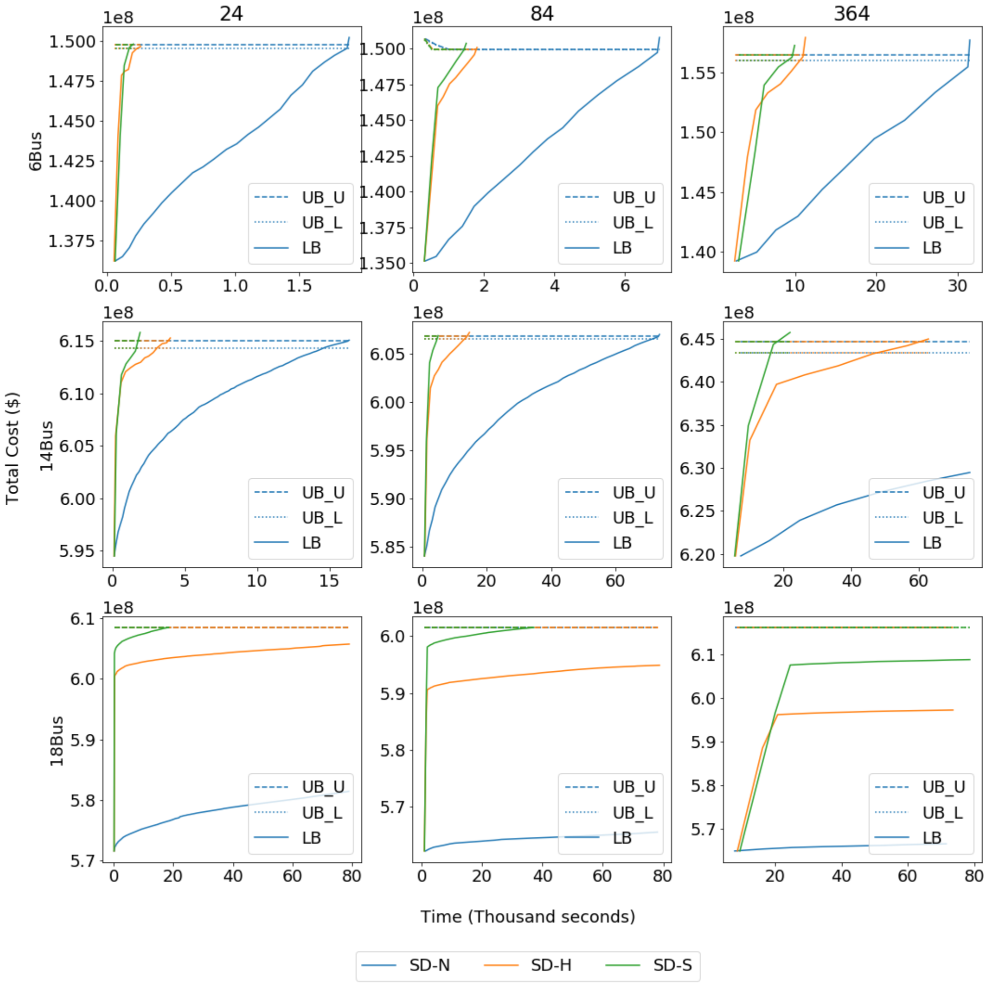


Figure 6.9: Performance of scenario decomposition with grouping

The difference in performance with two grouping mechanisms is also visible in Fig. 6.9, which illustrates the performance of the algorithms in terms of bound improvement. In the figure, SD-N represents scenario decomposition with no extensions, SD-H indicates scenario decomposition with heuristic grouping and SD-S denotes scenario decomposition with stable grouping. As shown in Fig. 6.9, grouping has improved the lower bound and the convergence rates of the SD algorithm significantly regardless of the grouping mechanism. In addition, according to 6 bus 84 day instance, when the best-known solution was not obtained in the first few iterations, grouping has aided to find it earlier in the process. This improvement in performance demonstrates that dissimilar grouping provides a number of benefits to the SD algorithm, (i) raise the lower bound significantly, (ii) improve the convergence rates, and (iii) provide high-quality solutions earlier in the process. When comparing the two schemes, for the 14-bus 84 day instance, stable grouping has reduced the time required to solve the instance by a factor of 15, while heuristic grouping reduced the time only by a factor of 5. This gap in performance also tends to increase with larger

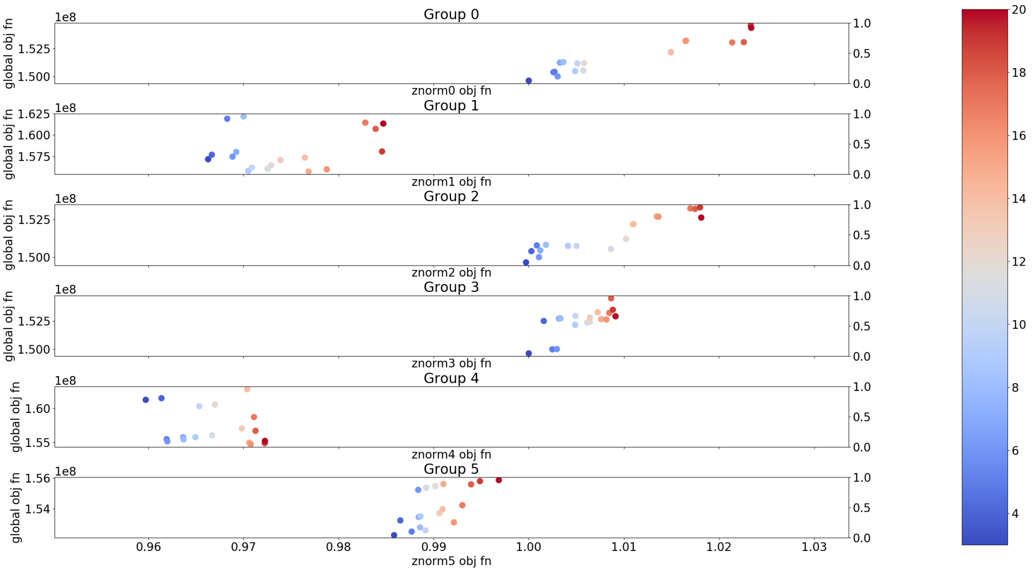


Figure 6.10: Progression of solutions with stable grouping

test cases like 18-bus system, as both 18-bus 24 day and 18-bus 84 day instances show that heuristic grouping is far away from reaching optimality compared to the stable approach.

One of the major reason for remarkable performance with stable grouping is its significant improvement in lower bound. According to the Fig. 6.9 stable grouping tends to provide much stronger bounds compared to the heuristic grouping approach. This is mainly due to the fact that stable groups contain scenarios that are more dissimilar to each other, hence they provide investment solutions that are better compromises than their individually preferred solutions. This is depicted in Fig. 6.10, which illustrate the progression of solutions for 6 bus 24 day instance with stable grouping. Note that the sub-problems now include groups of four scenarios, thus the figure presents the entire set of daily scenarios. Similar to the previous chapter, the progression is shown in terms of the global objective function value and normalised objective value for the individual group. As shown in Fig. 6.10, the first preferred solutions provided by half of the groups are located almost at the globally best-known objective (1.0 in the x-axis), while the other half initialise their preferred solutions quite close (0.96 or more) to the globally best-known objective. This evidently shows that the resulting lower bound is much stronger with grouping. Because the initial solutions are located quite close to the globally best-known objective, grouping has also reduced the number of evaluated solutions (both pre-optimal and post-optimal) required to reach the globally best known objective. As a result, in addition to the strong lower bound, reduction in evaluated solutions has also contributed towards the acceleration of convergence rates – evaluation is generally the more expensive step. Fig. 6.11 illustrates this reduction in explored search space with grouping, where grey dots indicate the solutions explored by the SD algorithm without grouping, while the coloured dots denote the solutions explored by the SD algorithm with stable grouping. It

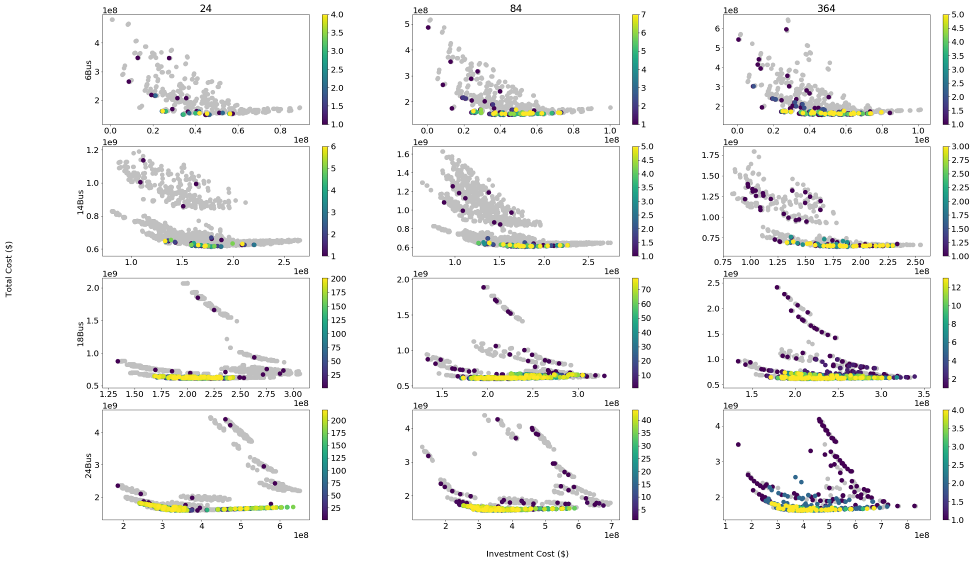


Figure 6.11: Explored solution space with stable grouping

is clearly visible that, grouping has reduced the number of explored solutions, especially outlying candidate solutions with extreme costs.

Another factor that has contributed to improving the convergence rates with stable grouping is the improved effectiveness in integer cuts. Since stable grouping bundles dissimilar scenarios in a more systematic manner, the groups (sub-problems) are more similar to each other. As a result, the integer cuts generated from a particular sub-problem have a higher impact on other sub-problems, because they cut off solutions not only preferred by themselves but also preferred by other sub-problems. Thus, dissimilar grouping further contributes to the algorithm convergence by improving the effectiveness of integer cuts implicitly. This similarity in sub-problems can be observed in Fig. 6.12, which shows the performance of solutions on other sub-problems. The green line indicates where a sub-problem is providing its own solutions. Fig. 6.12 clearly shows that solutions generated by all sub-problems perform well on other sub-problems as the maximum normalised objective for every group is quite low (for most of the sub-problems, maximum is 1.1).

Although grouping has improved the performance of the SD algorithm drastically, its application is limited. Grouping increases the sub-problem solving time, because the sub-problem instances are larger. Especially, sub-problems with dissimilar grouping, as more effort is required to agree on a solution when the properties within the sub-problem disagree with each other [150]. This performance with grouping is illustrated in Fig. 6.13, which depicts the phase performance for 14 bus 24 day instance and 24 bus 24 day instance. From the figure it can be observed that, grouping has significantly increased the time taken by the candidate solution generation phase and decreased the time taken by the evaluation phase. Especially for large systems like the 24-bus system (Fig. 6.13b), the time taken by the candidate solution generation phase in later iterations is extremely high. This suggests that increasing group size is limited to smaller test cases, although it improves the bounds

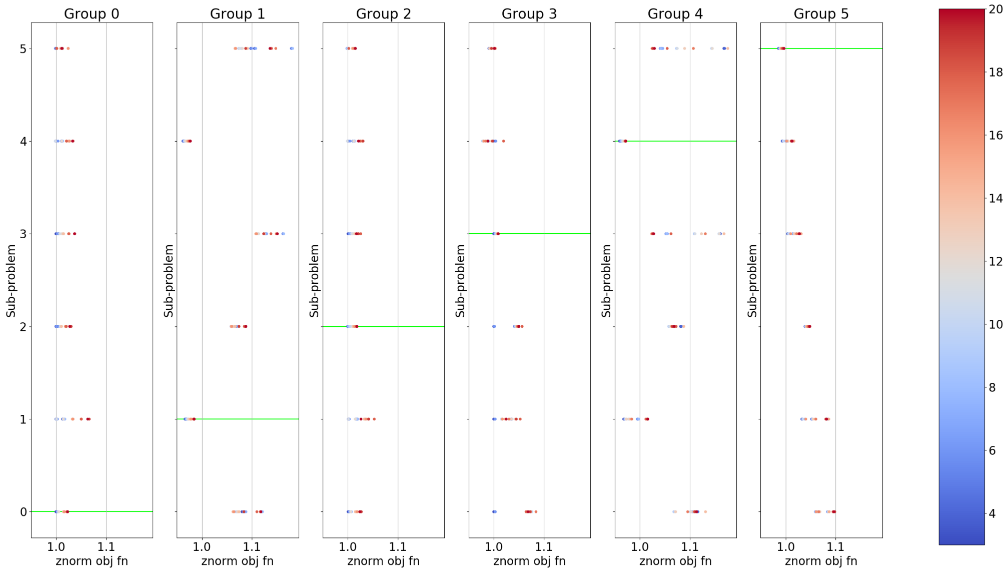


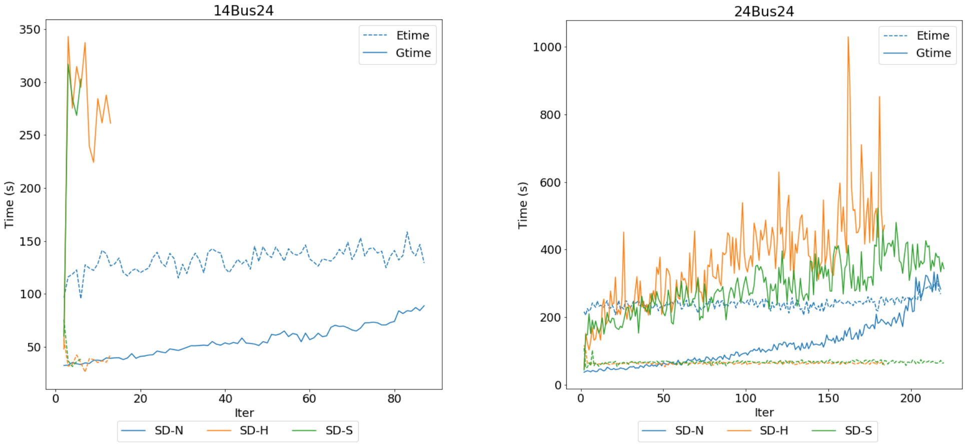
Figure 6.12: Solution performance on other sub-problems with grouping

and convergence rates, and has been already exploited in the SD context. Thus, any further improvement to the algorithm convergence requires additional mechanisms like branching or diving.

6.5.3 Impact of Branching

To investigate the impact of branching and compare the performance with diving, the scenario decomposition procedure was executed with both branching (local and multiple) and diving algorithms. The algorithms were implemented on the scenario decomposition framework with stable grouping. While diving algorithm was applied to all single-period instances, branching was applied only to 18 bus and 24 bus systems. Because branching requires a certain number of iterations to learn the frequent combinations, and all the instances of 6 bus and 14 bus systems converge within few iterations (with stable grouping), only larger test cases like 18 bus and 24 bus were utilised to examine the impact of branching with single-period instances. For two-period instances, both diving and branching were applied to all instances. Fig. 6.14 and Fig. 6.15 depict the obtained results with different algorithms, where SD-N refers to the basic scenario decomposition framework, SD-S indicates stable grouping, SD-D denotes stable grouping with diving, SD-L specifies stable grouping with local branching and finally, SD-V signifies stable grouping with multiple branching. In addition, the complete set of results for both single-period and two-period instances are provided in Appendix B.

As shown in Fig. 6.14 and Fig. 6.15, diving has not significantly improved the performance in many instances except for the 18 bus 24 single-period instance, in which diving shows a noticeable improvement compared to the SD algorithm with grouping. In fact, test cases with many days show adverse performance for diving by lagging behind and



(a) 14 Bus System (b) 24 Bus system

Figure 6.13: Phase performance with grouping

converging later than the grouping only algorithm (e.g. 6-bus 84 day and 14-bus 364 day instances in Fig. 6.14). Diving solves sub-problems repeatedly to generate and validate partial cuts. So for test cases with many days, partial cut validation process is expensive, due to the increased number of sub-problems. This poor performance is further evident in Fig. 6.15 with two-period formulation, where the sub-problems are twice the size of the single-period formulation and double the number. Another factor that has affected the poor performance of diving is the weak partial cuts. Since partial cuts generated by diving have many decisions, they are long and comparatively weak. Hence, many of them are required to raise the lower bound and improve the performance.

In contrast, both branching mechanisms show improved convergence rates in single-period and two-period instances. For example, in single-period 18-bus 84 day instance local branching has reduced the computational time by a factor of 5, while multiple branching has reduced the time by a factor of 3. Larger computational gains are observed with two-period instances, where 6-bus 24 day instance with local branching shows a reduction in computational time by a factor of 7, while multiple branching shows a reduction in computational time by a factor of 5. This improvement in convergence rates is mainly due to two reasons. First, branching fixes a set of decisions, so the time required to generate solutions decreases because of the reduced branch and bound tree. Second, branching forces all the sub-problems to explore a certain combination, so the ability to produce good quality solutions with that combination is determined earlier in the process. Thus, branching improves convergence rates by exploring fewer poor-quality solutions in total.

This elimination of poor-quality solutions is visible in Fig. 6.16 - Fig. 6.18, which illustrate the progression of solutions for the single-period 18 bus 24 day instance with stable grouping, stable grouping with local branching and stable grouping with multiple branching respectively. The colour ramp indicates the solutions generated by the main SD framework with grouping, while others denote the solutions generated during the branching mechanism. According to Fig. 6.16 - Fig. 6.18, local and multiple branching have avoided

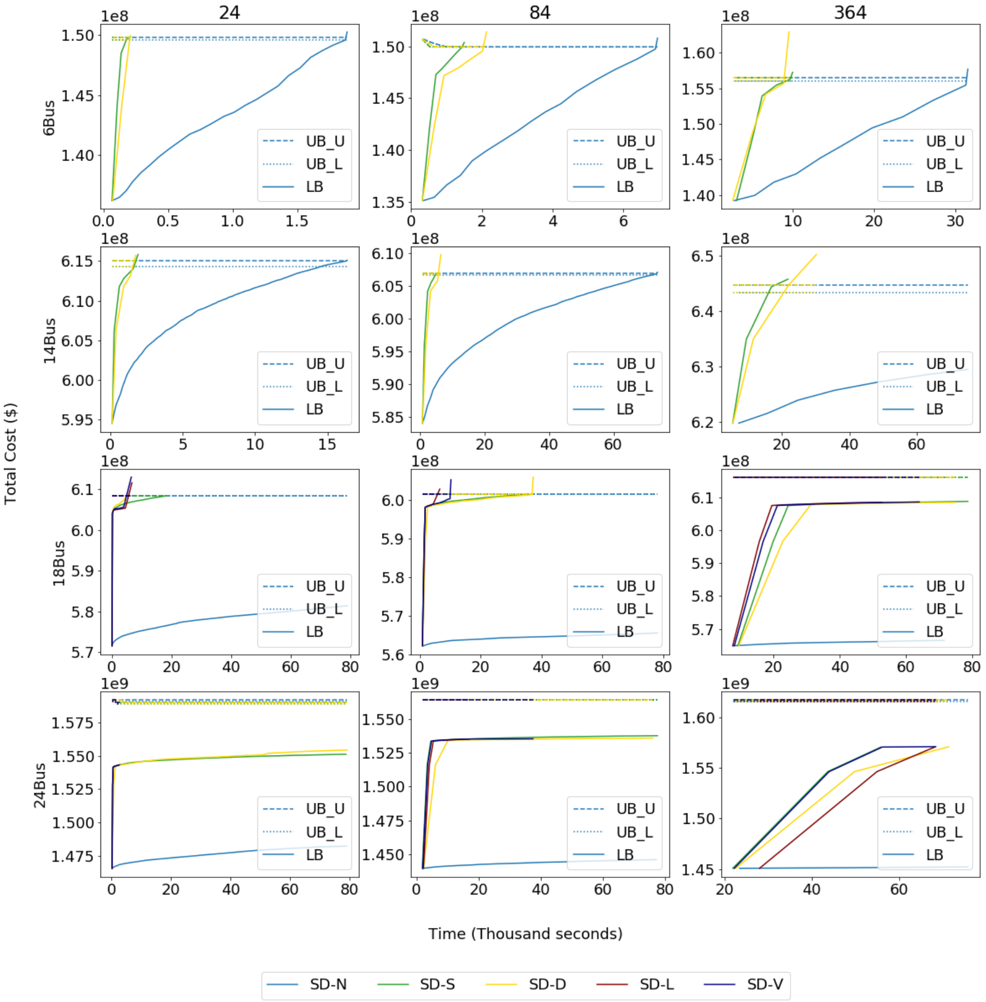


Figure 6.14: Performance of scenario decomposition with grouping and branching

multiple series of poor-quality solutions (higher global objective function values) that were generated and evaluated by the stable grouping only algorithm, especially at Group 0.

When comparing the two branching mechanisms, according to Fig. 6.14 and Fig. 6.15, local branching outperforms multiple branching in many instances except for single-period 18-bus 24 day and two-period 6-bus 84 day instances. According to Table 6.2, which summarises the results obtained for two-period 24-bus 24 days instance, local branching has found a new upper bound with 25 potential solutions in the pool. This performance from local branching is expected as it searches around the current best-known solution, where it is likely to have good-quality solutions. Therefore, local branching can be utilised to generate better quality solutions when the solution quality needs to be improved.

The differences in performance with local and multiple branching can be analysed through Fig. 6.17 and Fig. 6.18, which illustrate the progression of solutions with local and multiple branching. Fig. 6.17 depicts the solutions that were generated during the execution of two local branches (B1-10 and B2-11) at SD iteration 10 and 11 (Recall that

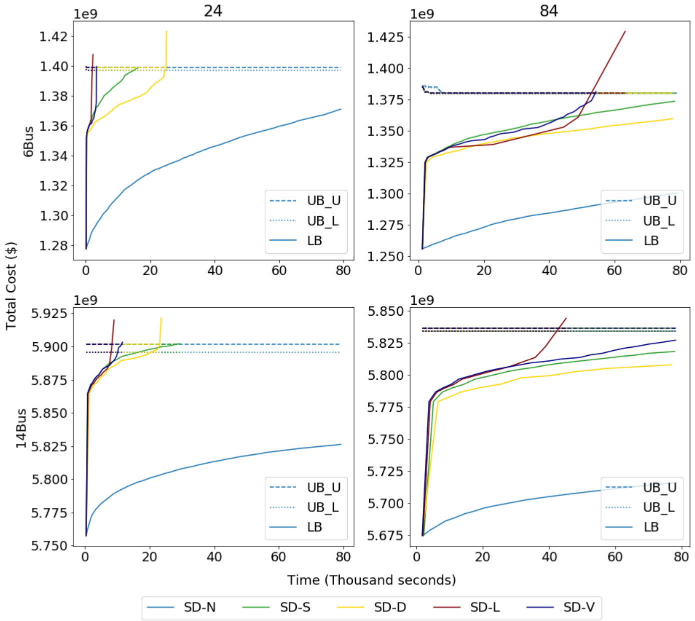


Figure 6.15: Performance of multi-period SD with grouping and branching

Table 6.2: Summary of results for two-period 24 bus 24 days instance

Type	Best Obj. (M$)	Opt. gap (%)	Res. gap. (%)	Num. sol.
Basic	14706.20	7.757	0.0802	14
Optimal G.	14621.78	2.947	0.0748	8
Diving	14627.42	2.980	0.0751	3
Local B.	14620.66	3.004	0.0740	25
Multiple B.	14621.78	3.009	0.0748	4

in the current setting branching initiates at iteration 10 in the SD algorithm). Fig. 6.18 illustrates the execution of multiple branches based on frequent combinations from SD iteration 10 -15. Recall that, in multiple branching, frequent combinations are obtained from individual groups and are prioritised based on length and prevalence (Refer to Section 6.4).

According to Fig. 6.17, local branching has avoided generation and evaluation of poor quality solutions compared to the stable grouping algorithm depicted in Fig. 6.16, especially at Group 0, 2 and 3. Instead, local branching has generated and evaluated a new series of better quality solutions (visible in the y-axis). In multiple branching, although it is not significant, a number of poor quality solutions have been explored. For example, when the frequent combination of group 0 (indicated by dark blue G0-10) is forced on to other groups, the obtained solutions are far beyond optimality (larger objective values in y-axis). Similar results can be observed with frequent combination of group 2 (indicated by light blue G2-11), where the generated solutions are outliers for other groups. These frequent combinations make the multiple branching process inefficient, and they suggest that the approach can be improved by filtering which frequent combinations to explore and/or by improving the execution order. In this regard, the main advantage that local

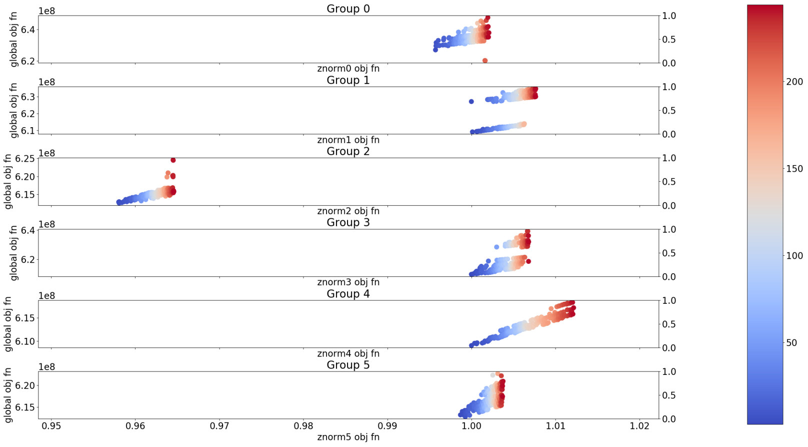


Figure 6.16: 18-bus 24 day instance with stable grouping

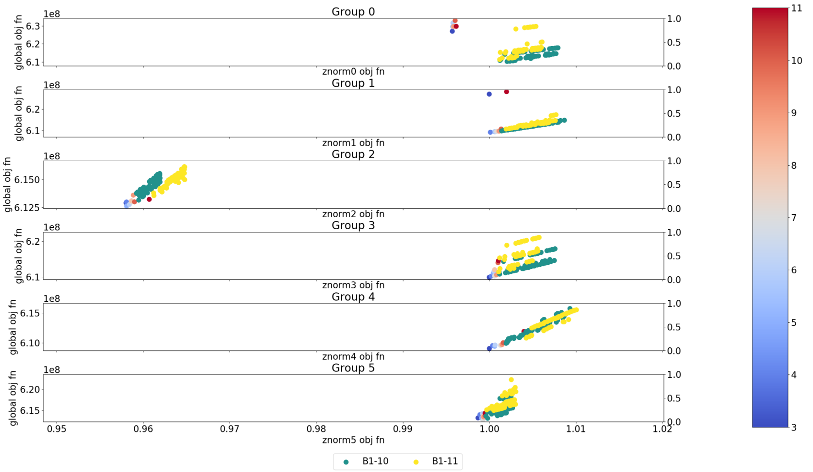


Figure 6.17: 18-bus 24 day instance with stable grouping and local branching

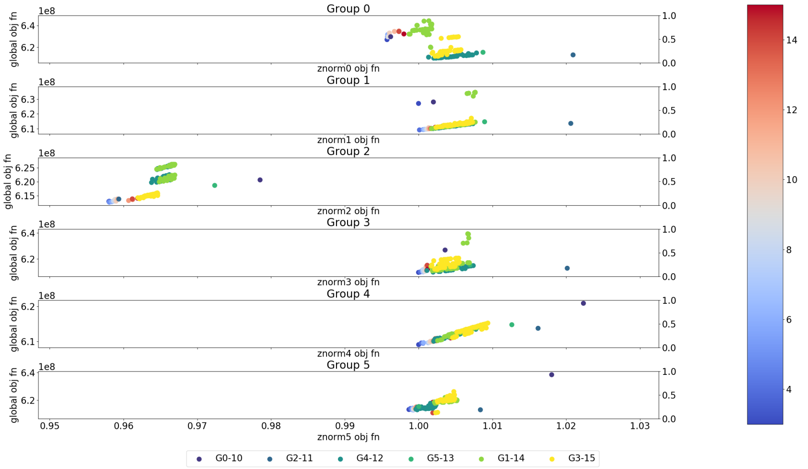


Figure 6.18: 18-bus 24 day instance with stable grouping and multiple branching

branching has over multiple branching is to discover new solutions that might be optimal during the branching process. However, as shown in Fig. 6.14 and Fig. 6.15, local branching has the risk of over-branching and eliminating solutions more than necessary. Since latter branching combinations consist only a few decisions, validation may require evaluation of many irrelevant solutions. Thus, a combined approach of local and multiple branching mechanisms may improve the efficiency of the branching process.

Therefore, to improve the convergence rate of the SD algorithm, the main challenge is to explore near-optimal solutions and eliminate them earlier in the process possibly without evaluation (via partial cuts), while minimising the exploration of average quality solutions.

6.6 Conclusion

This chapter proposed and explored two extensions i.e, grouping and branching, to tackle the issue of diversity in scenarios and to mitigate the issue of weak integer cuts.

The grouping method bundles dissimilar scenarios utilising the information revealed in the previous iterations. In particular, the chapter proposed two grouping strategies: heuristic and stable. The experimental analysis demonstrated drastic savings in computational time with grouping, due to the significant improvement in lower bound and reduction in explored solution space. Stable grouping showed improved performance compared to the heuristic grouping, as the method takes grouping preferences of all scenarios into account. However, the potential of grouping has already been exploited, as further grouping could deteriorate the overall performance of the SD algorithm because of the larger sub-problem solving times. This further emphasises the need for other mechanisms such as branching to improve the algorithm performance.

Branching leads to the generation of stronger partial cuts based on the identification of frequent combinations of decision variable values. In this chapter, we discussed two types of frequent combinations: one resulting from a pool of near-optimal solutions (local branching) and the other resulting from preferred regions of individual sub-problems (multiple branching). From the experimental analysis, it was observed that both branching mechanisms (with stable grouping) are effective, improve the efficiency of the SD algorithm and outperforms the "diving" algorithm. However, local branching outperformed multiple branching in many occasions. It found better quality solutions and converged prior to multiple branching. According to the outcomes, the execution of certain frequent combinations in multiple branching was not necessary. Thus, by modifying the approach to select the most critical set of frequent combinations, the efficiency of the multiple branching approach can be enhanced.

Even with grouping and branching, optimality gaps provided by larger instances are quite significant. Due to the sheer number of solutions lying between the preferred solutions and the optimal solution, and the fact that solutions are removed only through integer cuts, a large number of solutions must be evaluated and cut to prove optimality. This demonstrates that the key challenge in the scenario decomposition approach is not about finding good-quality solutions, rather it is eliminating solutions in an efficient manner in order to prove optimality.

Chapter 7

Conclusion and Future Work

7.1 Thesis Summary and Contributions

Energy system planning addresses the need for new capacities in generation, transmission and storage facilities. In general, these problems contain many inter-temporal dependencies, which are ignored due to computational limitations. However, with integration of large-scale renewable energy-based generation, consideration of these dependencies are becoming crucial in the planning stage, because they characterise the operational flexibility in a system and address the variability and uncertainty inherent to renewable energy-based generation.

The experimental analysis carried out in Chapter 3 showed that ignoring temporal dependencies in planning stage could lead to sub-optimal investment solutions. In particular, by considering three operational models with different details of operational flexibility: economic dispatch (no operational flexibility), economic dispatch with ramping (consider operational flexibility approximately via ramping constraints) and full unit commitment (accurately represent flexibility constraints), the analysis showed that traditionally used operational models like economic dispatch and economic dispatch with ramping provide sub-optimal investment solutions that are operationally expensive. More specifically, Chapter 3 showed that representing operational flexibility accurately requires unit commitment with chronological demand and renewable generation profiles as the embedded operational model.

Solving unit commitment in the planning context is computationally intensive, because of its numerous integer (binary) variables and inter-temporal constraints. Besides, a fine resolution is essential to accurately capture the rapid variations in renewable energy based generation. Thus, a planning problem that considers a representative year with hourly operational conditions often leads to a high-dimensional MIP problem, that is computationally limited even for small systems. Therefore, in this thesis, we developed computationally efficient methods to incorporate unit commitment in energy system planning problems. The thesis proposed solution methods to reduce the problem size and solve a complex MIP problem with a large number of binary variables in both investment and operational stages, and inter-temporal constraints.

In general, large-scale optimisation problems are solved using decomposition techniques based on mathematical programming. As shown in Chapter 2, application of decomposition techniques such as Bender's and Dantzig-Wolfe decomposition have been less effective on planning problems with unit commitment, because of the binary variables and inter-temporal constraints. Therefore, in Chapter 5 we proposed a decomposition framework based on an existing scenario decomposition (SD) method to decompose the deterministic GTEP-UC problem (generation and transmission expansion planning with unit commitment). Experimental analysis demonstrated that the proposed approach based on SD is capable of finding high-quality solutions in the first few iterations, and is computationally more efficient compared to both classic and multi-cut Bender's decomposition approaches. In addition, if implemented in a parallel processing environment, the SD framework has the potential to make GTEP-UC and related problems with binary variables and inter-temporal constraints much more tractable for practical size instances.

Moreover, further computational speed-ups can be gained in the sub-problem level and as a result in the overall problem, by utilising the adaptive resolution approach proposed in Chapter 4 i.e., Sliding Window with Backtracking (SWBT) algorithm. Unlike the coarse resolution with the fixed length, the proposed SWBT approach reduces the resolution only where it can be afforded, thus mitigates the risk of ignoring important time periods that characterises the flexibility of the system. The experimental analysis in Chapter 4 exhibited that the method provides high-quality solutions for unit commitment problems with significant gains in computational speeds, especially for larger systems, and is capable of capturing rapid variations in renewable energy-based generation.

In addition, two extensions were proposed in Chapter 6 to improve the performance of the scenario decomposition framework for GTEP-UC; grouping approaches to tackle the issue of diversity in scenarios and branching techniques to generate stronger integer cuts. Grouping approaches grouped dissimilar scenarios into one sub-problem to maintain similarity between the sub-problems. The dissimilarity was measured abstractly (e.g. penalty cost, solution distance) eliminating the need to carry out high-dimensional data analysis. The methods showed that, by grouping extremely different scenarios into one sub-problem, and forcing them to compromise their investment solutions, the lower bound and rate of convergence can be improved significantly. Especially, in the stable grouping approach, which takes differences in all sub-problems into account to ensure that extreme scenarios are grouped into one sub-problem.

On the other hand, branching mechanisms showed that by fixing sets of decisions, the outcomes of partial investment solutions can be realised earlier in the solution process avoiding any solutions with that combination being generated again. In particular, the local branching approach showed that by searching around the best-known solution, high-quality solutions can be generated and eliminated more efficiently. In addition, multiple branching showed that by forcing all the sub-problems to search a particular region preferred by a certain sub-problem simultaneously, preferred regions of sub-problems can be explored and eliminated earlier in the solution process improving the lower bound and convergence rates.

In this regard, the **contributions** of this thesis are as follows.

1. Through an experimental analysis that investigates the impact of unit commitment model in the planning context with respect to traditionally used economic dispatch model, the thesis showed that representing inter-temporal dependencies in operational conditions and limitations in thermal generators is influential in energy system planning with high shares of renewable energy-based generators and thermal generators. To the best of our knowledge, the study is the first of its kind to compare economic dispatch with full unit commitment model (with binary variables and detailed set of flexibility constraints) that includes one-year planning horizon at hourly resolution (without employing representative days and unit clustering) and a transmission network.

2. The thesis proposed a decomposition framework based on an existing scenario decomposition (SD) method to decompose the computationally intensive energy system planning with unit commitment problem. The proposed framework is computationally more efficient in terms of both run-time and memory compared to the Bender's decomposition. In addition, the scenario decomposition framework is capable of dealing with multi-period formulation with a relatively straightforward extension of single-period formulation with grouping. The SD method is also capable of finding high-quality solutions earlier in the solution process, especially when the sub-problems naturally deliver solutions that are near-optimal. Thus, SD offers a good heuristic to derive high-quality candidate options, when reaching optimality is hard and/or not necessary. Such an application is suitable for real-world problems that involve an enormous number of electric buses (nodes), generators and transmission lines. Since scenario decomposition was originally designed for stochastic programming problems, the proposed decomposition framework is also extensible for stochastic formulations.

3. The thesis investigated the scenario decomposition (SD) approach in detail and brought clarity to the fundamental elements of SD framework, as applied to deterministic planning problem with a large number of operational cases that consists binary variables and are bounded by inter-temporal constraints. The analysis introduced the notion of "pre-optimal" and "post-optimal" in generated solution, and showed that, for the SD algorithm to converge, at least one sub-problem must start exploring its "post-optimal" solutions. In addition, the analysis identified that diversity in scenarios and weak integer cuts are major factors that affect the performance of the SD algorithm.

4. The thesis proposed two grouping approaches: heuristic grouping and stable grouping, that group dissimilar scenarios to tackle the issue of diversity in scenarios and improve the performance of the SD framework. The proposed approaches measure dissimilarity abstractly eliminating the need to carry out high-dimensional data analysis. If a generic metric such as solution distance is chosen, the proposed grouping methods are also applicable to other problem domains with similar characteristics.

5. The thesis also proposed two branching mechanisms: local branching and multiple branching, that improve the performance of the scenario decomposition algorithm by removing sets of solutions in targeted regions, particularly near-optimal solutions (local branching) and preferred solutions of sub-problems (multiple branching). To identify these targeted regions, the thesis introduced the concept of frequent combinations. In addition, when closing the bound gap is difficult, local branching can be utilised to find better quality solutions.

6. The thesis proposed an algorithm named Sliding Window with Backtracking (SWBT) to obtain a resolution that is adaptive to the given data set. The algorithm is capable of reducing the size of the problem providing high-quality solutions and outperforms the commonly used fixed resolution approach. Thus, it is ideal for optimisation and simulation problems that involve high resolutions, to reduce the computational burden with minimum approximation errors. Such application is useful for planning and operational problems with shorter horizons such as rolling horizon based techniques and sub-problems in decomposition techniques.

7. Finally, the thesis presents four test cases that were developed to incorporate candidate investment options and technical parameters required for unit commitment problem. Existing test cases do not contain input data desired for both unit commitment model and capacity expansion planning. And, details of real systems are not publicly available. Thus, four test cases collected from multiple sources were upgraded to incorporate required input parameters. These test cases are currently available to the public via the Monash figshare repository.

With the transition towards renewable energy-based generation, I believe that the proposed methods and analytical discussions in this thesis will help to bridge the gap between the theoretical need to consider operational flexibility and the practical requirements of computational resources to incorporate unit commitment in planning problems, and provide a power system that is both reliable and economically efficient.

7.2 Future Work

In the scenario decomposition framework, large instances provided significant optimality gaps at the end of the maximum time limit even with grouping and branching. Due to the large number of solutions that needs to be cut off to improve bounds, the main challenge in scenario decomposition is not finding good quality solutions but rather eliminating solutions in an effective manner to raise the lower bound and close the optimality gap. Thus, it will be beneficial to investigate other methods that could be employed to improve the efficiency of the proposed scenario decomposition framework. For example, instead of solving all the sub-problems in every iteration, an approach could be devised to optimise which sub-problem to solve, to gain the largest bound improvement. In addition, existing solutions can be perturbed to generate new solutions rather than resolving sub-problems.

Although the current scenario decomposition framework is capable of handling intra-day inter-temporal constraints, the ability to deal with global inter-temporal constraints is limited. For example, hydro generators require consideration of consecutive months to accurately model seasonal storage and water flow constraints. In addition, constraints that represent environmental concerns such as renewable energy targets and carbon emission limits must account for power generation across the entire year. Thus in future, it will be useful to extend the approach to account for global constraints, as they will greatly enhance the utility of the proposed scenario decomposition framework in real-world applications.

All the experiments conducted in this thesis are based on test cases that were upgraded from existing operational test instances. Hence, for a more insightful study, it will be useful to gather all the elements discussed in this thesis, and conduct an experimental analysis using a real system (e.g. Australian energy system). However, the main challenge in this aspect would be collecting all the necessary data, especially the technical characteristics of thermal generators that are subject to confidentiality.

This thesis only considers a static multi-period formulation, where investment decisions for all the periods are made once in the solving process assuming perfect foresight. In reality, long-term expansion planning problems are associated with significant uncertainty and multi-period decisions are made over several stages when the uncertainty pertaining to the previous stage is realised. Such a dynamic formulation accounts for various future possibilities to provide more robust solutions. The proposed decomposition framework based on scenario decomposition permits a relatively straightforward extension for stochastic scenarios due to its stochastic programming origin. Therefore, it will be worthwhile to extend the current scenario decomposition approach to make decisions in multiple stages considering long-term uncertainty.

Appendix A

Test Cases

This chapter presents the four test-cases: 6-Bus, 14-Bus, 18-Bus and 24-Bus, that were updated to incorporate input parameters for energy system planning with unit commitment. The test cases contain investment costs for the generation and transmission expansion planning problem (candidate units), operational costs and technical parameters for the unit commitment problem with DC formulation, and hourly load and renewable generation profiles for a period of one year. These test cases are publicly available at Monash figshare repository [159, 160].

Nomenclature

Bus	Bus number
Ti	Technology Index
IntI	Initial existing status
C^{I}	Investment cost M$/MW
C^{AI}	Annualised cost (M$)
C^{M}	Fixed OM cost (M$/MW/yr)
C^{V}	Variable OM cost ($/MWh)
LS	Life span (yr)
γ	Discount rate (%)
	Nodes
Type	Bus Type
Bkv	Base kv
Lzone	Load Zone
LF	Load Factor
SF	Scaling factor
	Renewable Units
Rmax	Maximum available generation (MW)
Rzone	Renewable zone

	Thermal Unitst
Pmax	Maximum generation (MW)
Pmin	Minimum generation (MW)
IntS	Initial on off status
IntP	Initial generation (MW)
MinU/MinD	Minimum up and down times (hr)
RU/RD	Ramp up and down limits (MW)
SU	Start up limit (MW)
SD	Shut down limits (MW)
HR	Heat rate (MMBTU/MWh)
C^{F}	Fuel cost ($/MMBTU)
C^{G}	Generation cost ($/MWh)
C^{SU}	Star up cost ($/startup)
C^{C}	Commitment cost ($/hr)
	Transmission Lines
From/To	Connecting buses
Fmax	Maximum Flow (MW)
Fmin	Minimum Flow (MW)
X	Reactance (p.u.)

A.1 System Description

This section summarises the upgrading process conducted to generate test cases for energy system planning with unit commitment. In all test cases, the obtained data serve as the existing system, and candidate options were assigned to existing nodes/buses considering a range of technologies.

A.1.1 Bus Data

For each bus, bus type, Base kV, load distribution factor (LF), load zone (Lzone) and scaling factors (SF) are provided (e.g. Table A.4). For the bus type, Matpower case format was followed [161], where 1 indicates PQ bus, 2 refers to PV bus and 3 indicates reference bus. Load zone (Lzone) allows different demand profiles to be assigned for different demand nodes/buses. If one demand profile is chosen for several nodes/buses, load distribution factors (LF) are used to distribute the load among the nodes. For buses that do not contain any load, LF and Lzone are set to zero. The scaling factors (SF) were designed to promote new generating units and transmission lines in the panning problem. In addition, a Base MVA of 100 MVA was considered.

A.1.2 Unit Data

A technology index (Ti) was assigned to each unit according to Table A.1.

Table A.1: Technology index

1	Coal	5	Solar
2	CCGT	6	Battery Storage (Li-ion)
3	OCGT	7	Transmission lines (HV AC)
4	Wind		

Thermal Units

For thermal units (coal, CCGT and OCGT), cost and technical parameters that are necessary for the unit commitment model are provided (e.g. Table A.16). The technical parameters and cost parameters are based on Table A.2 [6] and Table A.3 respectively.

Table A.2: Technical parameters

Technology	Pmin	RU/RD	MinU	MinD
	p.u	p.u	hr	hr
Coal	0.5	0.3	24	12
CCGT	0.3	0.5	6	12
OCGT	0.25	1	1	1

Renewable Units

A renewable zone was assigned to all renewable units (e.g. Table A.6). Wind and solar profiles that determine the generation output of each unit are based on these zones. Note that these zones are different to load zones mentioned above and same zone number does not imply that the units and loads are co-located.

Transmission Lines

For transmission lines, the information required for DC power flow calculations are provided (e.g. Table A.5). The candidate lines were assigned to either increase the capacity in existing lines, or install completely new lines that connect previously unconnected nodes/buses changing the network topology.

A.1.3 Cost Data

The cost parameters for different technologies obtained from [6, 162] are summarised in Table A.3. Note that investment cost C^{I} shown here is the total cost of installation. Annualised investment cost was calculated using Eq (A.1). Total annual cost C^{AI} provided in test cases are the sum of annualised capital cost and fixed operational and maintenance cost for a year C^{M}. In addition, total generation cost C^{G} is considered as the sum of fuel cost C^{F} and variable OM cost C^{V}.

Table A.3: Cost parameters

Ti	C^{I}	LS	γ	HR	C^{F}	C^{V}	C^{M}	C^{SU+}	C^{G}
1	1.52	30	10	8	2.89	7.33	0.043	54.11	30.45
2	1.28	30	10	7.34	5.78	4.73	0.025	16.23	47.15
3	0.77	30	10	14.31	5.78	13.4	0.017	28.14	96.12
4	1.50	25	10	NA	NA	NA	0.060	NA	0
5	0.83	25	10	NA	NA	NA	0.015	NA	0
6	1.00	15	10	NA	NA	NA	0.012	NA	0
7	1100 *	50	7	NA	NA	NA	NA	NA	NA

* The provided value is in \$/MW/km.

+ The provided value is in \$/MW/startup.

$$\mathrm{C}^{AI} = \frac{\gamma\,\mathrm{C}^{I}}{1-(1+\gamma)^{-\mathrm{L}S}} \tag{A.1}$$

A.1.4 Demand, Solar and Wind Profiles

Hourly load and renewable generation profiles obtained from the Australian Energy Market Operator (*aemo.com.au*) for a period of one year are provided. For load profiles appropriate scaling factors were applied, and for renewable generation profiles, hourly capacity factors are presented. Due to the enormous amount of data, these profiles are not shown in this section and can be found in the repository [160].

A.2 Test Cases

This section provides the details of the four test cases: 6-Bus, 14-bus, 18-Bus, and 24-Bus. The technical parameters of the existing units were extended for the unit commitment problem, and candidate generators and transmission lines were added to update the test case for capacity expansion planning problems. The existing or candidate status of an unit is determined by the initial existing status “IntI”, where 1 indicates “existing” and 0 otherwise.

A.2.1 6-Bus System

The 6-bus system was obtained from *motor.ece.iit.edu*, and can be illustrated as follows.

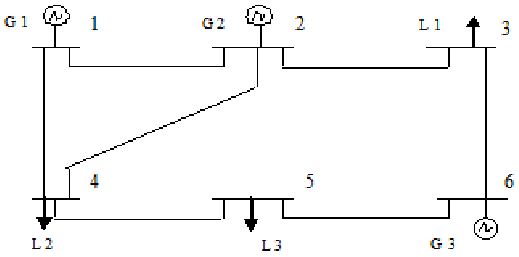


Figure A.1: 6-bus system

Table A.4: 6-bus nodes

Bus	Type	BkV	LF	Lzone	SF
1	3	230	0	0	0
2	2	230	0	0	0
3	2	230	1	1	0.0225
4	1	230	1	2	0.105
5	1	230	1	3	0.105
6	1	230	0	0	0

Table A.5: 6-bus transmission lines

Line	From	To	X	Fmax	Fmin	IntI	C^{AI}
1	1	2	0.17	200	-200	1	0
2	2	3	0.037	110	-110	1	0
3	1	4	0.258	100	-100	1	0
4	2	4	0.197	100	-100	1	0
5	4	5	0.037	100	-100	1	0
6	5	6	0.14	100	-100	1	0
7	3	6	0.018	100	-100	1	0
8	1	5	0.258	100	-100	0	0.39
9	4	5	0.037	100	-100	0	0.24
10	5	6	0.14	100	-100	0	0.56

Table A.6: 6-bus renewable generators

Gen	Bus	Ti	Rmax	Rzone	IntI	C^{AI}
1	4	4	70	1	0	15.77
2	5	4	50	1	0	11.26
3	6	5	115	3	0	12.24
4	5	5	100	3	0	10.64

A.2.2 14-Bus System

The 14 bus system is based on the IEEE 14 bus case, which was obtained from matpower test cases [161]. The 14 bus system is presented in Fig. A.2 [163].

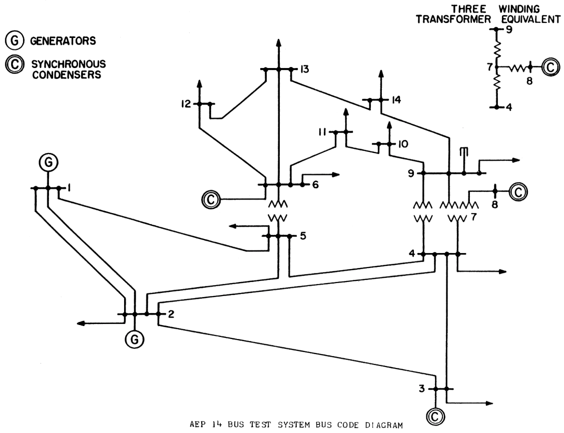

G GENERATORS
C SYNCHRONOUS CONDENSERS
THREE WINDING TRANSFORMER EQUIVALENT
9
7
8
4
13
12
14
11
10
9
1
6
5
8
7
4
2
3
AEP 14 BUS TEST SYSTEM BUS CODE DIAGRAM

Figure A.2: IEEE 14-bus system

Table A.7: 14-bus nodes

Bus	Type	BkV	LF	Lzone	SF
1	3	230	0	0	0
2	2	230	0.0838	1	0.1563
3	2	230	0.3637	1	0.1563
4	1	230	0.1846	1	0.1563
5	1	230	0.0293	1	0.1563
6	2	230	0.0432	1	0.1563
7	1	230	0	0	0
8	2	230	0	0	0
9	1	230	0.1139	1	0.1563
10	1	230	0.0347	1	0.1563
11	1	230	0.0135	1	0.1563
12	1	230	0.0235	1	0.1563
13	1	230	0.0521	1	0.1563
14	1	230	0.0575	1	0.1563

Table A.8: 14-bus renewable generators

Gen	Bus	Ti	Rmax	Rzone	IntI	C^{AI}
1	6	4	120	1	0	27.03
2	12	4	80	1	0	18.02
3	7	4	30	1	0	6.76
4	9	5	110	3	0	11.71
5	14	5	90	3	0	9.58
6	3	4	40	3	0	4.26

Table A.9: 14-bus transmission lines

Line	From	To	X	Fmax	Fmin	IntI	C^{AI}
1	1	2	0.05917	100	-100	1	0
2	1	5	0.22304	100	-100	1	0
3	2	3	0.19797	100	-100	1	0
4	2	4	0.17632	100	-100	1	0
5	2	5	0.17388	100	-100	1	0
6	3	4	0.17103	100	-100	1	0
7	4	5	0.04211	100	-100	1	0
8	4	7	0.20912	100	-100	1	0
9	4	9	0.55618	100	-100	1	0
10	5	6	0.25202	100	-100	1	0
11	6	11	0.1989	100	-100	1	0
12	6	12	0.25581	100	-100	1	0
13	6	13	0.13027	100	-100	1	0
14	7	8	0.17615	100	-100	1	0
15	7	9	0.11001	100	-100	1	0
16	9	10	0.0845	100	-100	1	0
17	9	14	0.27038	100	-100	1	0
18	10	11	0.19207	100	-100	1	0
19	12	13	0.19988	100	-100	1	0
20	13	14	0.34802	100	-100	1	0
21	1	2	0.05917	100	-100	0	0.40
22	2	3	0.19797	100	-100	0	0.24
23	1	3	0.34802	150	-150	0	1.08
24	10	9	0.19207	70	-70	0	0.22
25	10	14	0.27038	150	-150	0	0.83
26	6	8	0.04211	90	-90	0	0.14
27	8	9	0.19797	100	-100	0	0.28

A.2.3 18-Bus System

The 18 bus system represents a section of the IEEE 118 bus system obtained from *motor.ece.iit.edu*. In particular, it represents the buses 1-7, 11-19, 113 and 117. The 18 bus system is depicted in Fig. A.3 [164].

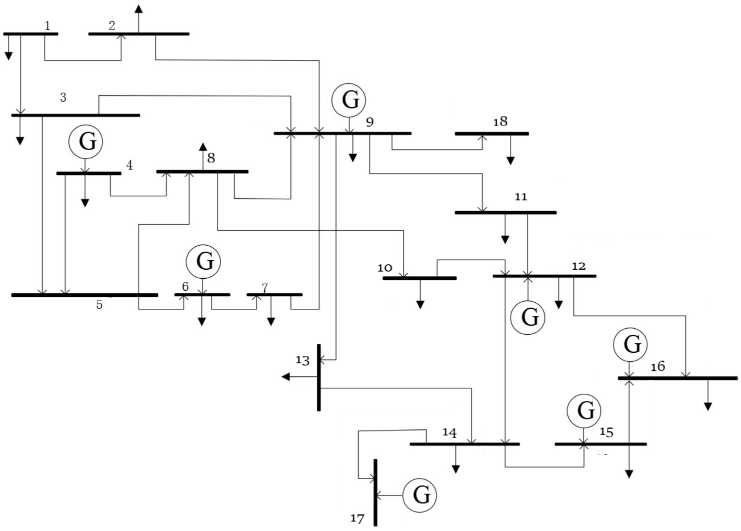


Figure A.3: 18-bus system

Table A.10: 18-bus nodes

Bus	Type	BkV	LF	Lzone	SF
1	3	138	0.081341368	1	0.2312
2	1	138	0.031896511	1	0.2312
3	1	138	0.062200454	1	0.2312
4	1	138	0.047852283	1	0.2312
5	1	138	0	0	0
6	1	138	0.082933937	1	0.2312
7	1	138	0.030303943	1	0.2312
8	1	138	0.111645305	1	0.2312
9	1	138	0.074956057	1	0.2312
10	1	138	0.054222568	1	0.2312
11	1	138	0.022326057	1	0.2312
12	1	138	0.143541822	1	0.2312
13	1	138	0.039874397	1	0.2312
14	1	138	0.01754834	1	0.2312
15	1	138	0.095689539	1	0.2312
16	1	138	0.071770908	1	0.2312
17	1	138	0	0	0
18	1	138	0.031896511	1	0.2312

Table A.11: 18-bus renewable generators

Gen	Bus	Ti	Rmax	Rzone	IntI	C^{AI}
1	7	4	70	1	0	15.77
2	4	4	140	1	0	31.53
3	14	4	250	1	0	56.31
4	1	4	135	1	0	30.41
5	11	5	80	3	0	8.51
6	10	5	145	3	0	15.43
7	8	5	120	3	0	12.77
8	18	5	135	3	0	14.37

Table A.12: 18-bus transmission lines

Line	From	To	X	Fmax	Fmin	IntI	C^{AI}
1	1	2	0.0999	175	-175	1	0
2	1	3	0.0424	175	-175	1	0
3	4	5	0.00798	500	-500	1	0
4	3	5	0.108	175	-175	1	0
5	5	6	0.054	175	-175	1	0
6	6	7	0.0208	175	-175	1	0
7	4	8	0.0688	175	-175	1	0
8	5	8	0.0682	175	-175	1	0
9	8	9	0.0196	175	-175	1	0
10	2	9	0.0616	175	-175	1	0
11	3	9	0.16	175	-175	1	0
12	7	9	0.034	175	-175	1	0
13	8	10	0.0731	175	-175	1	0

Line	From	To	X	Fmax	Fmin	IntI	C^{AI}
14	9	11	0.0707	175	-175	1	0
15	10	12	0.2444	175	-175	1	0
16	11	12	0.195	175	-175	1	0
17	9	13	0.0834	175	-175	1	0
18	12	14	0.0437	500	-500	1	0
19	13	14	0.1801	175	-175	1	0
20	14	15	0.0505	175	-175	1	0
21	15	16	0.0493	175	-175	1	0
22	16	17	0.117	175	-175	1	0
23	12	16	0.0394	175	-175	1	0
24	14	17	0.0301	175	-175	1	0
25	9	18	0.14	175	-175	1	0
26	4	8	0.0682	175	-175	0	0.97
27	8	9	0.0196	110	-110	0	0.37
28	13	14	0.1801	175	-175	0	1.55
29	14	17	0.0301	175	-175	0	1.36
30	14	15	0.0505	175	-175	0	1.16
31	9	18	0.14	110	-110	0	1.22

A.2.4 24-Bus System

The provided 24 bus system is based on the IEEE Reliability Test System - 96 [165]. The single-area version of this test case was updated by C. Ordoudis *et al.* [166] for electricity market and power system operation studies. In this section, the updated version is extended to incorporate capacity expansion data.

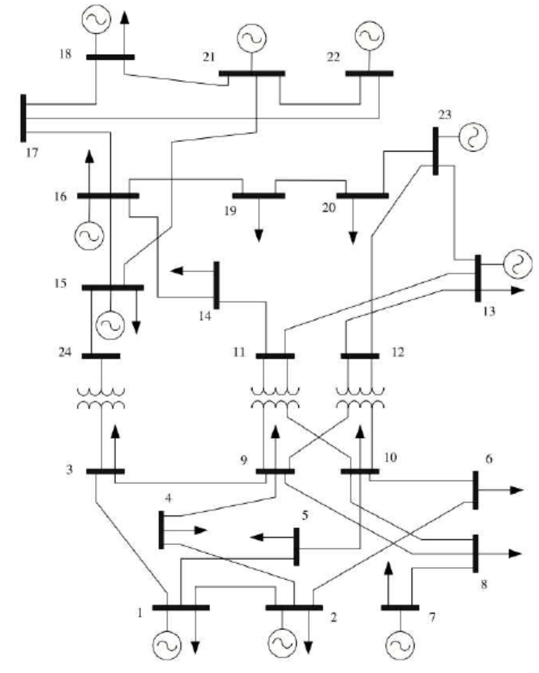


Figure A.4: IEEE RTS 24-bus system

Table A.13: 24-bus renewable generators

Gen	Bus	Ti	Rmax	Rzone	IntI	C^{AI}
1	3	4	420	1	0	94.61
2	5	4	315	1	0	70.95
3	7	4	455	1	0	102.49
4	16	4	245	1	0	55.19
5	21	4	175	1	0	39.42
6	23	4	420	1	0	94.61
7	13	5	115	3	0	12.24
8	14	5	100	3	0	10.64
9	15	5	220	3	0	23.42
10	21	5	110	3	0	11.71

Table A.14: 24-bus nodes

Bus	Type	BkV	LF	Lzone	SF
1	3		0.038	1	0.75
2	1		0.034	1	0.75
3	1		0.063	1	0.75
4	1		0.026	1	0.75
5	1		0.025	1	0.75
6	1		0.048	1	0.75
7	1		0.044	1	0.75
8	1		0.06	1	0.75
9	1		0.061	1	0.75
10	1		0.068	1	0.75
11	1		0	0	0.75
12	1		0	0	0.75
13	1		0.093	1	0.75
14	1		0.068	1	0.75
15	1		0.111	1	0.75
16	1		0.035	1	0.75
17	1		0	0	0.75
18	1		0.117	1	0.75
19	1		0.064	1	0.75
20	1		0.045	1	0.75
21	1		0	0	0.75
22	1		0	0	0.75
23	1		0	0	0.75
24	1		0	0	0.75

Table A.15: 24-bus transmission lines

Line	From	To	X	Fmax	Fmin	IntI	C^{AI}
1	1	2	0.0146	175	-175	1	0
2	1	3	0.2253	175	-175	1	0
3	1	5	0.0907	350	-350	1	0
4	2	4	0.1356	175	-175	1	0
5	2	6	0.205	175	-175	1	0
6	3	9	0.1271	175	-175	1	0
7	3	24	0.084	400	-400	1	0
8	4	9	0.111	175	-175	1	0
9	5	10	0.094	350	-350	1	0
10	6	10	0.0642	175	-175	1	0
11	7	8	0.0652	350	-350	1	0
12	8	9	0.1762	175	-175	1	0
13	8	10	0.1762	175	-175	1	0
14	9	11	0.084	400	-400	1	0
15	9	12	0.084	400	-400	1	0
16	10	11	0.084	400	-400	1	0
17	10	12	0.084	400	-400	1	0
18	11	13	0.0488	500	-500	1	0
19	11	14	0.0426	500	-500	1	0
20	12	13	0.0488	500	-500	1	0
21	12	23	0.0985	500	-500	1	0
22	13	23	0.0884	250	-250	1	0
23	14	16	0.0594	250	-250	1	0
24	15	16	0.0172	500	-500	1	0
25	15	21	0.0249	400	-400	1	0
26	15	24	0.0529	500	-500	1	0
27	16	17	0.0263	500	-500	1	0
28	16	19	0.0234	500	-500	1	0
29	17	18	0.0143	500	-500	1	0
30	17	22	0.1069	500	-500	1	0
31	18	21	0.0132	1000	-1000	1	0
32	19	20	0.0203	1000	-1000	1	0
33	20	23	0.0112	1000	-1000	1	0
34	21	22	0.0692	500	-500	1	0
35	1	2	0.0146	175	-175	0	0.70
36	6	10	0.0642	175	-175	0	0.42
37	7	8	0.0652	350	-350	0	2.23
38	13	23	0.0884	250	-250	0	1.39
39	14	16	0.0594	250	-250	0	1.20
40	15	21	0.0249	400	-400	0	3.19

Table A.16: 6-bus thermal generators

Gen	Bus	Ti	Pmax	Pmin	IntS	IntP	MinU	MinD	RU	RD	SU	SD	IntI	C^{AI}	C^{SU}	C^{G}	C^{C}
1	1	1	220	110	24	160	24	12	66	66	110	110	1	0	11904	30	0
2	2	2	100	30	6	50	6	12	50	50	50	50	1	0	1623	47	0
3	6	3	20	5	1	20	1	1	20	20	20	20	1	0	562	96	0
4	1	1	100	50	-12	0	24	12	30	30	50	50	0	20.48	5411	30	0
5	2	2	100	30	-12	0	6	12	50	50	50	50	0	16.11	1623	47	0
6	6	2	50	15	-12	0	6	12	25	25	25	25	0	8.05	811	47	0
7	5	3	50	10	-1	0	1	1	50	50	50	50	0	4.93	1407	96	0

Table A.17: 24-bus thermal generators

Gen	Bus	Ti	Pmax	Pmin	IntS	IntP	MinU	MinD	RU	RD	SU	SD	IntI	C^{AI}	C^{SU}	C^{G}	C^{C}
1	1	2	152	45.6	6	76	6	12	76	76	76	76	1	0	2466	47	0
2	2	3	152	38	1	152	1	1	152	152	152	152	1	0	4277	96	0
3	7	1	350	175	24	245	24	12	105	105	175	175	1	0	18938	30	0
4	13	1	591	295.5	24	413.7	24	12	177.3	177.3	295.5	295.5	1	0	31979	30	0
5	15	3	60	15	1	60	1	1	60	60	60	60	1	0	1688	96	0
6	15	2	155	46.5	6	77.5	6	12	77.5	77.5	77.5	77.5	1	0	2515	47	0
7	16	3	155	38.75	1	155	1	1	155	155	155	155	1	0	4361	96	0
8	18	1	400	200	24	280	24	12	120	120	200	200	1	0	21644	30	0
9	21	2	400	120	6	200	6	12	200	200	200	200	1	0	6492	47	0
10	22	1	300	150	24	210	24	12	90	90	150	150	1	0	16233	30	0
11	23	2	310	93	6	155	6	12	155	155	155	155	1	0	5031	47	0
12	23	1	350	175	24	245	24	12	105	105	175	175	1	0	18938	30	0
13	5	1	300	150	-12	0	24	12	90	90	150	150	0	61.44	16233	30	0
14	17	2	400	120	-12	0	6	12	200	200	200	200	0	64.42	6492	47	0
15	10	3	200	50	-1	0	1	1	200	200	200	200	0	19.73	5628	96	0
16	20	1	600	300	-12	0	24	12	180	180	300	300	0	122.88	32466	30	0
17	9	2	350	105	-12	0	6	12	175	175	175	175	0	56.37	5680	47	0
18	24	3	240	60	-1	0	1	1	240	240	240	240	0	23.67	6753	96	0

Table A.18: 14-bus thermal generators

Gen	Bus	Ti	Pmax	Pmin	IntS	IntP	MinU	MinD	RU	RD	SU	SD	IntI	C^{AI}	C^{SU}	C^{G}	C^{C}
1	1	1	330	165	24	231	24	12	99	99	165	165	1	0	17856	30	0
2	2	1	140	70	24	98	24	12	42	42	70	70	1	0	7575	30	0
3	3	2	100	30	6	50	6	12	50	50	50	50	1	0	1623	47	0
4	6	2	100	30	6	50	6	12	50	50	50	50	1	0	1623	47	0
5	8	3	100	25	1	100	1	1	100	100	100	100	1	0	2814	96	0
6	10	1	400	200	-24	0	24	12	120	120	200	200	0	81.92	21644	30	0
7	5	1	220	110	-24	0	24	12	66	66	110	110	0	45.06	11904	30	0
8	7	2	300	90	-6	0	6	12	150	150	150	150	0	48.32	4869	47	0
9	13	2	200	60	-6	0	6	12	100	100	100	100	0	32.21	3246	47	0
10	4	3	250	62.5	-1	0	1	1	250	250	250	250	0	24.66	7035	96	0
11	11	3	120	30	-1	0	1	1	120	120	120	120	0	11.84	3376	96	0

Table A.19: 18-bus thermal generators

Gen	Bus	Ti	Pmax	Pmin	IntS	IntP	MinU	MinD	RU	RD	SU	SD	IntI	C^{AI}	C^{SU}	C^{G}	C^{C}
1	4	1	30	15	24	21	24	12	9	9	15	15	1	0	1623	30	0
2	6	2	30	9	6	15	6	12	15	15	15	15	1	0	486	47	0
3	9	2	300	90	6	150	6	12	150	150	150	150	1	0	4869	47	0
4	12	1	30	15	24	21	24	12	9	9	15	15	1	0	1623	30	0
5	15	3	100	25	1	100	1	1	100	100	100	100	1	0	2814	96	0
6	16	3	30	7.5	1	30	1	1	30	30	30	30	1	0	844	96	0
7	17	2	100	30	6	50	6	12	50	50	50	50	1	0	1623	47	0
8	4	1	300	150	-24	0	24	12	90	90	150	150	0	61.44	16233	30	0
9	9	2	400	120	-6	0	6	12	200	200	200	200	0	64.42	6492	47	0
10	17	3	200	50	-1	0	1	1	200	200	200	200	0	19.72	5628	96	0
11	8	1	220	110	-24	0	24	12	66	66	110	110	0	45.06	11904	30	0
12	13	2	170	51	-6	0	6	12	85	85	85	85	0	27.38	2759	47	0
13	11	3	120	30	-1	0	1	1	120	120	120	120	0	11.84	3376	96	0

Appendix B

Detailed Results

Table B.1: Performance of scenario decomposition with single-period

Test Case	Instance	Type	Opt. obj.	Opt. gap	Res. gap	N. solutions
6 Bus	24	Basic	149.77	0	0.123	2
		Heuristic G.	149.77	0	0.123	2
		Stable G.	149.77	0	0.123	2
		Diving	149.77	0	0.123	2
	84	Basic	149.96	0	0.012	1
		Heuristic G.	149.96	0	0.012	1
		Stable G.	149.96	0	0.012	1
		Diving	149.96	0	0.012	1
	364	Basic	156.49	0	0.301	2
		Heuristic G.	156.49	0	0.301	2
		Stable G.	156.49	0	0.301	2
		Diving	156.49	0	0.301	2
14 Bus	24	Basic	615.05	0	0.117	3
		Heuristic G.	615.05	0	0.117	3
		Stable G.	615.05	0	0.117	3
		Diving	615.05	0	0.117	3
	84	Basic	606.86	0	0.042	1
		Heuristic G.	606.86	0	0.042	1
		Stable G.	606.86	0	0.042	1
		Diving	606.86	0	0.042	1
	364	Basic	644.73	2.20	0.206	4
		Heuristic G.	644.73	0	0.206	4
		Stable G.	644.73	0	0.206	4
		Diving	644.73	0	0.206	4
18 Bus	24	Basic	608.41	4.43	0.005	1
		Heuristic G.	608.41	0.45	0.005	1
		Stable G.	608.41	0	0.005	1
		Diving	608.41	0	0.005	1
		Local B.	608.41	0	0.005	1
		Multiple B.	608.41	0	0.005	1
	84	Basic	601.52	5.97	0.003	1
		Heuristic G.	601.52	1.10	0.003	1
		Stable G.	601.52	0	0.003	1
		Diving	601.52	0	0.003	1
		Local B.	601.52	0	0.003	1
		Multiple B.	601.52	0	0.003	1
	364	Basic	616.17	8.04	0.001	1
		Heuristic G.	616.17	3.06	0.001	1
		Stable G.	616.17	1.20	0.001	1
		Diving	616.17	1.247	0.001	1
		Local B.	616.17	1.242	0.001	1
		Multiple B.	616.17	1.226	0.001	1
24 Bus	24	Basic	1591.90	6.88	0.079	3
		Heuristic G.	1589.80	2.20	0.065	2
		Stable G.	1589.80	2.43	0.065	2
		Diving	1589.80	2.237	0.065	2
		Local B.	1589.80	2.95	0.065	2
		Multiple B.	1589.80	2.929	0.065	2
	84	Basic	1564.41	7.57	0.024	1
		Heuristic G.	1564.41	2.07	0.024	1
		Stable G.	1564.41	1.70	0.024	1
		Diving	1564.41	1.819	0.024	1
		Local B.	1564.41	1.867	0.024	1
		Multiple B.	1564.41	1.854	0.024	1
	364	Basic	1617.22	10.21	0.101	2
		Heuristic G.	1617.22	4.08	0.101	3
		Stable G.	1617.22	2.85	0.101	3
		Diving	1617.22	2.875	0.101	3
		Local B.	1617.22	2.866	0.101	3
		Multiple B.	1617.22	2.847	0.101	3

Table B.2: Performance of scenario decomposition with two-period

Test Case	Instance	Type	Opt. obj.	Opt. gap	Res. gap	N. solutions
6 Bus	24	Basic	1398.77	1.99	0.1347	4
		Stable G.	1398.77	0.00	0.1347	4
		Diving	1398.77	0.00	0.1347	4
		Local B.	1398.77	0.00	0.1347	4
		Multiple B.	1398.77	0.00	0.1347	4
	84	Basic	1380.31	5.81	0.0217	1
		Stable G.	1380.31	0.45	0.0217	1
		Diving	1380.31	1.48	0.0217	1
		Local B.	1380.31	0.00	0.0217	1
		Multiple B.	1380.31	0.00	0.0217	1
	364	Basic	1440.89	8.75	0.2592	3
		Stable G.	1440.89	2.99	0.2592	3
		Diving	1440.89	3.47	0.2592	3
		Local B.	1440.89	2.99	0.2592	3
		Multiple B.	1440.89	2.99	0.2592	3
14 Bus	24	Basic	5901.93	1.28	0.1051	3
		Stable G.	5901.93	0.00	0.1051	3
		Diving	5901.93	0.00	0.1051	3
		Local B.	5901.93	0.00	0.1051	3
		Multiple B.	5901.93	0.00	0.1051	3
	84	Basic	5836.35	2.06	0.0342	1
		Stable G.	5836.35	0.31	0.0342	1
		Diving	5836.35	0.49	0.0342	1
		Local B.	5836.35	0.00	0.0342	1
		Multiple B.	5836.35	0.16	0.0342	1
	364	Basic	6244.33	2.61	0.1676	4
		Stable G.	6244.33	0.52	0.1676	5
		Diving	6244.33	0.80	0.1676	4
		Local B.	6244.33	0.52	0.1676	5
		Multiple B.	6244.33	0.52	0.1676	5
18 Bus	24	Basic	5457.72	5.65	0.003	1
		Stable G.	5457.72	1.83	0.0033	1
		Diving	5457.72	1.88	0.0033	1
		Local B.	5457.72	1.95	0.0033	1
		Multiple B.	5457.72	1.94	0.0033	1
	84	Basic	5387.99	6.17	0.0021	1
		Stable G.	5387.99	1.75	0.0021	1
		Diving	5387.99	1.79	0.0021	1
		Local B.	5387.99	1.80	0.0021	1
		Multiple B.	5387.99	1.77	0.0021	1
	364	Basic	5517.13	7.95	0.0007	1
		Stable G.	5517.13	2.49	0.0007	1
		Diving	5517.13	2.50	0.0007	1
		Local B.	5517.13	2.49	0.0007	1
		Multiple B.	5517.13	2.49	0.0007	1
24 Bus	24	Basic	14706.20	7.76	0.0802	14
		Stable G.	14621.78	2.95	0.0748	8
		Diving	14627.42	2.98	0.0751	3
		Local B.	14620.66	3.00	0.074	25
		Multiple B.	14621.78	3.01	0.0748	4
	84	Basic	14276.62	7.50	0.0226	5
		Stable G.	14276.62	2.20	0.0226	2
		Diving	14276.62	2.21	0.0226	2
		Local B.	14276.62	2.21	0.0226	2
		Multiple B.	14276.62	2.20	0.0226	2
	364	Basic	14803.88	9.73	0.1315	1
		Stable G.	14803.88	3.70	0.1315	1
		Diving	14803.88	9.73	0.1315	1
		Local B.	14803.88	3.70	0.1315	1
		Multiple B.	14803.88	3.70	0.1315	1

References

[1] A. J. Conejo, L. Baringo, S. J. Kazempour, and A. S. Siddiqui, “Investment in electricity generation and transmission,” *Cham Zug, Switzerland: Springer International Publishing*, 2016.

[2] E. Ela, V. Diakov, E. Ibanez, and M. Heaney, “Impacts of variability and uncertainty in solar photovoltaic generation at multiple timescales - *National Renewable Energy Laboratory,*” 2013.

[3] J. Ma, V. Silva, R. Belhomme, D. S. Kirschen, and L. F. Ochoa, “Evaluating and planning flexibility in sustainable power systems,” in *2013 IEEE Power Energy Society General Meeting*, pp. 1–11, July 2013.

[4] J. Haas, F. Cebulla, K. Cao, W. Nowak, R. Palma-Behnke, C. Rahmann, and P. Mancarella, “Challenges and trends of energy storage expansion planning for flexibility provision in low-carbon power systems - a review,” *Renewable and Sustainable Energy Reviews*, vol. 80, pp. 603 – 619, 2017.

[5] N. E. Koltsaklis and A. S. Dagoumas, “State-of-the-art generation expansion planning: A review,” *Applied Energy*, vol. 230, pp. 563 – 589, 2018.

[6] B. Hua, R. Baldick, and J. Wang, “Representing operational flexibility in generation expansion planning through convex relaxation of unit commitment,” *IEEE Transactions on Power Systems*, vol. 33, pp. 2272–2281, March 2018.

[7] Z. Hu, F. Zhang, and B. Li, “Transmission expansion planning considering the deployment of energy storage systems,” in *2012 IEEE Power and Energy Society General Meeting*, pp. 1–6, IEEE, 2012.

[8] E. Talebizadeh, M. Rashidinejad, and A. Abdollahi, “Evaluation of plug-in electric vehicles impact on cost-based unit commitment,” *Journal of Power Sources*, vol. 248, pp. 545–552, 2014.

[9] Rocky Mountain Institute Boulder, Colorado, “Demand response: an introduction, overview of programs, technologies, and lessons learned,” 2006. Available at http://large.stanford.edu/courses/2014/ph240/lin2/docs/2440_doc_1.pdf.

[10] B. Palmintier and M. Webster, “Impact of unit commitment constraints on generation expansion planning with renewables,” in *2011 IEEE Power and Energy Society General Meeting*, pp. 1–7, July 2011.

[11] I. Sharan and R. Balasubramanian, “Generation expansion planning with high penetration of wind power,” *International Journal of Emerging Electric Power Systems*, vol. 17, no. 4, pp. 401–423, 2016.

[12] G. Morales-España, J. M. Latorre, and A. Ramos, “Tight and compact milp formulation for the thermal unit commitment problem,” *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4897–4908, 2013.

[13] Q. P. Zheng, J. Wang, and A. L. Liu, “Stochastic optimization for unit commitment—a review,” *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1913–1924, 2014.

[14] Q. Xu, S. Li, and B. F. Hobbs, “Generation and storage expansion co-optimization with consideration of unit commitment,” in *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, pp. 1–6, IEEE, 2018.

[15] N. Zhang, Z. Hu, C. Springer, Y. Li, and B. Shen, “A bi-level integrated generation-transmission planning model incorporating the impacts of demand response by operation simulation,” *Energy Conversion and Management*, vol. 123, pp. 84–94, 2016.

[16] D. Min, J.-h. Ryu, and D. G. Choi, “A long-term capacity expansion planning model for an electric power system integrating large-size renewable energy technologies,” *Computers & Operations Research*, vol. 96, pp. 244–255, 2018.

[17] B. F. Hobbs, “Optimization methods for electric utility resource planning,” *European Journal of Operational Research*, vol. 83, no. 1, pp. 1 – 20, 1995.

[18] E. Kahn, “Regulation by simulation: The role of production cost models in electricity planning and pricing,” *Operations Research*, vol. 43, no. 3, pp. 388–398, 1995.

[19] D. Phillips, “A mathematical model for determining generation plant mix,” in *Proceeding of the Third Power Systems Computation Conference*, 1969.

[20] F. Munoz, B. Hobbs, and J.-P. Watson, “New bounding and decomposition approaches for milp investment problems: Multi-area transmission and generation planning under policy constraints,” *European Journal of Operational Research*, vol. 248, no. 3, pp. 888 – 898, 2016.

[21] T. Zhang, R. Baldick, and T. Deetjen, “Optimized generation capacity expansion using a further improved screening curve method,” *Electric Power Systems Research*, vol. 124, pp. 47–54, 2015.

[22] Y. E. Güner, “The improved screening curve method regarding existing units,” *European Journal of Operational Research*, vol. 264, no. 1, pp. 310–326, 2018.

[23] F. Bessière, “The “investment’85” model of electricite de france,” *Management Science*, vol. 17, no. 4, pp. B–192, 1970.

[24] D. Anderson, “Models for determining least-cost investments in electricity supply,” *The Bell Journal of Economics and Management Science*, pp. 267–299, 1972.

[25] R. Turvey and D. Anderson, “Electricity economics: essays and case studies,” 1977.

[26] C. R. Scherer and L. Joe, “Electric power system planning with explicit stochastic reserves constraint,” *Management Science*, vol. 23, no. 9, pp. 978–985, 1977.

[27] S. Lumbreras and A. Ramos, “The new challenges to transmission expansion planning. survey of recent practice and literature review,” *Electric Power Systems Research*, vol. 134, pp. 19–29, 2016.

[28] A. J. Pereira and J. T. Saraiva, “A decision support system for generation expansion planning in competitive electricity markets,” *Electric power systems research*, vol. 80, no. 7, pp. 778–787, 2010.

[29] S. Dehghan, N. Amjady, and A. J. Conejo, “Reliability-constrained robust power system expansion planning,” *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2383–2392, 2016.

[30] R. Domínguez, A. J. Conejo, and M. Carrión, “Toward fully renewable electric energy systems,” *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 316–326, 2015.

[31] J. Hetzer, C. Y. David, and K. Bhattarai, “An economic dispatch model incorporating wind power,” *IEEE Transactions on energy conversion*, vol. 23, no. 2, pp. 603–611, 2008.

[32] P. Pisciella, M. T. Vespucci, M. Bertocchi, and S. Zigrino, “A time consistent risk averse three-stage stochastic mixed integer optimization model for power generation capacity expansion,” *Energy Economics*, vol. 53, pp. 203–211, 2016.

[33] J. Li, Z. Li, F. Liu, H. Ye, X. Zhang, S. Mei, and N. Chang, “Robust coordinated transmission and generation expansion planning considering ramping requirements and construction periods,” *IEEE Transactions on Power Systems*, vol. 33, pp. 268–280, Jan 2018.

[34] X. Zhang, L. Che, M. Shahidehpour, A. Alabdulwahab, and A. Abusorrah, “Electricity-natural gas operation planning with hourly demand response for deployment of flexible ramp,” *IEEE Transactions on Sustainable Energy*, vol. 7, no. 3, pp. 996–1004, 2016.

[35] A. Pina, C. A. Silva, and P. Ferrão, “High-resolution modeling framework for planning electricity systems with high penetration of renewables,” *Applied Energy*, vol. 112, pp. 215–223, 2013.

[36] D. G. Choi and V. M. Thomas, “An electricity generation planning model incorporating demand response,” *Energy Policy*, vol. 42, pp. 429–441, 2012.

[37] R. Hemmati, H. Saboori, and M. A. Jirdehi, “Multistage generation expansion planning incorporating large scale energy storage systems and environmental pollution,” *Renewable Energy*, vol. 97, pp. 636 – 645, 2016.

[38] A. van Stiphout, K. D. Vos, and G. Deconinck, “Operational flexibility provided by storage in generation expansion planning with high shares of renewables,” in *2015 12th International Conference on the European Energy Market (EEM)*, pp. 1–5, May 2015.

[39] N. Van Bracht, F. Grote, A. Fehler, and A. Moser, “Incorporating long-term uncertainties in generation expansion planning,” in *2016 13th International Conference on the European Energy Market (EEM)*, pp. 1–5, IEEE, 2016.

[40] M. R. Sheibani, G. R. Yousefi, M. A. Latify, and S. Hacopian Dolatabadi, “Energy storage system expansion planning in power systems: a review,” *IET Renewable Power Generation*, vol. 12, no. 11, pp. 1203–1221, 2018.

[41] N. E. Koltsaklis and M. C. Georgiadis, “A multi-period, multi-regional generation expansion planning model incorporating unit commitment constraints,” *Applied Energy*, vol. 158, pp. 310 – 331, 2015.

[42] A. Flores-Quiroz, R. Palma-Behnke, G. Zakeri, and R. Moreno, “A column generation approach for solving generation expansion planning problems with high renewable energy penetration,” *Electric Power Systems Research*, vol. 136, pp. 232 – 241, 2016.

[43] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power generation, operation, and control*. John Wiley & Sons, 2013.

[44] R. Hemmati, R.-A. Hooshmand, and A. Khodabakhshian, “State-of-the-art of transmission expansion planning: Comprehensive review,” *Renewable and Sustainable Energy Reviews*, vol. 23, pp. 312–319, 2013.

[45] L. Meeus, K. Purchala, D. Van Hertem, and R. Belmans, “Regulated cross-border transmission investment in europe,” *European Transactions on Electrical Power*, vol. 16, no. 6, pp. 591–601, 2006.

[46] R. Cruz-Rodriguez and G. Latorre-Bayona, “Hiper: interactive tool for mid-term transmission expansion planning in a deregulated environment,” *IEEE Power Engineering Review*, vol. 20, no. 11, pp. 61–62, 2000.

[47] R.-A. Hooshmand, R. Hemmati, and M. Parastegari, “Combination of ac transmission expansion planning and reactive power planning in the restructured power system,” *Energy Conversion and Management*, vol. 55, pp. 26–35, 2012.

[48] R.-C. Leou, “A multi-year transmission planning under a deregulated market,” *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 3, pp. 708–714, 2011.

[49] A. A. Foroud, A. A. Abdoos, R. Keypour, and M. Amirahmadi, “A multi-objective framework for dynamic transmission expansion planning in competitive electricity market,” *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 8, pp. 861–872, 2010.

[50] S. De La Torre, A. J. Conejo, and J. Contreras, “Transmission expansion planning in electricity markets,” *IEEE transactions on power systems*, vol. 23, no. 1, pp. 238–248, 2008.

[51] L. P. Garcés, A. J. Conejo, R. García-Bertrand, and R. Romero, “A bilevel approach to transmission expansion planning within a market environment,” *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1513–1522, 2009.

[52] M. Rider, A. Garcia, and R. Romero, “Power system transmission network expansion planning using ac model,” *IET Generation, Transmission & Distribution*, vol. 1, no. 5, pp. 731–742, 2007.

[53] R. A. Jabr, “Optimization of ac transmission system planning,” *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2779–2787, 2013.

[54] N. Alguacil, A. L. Motto, and A. J. Conejo, “Transmission expansion planning: A mixed-integer lp approach,” *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1070–1077, 2003.

[55] C. Coffrin and P. Van Hentenryck, “A linear-programming approximation of ac power flows,” *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, 2014.

[56] B. Stott, J. Jardim, and O. Alsaç, “Dc power flow revisited,” *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009.

[57] M. Soroush and J. D. Fuller, “Accuracies of optimal transmission switching heuristics based on dcopf and acopf,” *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 924–932, 2013.

[58] H. Zhang, G. T. Heydt, V. Vittal, and J. Quintero, “An improved network model for transmission expansion planning considering reactive power and network losses,” *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 3471–3479, 2013.

[59] J. Lavaei and S. H. Low, “Zero duality gap in optimal power flow problem,” *IEEE Transactions on Power systems*, vol. 27, no. 1, pp. 92–107, 2011.

[60] T. Akbari, A. Rahimi-Kian, and M. T. Bina, “Security-constrained transmission expansion planning: A stochastic multi-objective approach,” *International Journal of Electrical Power & Energy Systems*, vol. 43, no. 1, pp. 444–453, 2012.

[61] R. Mínguez and R. García-Bertrand, “Robust transmission network expansion planning in energy systems: Improving computational performance,” *European Journal of Operational Research*, vol. 248, no. 1, pp. 21–32, 2016.

[62] S. Dehghan and N. Amjady, "Robust transmission and energy storage expansion planning in wind farm-integrated power systems considering transmission switching," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 765–774, 2015.

[63] R. Hemmati, R. A. Hooshmand, and A. Khodabakhshian, "Comprehensive review of generation and transmission expansion planning," *IET Generation, Transmission Distribution*, vol. 7, pp. 955–964, Sept 2013.

[64] B. Alizadeh and S. Jadid, "Reliability constrained coordination of generation and transmission expansion planning in power systems using mixed integer programming," *IET Generation, Transmission & Distribution*, vol. 5, no. 9, pp. 948–960, 2011.

[65] Australian Energy Market Operator (AEMO), "Integrated system plan for the national electricity market," July 2018. Available at https://www.aemo.com.au/-/media/Files/Electricity/NEM/Planning_and_Forecasting/ISP/2018/Integrated-System-Plan-2018_final.pdf.

[66] Y.-H. Huang and J.-H. Wu, "A portfolio risk analysis on electricity supply planning," *Energy policy*, vol. 36, no. 2, pp. 627–641, 2008.

[67] F. J. De Sisternes, J. D. Jenkins, and A. Botterud, "The value of energy storage in decarbonizing the electricity sector," *Applied Energy*, vol. 175, pp. 368–379, 2016.

[68] B. S. Palmintier and M. D. Webster, "Impact of operational flexibility on electricity generation planning with renewable and carbon targets," *IEEE Transactions on Sustainable Energy*, vol. 7, pp. 672–684, April 2016.

[69] A. Moreira, D. Pozo, A. Street, and E. Sauma, "Reliable renewable generation and transmission expansion planning: co-optimizing system's resources for meeting renewable targets," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3246–3257, 2016.

[70] R. S. Go, F. D. Munoz, and J.-P. Watson, "Assessing the economic value of co-optimized grid-scale energy storage investments in supporting high renewable portfolio standards," *Applied Energy*, vol. 183, pp. 902–913, 2016.

[71] M. S. Sepasian, H. Seifi, A. A. Foroud, and A. Hatami, "A multiyear security constrained hybrid generation-transmission expansion planning algorithm including fuel supply costs," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1609–1618, 2009.

[72] J. Aghaei, N. Amjady, A. Baharvandi, and M.-A. Akbari, "Generation and transmission expansion planning: Milp–based probabilistic model," *IEEE Transactions on Power Systems*, vol. 29, no. 4, pp. 1592–1601, 2014.

[73] M. Qadrdan, H. Ameli, G. Strbac, and N. Jenkins, "Efficacy of options to address balancing challenges: Integrated gas and electricity perspectives," *Applied energy*, vol. 190, pp. 181–190, 2017.

[74] T. Li, M. Eremia, and M. Shahidehpour, "Interdependency of natural gas network and power system security," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1817–1824, 2008.

[75] Y. Zhu, Y. Li, and G. Huang, "An optimization decision support approach for risk analysis of carbon emission trading in electric power systems," *Environmental modelling & software*, vol. 67, pp. 43–56, 2015.

[76] O. Tietjen, M. Pahle, and S. Fuss, "Investment risks in power generation: A comparison of fossil fuel and renewable energy dominated markets," *Energy Economics*, vol. 58, pp. 174–185, 2016.

[77] E. Pean, M. Pirouti, and M. Qadrdan, "Role of the gb-france electricity interconnectors in integration of variable renewable generation," *Renewable energy*, vol. 99, pp. 307–314, 2016.

[78] R. Mena, M. Hennebel, Y.-F. Li, C. Ruiz, and E. Zio, "A risk-based simulation and multi-objective optimization framework for the integration of distributed renewable generation and storage," *Renewable and Sustainable Energy Reviews*, vol. 37, pp. 778–793, 2014.

[79] H. Sadeghi, M. Rashidinejad, and A. Abdollahi, "A comprehensive sequential review study through the generation expansion planning," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 1369–1394, 2017.

[80] J. Zhu and M. yuen Chow, "A review of emerging techniques on generation expansion planning," *IEEE Transactions on Power Systems*, vol. 12, pp. 1722–1728, Nov 1997.

[81] J. April, F. Glover, J. Kelly, and M. Laguna, "Simulation-based optimization: practical introduction to simulation optimization.," pp. 71–78, 01 2003.

[82] A. David and Z. Rongda, "An expert system with fuzzy sets for optimal planning (of power system expansion)," *IEEE Transactions on Power Systems*, vol. 6, no. 1, pp. 59–65, 1991.

[83] V. S. S. Vankayala and N. D. Rao, "Artificial neural networks and their applications to power systems—a bibliographical survey," *Electric Power Systems Research*, vol. 28, no. 1, pp. 67–79, 1993.

[84] I. M. De Mendonça, I. C. S. Junior, and A. L. Marcato, "Static planning of the expansion of electrical energy transmission systems using particle swarm optimization," *International Journal of Electrical Power & Energy Systems*, vol. 60, pp. 234–244, 2014.

[85] S. P. Torres and C. A. Castro, "Parallel particle swarm optimization applied to the static transmission expansion planning problem," in *2012 Sixth IEEE/PES Transmission and Distribution: Latin America Conference and Exposition (T&D-LA)*, pp. 1–6, IEEE, 2012.

[86] A. J. Conejo, E. Castillo, R. Minguez, and R. Garcia-Bertrand, *Decomposition techniques in mathematical programming: engineering and science applications.* Springer Science & Business Media, 2006.

[87] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[88] J. A. Bloom, "Solving an electricity generating capacity expansion planning problem by generalized benders' decomposition," *Operations Research*, vol. 31, no. 1, pp. 84–100, 1983.

[89] B. Gorenstin, N. Campodonico, J. Costa, and M. Pereira, "Power system expansion planning under uncertainty," *IEEE Transactions on Power Systems*, vol. 8, no. 1, pp. 129–136, 1993.

[90] L. Baringo and A. J. Conejo, "Wind power investment: A benders decomposition approach," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 433–441, 2011.

[91] F. D. Munoz, B. F. Hobbs, and J.-P. Watson, "New bounding and decomposition approaches for milp investment problems: Multi-area transmission and generation planning under policy constraints," *European Journal of Operational Research*, vol. 248, no. 3, pp. 888–898, 2016.

[92] M. Majidi-Qadikolai and R. Baldick, "A generalized decomposition framework for large-scale transmission expansion planning," *IEEE Transactions on Power Systems*, vol. 33, pp. 1635–1649, March 2018.

[93] S. Dehghan, N. Amjady, and A. Kazemi, "Two-stage robust generation expansion planning: a mixed integer linear programming model," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 584–597, 2014.

[94] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations research*, vol. 8, no. 1, pp. 101–111, 1960.

[95] A. P. Sanghvi and I. H. Shavel, "Investment planning for hydro-thermal power system expansion: Stochastic programming employing the dantzig-wolfe decomposition principle," *IEEE Transactions on Power Systems*, vol. 1, no. 2, pp. 115–121, 1986.

[96] K. J. Singh, A. B. Philpott, and R. K. Wood, "Dantzig-wolfe decomposition for solving multistage stochastic capacity-planning problems," *Operations Research*, vol. 57, no. 5, pp. 1271–1286, 2009.

[97] C.-P. Cheng, C.-W. Liu, and C.-C. Liu, "Unit commitment by lagrangian relaxation and genetic algorithms," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 707–714, 2000.

[98] F. Zhuang and F. D. Galiana, "Towards a more rigorous and practical unit commitment by lagrangian relaxation," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 763–773, 1988.

[99] W. Ongsakul and N. Petcharaks, “Unit commitment by enhanced adaptive lagrangian relaxation,” *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 620–628, 2004.

[100] H. Zeynal, L. X. Hui, Y. Jiazhen, M. Eidiani, and B. Azzopardi, “Improving lagrangian relaxation unit commitment with cuckoo search algorithm,” in *2014 IEEE International Conference on Power and Energy (PECon)*, pp. 77–82, IEEE, 2014.

[101] J. H. Roh, M. Shahidehpour, and Y. Fu, “Market-based coordination of transmission and generation capacity planning,” *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1406–1419, 2007.

[102] H. Chen, X. Wang, and X. Zhao, “Generation planning using lagrangian relaxation and probabilistic production simulation,” *International journal of electrical power & energy systems*, vol. 26, no. 8, pp. 597–605, 2004.

[103] C. Genesi, P. Marannino, M. Montagna, S. Rossi, I. Siviero, L. Desiata, and G. Gentile, “Risk management in long term generation planning,” in *2009 6th International Conference on the European Energy Market*, pp. 1–6, IEEE, 2009.

[104] S. Ahmed, “A scenario decomposition algorithm for 0-1 stochastic programs,” *Operations Research Letters*, vol. 41, no. 6, pp. 565 – 569, 2013.

[105] K. Ryan, D. Rajan, and S. Ahmed, “Scenario decomposition for 0-1 stochastic programs: Improvements and asynchronous implementation,” in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 722–729, May 2016.

[106] B. Basciftci, S. Ahmed, N. Z. Gebraeel, and M. Yildirim, “Stochastic optimization of maintenance and operations schedules under unexpected failures,” *IEEE Transactions on Power Systems*, vol. 33, pp. 6755–6765, Nov 2018.

[107] S. Lei, J. Wang, C. Chen, and Y. Hou, “Mobile emergency generator pre-positioning and real-time allocation for resilient response to natural disasters,” *IEEE Transactions on Smart Grid*, vol. 9, pp. 2030–2041, May 2018.

[108] T. Levin and A. Botterud, “Electricity market design for generator revenue sufficiency with increased variable generation,” *Energy Policy*, vol. 87, pp. 392–406, 2015.

[109] P. J. Ramírez, D. Papadaskalopoulos, and G. Strbac, “Co-optimization of generation expansion planning and electric vehicles flexibility,” *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1609–1619, 2015.

[110] Y. Li, C. Davis, Z. Lukszo, and M. Weijnen, “Electric vehicle charging in china’s power system: Energy, economic and environmental trade-offs and policy implications,” *Applied energy*, vol. 173, pp. 535–554, 2016.

[111] S. Pereira, P. Ferreira, and A. I. F. Vaz, "Generation expansion planning with high share of renewables of variable output," *Applied energy*, vol. 190, pp. 1275–1288, 2017.

[112] K. Bruninx, D. Madzharov, E. Delarue, and W. D'haeseleer, "Impact of the german nuclear phase-out on europe's electricity generation—a comprehensive study," *Energy Policy*, vol. 60, pp. 251–261, 2013.

[113] S. Jin, A. Botterud, and S. M. Ryan, "Temporal versus stochastic granularity in thermal generation capacity planning with wind power," *IEEE transactions on power systems*, vol. 29, no. 5, pp. 2033–2041, 2014.

[114] A. Shortt and M. O'Malley, "Quantifying the long-term impact of electric vehicles on the generation portfolio," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 71–83, 2013.

[115] B. Palmintier, "Flexibility in generation planning: Identifying key operating constraints," in *2014 Power Systems Computation Conference*, pp. 1–7, IEEE, 2014.

[116] C. I. Nweke, F. Leanez, G. R. Drayton, and M. Kolhe, "Benefits of chronological optimization in capacity planning for electricity markets," in *2012 IEEE International Conference on Power System Technology (POWERCON)*, pp. 1–6, Oct 2012.

[117] M. Welsch, P. Deane, M. Howells, B. Ó. Gallachóir, F. Rogan, M. Bazilian, and H.-H. Rogner, "Incorporating flexibility requirements into long-term energy system models-a case study on high levels of renewable electricity penetration in ireland," *Applied energy*, vol. 135, pp. 600–615, 2014.

[118] J. Hargreaves, E. K. Hart, R. Jones, and A. Olson, "Reflex: an adapted production simulation methodology for flexible capacity planning," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1306–1315, 2014.

[119] M. Wierzbowski, W. Lyzwa, and I. Musial, "Milp model for long-term energy mix planning with consideration of power system reserves," *Applied Energy*, vol. 169, pp. 93–111, 2016.

[120] A. Belderbos and E. Delarue, "Accounting for flexibility in power system planning with renewables," *International Journal of Electrical Power & Energy Systems*, vol. 71, pp. 33–41, 2015.

[121] S. Kamalinia, M. Shahidehpour, and A. Khodaei, "Security-constrained expansion planning of fast-response units for wind integration," *Electric Power Systems Research*, vol. 81, no. 1, pp. 107–116, 2011.

[122] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The benders decomposition algorithm: A literature review," *European Journal of Operational Research*, vol. 259, no. 3, pp. 801–817, 2017.

[123] A. Schwele, J. Kazempour, and P. Pinson, "Do unit commitment constraints affect generation expansion planning? a scalable stochastic model," *Energy Systems*, pp. 1–36, 2019.

[124] A. Flores-Quiroz, R. Palma-Behnke, G. Zakeri, and R. Moreno, "A column generation approach for solving generation expansion planning problems with high renewable energy penetration," *Electric Power Systems Research*, vol. 136, pp. 232–241, 2016.

[125] K. Poncelet, E. Delarue, D. Six, J. Duerinck, and W. D'haeseleer, "Impact of the level of temporal and operational detail in energy-system planning models," *Applied Energy*, vol. 162, pp. 631–643, 2016.

[126] A. Nogales, S. Wogrin, and E. Centeno, "Impact of technical operational details on generation expansion in oligopolistic power markets," *IET Generation, Transmission & Distribution*, vol. 10, no. 9, pp. 2118–2126, 2016.

[127] M. Carrión and J. M. Arroyo, "A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem," *IEEE Transactions on power systems*, vol. 21, no. 3, pp. 1371–1378, 2006.

[128] J. Ostrowski, M. F. Anjos, and A. Vannelli, "Tight mixed integer linear programming formulations for the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 39–46, 2011.

[129] M. F. Anjos, A. J. Conejo, *et al.*, "Unit commitment in electric energy systems," *Foundations and Trends® in Electric Energy Systems*, vol. 1, no. 4, pp. 220–310, 2017.

[130] C. Barrows, M.Hummon, W.Jones, and E. Hale, "Time domain partitioning of electricity production cost simulations," *National Renewable Energy Laboratory (NREL)*, 2014.

[131] Gurobi, "Gurobi optimizer reference manual," 2016.

[132] J. Wang, J. Wang, C. Liu, and J. P. Ruiz, "Stochastic unit commitment with subhourly dispatch constraints," *Applied energy*, vol. 105, pp. 418–422, 2013.

[133] J. Deane, G. Drayton, and B. Ó. Gallachóir, "The impact of sub-hourly modelling in power systems with significant levels of renewable generation," *Applied Energy*, vol. 113, pp. 152–158, 2014.

[134] E. Ela and M. O'Malley, "Studying the variability and uncertainty impacts of variable generation at multiple timescales," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1324–1333, 2012.

[135] H. Wu, I. Krad, A. Florita, B.-M. Hodge, E. Ibanez, J. Zhang, and E. Ela, "Stochastic multi-timescale power system operations with variable wind generation," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3325–3337, 2016.

[136] E. A. Bakirtzis, P. N. Biskas, D. P. Labridis, and A. G. Bakirtzis, “Multiple time resolution unit commitment for short-term operations scheduling under high renewable penetration,” *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 149–159, 2013.

[137] R. Domínguez, A. J. Conejo, and M. Carrion, “Toward fully renewable electric energy systems,” *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 316–326, 2014.

[138] J. H. Merrick, “On representation of temporal variability in electricity capacity planning models,” *Energy Economics*, vol. 59, pp. 261–274, 2016.

[139] R. Kannan and H. Turton, “A long-term electricity dispatch model with the times framework,” *Environmental Modeling & Assessment*, vol. 18, no. 3, pp. 325–343, 2013.

[140] K. Poncelet, E. Delarue, J. Duerinck, D. Six, and W. D’haeseleer, “The importance of integrating the variability of renewables in long-term energy planning models,” *Rome, Italy*, 2014.

[141] J. Abonyi, B. Feil, S. Nemeth, and P. Arva, “Modified gath–geva clustering for fuzzy segmentation of multivariate time-series,” *Fuzzy Sets and Systems*, vol. 149, no. 1, pp. 39–56, 2005.

[142] D. vom Stein, N. van Bracht, A. Maaz, and A. Moser, “Development of adaptive time patterns for multi-dimensional power system simulations,” in *2017 14th International Conference on the European Energy Market (EEM)*, pp. 1–5, IEEE, 2017.

[143] E. Keogh, S. Chu, D. Hart, and M. Pazzani, “Segmenting time series: A survey and novel approach,” in *Data mining in time series databases*, pp. 1–21, World Scientific, 2004.

[144] L. A. Leiva and E. Vidal, “Warped k-means: An algorithm to cluster sequentially-distributed data,” *Information Sciences*, vol. 237, pp. 196–210, 2013.

[145] M. Conforti, G. Cornuéjols, G. Zambelli, *et al.*, *Integer programming*, vol. 271. Springer, 2014.

[146] D. Gade, G. Hackebeil, S. M. Ryan, J.-P. Watson, R. J.-B. Wets, and D. L. Woodruff, “Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs,” *Mathematical Programming*, vol. 157, pp. 47–67, May 2016.

[147] S. Ahmed, J. Luedtke, Y. Song, and W. Xie, “Nonanticipative duality, relaxations, and formulations for chance-constrained stochastic programs,” *Mathematical Programming*, vol. 162, pp. 51–81, Mar 2017.

[148] D. Hemmi, G. Tack, and M. Wallace, "Scenario-based learning for stochastic combinatorial optimisation," in *Integration of AI and OR Techniques in Constraint Programming* (D. Salvagnin and M. Lombardi, eds.), (Cham), pp. 277–292, Springer International Publishing, 2017.

[149] K. Ryan, S. Ahmed, S. S. Dey, D. Rajan, A. Musselman, and J.-P. Watson, "Optimization-driven scenario grouping," *INFORMS Journal on Computing*, 2020.

[150] T. G. Crainic, M. Hewitt, and W. Rei, "Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design," *Computers and Operations Research*, vol. 43, pp. 90 – 99, 2014.

[151] Y. Deng, S. Ahmed, J. Lee, and S. Shen, "Scenario grouping and decomposition algorithms for chance-constrained programs," 2017.

[152] R. W. Irving, "An efficient algorithm for the "stable roommates" problem," *Journal of Algorithms*, vol. 6, no. 4, pp. 577 – 595, 1985.

[153] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[154] D. J. Abraham, P. Biró, and D. F. Manlove, ""almost stable" matchings in the roommates problem," in *International Workshop on Approximation and Online Algorithms*, pp. 1–14, Springer, 2005.

[155] R. W. Irving and D. F. Manlove, "The stable roommates problem with ties," *Journal of Algorithms*, vol. 43, no. 1, pp. 85–105, 2002.

[156] R. W. Irving and S. Scott, "The stable fixtures problem—a many-to-many extension of stable roommates," *Discrete Applied Mathematics*, vol. 155, no. 16, pp. 2118–2129, 2007.

[157] A. H. Land and A. G. Doig, *An Automatic Method for Solving Discrete Programming Problems*, pp. 105–132. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[158] M. Fischetti and A. Lodi, "Local branching," *Mathematical programming*, vol. 98, no. 1-3, pp. 23–47, 2003.

[159] S. Wijekoon, A. Liebman, and S. Dunstall, "Test Cases for Capacity Expansion Planning with Unit Commitment," 5 2019. Available at https://monash.figshare.com/articles/Test_Cases_for_Capacity_Expansion_Planning_with_Unit _Commitment/8135573.

[160] S. Wijekoon, A. Liebman, and S. Dunstall, "Data Set for Capacity Expansion Planning with Unit Commitment," 5 2019. Available at https://monash.figshare.com/articles/Data_Set_for_Capacity_Expansion_Planning_with_Unit_Commitment/8131529.

[161] R. Zimmerman and C. MurilloSanchez, *MATPOWER User's Manual version 7.0b1*. Available at http://www.pserc.cornell.edu/matpower/manual.pdf.

[162] S. Abdurrahman and S. M. Kristensen, "Technology data for the indonesian power sector." Available at https://www.ea-energianalyse.dk/wp-content/uploads/2020/02/1724_technology_data_indonesian_power_ector_dec2017-2.pdf.

[163] R. D. Christie, "Power systems test case archive." Available at http://labs.ece.uw.edu /pstca /pf14/pg _tca14bus.htm.

[164] Y. Fu, M. Shahidehpour, and Z. Li, "Long-term security-constrained unit commitment: hybrid dantzig-wolfe decomposition and subgradient approach," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 2093–2106, 2005.

[165] C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty, W. Li, R. Mukerji, D. Patton, N. Rau, D. Reppen, A. Schneider, M. Shahidehpour, and C. Singh, "The ieee reliability test system-1996. a report prepared by the reliability test system task force of the application of probability methods subcommittee," *IEEE Transactions on Power Systems*, vol. 14, pp. 1010–1020, Aug 1999.

[166] C. Ordoudis, P. Pinson, G. Morales, M. Juan, and M. Zugno, "An updated version of the ieee rts 24-bus system for electricity market and power system operation studies." Available at https://backend.orbit.dtu.dk/ws/portalfiles/portal/120568114/An_Updated_Version_of_the_IEEE_RTS_24Bus_System_for_Electricty_Market_an....pdf.