공학박사학위논문

Optimization Models and Decomposition Approaches for the Power System Operation under Uncertainty

불확실성 하에서의 전력시스템 운영에 대한 최적화 모형과 분해기법

2023 년 8 월

서울대학교 대학원

산업공학과

이 종 헌

Optimization Models and Decomposition Approaches for the Power System Operation under Uncertainty

불확실성 하에서의 전력시스템 운영에 대한 최적화 모형과 분해기법

지도교수 이 경 식

이 논문을 공학박사 학위논문으로 제출함

2023 년 5 월

서울대학교 대학원

산업공학과

이 종 헌

이종헌의 공학박사 학위논문을 인준함

2023 년 6 월

위 원 장	홍 성 필	(인)
부위원장	이 경 식	(인)
위 원	홍 유 석	(인)
위 원	박 경 철	(인)
위 원	조 남 욱	(인)

Abstract

Optimization Models and Decomposition Approaches for the Power System Operation under Uncertainty

Jongheon Lee
Department of Industrial Engineering
The Graduate School
Seoul National University

Power systems consist of generation, transmission, and distribution systems indicating the electricity network from its generation to the loads. Among various optimization problems in the system, the unit commitment (UC) problem is a fundamental problem that aims to minimize operation costs while meeting electricity demand by coordinating generation resources. To ensure effective and reliable operation, various uncertain factors such as renewable generation, load demand, and contingencies should appropriately be considered. However, since the computational burden increases as uncertain factors are incorporated into the optimization models, efficient models and solution approaches are needed. Thus, this dissertation aims to propose optimization models and decomposition methods to solve optimization problems in power system operation under uncertainty.

First, we investigate a novel modeling approach in two-stage stochastic programming, which is a widely used framework in power system operations under uncer-

tainty. We propose a partition-based risk-averse two-stage stochastic program that mitigates the drawbacks of the traditional two-stage stochastic programs with finite support. In the model, a set of scenarios is partitioned into several groups a second-stage cost is represented as the expectation of conditional value-at-risks of costs for each scenario group. We propose decomposition methods based on column-and-constraint generation to solve the model exactly for a given partition. In addition, a scenario partitioning method to enable the risk level of the model to be close to a given target is devised, and partitioning schemes for combining it with the proposed column-and-constraint generation algorithm are proposed. Numerical experiments were performed that demonstrated the effectiveness of the proposed partitioning schemes and the efficiency of the proposed solution approach.

Next, we address single-unit commitment (1UC) problems, which arise when an individual power producer bids a generator's schedule to the deregulated electricity market. Especially, we devise efficient dynamic programming algorithms to solve 1UC problems under stochastic electricity prices, by extending the results in the deterministic counterpart. Furthermore, leveraging the efficient algorithms on 1UC problems, we present two unit decomposition frameworks to solve the general UC problem under stochastic net load, which include a novel decomposition that has not been proposed. We propose a total of four solution approaches based on Lagrangian relaxation or column generation and analyze the dual bounds obtained from the methods. Through the numerical experiments, we demonstrate the efficiency of the proposed algorithms on 1UC problems. In addition, we analyze various unit decomposition methods for the stochastic UC problems and emphasize the scalability of the proposed novel column generation method with regard to the number of scenarios.

Finally, we study optimization models for microgrid operation under stochastic islanding and net load, where a microgrid is a localized power system with various distributed energy resources having the distinguishing feature that can be operated in an islanded or connected mode. Although multistage stochastic optimization models can address the dynamics and probabilistic nature of uncertainty, they suffer from the curse of dimensionality in that the number of scenarios grows exponentially with regard to the number of time periods. To overcome this drawback, we propose scalable optimization models to operate a microgrid under both uncertain factors. To deal with uncertain islanding events, a replanning procedure with models that consider at most one upcoming islanding event is proposed. To incorporate stochastic net load, the range of generation is determined in addition to commitment decisions to reduce the size of the model. Numerical experiments demonstrate that practical-sized instances can be solved using the proposed models, whereas they cannot be solved using the standard multistage model. The results also demonstrate the effectiveness of the solutions from the proposed models in an environment that both stochastic factors sequentially realize.

Keywords: Power system operation, Optimization under uncertainty, Unit commitment, Decomposition method, Microgrid, Stochastic optimization

Student Number: 2019-36357

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Backgrounds

1.1.1 Optimization problems in power systems

Power systems consist of generation, transmission, and distribution systems indicating from electricity production to the customer. The power system is a critical infrastructure in modern society that provides essential services to society, including providing power to homes, businesses, and industries. A reliable and efficient power system is essential for economic and environmental impacts since its importance is growing due to the adoption of renewable energy sources, urbanization, and the electrification of transportation and heating systems. As illustrated in Figure 1.1, the power system can be categorized into three main components: power generators which convert various sources of energy, such as coal, gas, nuclear, wind, or solar, into electrical energy, transmission systems responsible for transmitting the electricity from the generation stations to the load centers through a network of high-voltage transmission lines, and distribution systems that deliver electricity to nearby industrial and residential areas. There are numerous decision-making problems in power systems, which can be classified with regard to the time scale that they consider as illustrated in Figure 1.2. The *operation phase* refers to the problems that can occur

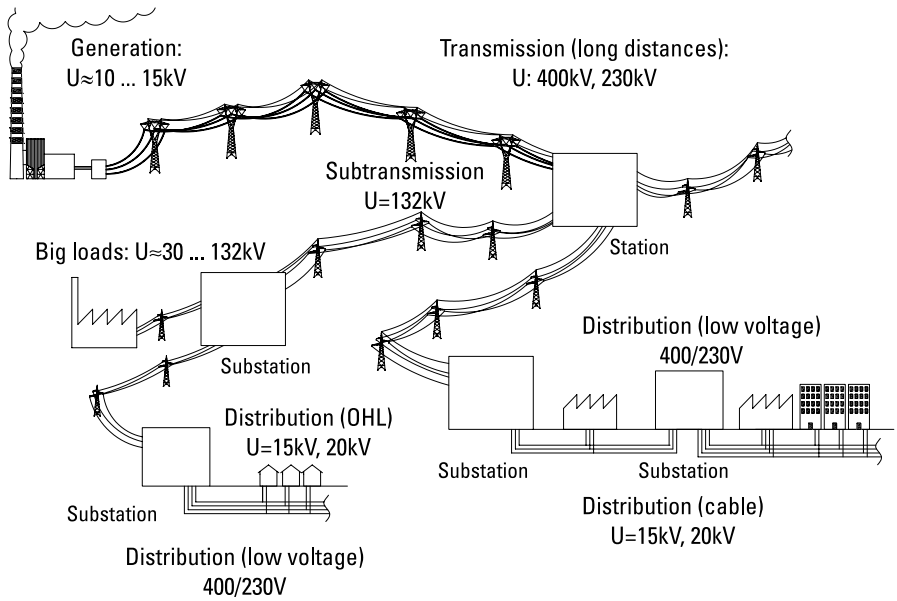


Figure 1.1: Illustration of power system structure (adapted from Delfino et al., 2018)

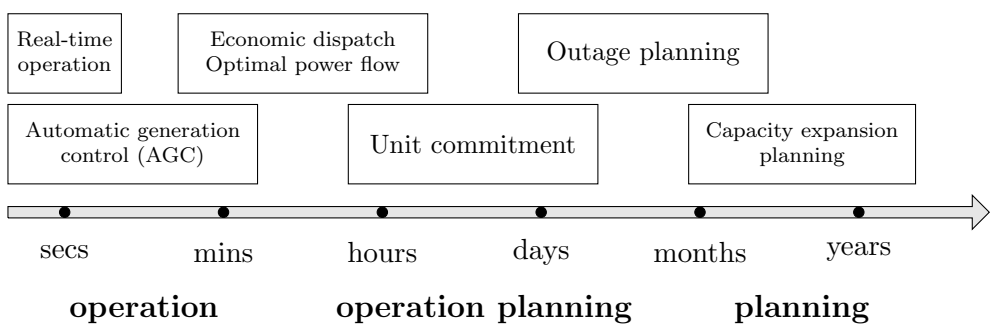


Figure 1.2: Optimization problems in power system operation by time scale (e.g., Akrami et al., 2019)

and be resolved in real-time or on the order of minutes, to maintain the system's stability and reliability. It includes balancing the demand and supply of electricity and responding to unexpected events and disturbances. It is crucial since electricity is hard to store a large amount of portion of the production in contrast to the other production systems. For example, the economic dispatch (ED) and optimal power flow (OPF) problems both determine the generation amount that minimizes operation costs by meeting electricity demand. The OPF problem extends the ED problem by considering more complex operational constraints for reliable operation considering the transmission line, such as voltage, reactive power, and line capacity limits (Cain et al., 2012).

On the other hand, the *planning phase* is concerned with long-term problems for the next few months to years, including investment decisions. For example, the capacity expansion problem is determining the optimal mix and size of power generation, transmission, and storage facilities required to meet future electricity demand while minimizing the cost of investment and generation. It is important to ensure a reliable and cost-effective power system for the future, where future electricity demand is estimated based on factors such as population growth, and economic development. Furthermore, the nuclear outage planning problem is related to the maintenance activities of nuclear power plants. It is important because nuclear power plants require periodic outages for refueling, maintenance, and repairs, and these shutdowns can last for several weeks or months (Griset et al., 2022). Since nuclear power plants often provide a large portion of the power system's baseload generation capacity, the outage schedule must be coordinated with other system operators to ensure the safe and reliable operation of the power system during the outage.

The *operation planning phase* is in the middle of the above-mentioned two phases in terms of the time scale, which has usually a planning horizon from a day to a week and decides hourly schedules by coordinating inner generation resources. The decisions in operation plans are essential as a baseline to make the real-time operation reliable and efficient. It is because the decisions that can be amended in a short time are restricted in practice since generators cannot instantly turn on or off. Therefore, the commitment decisions which refer to the on/off status of generators are important and must be planned in advance so that enough generation is always available to handle system demand with an adequate reserve margin in the event that generators or transmission lines go out or load demand increases (Zhu, 2015).

1.1.2 Unit commitment problem in power systems

The unit commitment (UC) problem is a fundamental optimization problem in the operation planning phase of the power system. The term unit represents a generator (generating unit) and the commitment indicates turning on a generator. It aims to minimize operational costs while satisfying system-wide requirements such as balancing supply and demand, and operational requirements of the generators. Typically, hourly decisions are made with a planning horizon from a day to a week. In a deregulated power system, the UC problem is usually solved by a coordinator such as an independent system operator (ISO), or a regional transmission organization (RTO), that is responsible for managing the power system in the region. The ISO typically receives bids from various generation companies (GENCOs) that own generation assets about their available capacity, operating costs, and other generating information relevant to the UC problem.

In general, the commitment and dispatch decisions of generators are determined through the UC problem. Commitment decisions are crucial in the UC problem since most generators have physical characteristics so changing their status in a short time is rarely possible. Dispatch decisions indicate the generation amount when the generator is on. Among the various types of generators, thermal generators which burn some sort of fuel to produce electricity are widely used. They are subject to numerous operational requirements. For example, minimum up (or down) requirements indicate the time periods that a generator must maintain its status when it starts (or stops) generating. In addition, ramping requirements limit the increase and decrease of the generation amount of a generator from a time period to an adjacent time period.

Considering those characteristics, solving UC problems is challenging. The basic UC problem, which focuses on meeting the demand by the generation only decided by the generators' on/off status, is shown to be NP-hard in Tseng (1996). It also has been shown to be strongly NP-hard (Bendotti et al., 2019) by additionally considering minimum up/down time requirements. To solve the model, classical and earlier approaches include dynamic programming (Lowery, 1966) and Lagrangian relaxation (Muckstadt and Koenig, 1977) approaches. Then, mixed-integer programming (MIP) approaches are widely used nowadays. It has been reported that the transition to MIP approaches has saved at least five billion USD annually in the United States (e.g., O'Neill, 2017; Knueven et al., 2020).

There are variants of the classical UC problem that considers specific power system environment. First, the single-unit commitment (1UC) problem arises when an individual power producer (IPP), who is responsible for managing the operation of

its individual generator, submits a bid to the deregulated electricity market (Pan and Guan, 2016). Instead of minimizing operating costs as in UC while demand and supply are met, the net profit, which is a subtraction of cost from the profit that it receives from the generation, is maximized in 1UC. In addition, compared to the UC problem which is strongly NP-hard, the 1UC problem is polynomially solvable although it is usually formulated as an MIP (Fan et al., 2002; Frangioni and Gentile, 2006b). Next, operation planning in a microgrid environment is of growing importance. A microgrid is a localized electric power system with a low voltage phase consisting of various distributed energy resources that can operate with connection to the centralized power system (main grid) or as an independent system while being disconnected from the main grid (Katiraei et al., 2008). Microgrids have been considered a key element of the future energy transition since they can increase the efficiency of distributed energy systems and facilitate the penetration of renewable energy sources (Moretti et al., 2020). In microgrids, locally distributed microgrid operators (MGOs) are responsible for managing their inner various distributed energy resources in the jurisdiction (Lee et al., 2021). There are specific characteristics that such as islanding events, which occur when a microgrid is disconnected from the main grid, need to be considered. Thus, to efficiently and reliably operate microgrids, those characteristics should be appropriately considered.

1.1.3 Optimization under uncertainty

To operate a power system in a reliable and efficient manner, various uncertain factors regarding the system should appropriately be considered. The uncertainty usually stems from the unpredictability of some components regarding the system,

such as electricity supply, demand, price, and so on. For example, the penetration of renewable electricity sources has brought significant challenges to power system operators since it is hard to predict precisely in advance because of their intermittent nature (Zhao and Guan, 2016). The next one is regarding system failure, especially unexpected outages of components, which typically are generators or transmission lines. A classical method in power system operation to deal with such uncertain factors is to provide spinning and operating reserves, which refer to a margin for available generation capacity that can be utilized at the realization of uncertainty. However, since it is an implicit and conservative approach that could lead to inefficient operation costs, various optimization models that explicitly consider uncertain factors have then been developed based on the optimization frameworks (Wu et al., 2007). We classify various optimization models under uncertainty in terms of the decision-making process and optimization frameworks.

First, the optimization models can be classified by the decision-making process. In the models, decision stages correspond to the group of decisions that are determined based on the same information on uncertainty. For example, a two-stage optimization model assumes the decision process with two stages, where the first-stage decisions are made before the realization of uncertainty, and the remaining second-stage decisions are determined based on the realization of uncertainty which occurs after the first-stage decisions. The multistage optimization models are the natural extension of the two-stage optimization models, which are adequate for sequential decision-making under uncertainty. Similarly to the two-stage models, each stage in the multistage model refers to a point where the decision is made based on the information on uncertainty at that point. Multistage models are more flexible

and can capture more complex decision-making processes, where the advantage is emphasized in Huang and Ahmed (2009), but they also tend to be more computationally demanding than two-stage models.

Next, we introduce representative optimization frameworks for modeling uncertain factors. Stochastic programming assumes that the uncertain parameter follows a certain probability distribution. Among the variations, the basic form is to minimize the expected costs for a given distribution. In practice, to avoid calculating expectations which is naturally intractable, a sample average approximation is widely used (Kleywegt et al., 2002). It is to replace the expectation with the sample average value based on sampling scenarios from the distribution. Then, the optimization model becomes a large-scale deterministic optimization model. On the other hand, robust optimization minimizes the worst-case performance over a set of possible values for uncertain parameters. It is a non-parametric method that seeks a solution that performs well under all possible values of the uncertain parameters within a specified set, which is called the *uncertainty set*. The uncertainty set must be carefully chosen to avoid the over-conservativeness of obtained solutions since extreme cases rarely occur. Various types of uncertainty sets have been considered such as box, ellipsoid, and Γ-robustness (Bertsimas and Sim, 2004). Finally, we note the hybrid models which can control the risk level between the two frameworks. Distributionally robust optimization (DRO) is a representative framework that minimizes the worst-case performance of expectation over a set of possible probability distributions for uncertain parameters, where the set is denoted by an *ambiguity set*. Compared to stochastic programming that minimizes expected costs for a certain distribution, it is less dependent on accurate estimation on probability distributions

and thus can hedge the risk of estimation error on the distribution. In addition, it can mitigate the over-conservativeness in robust optimization by utilizing distribution information that can be incorporated into the ambiguity sets. Various types of ambiguity sets are studied, including moment information-based (Delage and Ye, 2010) and statistical distance-based (e.g., Mohajerin Esfahani and Kuhn, 2018) sets. We lastly note that DRO can be related to a risk-averse stochastic program with a correspondence between a coherent risk measure and an ambiguity set (e.g., Artzner et al., 1999; Rahimian and Mehrotra, 2019).

1.2 Literature review

1.2.1 Studies on unit commitment problems

In UC problems, polyhedral studies have been widely done by focusing on the single-generator system. Lee et al. (2004) proposed valid inequalities for minimum up/down polytope with on/off decisions, further showing that they are sufficient to describe the convex hull of the polytope. Later, Rajan and Takriti (2005) extended the research to provide valid inequalities and convex hull descriptions when start-up decisions also exist. Morales-España et al. (2015) provided convex hull descriptions when the power generation limit and minimum up/down constraints exist. From then, various polyhedral studies are extended to incorporate ramping restrictions in single-generator system (Ostrowski et al., 2011; Damcı-Kurt et al., 2016). For valid inequalities for a multiple-generator system, Bendotti et al. (2018) proposed up-set inequalities, which are similar to the extended cover inequalities in the binary knapsack problem. Frangioni and Gentile (2006a) proposed a perspective cut to efficiently solve convex binary MIPs, and applied the method to the unit commitment problem

when the generation cost is a convex quadratic function. In addition, there have been algorithms for solving the 1UC problem directly, which can be solved in polynomial time. As reported in Pan and Guan (2016), there are two types of 1UC problems based on the bidding strategies of IPPs: self-commitment and self-scheduling. In the self-commitment problem, only commitment decisions for the planning horizon are decided, while the generation amount is also determined in the self-scheduling problem. Most of the studies focused on the self-scheduling problem. Fan et al. (2002) provided a dynamic programming-based polynomial-time algorithm to solve the self-scheduling problem with piece-wise linear variable cost. Then, Frangioni and Gentile (2006b) proposed a $\mathcal{O}(T^3)$-time algorithm with a convex quadratic variable cost function, where T is the number of periods in the planning horizon. Later, the algorithm is refined in the subsequent literature (Frangioni and Gentile, 2015; Guan et al., 2018; Wuijts et al., 2021). Further, based on the algorithm, several extended formulations that can describe the convex hull of the feasible solution set in higher dimension are proposed (Frangioni and Gentile, 2015; Guan et al., 2018; Knueven et al., 2018). Knueven et al. (2020) present a comprehensive comparison of various MIP formulations and proposed a new formulation regarding the UC problem.

To solve the UC problem, the above-mentioned valid inequalities are generated in advance or used in a branch-and-cut framework. In addition, decomposition approaches are also widely used. A well-known solution approach called *unit decomposition* (e.g., Van Ackooij et al., 2018) directly uses solutions of the 1UC problem. It is a method that relaxes system-wise requirements such as demand balance constraints and decomposes the remaining problem by each generator. Frangioni et al. (2008) applied the Lagrangian relaxation approach for a hydro-thermal system where cor-

responding self-scheduling subproblems are solved with their algorithm (Frangioni and Gentile, 2006b). Kim et al. (2018) propose a temporal decomposition approach, which is based on solving the problem by relaxing constraints regarding multi-period, for the UC problem with a longer planning horizon. We lastly note that there is a study that applies learning-based methods to enhance the solvability of large-scale security-constrained UC, by utilizing the practical context that the daily instance with similar characteristics is repeatedly solved (Xavier et al., 2021).

1.2.2 Optimization approaches under uncertainty in power system operation

There have been a lot of studies on optimization models and solution approaches for power system operation when uncertain factors are incorporated. For the stochastic UC problems, Carpentier et al. (1996) and Takriti et al. (1996) are earlier studies that explicitly considered various uncertain factors in the optimization models, where the electricity demand and outage of generators are considered uncertain in both. From then, various uncertain factors such as electricity prices (Takriti et al., 2000), renewable generation (Tuohy et al., 2009), and outages (Wu et al., 2007) are considered in the stochastic programs for UC problems. Several papers integrated the uncertainty in both renewable generation and electricity demand as net load (e.g., Bertsimas et al., 2013; Jiang et al., 2016), which is the load demand subtracted by the renewable generation.

To solve stochastic programs, decomposition approaches are widely used. Benders decomposition method (Benders, 1962), which is a decomposition based on partitioning variables and widely used to solve a variety of optimization problems

(Rahmaniani et al., 2017), is used to solve stochastic UC problems. Wang et al. (2013) presented a Benders decomposition method with the model with uncertain wind power generation and sub-hourly dispatch constraints. Zheng et al. (2013) proposed an elaborated Benders decomposition to solve two-stage stochastic UC with uncertain renewable generation when the discrete decisions are in both stages. Another decomposition approach is the scenario decomposition approach, which refers to relaxing so-called non-anticipativity constraints that the decisions in two scenarios are the same when they share the same information to a certain stage. Then, when the constraints are relaxed, the remaining problem can be decomposed for each scenario, which is a deterministic problem. In Wu et al. (2007), various uncertain factors including random outages and load forecasting errors are incorporated into the model and it is solved with scenario decomposition methods. Papavasiliou and Oren (2013) also proposed a scenario decomposition approach to solve the security-constrained UC problem with uncertain wind power generation. On the other hand, the unit decomposition approach, which can be used to solve the deterministic UC problem as in the previous subsection, can also be applied to stochastic UC problems, where the resulting subproblem can be decomposed to the stochastic self-scheduling problems. In Carpentier et al. (1996), an augmented Lagrangian approach is proposed and the subproblems are solved by dynamic programming. In addition, stochastic Lagrangian relaxation methods are also proposed in (Nowak and Römisch, 2000). Shiina and Birge (2004) solved the UC problem with uncertain demand and outage of generators by applying column generation techniques. Three types of uncertain factors, which are load, fuel, and electricity prices, are considered in Takriti et al. (2000), and the model is solved by Lagrangian relaxation, and

Benders' decomposition is applied to solve the subproblems.

To appropriately deal with the sequential realization of uncertain factors, Uçkun et al. (2015) proposed a hybrid optimization model that can incorporate dynamic decisions and obtain efficiency for stochastic wind power generation. They introduced a bucket that groups similar scenarios and imposed non-anticipativity requirements among the buckets, which can reduce the computational burden. Jiang et al. (2016) proposed strong valid inequalities for multistage stochastic unit commitment problem, where net load is considered uncertain. To solve the multistage model, stochastic dual dynamic programming (Pereira and Pinto, 1991) is widely used, which sequentially approximates the expected-to-go function and is applied to the energy planning problem. Zou et al. (2019) proposed a stochastic dual dynamic inter programming method, which is an extension to be used when binary decisions exist in every stage, to solve the problem with uncertain net load. For a comprehensive review on stochastic programming models, we refer to Zheng et al. (2015), Tahanan et al. (2015), and Van Ackooij et al. (2018).

For robust optimization models, Street et al. (2011) proposed a model that can accommodate up to k outages out of all N system components. This model was extended to accommodate various emergency situations and includes methods to model economic redispatch in case of emergency (Wang et al., 2013). Zhao and Zeng (2012) considered uncertainty of electricity price, wind-power generation, and the price-elastic demand curve. The model is solved by a column-and-constraint generation method, which has been represented as a generic solution approach to solve general two-stage robust optimization in the subsequent work (Zeng and Zhao, 2013). Bertsimas et al. (2013) proposed a robust optimization model that took the

uncertainty in the net load into consideration and solved the model based on Benders' decomposition techniques. Cho et al. (2019) proposed a new box-based robust optimization to enhance the feasibility of the solutions. Lorca et al. (2016) proposes a constraint generation approach to the multistage adaptive robust optimization model for unit commitment problem, where affine decision rules are adopted to enhance the tractability of the problem.

For DRO model, approximations based on the decision rules are widely used. Xiong et al. (2016) solved a two-stage DRO model with linear decision rule approximation. Duan et al. (2017) carefully designed the ambiguity set to make the corresponding DRO model tractable and scalable. Zhao and Guan (2016) propose a risk-averse stochastic optimization model with a finite number of scenarios and devise a Benders' decomposition approach to solve the problem. Benders decomposition and column-and-constraint generation methods are proposed based on the characteristic when the uncertainty is in the right-hand-side in the Wasserstein distance-based DRO model in Gamboa et al. (2021), where the tests are conducted to solve the UC problem.

Also in the microgrid environment, various optimization models including stochastic optimization (e.g., Farzin et al., 2017; Alvarado-Barrios et al., 2020), robust optimization (e.g., Moretti et al., 2020; Gholami et al., 2017), distributionally robust optimization (e.g., Yurdakul et al., 2021) have been proposed. Among the uncertain factors, the possibility of islanding events is one of the distinct features of microgrid operation compared to the general UC problem. Optimization models were proposed in Zacharia et al. (2019) for grid-connected and islanded modes with different objectives. A rolling horizon approach combined with a stochastic optimization model

was proposed in Bashir et al. (2019) to operate a microgrid for a one-year planning horizon. A stochastic optimization model was proposed in Farzin et al. (2017) to minimize the operation costs during unscheduled islanding where the islanding periods are uncertain. However, a few papers have considered it during the planning horizon, with an appropriate characterization of possible islanding events in advance. Khodaei (2013) proposed the $T - \tau$ criteria, which considers an islanding event of τ consecutive time periods during the given planning horizon with T time periods, and non-dispatchable generation was additionally considered in Khodaei (2014). The models in both studies aim to minimize power mismatches under worst-case realizations. Lee et al. (2021) proposed a multistage stochastic optimization approach considering the possibility of multiple islanding events in a planning horizon. Gholami et al. (2016) proposed a two-stage stochastic optimization model that considers both contingency-based and normal-operation-based uncertainty. Uncertain factors, including both islanding events and net load, were considered in a two-stage robust optimization framework in Gholami et al. (2017), Guo and Zhao (2018), and Mansouri et al. (2022), where the set of possible islanding events is predetermined.

1.3 Research motivations and contributions

Although there has been a considerable amount of research on power system operation under uncertainty, there are some areas where the research is not yet sufficient. Firstly, various types of power systems such as microgrids or single-generator systems need to be further investigated. For example, microgrids contain various distributed energy resources and have unique features such as islanding events that need to be

considered. Furthermore, optimization models and solution approaches under uncertainty need to be further investigated since traditional stochastic programs are insufficient to deal with distributional uncertainty and become more challenging to solve as the number of scenarios increases. Moreover, the computation burden increases when we further consider the multistage optimization models, although they can adequately represent the sequential realization of uncertain factors. To address these challenges, optimization models and solution approaches need to be extended and elaborated by considering the characteristics of power system operation and uncertain factors. To resolve these issues, this dissertation aims to extend the optimization methodologies used in power system operation under uncertainty. Specifically, we focus on various optimization problems that can occur in power system operation and propose efficient optimization methods to solve the problems. These efficient methods include constructing effective and scalable optimization models and proposing efficient solution approaches based on decomposition.

First, we propose a new optimization model in two-stage stochastic programming widely used in the literature. Typically, two-stage stochastic programs have been modeled and solved based on a finite support assumption, but a large number of scenarios makes it hard to solve, and there also are potential risks of inaccurate estimation of the underlying distribution. A new optimization model is proposed to mitigate the drawbacks and it is a risk-averse representation of a two-stage stochastic program with finite support, which we call a partition-based risk-averse two-stage stochastic program. In the model, a set of scenarios is partitioned into several groups, and the second-stage cost is defined as the expectation of risk levels for all groups. In particular, the conditional value-at-risk is considered a risk measure for each group,

and so the risk level of the model is affected by a quantile parameter or a partition of a given set of scenarios. One of the strong advantages of the model is that it enables efficient decomposition approaches, which is column-and-constraint generation. In addition, a scenario partitioning algorithm to enable the risk level of the model to be close to a given target is devised, and partitioning schemes for combining it with the proposed column-and-constraint generation algorithm are proposed.

Next, we investigate the 1UC problem under uncertain electricity prices. Especially, to maximize an IPP's net profit considering uncertainty in electricity prices, we study the stochastic self-scheduling problem and the self-commitment problem. To deal with a large number of price scenarios in the stochastic self-scheduling problem, we devise an efficient dynamic programming algorithm that is based on probing a finite number of generation amounts that can be optimal. For the self-commitment problem, we propose a dynamic programming algorithm whose complexity is linear with regard to the number of time periods. Another purpose of studying single-generator systems is to derive an efficient solution approach to solving general UC problems. By leveraging efficient algorithms on 1UC problems, we propose two unit decomposition methods to solve the UC problem under stochastic net load. We present Lagrangian relaxation and column generation methods to implement the methods, which include a novel decomposition that uses the self-commitment problem as a substructure and has not been proposed.

Finally, we study microgrid operation with the sequential realization of two uncertain factors: stochastic net load and islanding events. To address sequential realizations, operation plans need to be adaptable to the dynamics that these uncertain factors sequentially reveal in a given planning horizon. Although multistage stochas-

tic optimization models can address the dynamics and probabilistic nature of uncertainty, they suffer from the curse of dimensionality in that the number of scenarios grows exponentially with regard to the number of time periods. To overcome this drawback, we propose scalable optimization models for operating microgrids under stochastic islanding and net load, while ensuring that the solutions from the models are adaptable to sequential realization. In particular, integrated optimization models are presented to deal with both uncertain factors based on the combination of the proposed models for each of the two, which is scalable compared to the standard multistage model used in the literature.

1.4 Outline of the dissertation

The remainder of the dissertation is organized as follows.

- In Chapter 2, we study a general two-stage stochastic program that is widely used in the literature including power system operation. We propose a new model which we call a partition-based risk-averse two-stage stochastic program. We analyze the characteristics of the new model and provide exact and heuristic algorithms based on column-and-constraint generation. Next, we propose a partitioning algorithm to make the risk level of the model close to the pre-specified target, and a scheme that integrates the partitioning algorithm and the proposed solution approach. We conduct computational experiments to show the efficiency of the proposed algorithm and the effectiveness of the new model.

- In Chapter 3, to address 1UC problems under electricity price uncertainty,

two types of problems are studied. For the stochastic self-scheduling problem, we propose a dynamic programming algorithm that can efficiently incorporate a large number of scenarios. For the self-scheduling problem, we propose an efficient dynamic programming algorithm that can reduce the computational complexity proposed in the literature. Next, we propose efficient unit decomposition approaches to the UC problem under stochastic net load. We demonstrate the efficiency of the solution approaches through computational experiments.

- In Chapter 4, operating microgrids under two stochastic factors, net load and islanding events, where the uncertain factors sequentially realize. We first present a standard multistage stochastic optimization model to incorporate them. Next, for the main contribution, we propose scalable optimization models based on integrating each model considering each uncertain factor. Numerical experiments are constructed to demonstrate the scalability of the proposed models, and it also demonstrated that the proposed scheme is effective in microgrids operation.

- In Chapter 5, we summarize the results and contributions of the dissertation and present future research directions.

Chapter 2

Partition-based risk-averse two-stage stochastic program

2.1 Introduction

Two-stage stochastic programming is a representative modeling framework for describing sequential decision-making under uncertainty, wherein decision variables are partitioned into *here-and-now* decisions and *wait-and-see* decisions. It has a wide range of applications, such as operations in energy systems (Papavasiliou and Oren, 2013), healthcare systems (Kim and Mehrotra, 2015), network design (Santoso et al., 2005), and so on. In the literature, it is commonly assumed, based on the theoretical background of sample average approximation (e.g., Kleywegt et al., 2002), that the distribution of uncertain parameters has *finite support*. In other words, we have, in advance, a set of scenarios $\mathcal{S}$ that represents underlying uncertainties, with each scenario $s \in \mathcal{S}$ having its probability p_s. On this assumption, the two-stage stochastic program in the *risk-neutral* sense (SP) can be cast as below:

$$\text{(SP)} \quad \min_{x \in X} \left\{ c^T x + \sum_{s \in S} p_s Q(x, s) \right\},$$

where $Q(x, s) := \min\{q_s^T y_s \mid T_s x + W_s y_s \geq h_s\}$ is the optimal second-stage cost for each scenario $s \in \mathcal{S}$ with a given first-stage decision variable $x \in X \subseteq \mathbb{R}^n$. For each $s \in \mathcal{S}$, $y_s \in \mathbb{R}^d$ is the second-stage decision variable, and $q_s \in \mathbb{R}^d$, $h_s \in \mathbb{R}^m$, $T_s \in \mathbb{R}^{m \times n}$, and $W_s \in \mathbb{R}^{m \times d}$ are given data. The objective function is comprised of two parts: the first-stage cost $c^T x$ for a given $c \in \mathbb{R}^n$ and the second-stage cost which is the expectation of $Q(x, s)$ for all scenarios.

There is wide-ranging research on efficient solution methods for the SP, which is represented as a large-scale deterministic optimization model, usually based on constraint-generation or decomposition methods such as the dual decomposition (Carøe and Schultz, 1999), the L-shaped method (Birge and Louveaux, 2011), the progressive hedging (Rockafellar and Wets, 1991), and others. Nonetheless, solving the SP is computationally burdensome, because the size of the model is proportional to the number of scenarios, and a large number of scenarios are needed to represent underlying distribution in practice. In addition, accurate estimation of the underlying distributions of the random variables might not be possible in practice (Park and Lee, 2017). Recently, to mitigate the disadvantage of the SP, several alternative two-stage models that employ risk measures along with the associated solution algorithms have been proposed (Zeng and Zhao, 2013; Blanco and Morales, 2017; Mınguez et al., 2021).

In this chapter, we propose a new model which we call the *partition-based risk-averse two-stage stochastic program* (PSP for short), in which the second-stage cost is represented as the expectation of risk values for all *groups* instead of for all scenarios. Let $\mathcal{K} = \{S_1, \ldots, S_K\}$ be a partition of a set of scenarios $\mathcal{S}$ with $K \geq 1$ elements. We sometimes denote a partition $\mathcal{K}$ only by its indices, i.e. $\mathcal{K} = \{1, \ldots, K\}$, with a

slight abuse of notation. Throughout this chapter, we denote each element $k \in \mathcal{K}$ by *group* k, and the probability of each group k is defined by $\tilde{p}_k := \sum_{s \in S_k} p_s$. To represent the second-stage cost for each group, we define $\boldsymbol{Q_k}(x)$ as a random variable with its support on $\{Q(x,s)\}_{s \in S_k}$, and with its probability $p_{s|k} := p_s/\tilde{p}_k$ for $s \in S_k$ and $k \in \mathcal{K}$. The *conditional value-at-risk* (CVaR) with a quantile parameter $\alpha \in [0,1)$, which is widely used in risk-averse stochastic programs, is adopted as the risk measure for each group. It has been used in financial risk management and is a coherent risk measure that is known to have better mathematical properties than the value-at-risk (VaR) (Artzner et al., 1999). The model minimizes the sum of the first-stage cost and the second-stage cost, where the second-stage cost, which we also call the *risk level*, is represented as the expectation of CVaR values for all groups. For a given partition $\mathcal{K}$ of $\mathcal{S}$ and $\alpha \in [0,1)$, a specific model of the PSP, which is denoted as $\mathrm{PSP}(\mathcal{K}, \alpha)$, is defined as

$$(\mathrm{PSP}(\mathcal{K},\alpha)) \quad z^{\alpha}_{\mathcal{K}} = \min_{x \in X} \left\{ c^T x + \sum_{k \in \mathcal{K}} \tilde{p}_k \mathrm{CVaR}_{\alpha}\left(\boldsymbol{Q_k}(x)\right) \right\}. \tag{2.1}$$

For each scenario $s \in \mathcal{S}$, the second-stage cost $Q(x,s)$ is defined as in the definition of the SP. We assume that $Q(x,s)$ is feasible and bounded for any first-stage solution $x \in X$ and for all $s \in \mathcal{S}$, which is a widely applied assumption in the literature.

The PSP is a generalization of a number of two-stage stochastic (or robust) optimization models studied in the literature. For example, with $\mathcal{K} = \{\mathcal{S}\}$, the corresponding model $\mathrm{PSP}(\{\mathcal{S}\}, \alpha)$ becomes the model proposed by Mınguez et al. (2021). Specifically, with $\alpha = 0$, $\mathrm{PSP}(\{\mathcal{S}\}, 0)$ is the SP (the two-stage stochastic program in the risk-neutral sense). On the other hand, if we set $\alpha = 1 - \epsilon$ with a small positive

number ϵ such that $\epsilon \leq \min_{s\in\mathcal{S}} p_s$, then $\mathrm{CVaR}_{1-\epsilon}\left(\boldsymbol{Q_k}(x)\right) = \max_{s\in S_k} Q(x,s)$. Therefore, the model $\mathrm{PSP}(\mathcal{K}, 1-\epsilon)$ corresponds to the following model $\mathrm{RO}(\mathcal{K})$ studied by Blanco and Morales (2017) in a special application context of the unit commitment problem. In particular, $\mathrm{RO}(\{\mathcal{S}\})$ is the two-stage robust optimization model (e.g., Zeng and Zhao, 2013).

$$(\mathrm{RO}(\mathcal{K})) \quad \min_{x\in X}\left\{c^T x + \sum_{k\in\mathcal{K}} p_k \max_{s\in S_k} Q(x,s)\right\}. \tag{2.2}$$

Furthermore, the model PSP can be interpreted based on the following relationship with the distributionally robust optimization (DRO). Using the well-known correspondence between a coherent risk measure and an *ambiguity set* in the DRO literature (e.g., Artzner et al., 1999; Rahimian and Mehrotra, 2019), it can be shown that $\mathrm{PSP}(\mathcal{K},\alpha)$ is equivalent to the following $\mathrm{DRO}(\mathcal{K},\alpha)$ with an ambiguity set $\mathbb{P}(\mathcal{K},\alpha)$:

$$(\mathrm{DRO}(\mathcal{K},\alpha)) \quad \min_{x\in X}\left\{c^T x + \max_{p'\in\mathbb{P}(\mathcal{K},\alpha)} \sum_{s\in\mathcal{S}} p'_s Q(x,s)\right\},$$

where $\mathbb{P}(\mathcal{K},\alpha) = \{p' \in \mathbb{R}_+^{|\mathcal{S}|} \mid \sum_{s\in\mathcal{S}_k} p'_s = \tilde{p}_k, \forall k\in\mathcal{K},\ p'_s \leq p_s/(1-\alpha), \forall s\in\mathcal{S}\}$.

Let us call a partition $\bar{\mathcal{K}}$ a *refinement* of a given partition $\mathcal{K}$ if one can obtain $\bar{\mathcal{K}}$ by repeating partitioning for one or more groups of $\mathcal{K}$. From the equivalence between $\mathrm{PSP}(\mathcal{K},\alpha)$ and $\mathrm{DRO}(\mathcal{K},\alpha)$ with the fact that $\mathbb{P}(\bar{\mathcal{K}},\alpha) \subseteq \mathbb{P}(\mathcal{K},\alpha)$ when $\bar{\mathcal{K}}$ is a *refinement* of $\mathcal{K}$, it can be deduced that the risk level of $\mathrm{PSP}(\bar{\mathcal{K}},\alpha)$ is not higher than that of $\mathrm{PSP}(\mathcal{K},\alpha)$. In addition, it is also clear that the risk level of $\mathrm{PSP}(\mathcal{K},\alpha)$ is monotone non-decreasing as the value of α increases. Therefore, the salient feature of the model PSP is that one can control the risk level by the construction of a

partition $\mathcal{K}$ as well as the choice of α. The important implication of this is that, for a given α and $\mathcal{K}$, there might exist $\beta > \alpha$ and a refinement $\bar{\mathcal{K}}$ of $\mathcal{K}$ such that the risk level of PSP$(\bar{\mathcal{K}}, \beta)$ closely approximates that of PSP$(\mathcal{K}, \alpha)$. If it is the case, PSP$(\bar{\mathcal{K}}, \beta)$ can be used as a surrogate model for PSP$(\mathcal{K}, \alpha)$, and vice versa.

Our main purpose is to present the model PSP with the characteristics mentioned above and propose efficient solution algorithms to deal with a large number of scenarios along with effective scenario partitioning schemes to control the risk level of the model. The remainder of the chapter is organized as follows. We first review the related literature and summarize the contributions of our study in Section 2.2. In Section 2.3, we propose an efficient *column-and-constraint generation* algorithm to solve PSP$(\mathcal{K}, \alpha)$ for a given partition $\mathcal{K}$ and $\alpha \in [0, 1)$. In Section 2.4, we first present a heuristic algorithm to construct a partition with a desired risk level. Then, we present two partitioning schemes including a novel *adaptive partitioning scheme* that incorporates the heuristic algorithm into the column-and-constraint generation algorithm for PSP$(\mathcal{K}, \alpha)$. In Section 2.5, we demonstrate, through numerical experiments, the effectiveness of the proposed partitioning schemes and the efficiency of the proposed column-and-constraint generation algorithm. Finally, in Section 2.6, we summarize the contents of the chapter.

2.2 Literature review and contributions

There have been a number of studies on risk-averse stochastic programs. First of all, an important relationship between the risk-averse stochastic programs and the DRO is shown in Jiang and Guan (2018), which showed that an ℓ_1-distance-based ambiguity set in the DRO is equivalent to a risk-averse stochastic program with its objective

function being the weighted sum of a supremum and the CVaR functional. For solution approaches to risk-averse stochastic programs with finite support, the L-shaped methods have been proposed to solve a single-stage problem with the CVaR functional (Künzi-Bay and Mayer, 2006) or a two-stage problem with binary variables in both stages with various ambiguity sets in the DRO framework (Bansal et al., 2018). Another well-known solution approach to a risk-averse stochastic program is the *column-and-constraint generation* method, which was first proposed in Zeng and Zhao (2013) to solve the two-stage robust optimization model RO($\{\mathcal{S}\}$), which is the same as PSP($\{\mathcal{S}\}, 1-\epsilon$). They demonstrated that this method requires a smaller number of iterations to converge than does the Benders decomposition method. It has been applied to solve a number of risk-averse stochastic programs. For example, An and Zeng (2015) extended the method to the two-stage robust optimization model with multiple polyhedral uncertainty sets. Blanco and Morales (2017) proposed RO($\mathcal{K}$) for the unit commitment problem, which is the same as PSP($\mathcal{K}, 1-\epsilon$), that considers a partition of the set of scenarios and the maximum function as the risk measure, and devised an algorithm based on column-and-constraint generation. Mınguez et al. (2021) proposed a generic risk-averse two-stage stochastic program that considers the CVaR as the risk measure, which is the same as PSP($\{\mathcal{S}\}, \alpha$), and devised a column-and-constraint generation algorithm for it.

In the stochastic programming literature, partitioning the set of scenarios, *scenario partitioning* for short, has been used mainly in obtaining primal and dual bounds or accelerating existing solution approaches. It is also called scenario *grouping* or *clustering*. In Birge (1982), *group subproblems* are proposed to obtain better primal and dual bounds of two-stage stochastic linear programs, and a bound hi-

erarchy among different group subproblems is shown. Later, the idea is extended to multi-stage settings in Sandıkçı et al. (2013) and Sandikçi and Özaltin (2017). Scenario partitioning also has been used to enhance existing solution approaches. In Escudero et al. (2013), it is used to get an improved dual bound of a Lagrangian relaxation-based solution approach in the two-stage mixed-binary stochastic program. In Crainic et al. (2014), it is used to devise a progressive hedging-based heuristic algorithm for stochastic network design problems. In those studies mentioned above, a partition is usually constructed *a priori* with similarity-based clustering methods (e.g., Crainic et al., 2014; Blanco and Morales, 2017), such as the k-means clustering algorithm. Since the scenario partitioning procedure itself could be improved by further investigating its objective and optimizing its goal, there have been a few studies on how to make *good* partitions. Ryan et al. (2020) presented a mixed-integer program to construct a partition that maximizes dual bound improvement for a given set of incumbent solutions in a two-stage stochastic program. Deng et al. (2020) proposed a bilevel mixed-integer program problem and a branch-and-cut method to obtain the tightest *quantile bound* by utilizing the fact that the quantile bound for a chance-constrained stochastic program can be improved by scenario partitioning. Lastly, Song and Luedtke (2015) and its follow-up study by Ackooij et al. (2018) proposed adaptive partitioning methods to exactly solve two-stage stochastic linear programs in the risk-neutral sense.

Now, we present the motivations and contributions in comparison with the previous studies in the literature mentioned above.

- As mentioned so far, several risk-averse two-stage stochastic (or robust) optimization models along with associated solution algorithms have been studied

(Zeng and Zhao, 2013; Blanco and Morales, 2017; Mınguez et al., 2021). However, studies on a unified modeling and solution approach encompassing those previous studies have been limited to the best of our knowledge. To fill this gap, we propose the model PSP with an efficient solution algorithm, which is a generalization of the previous studies in terms of modeling capacity and solution approaches. As mentioned in Section 2.1, risk-averse stochastic optimization models studied by Blanco and Morales (2017), and Mınguez et al. (2021) including robust optimization model (e.g., Zeng and Zhao, 2013) are special cases of the PSP, and the proposed solution algorithm can be directly applied to those models.

- Although the PSP can be solved by extending the algorithm proposed in Mınguez et al. (2021), which was originally proposed to solve PSP($\{\mathcal{S}\}, \alpha$), we propose a column-and-constraint generation algorithm and theoretically show that the performance of the proposed algorithm is at least as good as that of an extension of their algorithm. In addition, our algorithm has a controllable parameter that can manage the trade-off between optimality and computational time. Through extensive computational experiments, we also demonstrate the efficiency of the proposed algorithm.

- Scenario partitioning in the PSP is used to model the risk level of a specific model PSP($\mathcal{K}, \alpha$) together with $\alpha \in [0, 1)$, whereas, in the existing studies, it has been mainly used for the purpose of obtaining upper or lower bounds for a given two-stage stochastic program. From the perspective of modeling the risk level, Blanco and Morales (2017) proposed RO($\mathcal{K}$) mentioned above in which

the risk level may vary according to scenario partitioning, and used the k-means clustering algorithm to form a partition. However, there is a limitation in that it is difficult for decision makers to understand what they are trying to optimize and that it has not been sufficiently investigated how well a partition is constructed. To address this modeling issue, in this study, we defined a scenario partitioning problem to form a partition from the point of view of a decision maker by making the risk level of the PSP to be close to a given target risk level and devised a heuristic algorithm whose effectiveness is demonstrated through computational experiments.

- We also propose two *partitioning schemes* for combining the proposed partitioning method with the column-and-constraint generation algorithm for the PSP. The first scheme, which we call the *a priori* partitioning scheme, is to form a partition with a desired risk level using the proposed partitioning algorithm in advance as in the previous studies. The second one is a novel partitioning scheme, which we call the *adaptive partitioning scheme*, whereby, to obtain a partition $\mathcal{K}$ with a desired risk level as well as an optimal solution of $\mathrm{PSP}(\mathcal{K}, \alpha)$, a partition is gradually re-constructed by making use of incumbent solutions obtained in the process of the proposed column-and-constraint generation algorithm. Computational experiments show that the proposed partitioning schemes are effective.

2.3 Column-and-constraint generation approaches for the PSP

In this section, we propose efficient column-and-constraint generation algorithms to exactly solve the model $\mathrm{PSP}(\mathcal{K}, \alpha)$ with a given partition $\mathcal{K}$. Recall that it has originally been proposed in Zeng and Zhao (2013) to solve the two-stage robust optimization model ($\mathrm{RO}(\{\mathcal{S}\}) = \mathrm{PSP}(\{\mathcal{S}\}, 1-\epsilon)$). Especially, we propose a *primal* column-and-constraint generation algorithm in Section 2.3.1, and present the *dual* counterpart in Section 2.3.2. The latter is an extension of the algorithm in Mınguez et al. (2021), and the lower bounds of the two algorithms are compared. Before demonstrating the algorithm, we first present an extensive formulation of $\mathrm{PSP}(\mathcal{K}, \alpha)$, by using a well-known representation of a CVaR value as an optimal objective value of a linear program.

Proposition 2.1. *The extensive formulation of the model* $\mathrm{PSP}(\mathcal{K}, \alpha)$ *is as follows.*

$$
\begin{aligned}
(\mathrm{PSP}(\mathcal{K}, \alpha)) \quad z^{\alpha}_{\mathcal{K}} = \min \;\; & c^T x + \sum_{k \in \mathcal{K}} \tilde{p}_k \eta_k + \frac{1}{1-\alpha} \sum_{s \in \mathcal{S}} p_s v_s \\
\text{s.t.} \;\; & x \in X, \\
& v_s + \eta_k \geq q_s^T y_s \quad \forall s \in S_k, k \in \mathcal{K}, \\
& T_s x + W_s y_s \geq h_s \quad \forall s \in \mathcal{S}, \\
& v_s \geq 0 \quad \forall s \in \mathcal{S}.
\end{aligned}
$$

Proof. By using the well-known result that obtaining a CVaR value of a random variable can be formulated as a linear program problem (Rockafellar and Uryasev, 2000), the CVaR value of second-stage costs for a group $k \in \mathcal{K}$ can be represented

as follows:

$$\begin{aligned} \mathrm{CVaR}_\alpha(\boldsymbol{Q_k}(x)) = \min \ & \eta_k + \frac{1}{1-\alpha} \sum_{s \in S_k} p_{s|k} v_s \\ \text{s.t.} \ & v_s + \eta_k \geq q_s^T y_s \quad \forall s \in S_k, \\ & T_s x + W_s y_s \geq h_s \quad \forall s \in S_k, \\ & v_s \geq 0 \quad \forall s \in S_k. \end{aligned} \tag{2.3}$$

Then, incorporating the formulation into the model (2.1) yields the desired result.

□

2.3.1 Primal column-and-constraint generation algorithm

In the column-and-constraint generation algorithm, variables and constraints corresponding to certain scenarios are gradually added to the restricted master problem, until the upper and lower bounds coincide. The algorithm starts from restricted master problem $\mathrm{RMP}_{\mathcal{K}}(\mathcal{S}')$ with a scenario subset $\mathcal{S}' \subseteq \mathcal{S}$ as follows.

$$\begin{aligned} (\mathrm{RMP}_{\mathcal{K}}(\mathcal{S}')) \quad z(\mathcal{S}') = \min \quad & c^T x + \sum_{k \in \mathcal{K}} \tilde{p}_k \eta_k + \frac{1}{1-\alpha} \sum_{s \in \mathcal{S}} p_s v_s \\ \text{s.t.} \quad & x \in X \\ & v_s + \eta_k \geq \underline{\eta} \quad \forall k \in \mathcal{K}, s \in S_k \cap (\mathcal{S} \setminus \mathcal{S}') \\ & v_s + \eta_k \geq q_s^T y_s \quad \forall k \in \mathcal{K}, s \in S_k \cap \mathcal{S}' \qquad (2.4) \\ & T_s x + W_s y_s \geq h_s \quad \forall s \in \mathcal{S}' \qquad (2.5) \\ & v_s \geq 0 \quad \forall s \in \mathcal{S}, \end{aligned}$$

where $\underline{\eta}$ is a lower bound of variable η_k that is small enough. We remark that one can also obtain an optimal solution of $\mathrm{PSP}(\mathcal{K}, \alpha)$ by solving the master problem

(MP:=$\text{RMP}_{\mathcal{K}}(\mathcal{S})$) directly using a commercial solver. Since $\text{RMP}_{\mathcal{K}}(\mathcal{S}')$ is a relaxation of MP, the optimal objective value $z(\mathcal{S}')$ is a valid lower bound of MP.

When one has an incumbent first-stage solution $\bar{x}$ obtained by solving a restricted master problem, one can also obtain corresponding optimal second-stage cost $Q(\bar{x}, s)$ for all scenarios $s \in \mathcal{S}$. Then, a primal objective value can be derived as $c^T\bar{x} + \sum_{k\in\mathcal{K}} p_k \text{CVaR}_\alpha(\boldsymbol{Q}_k(\bar{x}))$, and becomes the upper bound of the master problem.

The main step of the algorithm is to generate scenarios that have not already been generated for the restricted master problem unless the lower and upper bounds converge. Whether the algorithm is exact or not depends on how scenarios to be generated are selected. To indicate which scenarios to be added, we define a vector $\pi(x)$ which depends on the second-stage costs $Q(x, s)$ for $s \in \mathcal{S}$, as follows:

$$\pi(x) := \underset{\pi\in\mathbb{R}^{|\mathcal{S}|}}{\text{argmax}} \left\{ \sum_{s\in\mathcal{S}} Q(x,s)\pi_s \;\Big|\; 0 \le \pi_s \le \frac{p_{s|k}}{1-\alpha} \;\forall s \in S_k, k \in \mathcal{K},\; \sum_{s\in S_k} \pi_s = \Gamma \;\forall k \in \mathcal{K} \right\}. \tag{2.6}$$

In (2.6), $\Gamma \in [0, \frac{1}{1-\alpha}]$ is a pre-defined parameter that can control the number of scenarios to be added. We mention that the optimization problem in (2.6) can easily be solved as follows: for each group $k \in \mathcal{K}$, sort the scenarios according to the second-stage cost and assign each scenario by its maximum possible value ($\pi_s = \frac{p_{s|k}}{1-\alpha}$) from highest to lowest, until the summation of corresponding π_s values is satisfied for Γ with equality, and let the π_s values for the rest of the scenarios be zero. For a given $\pi(x)$, we define a set of scenarios

$$\mathcal{S}(x) := \{s \in \mathcal{S} \mid \pi_s(x) > 0\} \tag{2.7}$$

to indicate which scenarios can be added. Our scenario generation criterion is to

Algorithm 2.1 A primal column-and-constraint generation algorithm for PSP($\mathcal{K}, \alpha$) with given $\mathcal{K}$

Let $z^U = +\infty$, $z^L = -\infty$, $\mathcal{S}' = \emptyset$, and $i = 0$.
while $z^L < z^U$ **do**
  $i = i + 1$
  obtain x^i as an optimal first-stage solution, update lower bound $z^L = z(\mathcal{S}')$ by solving (RMP$_{\mathcal{K}}(\mathcal{S}')$)
  obtain optimal second-stage cost $Q(x^i, s)$ for each scenario $s \in \mathcal{S}$
  obtain each group's second-stage risk CVaR$_\alpha(\boldsymbol{Q}_k(x^i))$ by (2.3) for all $k \in \mathcal{K}$
  let $z^I = c^T x^i + \sum_{k\in\mathcal{K}} \tilde{p}_k \text{CVaR}_\alpha(\boldsymbol{Q}_k(x^i))$ and update upper bound by $z^U = \min\{z^I, z^U\}$
  if $z^L < z^U$ **then**
    calculate $\pi(x^i)$ for $k \in \mathcal{K}$ by (2.6)
    obtain $\mathcal{S}(x^i)$ and generate y_s for $s \in \mathcal{S}(x^i) \setminus \mathcal{S}'$ ▷ add corresponding constraints (2.4) and (2.5) for the restricted master problem
    let $\mathcal{S}' = \mathcal{S}' \cup \mathcal{S}(x^i)$
  end if
end while
return $z^U, z^L, \mathcal{S}', i, x^i$ and $Q(x^i, s)\ \forall s \in \mathcal{S}$

select scenarios with regard to second-stage costs induced by incumbent first-stage solution $\bar{x}$: once second-stage costs $Q(\bar{x}, s)$ for all scenarios $s \in \mathcal{S}$ are obtained, scenarios in $\mathcal{S}(\bar{x})$ can be generated. To be more specific, for the newly generated scenarios, the set of which is $\mathcal{S}(\bar{x}) \setminus \mathcal{S}'$, corresponding second-stage variables y_s and constraints (2.4) and (2.5) are generated for the restricted master problem. At the following iteration, RMP$_{\mathcal{K}}(\mathcal{S}')$ with $\mathcal{S}' = \mathcal{S}' \cup \mathcal{S}(\bar{x})$ will be solved, and the algorithm will continue until the upper and lower bounds coincide, and the process is summarized in Algorithm 2.1.

We now show that for $\Gamma \geq 1$, the lower and upper bounds converge to the optimal objective value in a finite number of iterations according to the scenario-generation criterion. Before presenting the main result, we first present a basic observation. For a given group $k \in \mathcal{K}$ and a given $\boldsymbol{Q}_k(x)$, we define another discrete random variable

$\boldsymbol{Q}'_k(x)$ as follows, where only the realizations are different between the two.

$$Q'(x,s) = \begin{cases} Q(x,s), & \text{if } s \in \mathcal{S}(x), \\ \underline{\eta}, & \text{if } s \notin \mathcal{S}(x), \end{cases} \tag{2.8}$$

where $\underline{\eta}$ is an arbitrarily small value that satisfies $\underline{\eta} \leq \min_{s \in S_k} Q(x,s)$. It trivially holds that $\mathrm{CVaR}_\alpha(\boldsymbol{Q}_k(x)) \geq \mathrm{CVaR}_\alpha(\boldsymbol{Q}'_k(x))$, in that $Q(x,s) \geq Q'(x,s)$ for all $s \in S_k$. We next show that the two risk values are the same.

Proposition 2.2. *For any $k \in \mathcal{K}$, consider discrete random variables $\boldsymbol{Q}_k(x)$ and $\boldsymbol{Q}'_k(x)$ as defined in* (2.8) *where $\mathcal{S}(x)$ is defined in* (2.7)*. Then, for $\Gamma \geq 1$, $\mathrm{CVaR}_\alpha(\boldsymbol{Q}_k(x))$ $= \mathrm{CVaR}_\alpha(\boldsymbol{Q}'_k(x))$.*

Proof. Since the value $\mathrm{CVaR}_\alpha(\boldsymbol{Q}'_k(x))$ is non-decreasing as Γ increases, it suffices to show the equivalence between $\mathrm{CVaR}_\alpha(\boldsymbol{Q}_k(x))$ and $\mathrm{CVaR}_\alpha(\boldsymbol{Q}'_k(x))$ for $\Gamma = 1$. Consider obtaining a CVaR value of $\boldsymbol{Q}_k(x)$, where the optimization problem is represented as a dual linear program as follows:

$$\begin{aligned} &\mathrm{CVaR}_\alpha(\boldsymbol{Q}_k(x)) \\ =&\max\left\{ \sum_{s \in S_k} Q(x,s)\pi_s \;\middle|\; 0 \leq \pi_s \leq \frac{p_{s|k}}{1-\alpha}\ \forall s \in S_k,\ \sum_{s \in S_k} \pi_s = 1 \right\} \\ =&\max\left\{ \sum_{s \in S_k \cap S(x)} Q(x,s)\pi_s + \sum_{s \in S_k \setminus S(x)} \underline{\eta}\pi_s \;\middle|\; 0 \leq \pi_s \leq \frac{p_{s|k}}{1-\alpha}\ \forall s \in S_k,\ \sum_{s \in S_k} \pi_s = 1 \right\} \\ =&\mathrm{CVaR}_\alpha(\boldsymbol{Q}'_k(x)). \end{aligned}$$

The first equation is from the dual representation of the CVaR value and the second equation is from the definition of $S(x)$ in (2.7). Thus, the equivalence holds for

$\Gamma \geq 1$. $\square$

Proposition 2.2 implies that not all the optimal second-stage costs are needed to obtain the exact CVaR cost defined on a whole set of scenarios. Therefore, one can expect that only a subset of scenarios is needed to solve the model exactly and that the proposed solution approach leverages the result. Now, we present the main result, which shows the exactness of the algorithm when $\Gamma \geq 1$.

Proposition 2.3. Algorithm 2.1 *with* $\Gamma \geq 1$ *is exact, i.e.* $z^U = z^L$ *holds in a finite number of iterations.*

Proof. It suffices to show that at least one scenario is generated for the restricted master problem whenever $z^L < z^U$. We assume that no scenarios are added for the first time at iteration $t \in \mathbb{Z}_+$. The incumbent primal objective value z^t at the iteration t, with an optimal first-stage solution x^t, becomes

$$\begin{aligned} z^t = c^T x^t + \min \sum_{k \in \mathcal{K}} \tilde{p}_k \eta_k + \frac{1}{1-\alpha} \sum_{s \in \mathcal{S}} p_s v_s \\ \text{s.t. } v_s + \eta_k \geq Q(x^t, s) \quad \forall s \in S_k, k \in \mathcal{K} \\ v_s \geq 0 \quad \forall s \in \mathcal{S}. \end{aligned}$$

We now show that if the algorithm proceeds to the next iteration, the lower and upper bounds become equal to each other. At the next $t+1$th iteration, since no scenarios are added in the tth iteration, we obtain the same optimal first-stage solution, the lower bound, and the optimal second-stage costs for all scenarios, i.e.

x and $Q(x,s)$ remains the same for all $s \in \mathcal{S}$. Therefore,

$$
\begin{aligned}
z^L = \min \quad & c^T x + \sum_{k\in\mathcal{K}} \tilde{p}_k \eta_k + \frac{1}{1-\alpha} \sum_{s\in\mathcal{S}} p_s v_s \\
\text{s.t.} \quad & x \in X \\
& v_s + \eta_k \geq q_s^T y_s \quad \forall s \in S_k \cap \mathcal{S}', k \in \mathcal{K} \\
& v_s + \eta_k \geq \underline{\eta} \quad \forall s \in S_k \cap (\mathcal{S} \setminus \mathcal{S}'), k \in \mathcal{K} \\
& T_s x + W_s y_s \geq h_s \quad \forall s \in \mathcal{S}' \\
& v_s \geq 0 \quad \forall s \in \mathcal{S} \\
= c^T x^t + \min \quad & \sum_{k\in\mathcal{K}} \tilde{p}_k \eta_k + \frac{1}{1-\alpha} \sum_{s\in\mathcal{S}} p_s v_s \qquad (2.9) \\
\text{s.t.} \quad & v_s + \eta_k \geq Q(x^t, s) \quad \forall s \in S_k \cap \mathcal{S}', k \in \mathcal{K} \\
& v_s + \eta_k \geq \underline{\eta} \quad \forall s \in S_k \cap (\mathcal{S} \setminus \mathcal{S}'), k \in \mathcal{K} \\
& v_s \geq 0 \quad \forall s \in \mathcal{S} \\
= c^T x^t + \min \quad & \sum_{k\in\mathcal{K}} \tilde{p}_k \eta_k + \frac{1}{1-\alpha} \sum_{s\in\mathcal{S}} p_s v_s \qquad (2.10) \\
\text{s.t.} \quad & v_s + \eta_k \geq Q(x^t, s) \quad \forall s \in S_k, k \in \mathcal{K} \\
& v_s \geq 0 \quad \forall s \in \mathcal{S} \\
= z^t \ \geq \ & z^U
\end{aligned}
$$

holds, which means that the two bounds are the same, where the equality in (2.9) is satisfied because x^t is an optimal first-stage solution, and the equality in (2.10) holds by Proposition 2.2 with the fact that $\mathcal{S}'$ includes a set of scenarios $\mathcal{S}(x^t)$ to calculate the exact CVaR value. Further, since at each iteration at least one scenario is added or the algorithm terminates, the number of iterations is no greater than

$|\mathcal{S}|$. Thus, the upper and lower bounds coincide in a finite number of iterations. □

We now note some additional directions for scenario generation in the algorithm. First, when the probabilities of scenarios are equal to each other, i.e. $p_s = 1/|\mathcal{S}|$ for all $s \in \mathcal{S}$, which is generally the case in practice, obtaining $\mathcal{S}(x)$ is much simpler. In that case, since $\tilde{p}_k = n_k/|\mathcal{S}|$ for each group $k \in \mathcal{K}$, where n_k denotes the number of scenarios in group k, it can be deduced that the set $\mathcal{S}(x)$ can be obtained by choosing the highest $\lceil n_k(1-\alpha)\Gamma \rceil$ scenarios for each group k with regard to its second-stage cost $Q(x, s)$. Another notable point is that when the risk function for each group is a maximum function, e.g. $\mathrm{PSP}(\mathcal{K}, 1-\epsilon)$ as in Blanco and Morales (2017). Since ϵ indicates a sufficiently small value that $\epsilon \leq \min_{s\in\mathcal{S}} p_s$, at most one scenario which has the maximum second-stage cost for each group is generated in this case when $\Gamma = 1$.

Finally, we note some practical guidelines for using the controllable parameter Γ. Parameter Γ affects the number of scenarios to be added in one iteration. The simplest implementation is to set $\Gamma = 1$, which indicates a minimal set of scenarios added to ensure optimality. One could expect that the number of iterations has a negative relationship with the number of scenarios added in one iteration. When $\Gamma > 1$, since more scenarios are added in one iteration, the algorithm may converge in fewer iterations. On the other hand, when $\Gamma < 1$, the algorithm is not guaranteed to drive an optimal solution or terminate. For this case, we present a slight modification to the termination condition in Algorithm 2.1. It is to terminate when no scenarios are added for any iteration. Then, it can be used as a heuristic algorithm that the obtained upper bound z^U at the end of the algorithm can be used. But since fewer scenarios are generated in one iteration, the algorithm can terminate in a shorter

time while obtaining a good quality solution. It is worthwhile to analyze the practical choice of the parameter Γ in the proposed column-and-constraint generation method.

2.3.2 Dual column-and-constraint generation and comparison

In this section, we present another column-and-constraint-generation algorithm for $\mathrm{PSP}(\mathcal{K}, \alpha)$ and we compare the two methods. This algorithm is a straightforward extension of the algorithm proposed in Mınguez et al. (2021) for solving $\mathrm{PSP}(\{\mathcal{S}\}, \alpha)$, e.g., the second-stage cost is the CVaR value for the set of whole scenarios. First, we can rewrite the restricted master problem as

$$
\begin{aligned}
(\mathrm{RMP}_{\mathcal{K}}(\mathcal{S}')) \quad \min \quad & c^T x + \sum_{k \in \mathcal{K}} P_k(y) \\
\text{s.t.} \quad & x \in X \\
& T_s x + W_s y_s \geq h_s \quad \forall s \in \mathcal{S}'
\end{aligned}
$$

where $P_k(y)$ is defined as an optimal second-stage risk value for each group $k \in \mathcal{K}$, i.e.

$$
\begin{aligned}
P_k(y) := \quad \min \quad & \tilde{p}_k \eta_k + \frac{1}{1-\alpha} \sum_{s \in S_k} p_s v_s \\
\text{s.t.} \quad & v_s + \eta_k \geq q_s^T y_s \quad \forall s \in S_k \cap \mathcal{S}' \\
& v_s + \eta_k \geq \underline{\eta} \quad \forall s \in S_k \cap \left(\mathcal{S} \setminus \mathcal{S}'\right) \\
& v_s \geq 0 \quad \forall s \in S_k
\end{aligned}
$$

In addition, the problem $P_k(y)$ can be obtained as a maximization problem using linear programming duality, i.e. $P_k(y) = \max\left\{\bar{P}_k(\psi, y) \mid \psi \in \Psi^k\right\}$, where

$$\bar{P}_k(\psi, y) := \sum_{s \in S_k \cap \mathcal{S}'} \psi_s q_s^T y_s + \sum_{s \in S_k \cap (\mathcal{S} \setminus \mathcal{S}')} \psi_s \underline{\eta},$$

$$\Psi^k := \left\{ \psi \in \mathbb{R}^k \;\middle|\; 0 \leq \psi_s \leq \frac{1}{1-\alpha} p_s \; \forall s \in \mathcal{S}_k, \; \sum_{s \in S_k} \psi_s = \tilde{p}_k \right\}.$$

For any $\psi \in \Psi^k$, $\bar{P}_k(\psi, y)$ can be used as a lower estimate of $P_k(y)$ since $P_k(y) \geq \bar{P}_k(\psi, y)$. We can present another column-and-constraint generation algorithm where the lower estimate is gradually updated. In the algorithm, the restricted master problem is slightly modified, which is $\mathrm{RMP}^l_{\mathcal{K}}(\mathcal{S}')$ at iteration $l \in \mathbb{Z}_+$ as below.

$$\begin{aligned}
(\mathrm{RMP}^l_{\mathcal{K}}(\mathcal{S}')) \quad z^l(\mathcal{S}') = \min \quad & c^T x + \sum_{k \in \mathcal{K}} r_k \\
\text{s.t.} \quad & x \in X \\
& T_s x + W_s y_s \geq h_s \quad \forall s \in \mathcal{S}' \\
& r_k \geq \bar{P}_k(\psi^\ell, y) \quad \forall k \in \mathcal{K}, \; \ell = 1, \ldots, l-1,
\end{aligned}$$

where the decision variable r_k indicates an estimate of $P_k(y)$. When the first-stage solution is obtained by x^l at the iteration, the lower estimation of the second-stage risk can be obtained by $\bar{P}_k(\psi^l, y)$, where $\psi^l := \mathrm{argmax}\left\{\sum_{s \in \mathcal{S}} Q(x^l, s)\psi_s \mid \psi \in \Psi^k\right\}$. Any scenario generation criteria that can yield the optimal solutions make the algorithm an exact solution approach for $\mathrm{PSP}(\mathcal{K}, \alpha)$. The exactness of the algorithm whose scenario generation criterion corresponds to that in Section 2.3.1 with $\Gamma = 1$ has shown in Mınguez et al. (2021). We denote the approach as *dual implementa-*

tion of column-and-constraint generation since the second-stage cost is represented in dual form, unlike *primal implementation* in the case of the proposed method in Section 2.3.1. This *dual* implementation has the advantage that a smaller number of variables are needed to represent the second-stage cost for each group $k \in \mathcal{K}$ by r_k, instead of decision variables v_s and η_k in *primal* implementation. On the other hand, the lower bounds of the two restricted master problems can easily be compared. Let $z(\mathcal{S}')$ be an optimal objective value of $\mathrm{RMP}_{\mathcal{K}}(\mathcal{S}')$ and let $z^l(\mathcal{S}')$ be an optimal objective value of $\mathrm{RMP}^l_{\mathcal{K}}(\mathcal{S}')$. Since $P_k(y) \geq \bar{P}_k(\psi^l, y)$, it can trivially be known that $z(\mathcal{S}') \geq z^l(\mathcal{S}')$ holds, and we state the result as follows:

Proposition 2.4. *For a given set of scenarios $\mathcal{S}' \subseteq \mathcal{S}$, $z(\mathcal{S}') \geq z^l(\mathcal{S}')$ for any $l \in \mathbb{Z}_+$.*

Proposition 2.4 points out that when we have the same sets of scenarios for restricted master problems until the iteration $l \in \mathbb{Z}_+$, the primal implementation can yield a better lower bound compared to the dual implementation. Thus, the proposed primal implementation can speed up the column-and-constraint generation algorithm by reducing the number of iterations.

2.4 Scenario partitioning methods

In this section, we propose a framework for effectively and efficiently constructing a partition $\mathcal{K}$ that has a critical role in model $\mathrm{PSP}(\mathcal{K}, \alpha)$. As noted earlier, the risk level of model $\mathrm{PSP}(\mathcal{K}, \alpha)$ is affected by both α and $\mathcal{K}$. The effect of quantile parameter α on the model's risk is straightforward: when α is higher, one can obtain a more risk-averse model; on the other hand, when α is close to zero, the risk of the model

tends towards that of the risk-neutral stochastic program. However, whereas it was mentioned in Section 2.1 that different partitions could lead to different risk levels of PSP, it is still not clear how a decision maker can construct a partition $\mathcal{K}$ to obtain the desired risk level. To this end, we herein define a partitioning problem along with the criteria, and devise an efficient partitioning algorithm to obtain such a partition in a reasonable time; further, we discuss how the proposed partitioning method can be integrated with the proposed solution approach on PSP$(\mathcal{K}, \alpha)$. Throughout the section, a partition $\mathcal{K}$ is chosen among a set of possible partitions $\mathbb{K}$, which is a subset of all possible partitions, with some restrictions. We let $\mathbb{K}$ be a tuple consisting of two scalar values, i.e. $\mathbb{K} = (K, N)$. The first value K indicates the number of groups in a partition, and the second value N denotes the maximum number of scenarios in each group, similar to the settings in the literature (e.g., Ryan et al., 2020; Deng et al., 2020).

First, we propose partitioning criteria, which is to make the risk level of model PSP$(\mathcal{K}, \alpha)$ close to the pre-defined risk level, which we call *target risk*. Precisely, since the risk level of model PSP$(\mathcal{K}, \alpha)$ is affected by the second-stage cost function, it can be compared to the pre-determined risk level without partitioning, i.e. $\mathcal{K} = \{\mathcal{S}\}$. In other words, the goal is to construct a partition where the corresponding model's second-stage cost is closest to the pre-determined *target risk* in the whole set of scenarios for first-stage solutions in X. It is so generic in that one can choose various risk positions by setting different target risks. For example, when the target risk is high, the goal is to find a partition that is close to the risk-averse one; and the target risk is low, the goal is to find a partition that is similar to the risk-neutral one.

We now formally describe the optimization model with the criteria. For second-

stage scenario costs $Q(x, s)$ for all scenarios $s \in \mathcal{S}$ and given $x \in X$, the *target risk* is defined as a functional value of *any* pre-determined real-valued risk function in the whole scenario set $\mathcal{S}$, that can be easily calculated. Here, we let the target risk function be $\text{CVaR}_\beta(\cdot)$ with $\beta \in [0, 1)$ for the consistency with the risk function in $\text{PSP}(\mathcal{K}, \alpha)$. Then, the objective is to find a partition that minimizes the maximum deviation among the first-stage solutions $x \in X$, where the deviation is defined as the difference between the target risk and the actual second-stage cost. The partitioning problem $\mathcal{P}(X)$ can be represented as follows, where $\boldsymbol{Q}(x)$ is a random variable with its support $\{Q(x, s)\}_{s\in\mathcal{S}}$ with probability p_s for $s \in \mathcal{S}$ and $\boldsymbol{Q_k}(x)$ is a random variable with its support $\{Q(x, s)\}$ on $s \in S_k$ with probability $p_{s|k}$ for $k \in \mathcal{K}$.

$$(\mathcal{P}(X)) \quad \min_{\mathcal{K}\in\mathbb{K}} \left\{ \max_{x\in X} \left| \mathbb{E}_{\mathcal{K}} \left[\text{CVaR}_\alpha(\boldsymbol{Q_k}(x)) \right] - \text{CVaR}_\beta(\boldsymbol{Q}(x)) \right| \right\}.$$

It can easily be seen that the optimal objective value of the model is always greater than or equal to zero. When the objective value is zero, it means that the second-stage cost of a partition is the same, with a given target risk, for *any* possible first-stage solution in X, in which case, we can consider the partition to be *ideal*. However, it is rarely the case that the *ideal* partition exists, and it is unrealistic that we know all of the first-stage solutions in advance. Rather, in practice, it is quite natural that we have some of the first-stage solutions, the set of which we denote $\bar{X} \subseteq X$, rather than having all of them. In the following subsection, we propose an efficient algorithm to solve $\mathcal{P}(\bar{X})$.

2.4.1 Partitioning algorithm

Although the problem $\mathcal{P}(\bar{X})$ can be solved through enumeration-based methods, it may be undesirable to exactly solve the problem since the computational burden increases when the number of groups or scenarios increases. Thus, in this section, we instead propose, as motivated by various methods in clustering literature, an efficient heuristic to deal with the problem with a large number of scenarios and groups to be assigned. It is a local search-based heuristic whereby a partition is iteratively reconstructed while no improvements, in terms of deviation, are found. It is similar to the algorithms in clustering literature, such as in partitioning around medoids (Kaufman and Rousseeuw, 1990) in k-medoids clustering. Note that the algorithm leverages the fact that for a given partition $\mathcal{K} := \{S_1, \ldots, S_k\}$, it is relatively easy to calculate the maximum deviation $D_{\mathcal{K}}$, which can be defined as

$$D_{\mathcal{K}} := \max_{x^i \in \bar{X}} \left| \sum_{k \in \mathcal{K}} \tilde{p}_k \mathrm{CVaR}_\alpha(\boldsymbol{Q_k}(x^i)) - \mathrm{CVaR}_\beta(\boldsymbol{Q}(x^i)) \right|, \tag{2.11}$$

because calculating the CVaR value of a discrete random variable can be easily done by solving a linear program or by sorting, as mentioned in Section 2.3.1.

The generic procedure is summarized in Algorithm 2.2. The algorithm starts from an initial partition $\mathcal{K}_0 = \{S_1^0, \ldots, S_k^0\}$ in $\mathbb{K}$. In the algorithm, the maximum deviations are evaluated among *neighborhood* $\mathcal{N}(\mathcal{K})$, which is a set of partitions, for a given partition $\mathcal{K}$. It consists of partitions constructed by two operations, where one is an *exchange* operation and the other is a *transfer* operation. For a partition $\mathcal{K}_t = \{S_1^t, \ldots, S_k^t\}$ at iteration $t \in \mathbb{Z}_+$, in the exchange operation, a scenario in a group is exchanged with a different scenario in a different group, i.e. $S_i^{t+1} =$

Algorithm 2.2 A generic partitioning heuristic to solve $\mathcal{P}(\bar{X})$

1: An initial partition $\mathcal{K}_0 \in \mathbb{K}$ with $\tilde{p}_k = \sum_{s \in S_k} p_s$ given.
2: $\mathcal{K}_{best} = \mathcal{K}_0$, $D_{best} = D_{\mathcal{K}_0}$, $i = 0$ and `improved` = true ▷ $D_{\mathcal{K}_0}$ is calculated by (2.11)
3: **while** `improved` **do**
4: i += 1
5: construct a set of neighborhood $\mathcal{N}(\mathcal{K}_{i-1})$ and calculate $D_{\mathcal{K}}$ for all $\mathcal{K} \in \mathcal{N}(\mathcal{K}_{i-1})$
6: **if** $\min\{D_{\mathcal{K}} \mid \mathcal{K} \in \mathcal{N}(\mathcal{K}_{i-1})\} \geq D_{best}$ **then**
7: `improved` = false
8: **else**
9: move on to the partition $\mathcal{K}_i \in \mathcal{N}(\mathcal{K}_{i-1})$ and $\mathcal{K}_{best} = \mathcal{K}_i$ ▷ with move-on rule
10: **end if**
11: **end while**
12: **return** a partition $\mathcal{K}_{best}$ and its maximum deviation D_{best}

$\left(S_i^t \setminus \{s_i^t\}\right) \cup \{s_j^t\}$ and $S_j^{t+1} = \left(S_i^t \setminus \{s_j^t\}\right) \cup \{s_i^t\}$ for $s_i^t \in S_i^t, s_j^t \in S_j^t$ for $S_i^t, S_j^t \in \mathcal{K}_t$. In the transfer operation, a scenario in a group is transferred to another group, if possible, i.e. $S_i^{t+1} = S_i^t \setminus \{s_i^t\}$ and $S_j^{t+1} = S_j^t \cup \{s_i^t\}$ for $s_i^t \in S_i^t, s_j^t \in S_j^t$ for $S_i^t, S_j^t \in \mathcal{K}_t$. The partition moves on to the next partition, which is one of the neighborhoods that shows improvements in terms of maximum deviation. There are various implementation methods for selecting a partition among neighborhoods showing improvement. In this chapter, we suggest three generic rules: the first one is to move to a partition with the best improvement (*move-best*), the second one is to move to the first partition that shows a decrease (*move-first*), and the third one is to choose a partition randomly among those improved (*move-random*). The procedure continues until no improvements are found, i.e. the deviation no longer decreases. It has an advantage in that we can obtain a feasible partition at any of its iterations, whenever we have an initial partition.

2.4.2 Partitioning schemes

Next, we propose partitioning schemes, in which we denote the entire procedure from constructing a desired partition $\mathcal{K}$ to solving the corresponding model PSP($\mathcal{K}, \alpha$). Recall that one can construct a partition with criteria and an algorithm in Section 2.4.1, and solve the problem by the proposed method in Section 2.3. Here, we discuss two partitioning schemes by which the proposed methodologies can be integrated. One is an *a priori partitioning scheme*, whereby a partition is constructed before solving the model, and the other is an *adaptive partitioning scheme*, in which a partition is adaptively re-constructed within the proposed solution approach.

A priori partitioning scheme

One straightforward way is to solve a model PSP($\mathcal{K}, \alpha$) with the proposed solution approach after constructing a partition $\mathcal{K}$; in other words, a partition is constructed *a priori* by using information that can be obtained in advance. It is a natural and common method widely employed in the literature. When solving the partitioning problem $\mathcal{P}(\bar{X})$ with Algorithm 2.2, it is important to obtain a set of incumbent solutions $\bar{X}$. We note that it could depend on the specific problem one considers. Here, we briefly introduce two generic ways of constructing a set $\bar{X}$, which are based on solving the deterministic counterparts of the stochastic program and independent of specific problem characteristics.

One way is to let $\bar{X} = X^{EV}$, where $X^{EV} = \{x^{EV}\}$ and x^{EV} is defined as an optimal first-stage solution obtained by solving the *expected value problem*, which is well-known in the stochastic programming literature (e.g., Birge, 1982). The solution

x^{EV} is calculated as

$$x^{EV} := \underset{x \in X}{\operatorname{argmin}} \left\{ c^T x + Q(x, \bar{s}) \right\},$$

where the scenario $\bar{s}$ indicates the scenario with the expected values, i.e. the corresponding matrices and vectors are defined as $(q_{\bar{s}}, T_{\bar{s}}, W_{\bar{s}}, h_{\bar{s}}) := (\mathbb{E}[q_s], \mathbb{E}[T_s], \mathbb{E}[W_s], \mathbb{E}[h_s])$.

The second way is to use multiple *single-scenario problems* to obtain a set of first-stage solutions, i.e. $\bar{X} = X^{MS}$, where $X^{MS} = \{x_s\}_{s \in \mathcal{S}}$. In the set, x_s for scenario $s \in \mathcal{S}$ is calculated as

$$x_s := \underset{x \in X}{\operatorname{argmin}} \left\{ c^T x + Q(x, s) \right\}.$$

The latter implementation makes use of various scenarios, which contrasts with the former implementation which considers only one scenario (the expected value scenario). However, it could be burdensome to implement the latter case as the number of scenarios gets larger. In that situation, one could use a subset of $\mathcal{S}$ instead of considering all of the scenarios.

Adaptive partitioning scheme via column-and-constraint-generation

In the *a priori* scheme, the partitioning procedure can be regarded as a preprocessing step in solving model PSP$(\mathcal{K}, \alpha)$. However, the partitioning procedure itself can be computationally demanding, and it depends on the set $\bar{X}$ obtained a priori, which contains only partial information on first-stage solutions. For this purpose, we propose another novel partitioning scheme, which we call *adaptive partitioning*, to construct a partition by making use of information within the column-and-constraint generation proposed in Section 2.3. The main feature of the scheme is that it does not need to choose a set $\bar{X}$ in advance since the set is gradually constructed within the

procedure. In addition, one can obtain a partition $\mathcal{K}$ along with an optimal solution of PSP$(\mathcal{K}, \alpha)$ at the end of the entire procedure. The scheme is similar to the proposed column-and-constraint generation algorithm in Algorithm 2.1, except that it has an additional *re-constructing phase* at each iteration. In that phase, the given partition is updated using incumbent first-stage solutions obtained by solving the restricted master problem. When a partition does not change in the re-constructing phase, the remaining algorithm proceeds in the same manner as the solution approach.

The overall procedure is summarized in Algorithm 2.3, and we state the procedure more specifically. Differing from an *a priori partitioning scheme* that starts with a set of incumbent solutions X_0, it is initialized with an empty set, and an initial partition $\mathcal{K}_0$ in $\mathbb{K}$. At the ith iteration, an incumbent first-stage solution x^i is obtained by solving RMP$_{\mathcal{K}}(\mathcal{S}')$. Then, in the *re-constructing phase*, an incumbent first-stage solution is added, i.e. $X_i = X_{i-1} \cup \{x^i\}$, and a partition $\mathcal{K}_i$ is constructed by solving $\mathcal{P}(X_i)$ by Algorithm 2.2. In the algorithm, $\mathcal{K}_{i-1}$ is set to an initial partition $\mathcal{K}_0$. When the partition remains the same, i.e. $\mathcal{K}_i = \mathcal{K}_{i-1}$, a partition is determined and the re-constructing phase does not proceed until the end of the algorithm. Otherwise, the phase is revisited in subsequent iterations. We note that scenarios are also generated for the restricted master problem for a new partition $\mathcal{K}_i$, which leads to a new incumbent first-stage solution at the next iteration. The obtained upper and lower bounds (z_i^U and z_i^L respectively) become the *true* upper and lower bounds (z^U and z^L respectively) of PSP$(\mathcal{K}, \alpha)$ after partition $\mathcal{K}$ is finalized.

We also show that the lower and upper bounds from in Algorithm 2.3 coincide in a finite number of iterations by extending the result in Proposition 2.3.

Proposition 2.5. Algorithm 2.3 *terminates with* $z^U = z^L$ *in a finite number of*

Algorithm 2.3 An adaptive partitioning scheme to obtain $\mathcal{K} \in \mathbb{K}$ and solve $\mathrm{PSP}(\mathcal{K}, \alpha)$

1: an initial partition $\mathcal{K}_0 \in \mathbb{K}$, $X_0 = \emptyset$, $\mathcal{S}' = \emptyset$
2: set $z^U = +\infty$, $z^L = -\infty$, $i = 0$, $\mathcal{K} = \mathcal{K}_0$, and `modify` = true
3: **while** $z^L < z^U$ **do**
4: $i = i + 1$
5: obtain an optimal first-stage solution x^i and update $z_i^L = z_{\mathcal{K}}(\mathcal{S}')$ by solving $(\mathrm{RMP}_{\mathcal{K}}(\mathcal{S}'))$
6: obtain optimal second-stage cost $Q(x^i, s)$ for each scenario $s \in \mathcal{S}$
7: obtain each group's second-stage risk $\mathrm{CVaR}_\alpha(\boldsymbol{Q}_k(x^i))$ by (2.3) for all $k \in \mathcal{K}$
8: let $z_i^I = c^T x^i + \sum_{k \in \mathcal{K}} \tilde{p}_k \mathrm{CVaR}_\alpha(\boldsymbol{Q}_k(x^i))$ and update $z_i^U = \min\{z_i^I, z_i^U\}$
9: **if** `modify` **then** ▷ re-construction phase
10: $X_i = X_{i-1} \cup \{x^i\}$
11: construct $\mathcal{K}_i$ by solving $\mathcal{P}(X_i)$ with $\mathcal{K}_{i-1}$ as an initial partition and $\tilde{p}_k = \sum_{s \in S_k} p_s$ for $k \in \mathcal{K}$
12: **if** $\mathcal{K}_i = \mathcal{K}_{i-1}$ **then**
13: `modify` = false, $(z^U, z^L) = (z_i^U, z_i^L)$, $\mathcal{K} = \mathcal{K}_i$
14: **end if**
15: **else**
16: $(z^U, z^L) = (z_i^U, z_i^L)$
17: **end if**
18: obtain $\mathcal{S}(x^i)$ by calculating $\pi(x^i)$ by (2.6)
19: **if** $z^L < z^U$ **then**
20: generate y_s for $s \in \mathcal{S}(x^i) \setminus \mathcal{S}'$ and corresponding constraints (2.4) and (2.5) for the restricted master problem
21: let $\mathcal{S}' := \mathcal{S}' \cup \mathcal{S}(x^i)$
22: **end if**
23: **end while**
24: **return** $\mathcal{K}, z^U, z^L, \mathcal{S}', i, x^i$, and $Q(x^i, s)\ \forall s \in \mathcal{S}$

iterations.

Proof. It suffices to show that the re-construction phase (line 9 to 17 in Algorithm 2.3) runs finite times since $z^U = z^L$ in a finite number of iterations has been shown in Proposition 2.3. Suppose that re-constructing continues and `modify`=true. If no scenarios are generated at that iteration, the first-stage solution remains the same at the following iteration. Since it makes the partition remain the same, the re-

construction phase terminates. On the other hand, if at least one scenario is generated, the procedure continues but this can occur finite times since the number of scenarios is finite. Thus, the re-constructing phase is repeated finite times, and thus the algorithm terminates in a finite number of iterations. □

2.5 Computational experiments

We conduct extensive numerical experiments to demonstrate the effectiveness of the proposed partitioning methods and the efficiency of the proposed solution approach. As mentioned, the methodologies in this chapter address generic two-stage stochastic programs with finite support, therefore it can be applied to the various problems having two-stage decision processes including power system operation. Here, we conduct and report computational tests with the two-stage stochastic unit commitment problem, which is a fundamental optimization problem in power system operation. The experiments which are also done for the well-known facility location problem are reported in Appendix A. In the remainder of this section, we first briefly introduce the mathematical formulation of the two-stage stochastic unit commitment problem in Section 2.5.1. Then, we describe the experimental setup including the implementation details and the settings that are used, in Section 2.5.2. Lastly, the computational results are demonstrated in Section 2.5.3.

2.5.1 Two-stage stochastic unit commitment problem

In the unit commitment problem, the decision-maker decides the on/off status along with the generation amount of the set of generators $\mathcal{G}$ for the planning horizon $\mathcal{T} := \{1, \dots, T\}$ to meet electricity demand $d_{s,t}$ for $s \in \mathcal{S}, t \in \mathcal{T}$. Specifically, for a generator $g \in \mathcal{G}$ and each time period $t \in \mathcal{T}$, $x_{g,t}, x^U_{g,t}$ and $x^D_{g,t}$ indicate the on, start-

up, and shut-down status, respectively. Note that $x_{g,0}$ is a given initial on/off state for $g \in \mathcal{G}$. In addition, $y_{g,s,t}$ and $y^0_{s,t}$ indicate the generation amount and amount of load shedding for each scenario $s \in \mathcal{S}$, respectively. For each generator $g \in \mathcal{G}$, $P^{\min}_g$ (or $P^{\max}_g$) represents the minimum (or maximum) amount that can be generated when the generator is on, and t^U_g (or t^D_g) represents the minimum time periods that a generator must be on (or off). In addition, R_g and $\bar{R}_g$ indicate the ramp-up/down and start-up/shut-down ramp limits. The cost parameters $C^{ON}_{g,t}, C^U_{g,t}, C^D_{g,t}, C^V_{g,t}$ and K_t indicate coefficients of fixed, start-up, shut-down, variable generation, and load shedding, respectively. The mathematical formulation of the two-stage risk-averse unit commitment problem is presented below:

$$\min \quad \sum_{g\in\mathcal{G},t\in\mathcal{T}} \left(C^{ON}_{g,t}x_{g,t} + C^U_{g,t}x^U_{g,t} + C^D_{g,t}x^D_{g,t}\right) + \sum_{k\in\mathcal{K}} \tilde{p}_k\eta_k + \frac{1}{1-\alpha}\sum_{s\in\mathcal{S}} p_s v_s \tag{2.12a}$$

$$\text{s.t.} \quad v_s + \eta_k \geq \sum_{g\in\mathcal{G},t\in\mathcal{T}} \left(C^V_{g,t}y_{g,s,t} + K_t y^0_{s,t}\right) \quad \forall s \in S_k, k \in \mathcal{K}, \tag{2.12b}$$

$$\sum_{g\in\mathcal{G}} y_{g,s,t} + y^0_{s,t} \geq d_{s,t} \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{2.12c}$$

$$P^{\min}_g x_{g,t} \leq y_{g,s,t} \leq P^{\max}_g x_{g,t} \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \tag{2.12d}$$

$$y_{g,s,t} \leq y_{g,s,t-1} + R_g x_{g,t-1} + (1 - x_{g,t-1})\bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \tag{2.12e}$$

$$y_{g,s,t} \geq y_{g,s,t-1} - R_g x_{g,t} - (1 - x_{g,t})\bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \tag{2.12f}$$

$$x^U_{g,t} - x^D_{g,t} = x_{g,t} - x_{g,t-1} \quad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{2.12g}$$

$$x_{g,k} \geq x_{g,t} - x_{g,t-1} \quad \forall g \in \mathcal{G}, k \in \{t+1, \ldots, \min\{t + t^U_g - 1, T\}\}, t \in \mathcal{T}, \tag{2.12h}$$

$$1 - x_{g,k} \geq x_{g,t-1} - x_{g,t} \quad \forall g \in \mathcal{G}, k \in \{t+1, \ldots, \min\{t + t^D_g - 1, T\}\}, t \in \mathcal{T}, \tag{2.12i}$$

$$x_{g,t}, x^U_{g,t}, x^D_{g,t} \in \{0,1\},\ y_{g,s,t}, y^0_{s,t}, v_s \geq 0 \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}. \tag{2.12j}$$

In the formulation, the objective function (2.12a) minimizes the total cost consisting of fixed costs and second-stage costs. Constraints (2.12b) represent a relationship

between second-stage risk and cost for each scenario, where the right-hand side consists of variable generation costs and penalty costs for unmet demand. Constraints (2.12c) mean that the generation amount including load shedding should satisfy the load demand for each time period. Constraints (2.12d) restrict the minimum and maximum amounts of generation. Constraints (2.12e) and (2.12f) indicate the ramp-up and down limits of each generator. Constraints (2.12g) demonstrate the logical relationships among the on, start-up, and shut-down variables. Constraints (2.12h) and (2.12i) require the minimum up (down) time periods for each generator whenever on (off). Constraints (2.12j) indicate the domain for each decision variable.

2.5.2 Experimental setup

In the experiments, an illustrative 10-unit system for 24 periods of the planning horizon is used, the data of which is from Kazarlis et al. (1996) with demand scaled by 0.5. A total of 500 scenarios for stochastic demand d are generated from a normal distribution $N(\bar{d}, (\frac{\bar{d}}{4})^2)$, where $\bar{d}$ indicates nominal demand. We consider that the set of scenarios $\mathcal{S}$ has equi-probable scenarios, i.e. $p_s = 1/|\mathcal{S}|$ for $s \in \mathcal{S}$.

The *a priori* partitioning scheme is basically applied for constructing a partition $\mathcal{K}$. Recall that in this scheme, a partition is constructed in advance based on a predefined set of incumbent first-stage solutions $\bar{X}$. In the partitioning problem, we let $\beta = 2\alpha - 1$ for each α so as to make a deviation between β and 1 equal to twice that between α and 1. Therefore, the goal of scenario partitioning in this experiment is to determine a partition $\mathcal{K}$, where the risk level of the corresponding model $\mathrm{PSP}(\mathcal{K}, \alpha)$ is the closest to that of $\mathrm{PSP}(\{\mathcal{S}\}, \beta)$. We also consider the case wherein $|\mathcal{S}|/K \in \mathbb{Z}_+$ and the group sizes are all equal to each other. That is, the number of scenarios for

each group only depends on the number of groups K, i.e. $\mathbb{K}{=}(K, |\mathcal{S}|/K)$.

We also devised two basic partitioning methods for comparison with the proposed partitioning method. The methods use rank function $r(\cdot) : \mathcal{S} \mapsto \mathcal{S}$ to assign each scenario to each group; when the rank of a scenario $s \in \mathcal{S}$ is $r(s)$, the scenario is assigned to the $k(s)$th group with $k(s) := \lceil \frac{r(s)K}{|\mathcal{S}|} \rceil$. As for the two methods, the `rand` method lets $r(s) = s$, which is to say that a group is constructed in random order since a scenario is generated randomly. On the other hand, in the `dagg` method, $r(s)$ is determined by the aggregated demand of each scenario, where the corresponding value is $||d_s||$, the distance is ℓ_1-norm, and the rank is determined by the relative position of each scenario s: i.e. by sorting each scenario according to demand. A partition made by `rand` is also used as the initial partition in the proposed partitioning methods, and we use *move-first* to implement the methods.

Various numbers of groups in $\{1, 5, 10, 20\}$ and α values in $\{0.7, 0.8, 0.9\}$ are tested. The time limit is set to 3,600 seconds for each combination of (α, β) and K. For each combination, the averaged value among five replicated sets of scenarios is reported. All the models and algorithms are implemented with C++ using CPLEX 20.1 as a mixed-integer program solver with its default parameter setting. Finally, all the computational experiments were conducted on an Intel i7-8700 3.20 gigahertz machine with 32 gigabytes RAM.

2.5.3 Experimental results

Effectiveness of proposed partitioning methods

We analyze the effectiveness of the four partitioning methods, including our two proposed methods, in *a priori* partitioning scheme. We use $\bar{X} = X^{EV}$ and $\bar{X} =$

X^{WS}, where X^{WS} consists of an incumbent first-stage solution x^{WS} with a fictitious worst-case scenario $\hat{s}$, where the scenario is defined as $(q_{\hat{s}}, T_{\hat{s}}, W_{\hat{s}}, h_{\hat{s}}) := (\max_{s\in\mathcal{S}} q_s, \max_{s\in\mathcal{S}} T_s, \max_{s\in\mathcal{S}} W_s, \max_{s\in\mathcal{S}} h_s)$. In other words,

$$x^{WS} := \underset{x\in X}{\operatorname{argmin}} \left\{ c^T x + Q(x, \hat{s}) \right\}.$$

The methods, with two sets of $\bar{X}$s, are compared with two other methods: `dagg` and `rand` mentioned in Section 2.5.2. Table 2.1 reports the relative objective value (`relobj`), the computation time (`time`), and the number of iterations (`iter`) in the partitioning algorithm along with the actual deviation (`dev`) for all of the partitioning methods. Among them, `relobj` is calculated as

$$\texttt{relobj} := \frac{D_{\mathcal{K}_{best}}}{D_{\mathcal{K}_0}},$$

where $\mathcal{K}_{best}$ is a partition obtained from Algorithm 2.2, and $\mathcal{K}_0$ is an initial partition, which indicates how the algorithm can reduce the maximum deviation ($D_{\mathcal{K}}$) from the initial value. In addition, `dev` is calculated as $|R_{\mathcal{K}} - R_t| / R_t$, where $R_{\mathcal{K}} = \sum_{k\in\mathcal{K}} p_k$ $\text{CVaR}_\alpha(\boldsymbol{Q}_k(x))$ is defined as the actual second-stage risk level, and R_t=$\text{CVaR}_\beta(\boldsymbol{Q}(x))$ indicates the actual target risk as a value of the risk function with β.

First of all, the proposed partitioning methods show a low `relobj` value in a reasonable time, which is nearly close to zero. Except for the cases when $\bar{X} = X^{WS}$ and $(\alpha, K) = (0.7, 5)$ or (0.7,10), the proposed partitioning methods show similar performance for `relobj`, `dev`, and `time`, where in those cases $\bar{X} = X^{WS}$ shows worse performance. Overall, $\bar{X} = X^{EV}$ has stable performance overall in terms of `dev` and computation time, and $\bar{X} = X^{WS}$ shows the lowest `dev` when $\alpha = 0.8$ and 0.9. Both

Table 2.1: Partitioning statistics for four partitioning methods

α	K	Proposed partitioning methods $\bar{X} = X^{EV}$				Proposed partitioning methods $\bar{X} = X^{WS}$				dagg	rand
		relobj	time (s)	iter	dev (%)	relobj	time (s)	iter	dev (%)	dev (%)	dev (%)
0.7	5	1.6×10^{-6}	127.7	534.2	0.89	29.1×10^{-2}	795.5	1,002.6	1.31	1.12	2.72
	10	3.4×10^{-6}	43.5	728.2	0.69	46.6×10^{-2}	242.9	1,350.2	2.03	1.61	2.67
	20	0.6×10^{-6}	12.9	474.6	1.15	0.4×10^{-6}	12.8	461.0	0.23	1.87	2.57
	avg	1.9×10^{-6}	61.4	579.0	0.91	25.2×10^{-2}	350.4	937.9	1.19	1.53	2.65
0.8	5	8.3×10^{-6}	82.0	242.6	0.76	4.8×10^{-6}	81.9	224.8	0.09	2.37	2.10
	10	2.6×10^{-6}	31.7	280.6	0.87	1.9×10^{-6}	31.6	253.0	0.09	2.96	2.02
	20	1.9×10^{-6}	13.2	252.6	0.86	3.5×10^{-6}	13.1	254.8	0.09	3.26	1.93
	avg	4.3×10^{-6}	42.3	258.6	0.83	3.4×10^{-6}	42.2	244.2	0.09	2.87	2.02
0.9	5	9.8×10^{-6}	87.2	127.0	0.65	29.1×10^{-6}	88.1	105.0	0.04	3.79	1.56
	10	14.3×10^{-6}	32.8	102.4	0.53	15.3×10^{-6}	33.4	103.4	0.04	4.51	1.42
	20	13.2×10^{-6}	13.6	103.4	0.56	7.5×10^{-6}	13.6	90.2	0.01	4.90	1.21
	avg	12.4×10^{-6}	44.5	110.9	0.58	17.3×10^{-6}	45.0	99.5	0.03	4.40	1.40

methods show much lower actual deviation (`dev`) than do the other two methods (`dagg` and `rand`). For the other two methods, `dagg` shows better deviation values than `rand` when $\alpha = 0.7$, but the performance gets worse as α increases. It indicates making a partition with similar scenarios has difficulty in representing risk-averse objectives. On the other hand, the deviation values of `rand` naturally decrease as α increases, since the values between α and β get smaller when α increases.

Effectiveness of proposed solution approach

The performance of the proposed solution approach, the primal implementation of the column-and-constraint-generation method in Section 2.3.1, is compared with two other methods: one is the dual implementation of column-and-constraint generation (see Section 2.3.2), the other is solving extensive formulation (MP=RMP($\mathcal{S}$)) by commercial solver CPLEX ('CPLEX'). The computational times (`time`) are reported for all solution approaches, and the total number of iterations (`iter`), the total number of added scenarios (`totscn`), and the total computation time (`time`) are

additionally reported for the column-and-constraint-generation methods, with the Γ basically set to 1.

The results in Table 2.2 show the computational aspects of the three solution approaches. First of all, both the column-and-constraint-generation methods show less computation time than CPLEX for all α and K combinations. The difference between them increases for higher α values, since the computation times of the former decline as α gets larger, while those of the latter slightly increase. It shows the advantages of column-and-constraint generation methods for higher α values. On the other hand, the effect of K on computation time is not clear for any of the methods. As for the two column-and-constraint generation methods, although the number of scenarios added is nearly the same, the primal implementation shows a much smaller number of iterations than the dual implementation. This leads faster convergence of the algorithm, and it also shows why the proposed algorithm is more efficient than the dual implementation.

Effect of Γ on proposed solution approach

The computational performance of the proposed solution approach with various Γ values in $\{0.6, 0.8, 1.0, 1.2, 1.4\}$ is evaluated. Recall that the parameter Γ controls the number of scenarios that can be added in one iteration. We also record the total number of iterations (`iter`), the total number of added scenarios (`scn`), and the computation time (`time`), as previously. Since the algorithm is not exact when $\Gamma < 1$, as mentioned in Section 2.3.1, we additionally report the `gap` for $\Gamma < 1$ to

Table 2.2: Computational performance of three solution approaches

α	K	column-and-constraint generation primal implementation			column-and-constraint generation dual implementation			CPLEX
		`iter`	`totscn`	`time` (s)	`iter`	`totscn`	`time` (s)	`time` (s)
0.7	1	3.6	163.4	361.0	4.2	163.4	576.4	987.9
	5	4.0	172.8	504.4	5.2	173.4	976.5	1,279.5
	10	4.2	181.4	608.0	5.4	182.2	1,133.5	1,161.7
	20	4.4	184.4	652.4	4.8	183.0	740.3	1,218.4
	avg	4.1	175.5	531.5	4.9	175.5	856.7	1,161.9
0.8	1	4.2	113.6	315.1	5.0	113.4	526.0	1,040.5
	5	4.2	124.0	362.3	5.6	124.0	834.9	1,241.1
	10	4.2	123.6	401.6	5.0	123.6	769.2	1,401.2
	20	4.0	120.2	319.6	4.2	120.2	372.6	1,335.7
	avg	4.2	120.4	349.7	5.0	120.3	625.7	1,254.6
0.9	1	4.0	68.2	157.8	6.4	68.8	770.7	1,248.4
	5	4.2	77.2	226.5	6.2	81.0	810.5	1,345.8
	10	4.2	77.4	221.3	5.2	77.6	556.4	1,545.0
	20	4.4	85.4	261.8	5.0	84.6	409.6	1,404.2
	avg	4.2	77.1	216.9	5.7	78.0	636.8	1,385.9

evaluate the quality of the upper bound, where the value is calculated as

$$\texttt{gap} = \frac{z(\Gamma) - z(1)}{z(1)},$$

where $z(\Gamma)$ is the optimal objective value from the proposed column-and-constraint generation with Γ. Table 2.3 shows the computational aspects of the proposed solution approach for various Γ values. First of all, the total number of scenarios is proportional to Γ, whereas the number of iterations has a negative relationship with Γ. Compared with $\Gamma = 1$, computational gain by fewer iterations is not obvious for $\Gamma \in \{1.2, 1.4\}$, since more scenarios added at each iteration incur a computational burden in solving the restricted master problem. On the other hand, the computation times are much reduced for $\Gamma \in \{0.6, 0.8\}$, and a significantly low optimality gap (`gap`), less than 10^{-4} for nearly all combinations, is observed. Especially, one can obtain near-optimal solutions, whose average optimality gap is 6.2×10^{-5}, within 50% of the computation time required in the cases where $\Gamma = 1$. The results summarize that one can efficiently obtain good quality solutions by setting Γ less than 1.

Effectiveness of adaptive partitioning scheme

We lastly analyze the computational aspects of the adaptive partitioning scheme by comparing them with the *a priori* partitioning scheme. Partitions in the *a priori* partitioning scheme are constructed by solving $\mathcal{P}(\bar{X})$ with $\bar{X} = X^{EV}$. Table 2.4 shows the computational performance of the two schemes, including the objective value (`objval`), the number of iterations (`iter`), the number of added scenarios

Table 2.3: Computational performance of proposed column-and-constraint generation algorithm for various Γ values

α	K	Γ=0.6				Γ=0.8				Γ=1			Γ=1.2			Γ=1.4		
		iter	scn	gap ($\times 10^{-5}$)	time (s)	iter	scn	gap ($\times 10^{-5}$)	time (s)	iter	scn	time (s)	iter	scn	time (s)	iter	scn	time (s)
0.7	1	4.2	105.2	14.3	138.5	4.2	133.0	0.0	230.0	3.6	163.4	369.0	3.4	191.0	429.2	3.0	219.8	495.0
	5	4.4	119.6	11.7	206.5	4.2	143.2	6.4	278.1	4.0	174.2	557.7	3.8	206.8	670.2	3.4	234.0	696.3
	10	4.2	121.0	11.6	189.2	4.4	150.6	0.0	367.9	4.2	181.8	556.5	3.4	207.6	601.8	3.2	234.6	691.0
	20	4.2	131.0	12.8	234.4	4.4	151.0	0.0	351.0	4.2	185.8	620.3	4.0	206.0	772.7	3.6	247.4	802.7
	avg	4.3	119.2	12.6	192.1	4.3	144.5	1.6	306.8	4.0	176.3	525.9	3.7	202.9	618.4	3.3	234.0	671.2
0.8	1	4.0	77.6	10.9	85.5	4.2	95.2	0.0	160.0	4.2	113.6	316.3	3.4	132.2	296.3	3.4	152.6	339.3
	5	4.2	83.6	0.0	103.7	4.2	104.2	0.0	202.8	4.2	125.4	370.5	3.4	139.8	338.0	3.4	159.4	412.8
	10	4.4	87.6	0.0	143.8	4.2	107.2	0.0	239.0	4.2	125.0	351.6	3.4	142.8	332.5	3.6	159.4	534.6
	20	4.2	86.0	0.0	132.2	4.0	102.0	0.0	202.0	4.0	121.2	348.0	3.6	141.8	352.3	3.4	160.2	406.7
	avg	4.2	83.7	2.7	116.3	4.2	102.2	0.0	200.9	4.2	121.3	346.6	3.5	139.2	329.8	3.5	157.9	423.4
0.9	1	4.2	51.4	1.1	64.5	4.4	61.0	1.1	109.1	4.0	68.2	155.0	3.4	76.2	155.0	3.4	83.6	185.4
	5	4.6	62.0	8.5	110.0	5.0	73.6	0.0	200.7	4.2	77.6	226.9	3.8	84.0	219.7	3.6	93.6	240.3
	10	4.6	65.0	0.0	117.2	4.6	72.4	0.4	176.8	4.2	78.2	217.3	3.6	83.8	196.0	3.4	89.8	210.6
	20	4.4	65.6	4.0	128.9	4.4	65.6	4.0	130.1	4.2	82.8	243.0	4.2	82.8	242.0	3.8	100.0	290.1
	avg	4.5	61.0	3.4	105.2	4.6	68.2	1.4	154.2	4.2	76.7	210.6	3.8	81.7	203.2	3.6	91.8	231.6

(`totscn`), the actual deviation (`dev`) and computation time (`time`). For the *a priori* partitioning scheme, we separate the computation time into the time for solving the model (`stime`) and the time for constructing a partition (`ptime`).

Based on the table data, the performance of the adaptive partitioning scheme relative to that of *a priori* partitioning scheme is analyzed. First of all, the objective values from the schemes are similar and those of adaptive partitioning being slightly lower, which implies that a partition from adaptive partitioning scheme is less risk-averse. The numbers of iterations and added scenarios are larger for adaptive partitioning, which incurs a greater computational burden, even when taking partitioning time for a priori partitioning (`ptime`) into account. This is due to the fact that the re-constructing phase takes more computation time whenever an incumbent solution is added at each iteration. However, the deviation from adaptive partitioning is lower in its *a priori* counterpart; especially, it has the lowest `dev` when $\alpha = 0.8$ or 0.9, which have shown exactly zeros for those cases. To sum up, in order to take proper advantage of the adaptive partitioning scheme, it needs to be elaborated in terms of computational time, since constructing a partition becomes

Table 2.4: Overall statistics of two partitioning schemes

α	K	adaptive partitioning					a priori partitioning					
		objval	iter	totscn	time (s)	dev (%)	objval	iter	totscn	ptime (s)	stime (s)	dev (%)
0.7	5	303,753.8	4.2	248.2	2,589.4	1.33	308,992.6	4.0	172.8	127.7	529.5	0.89
	10	302,059.4	4.2	269.6	1,635.7	2.05	308,598.2	4.2	181.4	43.5	597.8	0.69
	20	306,890.2	4.2	282.0	1,598.8	0.00	309,700.4	4.4	184.4	12.9	643.5	1.15
	avg	304,234.5	4.2	266.6	1,941.3	1.12	309,097.1	4.2	179.5	61.4	590.3	0.91
0.8	5	311,351.4	4.4	189.6	1,854.9	0.00	313,167.0	4.2	124.0	82.0	379.6	0.76
	10	311,351.4	4.6	192.8	1,247.8	0.00	313,432.0	4.2	123.6	31.7	411.3	0.87
	20	311,354.8	4.4	202.0	1,028.1	0.00	313,416.0	4.0	120.2	13.2	316.2	0.86
	avg	311,352.5	4.5	194.8	1,376.9	0.00	313,338.3	4.1	122.6	42.3	369.0	0.83
0.9	5	316,757.4	4.6	118.8	1,758.7	0.00	318,282.0	4.2	77.2	87.2	228.7	0.65
	10	316,695.2	5.2	122.2	1,264.0	0.00	318,071.2	4.2	77.4	32.8	228.3	0.53
	20	316,757.4	5.6	128.8	930.5	0.00	318,078.2	4.4	85.4	13.6	266.5	0.56
	avg	316,736.7	5.1	123.3	1,317.8	0.00	318,143.8	4.3	80.0	44.5	241.1	0.58

more burdensome as the number of iterations increases.

2.6 Summary

In this chapter, we presented a novel risk-averse two-stage stochastic program with finite support, which is based on partitioning the set of scenarios. It is a generalized model that can represent various models with finite support in the literature. We proposed an efficient and generic solution approach, which is based on column-and-constraint generation, to solve the model with a given partition. In addition, we devised a partitioning algorithm to construct a partition whose risk is closest to the pre-defined target. We also present partitioning schemes on how to consolidate the proposed solution approach and partitioning algorithms, including a novel adaptive partitioning scheme where a partition is iteratively re-constructed within the column-and-constraint generation procedure. Numerical experiments are conducted to show the effectiveness of the proposed partitioning methods, as indicated by the significantly low deviation from the target compared to other existing methods. Further,

the proposed column-and-constraint generation is shown to be efficient, compared to solving extensive formulations or extensions of the method in the literature. In addition, the practical usefulness of a controllable parameter Γ also is demonstrated that the models can be solved much faster for $\Gamma < 1$ with a significantly small optimality gap. Finally, the computational aspects of the adaptive partitioning scheme show its potential utility over an *a priori* partitioning scheme.

Chapter 3

Single unit commitment under uncertainty and its application to unit decomposition approaches

3.1 Introduction

Single-unit commitment (1UC) problem is a problem of deciding the generation schedule of a single generator in electric power systems. It usually arises in a deregulated electricity market that allows a variety of individual power producers (IPPs) bid their generator's schedules (MISO, 2022). It has been observed that two major bidding strategies are possible: *self-scheduling* and *self-commitment* (Pan and Guan, 2016). In the *self-commitment* approach, an IPP decides only the commitment states of a generator, whereas generation amounts are also determined by an IPP in the *self-scheduling* approach. In both approaches, the goal of an IPP is to maximize its net profit under the given electricity price is given, considering the operational requirements of an internal generator. The 1UC problem has widely been studied in the literature since it also has a critical role in solving the unit commitment (UC) problem, which is the problem solved by the system operator minimizing total operation cost while satisfying system-wise load demand by coordinating *multiple* generators. Analyses on the single-generator system have been successful in efficiently solving UC problems (e.g., Ostrowski et al., 2011; Yan et al., 2020). In

addition, a well-known solution approach called *unit decomposition* (e.g., Frangioni et al., 2008; Van Ackooij et al., 2018) directly uses solutions approaches for the 1UC problem. It is a method that relaxes system-wise requirements such as demand balance constraints and decomposes the remaining problem by each generator.

In the literature, various types of generators are considered in 1UC problems, such as combined-cycle (Pan and Guan, 2017; Papavasiliou et al., 2014), or pumped-hydro storage generators (Qu et al., 2022). Among them, thermal generators have been widely focused on 1UC problems. Lee et al. (2004) proposed valid inequalities for minimum up/down polytope where on/off decisions exist, further showing that they are sufficient to describe the convex hull of the polytope. Later, Rajan and Takriti (2005) extended the research to provide valid inequalities when start-up decisions and corresponding costs also exist. Morales-España et al. (2015) present valid inequalities that can describe the convex hull under the power generation limit and minimum up/down requirements. Using the results, it is shown that the self-commitment problem can be solved in a polynomial time. For the self-scheduling problem, where the generation amount is also determined, generation ramping requirements, which restrict the increase or decrease of generation amount in two adjacent time periods, make the problem not trivial to solve. Various studies including valid inequalities (e.g., Ostrowski et al., 2011; Damcı-Kurt et al., 2016) and a systematic formulation tightening approach (Yan et al., 2020) are proposed by focusing on the single-generator system with ramping requirements. However, it has been shown that the problem also can be solved in a polynomial time even with the ramping restrictions. Methods that directly solve self-scheduling problems are mostly based on dynamic programming approaches. Fan et al. (2002) provided a

dynamic programming-based polynomial-time algorithm to solve the self-scheduling problem with piece-wise linear variable cost. Then, Frangioni and Gentile (2006b) proposed a $\mathcal{O}(T^3)$-time algorithm with a convex quadratic variable cost function, where T is the number of periods in the planning horizon. The algorithm is based on constructing a state-space graph indicating on/off status and deciding on/off status by solving the shortest path problem. From then, several further works are made based on the algorithm. In Frangioni and Gentile (2015) and Guan et al. (2018), the complexity of obtaining commitment decisions is improved by refining the state-space graph. In addition, extended formulations that can provide integral solutions are proposed (Frangioni and Gentile, 2015; Guan et al., 2018; Knueven et al., 2018). Further, a refined algorithm that can reduce the computational burden is proposed in Wuijts et al. (2021). In addition, Frangioni et al. (2008) applied the Lagrangian relaxation approach to solve the Uc problem with hydro-thermal generators by unit decomposition method, where the algorithm in Frangioni and Gentile (2006b) is used to solve the decomposed self-scheduling problem.

Although the above-mentioned studies have made considerable contributions to solving 1UC problems, the given electricity price is considered certain in advance. However, since increasing penetration of renewable energy makes the electricity price uncertain and volatile (e.g., Pan and Guan, 2016), an IPP needs to consider it when submitting the offer. Regarding this, few studies considered electricity price uncertainty in 1UC problems. For the stochastic price, Pan and Guan (2016) proposed facet-defining inequalities for multi-stage stochastic optimization model, and Guan et al. (2018) presented a dynamic programming algorithm that extends the results in the deterministic counterpart. Lu et al. (2022) proposed an extended formulation

that can describe a convex hull in a two-stage stochastic 1UC problem. In addition, we note that earlier studies have utilized unit decomposition methods for the UC problem when net load demand is uncertain, which has the subproblem structure as stochastic 1UC problems. In Carpentier et al. (1996), an augmented Lagrangian approach is proposed. In addition, stochastic Lagrangian relaxation methods (Nowak and Römisch, 2000), and column generation methods (Shiina and Birge, 2004) are also proposed.

In this chapter, we study 1UC problems with stochastic electricity prices and extend our analysis on 1UC to efficiently deal with stochastic unit commitment problems under net load uncertainty using unit decomposition methods. Our motivation is on aspects that have not been sufficiently addressed in the existing literature. First, although few studies have proposed to solve the stochastic self-scheduling problems, the efficiency is not evaluated (Guan et al., 2018) or the algorithms are not enough to deal with a large number of scenarios (Pan and Guan, 2016). Therefore, we propose an efficient implementation to deal with many scenarios, where only the generation amount is adaptable to the electricity price realization. Further, we refine an algorithm to solve the self-commitment problem consisting of minimum up/down requirements, whose complexity is $\mathcal{O}(T)$, which is advantageous over the previous algorithms in the literature (e.g., Guan et al., 2018; Frangioni and Gentile, 2015).

Next, using the results, we propose unit decomposition approaches to efficiently deal with the stochastic unit commitment problem under net load uncertainty. Note that although some earlier studies (e.g., Carpentier et al., 1996; Nowak and Römisch, 2000; Shiina and Birge, 2004) have investigated unit decomposition approaches on stochastic unit commitment, ramping requirements are ignored in the problem,

which makes the analysis on the subproblem structure or the computation much easier. Thus, solution approaches in unit decomposition framework under generation ramping requirements are worthwhile to be further investigated. In this study, we present two unit decomposition frameworks: *schedule decomposition* and *commitment decomposition*. Each of the frameworks corresponds to the resulting subproblem structure. The first one, which we call *schedule decomposition*, utilizes the stochastic self-scheduling problem as a substructure and is a straightforward decomposition that is widely used in deterministic settings. On the other hand, the other one which we denote *commitment decomposition*, which uses the self-commitment problem as a substructure, is a novel decomposition that has rarely been considered in the literature. Further, we propose two solution approaches for each decomposition framework, which are based on Lagrangian relaxation and column generation.

The contribution of the chapter can be summarized as follows.

- First, we develop efficient dynamic programming algorithms for 1UC problems. For the stochastic self-scheduling problem, we extend the dynamic programming approach which was developed to solve the deterministic counterpart. It is based on proposing another dynamic programming approach in the substructure by characterization of an optimal solution, to efficiently deal with a number of scenarios. Furthermore, for the self-commitment problem, we propose a dynamic programming algorithm that has an enhanced computational complexity compared to the previous works.

- Next, by using the proposed algorithms on 1UC problems, we propose two unit decomposition frameworks for the UC problem under net load uncertainty. The *schedule decomposition* method utilizes the substructure as the

self-scheduling problem, while the *commitment decomposition* method which utilizes the self-commitment problem as a substructure is a novel approach that has not been proposed. Lagrangian relaxation and column generation-based solution approaches are proposed for each decomposition method, and we also compare theoretical dual bounds from the approaches.

- Through extensive computational experiments, we first conduct algorithms on 1UC problems to show the efficiency of the proposed algorithms when the number of scenarios or the time periods increase. Next, we compare a total of four solution approaches to solve the UC problem with the stochastic net load. Finally, we emphasize the efficiency of the commitment decomposition with a column generation-based solution approach, by comparing the extensive formulation approach as the number of scenarios increases.

The remainder of the chapter is written as follows. In Section 3.2, we present dynamic programming algorithms for the stochastic self-scheduling and self-commitment problems. Next, in Section 3.3, we propose two unit decomposition methods for the UC problem under stochastic net load, along with the Lagrangian relaxation and column generation-based solution approaches. In Section 3.4, we conduct computational experiments to demonstrate the efficiency of our approach to 1UC problems and unit decomposition approaches. In Section 3.5, we conclude the chapter by summarizing the results.

3.2 Algorithms for the single-unit commitment problem under price uncertainty

In this section, we study two types of 1UC problems, especially stochastic self-scheduling and self-commitment problems. We present notations to describe the mathematical formulations in Table 3.1. Note that in this section we omit generator information in decision variables and parameters since we focus on a specific generator when dealing with 1UC problems. With a slight abuse of notation, we will extend the dimensions of variables and parameters to indicate generator information in the following sections which deal with a set of generators. In addition, we denote $[a, b] := \{t \in \mathbb{Z} \mid a \leq t \leq b\}$ for two integers a and b and we use boldface letter to represent vectors. Furthermore, throughout the section, we represent the 1UC problems as the minimization form where the profit is regarded as the negative cost, for convenience and consistency with the UC problems in the following section.

3.2.1 Dynamic programming algorithm for the stochastic self-scheduling problem

We address the stochastic self-scheduling problem where the uncertain electricity prices for a given planning horizon are given in advance with a set of scenarios $\mathcal{S}$. Since the commitment states are hard to modify in real-time operation, we focus on the two-stage setting where the commitment decisions are determined before the realization of uncertain prices, and the generation amounts are decided according to the realization. For each scenario $s \in \mathcal{S}$, $(r_{s,t})_{t\in\mathcal{T}}$ represents electricity prices and the probability of each scenario is p_s. For convenience, we let $b_{s,t} := C_t^V - r_{s,t}$ be a linear net cost coefficient for a unit generation amount for $s \in \mathcal{S}, t \in \mathcal{T}$. The stochastic

Table 3.1: Nomenclature

Sets and Indices	
$\mathcal{T}$	set of time periods, $t \in \mathcal{T} = \{1, 2, \ldots, T\}$
$\mathcal{S}$	set of scenarios, $s \in \mathcal{S} = \{1, 2, \ldots, S\}$
Parameters	
C_t^V	variable generation cost at period t
C_t^{ON}	fixed generation cost at period t
C_t^U	start-up cost at period t
C_t^D	shut-down cost at period t
p_s	probability of scenario s
$r_{s,t}$	electricity price for scenario s at period t
ρ_t	commitment profit at period t
$P^{\min}$ ($P^{\max}$)	minimum (maximum) generation limit
R	ramp-up/down limit
$\bar{R}$	start-up/shut-down ramp limit ($\bar{R} \geq P^{\min}$)
t^U (t^D)	minimum up (down) time
Decision Variables	
x_t	1 if a generator is on at period t, 0 otherwise
x_t^U	1 if a generator is starting up at period t, 0 otherwise
x_t^D	1 if a generator is shutting down at period t, 0 otherwise
$y_{s,t}$	generation amount of scenario s at period t

self-scheduling problem minimizes the expected net cost for all scenarios by deciding each scenario's generation amount. The mathematical formulation of the problem can be written as follows:

$$\min \quad \sum_{t\in\mathcal{T}} \left(C_t^{ON} x_t + C_t^U x_t^U + C_t^D x_t^D\right) + \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s b_{s,t} y_{s,t} \tag{3.1a}$$

$$\text{s.t.} \quad x_k \geq x_t - x_{t-1} \quad \forall k \in \{t+1,\ldots,\min\{t+t^U-1,T\}\}, t\in\mathcal{T} \tag{3.1b}$$

$$1 - x_k \geq x_{t-1} - x_t \quad \forall k \in \{t+1,\ldots,\min\{t+t^D-1,T\}\}, t\in\mathcal{T} \tag{3.1c}$$

$$x_t^U - x_t^D = x_t - x_{t-1} \quad \forall t\in\mathcal{T} \tag{3.1d}$$

$$P^{\min} x_t \leq y_{s,t} \leq P^{\max} x_t \quad \forall s\in\mathcal{S}, t\in\mathcal{T} \tag{3.1e}$$

$$y_{s,t} \leq y_{s,t-1} + R x_{t-1} + \bar{R}(1 - x_{t-1}) \quad \forall s\in\mathcal{S}, t\in\mathcal{T} \tag{3.1f}$$

$$y_{s,t-1} \leq y_{s,t} + R x_t + \bar{R}(1 - x_t) \quad \forall s\in\mathcal{S}, t\in\mathcal{T} \tag{3.1g}$$

$$x_t, x_t^U, x_t^D \in \{0,1\}, y_{s,t} \geq 0 \quad \forall s\in\mathcal{S}, t\in\mathcal{T} \tag{3.1h}$$

The objective function (3.1a) is the expected cost which consists of commitment and variable generation costs. Constraints (3.1b) and (3.1c) indicate minimum up and down requirements when the generator starts up or shuts down. Constraints (3.1d) demonstrate the logical relationship between on/off and start-up/shut-down variables. Next, constraints (3.1e) restrict the range of generation when a generator is on. Lastly, constraints (3.1f) (or (3.1g)) indicate the ramping constraints that limit the maximum difference when the generation amount increases (or decreases), where x_0 and $y_{s,0}$ denote initial states.

Although the problem is a mixed-integer linear programming (MIP) problem that can be directly solved by a commercial solver, the computational burden in-

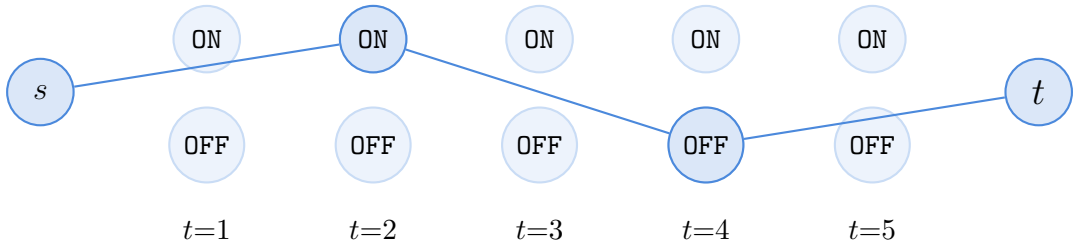


Figure 3.1: Illustration of a state-space graph on obtaining commitment decisions

creases as the number of scenarios increases. In this section, we devise a dynamic programming algorithm to solve the problem, which extends widely-used dynamic programming approaches (e.g., Frangioni and Gentile, 2006b; Frangioni and Gentile, 2015). The idea is based on constructing a state-space graph where each node indicates a generator's on/off status and represents the problem to the shortest path problem on the graph, as illustrated in Figure 3.1. For each period $t \in \mathcal{T}$, two types of nodes (states) are defined: $\texttt{ON}_t$, $\texttt{OFF}_t$. The node $\texttt{ON}_t$ indicates that a generator starts up at period t, which means the generator is off at period $t-1$. Similarly, the node $\texttt{OFF}_t$ indicates that a generator becomes off right after the period t, meaning that it is on until period t and turns off at the period $t+1$. Further, arcs between different types of nodes can only be constructed, and when the minimum up/down requirements are satisfied. For $h, k \in \mathcal{T}$, an arc from the node $\texttt{ON}_h$ to node $\texttt{OFF}_k$ can be constructed only if $k \geq h + t^U$. Similarly, an arc from node $\texttt{OFF}_h$ to node $\texttt{ON}_k$ can be constructed only if $k \geq h + t^D + 1$. Since the number of nodes is $\mathcal{O}(T)$ and that of arcs is $\mathcal{O}(T^2)$ in a dynamic acyclic graph, the optimal cost of the problem can be calculated in $\mathcal{O}(T^2)$ once all arc costs are calculated in advance. One of the important tasks is to calculate all the costs of arcs representing on-periods efficiently. For an arc from node $\texttt{OFF}_h$ and $\texttt{ON}_k$ where $k \geq h + t^D + 1$, since the transition indicates

the generator starts up at period k, the corresponding cost is C_k^U. On the other hand, for an arc from node $\texttt{ON}_h$ and $\texttt{OFF}_k$ where $k \geq h + t^U - 1$, since the generator is on for the periods $[h, k]$, the corresponding arc cost is the summation of fixed generation cost (which is $\sum_{t\in[h,k]} C_t^{ON}$), and the expected variable generation cost for $[h, k]$. The expected variable generation cost for $[h, k]$ is an expectation of each for all scenarios, which is $\sum_{s\in\mathcal{S}} p_s z^D(s, h, k)$. Each cost $z^D(s, h, k)$ is defined as an optimal cost of solving problem $\mathrm{D}(s, h, k)$ defined as follows.

$$
\begin{aligned}
z^D(s,h,k) \;:=\; \min \quad & \sum_{t\in[h,k]} b_{s,t} w_t \\
\text{s.t.} \quad & P^{\min} \leq w_t \leq P^{\max} \quad \forall t \in [h,k], \\
& w_h \leq \bar{R}, \\
& w_k \leq \bar{R}, \\
& w_t - R \leq w_{t+1} \leq w_t + R \quad \forall t \in [h,k-1].
\end{aligned}
$$

We note that the problem can be easily solved by a linear programming (LP) solver. However, it may be undesirable when dealing with all $[h, k]$ pairs for $h, k \in \mathcal{T}$ and all of the scenarios in $\mathcal{S}$. Thus, we propose another dynamic programming algorithm for $\mathrm{D}(s, h, k)$, which is efficient and advantageous when calculating the costs for all pairs $[h, k]$ for $h, k \in \mathcal{T}$, further helpful when solving the problem for a number of scenarios. Our dynamic programming approach is based on characterizing optimal solutions of $\mathrm{D}(s, h, k)$, which has been mentioned in the literature. The following proposition can be derived based on the results in Guan et al. (2018) that there exists an optimal solution in a set of a finite number of generation amounts.

Proposition 3.1. *Let $\mathcal{Q}$ be a set of generation amounts that each point $q \in \mathcal{Q}$ satisfies $P^{\min} \leq q \leq P^{\max}$, and equals to the one of those: $P^{\min}+nR, P^{\max}-nR, \bar{R}+nR, \bar{R}-nR$, where $n \in \mathbb{Z}_+$. Then, there exists an optimal solution $(w_t)_{t\in[h,k]}$ that satisfies $w_t \in \mathcal{Q}$ for all time periods $t \in [h,k]$.*

The result implies an efficient dynamic programming approach to solve $\mathrm{D}(s,h,k)$ for given h and $k \in \mathcal{T}$ for any given scenario $s \in \mathcal{S}$. For each node $q \in \mathcal{Q}$, we denote a set $\mathcal{Q}(q)$ be a set of points that can be connected, in terms of satisfying ramping requirements, to point q, i.e. $\mathcal{Q}(q) := \{q' \in \mathcal{Q} \mid q - R \leq q' \leq q + R\}$. We denote an optimal cost when the generation amount is q at the period k by $z'(s,h,k,q)$. Then, for a given scenario $s \in \mathcal{S}$ and $h \in \mathcal{T}$, the recursion can be written as follows,

$$\begin{aligned} z'(s,h,h,q) =& b_{s,h}q && \forall q \in \mathcal{Q}, \\ z'(s,h,k,q) =& \min_{q'\in\mathcal{Q}(q)} \left\{z'(s,h,k-1,q') + b_{s,k}q'\right\} && \forall k \in [h+1,T], q \in \mathcal{Q}. \end{aligned} \tag{3.2}$$

Then, the optimal cost $z^D(s,h,k)$ becomes the minimum cost of $z'(s,h,k,q)$ that the $q \in \mathcal{Q}$ satisfies the ramping constraint in the last period k as follows.

$$z^D(s,h,k) = \min_{q\in\mathcal{Q}} \left\{z'(s,h,k,q) \mid q \leq \bar{R}\right\} \quad \forall k \in [h,T] \tag{3.3}$$

Using the recursion, $z^D(s,h,k)$ can be obtained by solving dynamic programs, as illustrated in Figure 3.2. The recursion also implies that it is easier to induce the optimal cost $z^D(s,h,k)$ from the $z^D(s,h,k-1)$ when using the dynamic programming procedure. Combining the recursions (3.2) and (3.3), $z^D(s,h,k)$ can be obtained while calculating $\{z'(s,h,k,q)\}_{q\in\mathcal{Q}}$ from $\{z'(s,h,k-1,q)\}_{q\in\mathcal{Q}}$. It demonstrates that

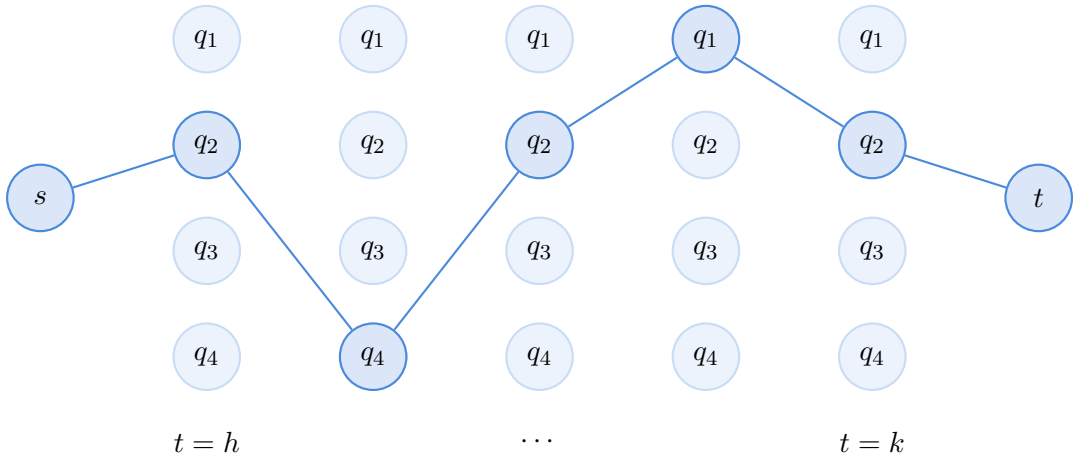


Figure 3.2: Illustration of a dynamic programming approach to solve $\mathrm{D}(s, h, k)$

one can obtain a series of optimal costs $\{z^D(s, h, k)\}_{k\in[h,T]}$ by doing only one recursion from $k = h$ to $k = T$, which is similar to the implementation in Frangioni and Gentile (2006b) for the deterministic setting. It means that the advantage of the dynamic programming approach increases when solving multiple dispatch problems with all pairs (h, k) for h and k in $\mathcal{T}$.

3.2.2 Dynamic programming algorithm for the self-commitment problem

Next, we consider the self-commitment problem, where the on/off status of a generator for the planning horizon is determined to maximize the revenue. In the problem, an IPP gets a constant profit ρ_t for each time period $t \in \mathcal{T}$ when the generator is on, regardless of the amount of generation. Since a constant generation cost when a generator is on is C_t^{ON}, we can denote the net cost as $C_t := C_t^{ON} - \rho_t$ for each period $t \in \mathcal{T}$. The objective is to minimize the net profit for the planning horizon by deciding on/off status satisfying minimum up/down requirements. The self-commitment

problem can be formulated as follows.

$$
\begin{aligned}
\min \quad & \sum_{t\in\mathcal{T}} \left(C_t x_t + C_t^U x_t^U + C_t^D x_t^D\right) \\
\text{s.t.} \quad & x_t^U - x_t^D = x_t - x_{t-1} \quad \forall t \in \mathcal{T}, \\
& x_k \geq x_t - x_{t-1} \quad \forall k \in \{t+1, \ldots, \min\{t + t^U - 1, T\}\}, t \in \mathcal{T}, \\
& 1 - x_k \geq x_{t-1} - x_t \quad \forall k \in \{t+1, \ldots, \min\{t + t^D - 1, T\}\}, t \in \mathcal{T}, \\
& x_t, x_t^U, x_t^D \in \{0, 1\} \quad \forall t \in \mathcal{T},
\end{aligned}
$$

where x_0 is the initial on/off status before the beginning of the planning horizon. We note that it can be solved in a polynomial time although it is an integer linear program. Lee et al. (2004) proposed valid inequalities which can characterize the convex hull of the feasible solution set, when only on $(x_t)_{t\in\mathcal{T}}$ variables exist. Rajan and Takriti (2005) extend the results to be used when the start-up variables and corresponding costs exist. Algorithms to solve the self-commitment problem are studied as a substructure in solving self-scheduling problems. A $\mathcal{O}(T^3)$-time dynamic programming algorithm was proposed in Frangioni and Gentile (2006b). An improved $\mathcal{O}(T^2)$ algorithm, which is used to obtain the commitment decisions in self-scheduling problems as in Figure 3.1, is proposed in recent literature (e.g., Frangioni and Gentile, 2015; Guan et al., 2018). In this chapter, we propose an $\mathcal{O}(T)$-time algorithm for the problem, by using the fact that only the information that indicates whether a generator is sufficiently on or off is needed in the self-commitment problem. Here, we state the algorithm and assume that the generator has been off for a sufficient time, for ease of exposition.

The key is to define four types of states for each time period: SU, ON, SD, OFF. For each period $t \in \mathcal{T}$, the state $\texttt{SU}_t$ means that a unit starts up at period t, meaning that it is off when the period $t-1$. Next, the state $\texttt{ON}_t$ indicates that a unit has been on for a sufficient time at the period, which is at least t^U consecutive periods. Further, the state $\texttt{SD}_t$ indicates that the generator shuts down at period t, where it is on at period $t-1$. Lastly, the state $\texttt{OFF}_t$ means that a unit is off for a sufficient time at the period, which at least t^D periods. We let $V^{\texttt{S}}(t)$ be an optimal cost from the beginning of the planning horizon to the period t when the state at period is S. With the defined states, the dynamic programming recursion can be written as follows.

$$V^{\texttt{SU}}(t) = V^{\texttt{OFF}}(t-1) + C_t^U \quad \forall t \in \mathcal{T} \tag{3.4a}$$

$$V^{\texttt{SD}}(t) = V^{\texttt{ON}}(t-1) + C_t^D \quad \forall t \in \mathcal{T} \tag{3.4b}$$

$$V^{\texttt{ON}}(t) = \begin{cases} V^{\texttt{ON}}(t-1) + C_t & \text{if } t < t^U \\ \min\left\{V^{\texttt{ON}}(t-1) + C_t, V^{\texttt{SU}}(t-t^U+1) + V^{\texttt{ON2}}(t)\right\} & \text{if } t \geq t^U \end{cases} \quad \forall t \in \mathcal{T} \tag{3.4c}$$

$$V^{\texttt{OFF}}(t) = \begin{cases} V^{\texttt{OFF}}(t-1) & \text{if } t < t^D \\ \min\left\{V^{\texttt{OFF}}(t-1), V^{\texttt{SD}}(t-t^D+1)\right\} & \text{if } t \geq t^D \end{cases} \quad \forall t \in \mathcal{T} \tag{3.4d}$$

where $V^{\texttt{ON2}}(t) := \sum_{k \in [t-t^U+1, t]} C_k$ indicates a cumulative net cost when the generator is on for period $[t-t^U+1, t]$. The values $V^{\texttt{ON2}}(t)$ for $t \geq t^U$ can be calculated at $\mathcal{O}(T)$ time, since $V^{\texttt{ON2}}(t) = V^{\texttt{ON2}}(t-1) + C_t - C_{t-1-t^U}$. In the recursion equations (3.4a) (and (3.4b)) indicates that the state SU (SD) only be possible at time

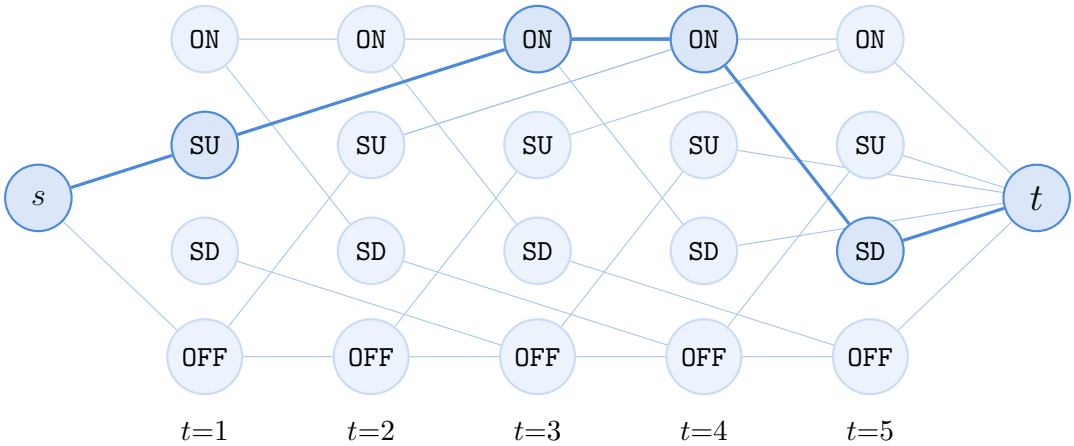


Figure 3.3: Illustration of the self-commitment problem when a generator is on for [1,4] for $t^U = t^D = 3$

period t when the unit is sufficiently off (on) at the period $t-1$. Equations (3.4c) imply that the state ON is possible when it is ON at the previous time period or it is ST at $t - t^U$. Similarly, equations (3.4d) indicate that the OFF state is possible when it is OFF at the previous time period or it is SD at $t - t^D$. The optimal objective value becomes the minimum value of four states in the last time period, i.e. $\min\left\{V^{\texttt{SU}}(T), V^{\texttt{SD}}(T), V^{\texttt{ON}}(T), V^{\texttt{OFF}}(T)\right\}$. It can easily be seen that the calculation of the optimal cost can be on in $\mathcal{O}(T)$ since for each time period the cost functions can be calculated in $\mathcal{O}(1)$-time. The illustration, which is represented as a shortest path problem, is in Figure 3.3 for a generator with $t^U = t^D = 3$, where it is initially off. The solution indicates that the generator starts up at period 1. Since the minimum up time is 3 periods, it is sufficiently on at period 3 and it can turn off from the period. Then, it remains on until period 4 and changes its state to off at period 5.

3.3 Unit decomposition approaches for the unit commitment problem under stochastic net load

Now, we present the unit decomposition approaches to the UC problem under stochastic net load. Recall that the net load is a subtraction of renewable generation from the load demand and is uncertain since the increasing penetration of renewable generation resources makes it hard to accurately predict the exact amount. In this section, we define a set of generators by $\mathcal{G} := \{1, \ldots, G\}$, we use additional subscript g to indicate every parameter and decision variable in Table 3.1 regarding a generator $g \in \mathcal{G}$. For $s \in \mathcal{S}, t \in \mathcal{T}$, the net load value for scenario s in period t is denoted by $d_{s,t}$. In addition, we define a decision variable indicating the amount of load shedding to $y^0_{s,t}$, with corresponding unit penalty cost coefficient K_t. We present a two-stage stochastic optimization model for the UC problem under net load uncertainty. In the model, the commitment states of all generators are determined before the net load realization, and the generation amount is determined to meet the net load after the realization. The extensive formulation (EXTS) is presented as follows.

$$\min \quad \sum_{g\in\mathcal{G},t\in\mathcal{T}} \left(C^{ON}_{g,t}x_{g,t} + C^{U}_{g,t}x^{U}_{g,t} + C^{D}_{g,t}x^{D}_{g,t}\right) + \sum_{g\in\mathcal{G},s\in\mathcal{S},t\in\mathcal{T}} p_s \left(C^{V}_{g,t}y_{g,s,t} + K_t y^0_{s,t}\right) \tag{3.5a}$$

$$\text{s.t.} \quad \sum_{g\in\mathcal{G}} y_{g,s,t} + y^0_{s,t} \geq d_{s,t} \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{3.5b}$$

$$y_{g,s,t} \geq P^{\min}_g x_{g,t} \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \tag{3.5c}$$

$$y_{g,s,t} \leq P^{\max}_g x_{g,t} \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \tag{3.5d}$$

$$y_{g,s,t} \leq y_{g,s,t-1} + R_g x_{g,t-1} + (1 - x_{g,t-1})\bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \tag{3.5e}$$

$$y_{g,s,t} \geq y_{g,s,t-1} - R_g x_{g,t} - (1 - x_{g,t})\bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \tag{3.5f}$$

$$x^U_{g,t} - x^D_{g,t} = x_{g,t} - x_{g,t-1} \quad \forall g \in \mathcal{G}, t \in \mathcal{T} \tag{3.5g}$$

$$x_{g,k} \geq x_{g,t} - x_{g,t-1} \quad \forall g \in \mathcal{G}, k \in \{t+1, \ldots, \min\{t + t^U_g - 1, T\}\}, t \in \mathcal{T}, \tag{3.5h}$$

$$1 - x_{g,k} \geq x_{g,t-1} - x_{g,t} \quad \forall g \in \mathcal{G}, k \in \{t+1, \ldots, \min\{t + t^D_g - 1, T\}\}, t \in \mathcal{T}, \tag{3.5i}$$

$$x_{g,t}, x^U_{g,t}, x^D_{g,t} \in \{0,1\},\ y_{g,s,t}, y^0_{s,t} \geq 0 \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}. \tag{3.5j}$$

In the formulation, the objective function (3.5a) minimizes the commitment cost and the expected variable generation cost for all scenarios. The balance constraints (3.5b) indicate the amount generated must be not less than the net load, otherwise, the load shedding occurs. Constraints (3.5c) and (3.5d) indicate the minimum and maximum generation amount when a generator is on. Constraints (3.5e) and (3.5f) demonstrate up and down ramp limit when the generation amount at a certain time period increases and decreases. Next, constraints (3.5g) indicate the logical relationship between the on/off variable and start-up/shut-down variables. Next, constraints (3.5h) and (3.5i) indicate minimum up/down requirements, where $x_{g,0}$ and $y_{g,s,0}$ are initial conditions.

Although the model can be directly solved by an off-the-shelf solver, the computational burden increases as the number of scenarios grows. A variety of research has focused on decomposition methods to solve the problem (see Zheng et al., 2015; Van Ackooij et al., 2018, for details). Here, we present two unit decomposition frameworks for the problem, which are *schedule decomposition* and *commitment decomposition*, where each framework is related to the subproblem structure when constraints are relaxed. In addition, two solution approaches, Lagrangian relaxation-based and column generation-based solution approaches are presented for each framework. Thus, a total of four solution approaches will be proposed: `SD-LR`, `SD-CG`, `CD-LR`, and `CD-CG`. Among the four methods, schedule decomposition is used in `SD-LR` and `SD-CG`, while commitment decomposition is used in `CD-LR` and `CD-CG`. In addition, Lagrangian relaxation-based methods are used in `SD-LR` and `CD-LR`, while column generation-

based methods are used in `SD-CG` and `CD-CG`.

3.3.1 Schedule decomposition

The schedule decomposition utilizes the substructure of the stochastic self-scheduling problem. At first, the method `SD-LR` is based on Lagrangian relaxation of (EXTS) obtained by relaxing the demand balance constraints (3.5b). The idea behind the approach is that when we relax the demand balance constraints (3.5b), the problem can be decomposed with the generators. When the Lagrangian multiplier $\boldsymbol{\lambda} := (\lambda_{s,t})_{s\in\mathcal{S},t\in\mathcal{T}}$ for the constraints, the corresponding Lagrangian relaxation problem can be written as follows. It can easily be seen that the decomposed problem is a stochastic self-scheduling problem for each generator.

$$z_{LR}^{SD}(\boldsymbol{\lambda}) := \min \left\{ \sum_{s\in\mathcal{S},t\in\mathcal{T}} \left(\lambda_{s,t}\left(d_{s,t} - y_{s,t}^0\right) + p_s K_t y_{s,t}^0\right) + \sum_{g\in\mathcal{G}} z^{SS}(g,\boldsymbol{\lambda}) \right\},$$

where

$$\begin{aligned}
& z^{SS}(g,\boldsymbol{\lambda}) \\
:= \min \quad & \sum_{t\in\mathcal{T}} \left(C_{g,t}^{ON} x_t + C_{g,t}^{U} x_t^{U} + C_{g,t}^{D} x_t^{D}\right) + \sum_{s\in\mathcal{S},t\in\mathcal{T}} \left(p_s C_{g,t}^{V} - \lambda_{s,t}\right) y_{s,t} \\
\text{s.t.} \quad & P_g^{\min} x_t \le y_{s,t} \le P_g^{\max} x_t \quad \forall s\in\mathcal{S}, t\in\mathcal{T}, && \text{(3.6a)} \\
& y_{s,t} \le y_{s,t-1} + R_g x_{t-1} + (1-x_{t-1})\bar{R}_g \quad \forall s\in\mathcal{S}, t\in\mathcal{T}, && \text{(3.6b)} \\
& y_{s,t} \ge y_{s,t-1} - R_g x_t - (1-x_t)\bar{R}_g \quad \forall s\in\mathcal{S}, t\in\mathcal{T}, && \text{(3.6c)} \\
& x_t^U - x_t^D = x_t - x_{t-1} \quad \forall t\in\mathcal{T} && \text{(3.6d)} \\
& x_k \ge x_t - x_{t-1} \quad \forall k\in\{t+1,\ldots,\min\{t+t_g^U-1,T\}\}, t\in\mathcal{T}, && \text{(3.6e)}
\end{aligned}$$

$$1 - x_k \geq x_{t-1} - x_t \quad \forall k \in \{t+1, \ldots, \min\{t + t_g^D - 1, T\}\}, t \in \mathcal{T}, \quad (3.6f)$$

$$x_t, x_t^U, x_t^D \in \{0,1\},\ y_{s,t}, y_{s,t}^0 \geq 0 \quad \forall s \in \mathcal{S}, t \in \mathcal{T}. \quad (3.6g)$$

We can obtain the Lagrangian dual z_{LD}^{SD} by maximizing the Lagrangian relaxation with regard to the multiplier as follows,

$$z_{LD}^{SD} = \max_{\boldsymbol{\lambda} \geq \mathbf{0}} z_{LR}^{SD}(\boldsymbol{\lambda}).$$

Next, the method `SD-CG` utilizes the substructure as the stochastic self-scheduling problem, which is based on column generation. For this, we present a pattern-based formulation which we call *schedule-pattern-based* formulation, where a *pattern* corresponds to a generation schedule that includes on/off status and generation amount for all scenarios and all periods for a generator. For each generator $g \in \mathcal{G}$, we define a set of patterns $\mathcal{SP}(g)$. We use superscript p to indicate the parameters regarding the pattern $p \in \mathcal{SP}(g)$. A pattern p corresponds to a vector $\left(x_{g,t}^p, x_{g,t}^{p,U}, x_{g,t}^{p,D}, y_{g,s,t}^p\right)$, which consists of on/off status $x_{g,t}^p$, start-up status $x_{g,t}^{p,U}$, shut-down status $x_{g,t}^{p,D}$, generation amount for each scenario $y_{g,s,t}^p$. Then, the cost when the pattern $p \in \mathcal{SP}(g)$ is used is denoted by c_g^p and calculated as follows.

$$c_g^p := \sum_{t \in \mathcal{T}} \left(C_{g,t}^{ON} x_{g,t}^p + C_{g,t}^U x_{g,t}^{p,U} + C_{g,t}^D x_{g,t}^{p,D} \right) + \sum_{s \in \mathcal{S}, t \in \mathcal{T}} p_s C_{g,t}^V y_{g,s,t}$$

We use a binary decision vector λ_g^p to indicate whether a pattern $p \in \mathcal{SP}(g)$ is used or not. Then, the schedule-pattern-based formulation (PS) can be represented as

follows.

$$
\begin{aligned}
\text{(PS)} \quad \min \quad & \sum_{g\in\mathcal{G}} \sum_{p\in\mathcal{SP}(g)} c_g^p \lambda_g^p + \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s K_t y_{s,t}^0 \\
\text{s.t.} \quad & \sum_{g\in\mathcal{G}} \sum_{p\in\mathcal{SP}(g)} y_{g,s,t}^p \lambda_g^p + y_{s,t}^0 \geq d_{s,t} \quad \forall s \in \mathcal{S}, t \in \mathcal{T} \\
& \sum_{p\in\mathcal{SP}(g)} \lambda_g^p = 1 \quad \forall g \in \mathcal{G} \\
& \lambda_g^p \in \{0,1\}, y_{s,t}^0 \geq 0 \quad \forall p \in \mathcal{SP}(g), g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}
\end{aligned}
$$

Since the model needs a very large number of decision variables, the proposed approach is based on solving its linear programming relaxation of (PS), which we denote (PS-LM), in addition to deriving primal solutions. To solve the model (PS-LM), we use the column generation method, which iteratively adds columns to the restricted master problem. For a given pattern subset $\mathcal{SP}'(g) \subseteq \mathcal{SP}(g)$, the restricted master problem (PS-RLM) can be written as follows.

$$
\begin{aligned}
\text{(PS-RLM)} \quad \min \quad & \sum_{g\in\mathcal{G}} \sum_{p\in\mathcal{SP}'(g)} c_g^p \lambda_g^p + \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s K_t y_{s,t}^0 \\
\text{s.t.} \quad & \sum_{g\in\mathcal{G}} \sum_{p\in\mathcal{SP}'(g)} y_{g,s,t}^p \lambda_g^p + y_{s,t}^0 \geq d_{s,t} \quad \forall s \in \mathcal{S}, t \in \mathcal{T} \qquad (\pi_{s,t}) \\
& \sum_{p\in\mathcal{SP}'(g)} \lambda_g^p = 1 \quad \forall g \in \mathcal{G} \qquad (\phi_g) \\
& \lambda_g^p, y_{g,s,t}, y_{s,t}^0 \geq 0 \quad \forall p \in \mathcal{SP}'(g), g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}
\end{aligned}
$$

In the formulation, we denote dual variables $\boldsymbol{\pi}$ and $\boldsymbol{\phi}$ next to the constraints. When we obtain the dual optimal solutions $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\phi}})$ to the (PS-RLM), we can check whether

it is optimal to (PS-LM) by calculating the reduced cost. It can easily be seen that the $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\phi}})$ is optimal to (PS-LM) when

$$c_g^p - \sum_{s\in\mathcal{S},t\in\mathcal{T}} \hat{\pi}_{s,t} y_{g,s,t}^p \geq \hat{\phi}_g \quad \forall p \in \mathcal{SP}(g), g \in \mathcal{G}. \tag{3.7}$$

We can calculate the minimum value of the left-hand-side of (3.7) by solving the following subproblem for each generator $g \in \mathcal{G}$. In addition, it is observed that the problem is a stochastic self-scheduling problem (in Section 3.2.1), having a slightly different variable generation cost function.

$$\begin{aligned} z_{SP}^{SD}(g) \ = \min \quad & \sum_{t\in\mathcal{T}} \left(C_{g,t}^{ON} x_t + C_{g,t}^{U} x_t^U + C_{g,t}^{D} x_t^D\right) + \sum_{s\in\mathcal{S},t\in\mathcal{T}} \left(p_s C_{g,t}^{V} - \hat{\pi}_{s,t}\right) y_{s,t} \\ \text{s.t.} \quad & (\boldsymbol{x}, \boldsymbol{y}) \text{ satisfies (3.6a)} - \text{(3.6g)} \end{aligned}$$

When $z_{SP}^{SD}(g) \geq \hat{\phi}_g$ for all generators, the optimal solution of (PS-RLM) with $\mathcal{SP}'(g)$ $g \in \mathcal{G}$ is optimal for (PS-LM). If not, there exists at least one generator that $z_{SP}^{SD}(g) < \hat{\phi}_g$, and the optimal solution of the problem becomes a new pattern $(\boldsymbol{x}^p, \boldsymbol{y}^p)$ that needs to be added to the (PS-RLM). The procedure repeats until no patterns (columns) are added to the (PS-RLM), i.e. $z_{SP}^{SD}(g) \geq \hat{\phi}_g$ for all generators $g \in \mathcal{G}$.

3.3.2 Commitment decomposition

Here, we propose another unit decomposition framework which we denote *commitment decomposition* since it utilizes the self-commitment problem as a subproblem. First, we present the Lagrangian relaxation of (EXTS) in `CD-CG`. Here, compared

to the schedule decomposition that relaxes (3.5b), constraints (3.5c)-(3.5f), which include generation ramping and generation limit requirements, are also relaxed in the commitment decomposition. We denote Lagrangian multipliers corresponding to the constraints from (3.5b) to (3.5f) by from $\boldsymbol{\lambda}^1$ to $\boldsymbol{\lambda}^5$. Then, the Lagrangian relaxation in commitment decomposition is written as follows.

$$z_{LR}^{CD}(\boldsymbol{\lambda}) := \min \left\{ \delta + \sum_{g\in\mathcal{G},s\in\mathcal{S},t\in\mathcal{T}} \beta_{g,s,t} y_{g,s,t} + \sum_{s\in\mathcal{S},t\in\mathcal{T}} \gamma_{s,t} y_{s,t}^0 + \sum_{g\in\mathcal{G}} z^{SC}(g,\boldsymbol{\lambda}) \right\},$$

and the remaining problem is

$$\begin{aligned}
& z^{SC}(g,\boldsymbol{\lambda}) \\
& := \min \quad \sum_{t\in\mathcal{T}} \left(\alpha_{g,t} x_t + C_{g,t}^U x_t^U + C_{g,t}^D x_t^D\right) \\
& \quad \text{s.t.} \quad x_t^U - x_t^D = x_t - x_{t-1} \quad \forall t \in \mathcal{T} && \text{(3.8a)} \\
& \qquad x_k \geq x_t - x_{t-1} \quad \forall k \in \{t+1,\ldots,\min\{t+t_g^U-1,T\}\}, t\in\mathcal{T}, && \text{(3.8b)} \\
& \qquad 1 - x_k \geq x_{t-1} - x_t \quad \forall k \in \{t+1,\ldots,\min\{t+t_g^D-1,T\}\}, t\in\mathcal{T}, && \text{(3.8c)} \\
& \qquad x_t, x_t^U, x_t^D \in \{0,1\} \quad \forall t \in \mathcal{T}, && \text{(3.8d)}
\end{aligned}$$

where the parameters $(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\delta)$ are

$$\begin{aligned}
\alpha_{g,t} := {} & C_{g,t}^{ON} + P_g^{\min}\left(\sum_{s\in\mathcal{S}} \lambda_{g,s,t}^2\right) - P_g^{\max}\left(\sum_{s\in\mathcal{S}} \lambda_{g,s,t}^3\right) \\
& + (\bar{R}_g - R_g)\left(\sum_{s\in\mathcal{S}} \lambda_{g,s,t+1}^4 \mathbb{I}_t\right) + (\bar{R}_g - R_g)\left(\sum_{s\in\mathcal{S}} \lambda_{s,g,t}^5\right) \qquad \forall g\in\mathcal{G}, t\in\mathcal{T} \\
\beta_{g,s,t} := {} & p_s C_{g,t}^V - \lambda_{s,t}^1 - \lambda_{g,s,t}^2 + \lambda_{g,s,t}^3 \\
& + \left(\lambda_{g,s,t}^4 - \lambda_{g,s,t+1}^4 \mathbb{I}_t\right) - \left(\lambda_{g,s,t}^5 - \lambda_{g,s,t+1}^5 \mathbb{I}_t\right) \qquad \forall g\in\mathcal{G}, s\in\mathcal{S}, t\in\mathcal{T}
\end{aligned}$$

$$\gamma_{s,t} := p_s K_t - \lambda^1_{s,t} \qquad \forall s \in \mathcal{S}, t \in \mathcal{T}$$

$$\delta := \sum_{s\in\mathcal{S},t\in\mathcal{T}} \lambda^1_{s,t} d_{s,t} - \sum_{g\in\mathcal{G},s\in\mathcal{S},t\in\mathcal{T}} \bar{R}_g \left(\lambda^4_{g,s,t+1}\mathbb{I}_t + \lambda^5_{g,s,t}\right),$$

and $\mathbb{I}_t$ is a function that has value 1 when $t < T$ and 0 otherwise.

It can be observed that the problem is the self-commitment problem for each generator g, with a slightly modified cost coefficient on commitment decisions. Then, we can obtain the Lagrangian dual z^{CD}_{LD} by maximizing the Lagrangian relaxation value as follows,

$$z^{CD}_{LD} = \max_{\boldsymbol{\lambda}\geq\mathbf{0}} z^{CD}_{LR}(\boldsymbol{\lambda}).$$

Next, we present another column generation-based solution approach for the commitment decomposition method, which is `CD-CG`, by presenting a new pattern-based formulation which we call *commitment-pattern-based formulation*. In the formulation, each decision variable (pattern) corresponds to the commitment states of each generator for all time periods. The formulation is constructed similarly to the construction of schedule-pattern-based formulation in Section 3.3.1. We define a set of all commitment patterns by $\mathcal{CP}(g)$ for each generator $g \in \mathcal{G}$. Each pattern p in $\mathcal{CP}(g)$ corresponds to feasible commitment states and the corresponding vector $(x^p_{g,t}, x^{p,U}_{g,t}, x^{p,D}_{g,t})$ indicates the on/off status, start-up status, and the shut-down status, respectively. For the pattern,

$$c^p_g := \sum_{t\in\mathcal{T}} \left(C^{ON}_{g,t} x^p_{g,t} + C^U_{g,t} x^{p,U}_{g,t} + C^D_{g,t} x^{p,D}_{g,t}\right)$$

represents cost when the pattern p is used. We let a decision variable λ^p_g indicate

whether a pattern $p \in \mathcal{CP}(g)$ is used or not. Then, the mathematical formulation of commitment-pattern-based formulation (PC) is as follows.

$$
\begin{aligned}
\text{(PC)} \quad \min \ & \sum_{g\in\mathcal{G}} \sum_{p\in\mathcal{CP}(g)} c_g^p \lambda_g^p + \sum_{g\in\mathcal{G},s\in\mathcal{S},t\in\mathcal{T}} p_s C_{g,t}^V y_{g,s,t} + \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s K_t y_{s,t}^0 \\
\text{s.t.} \ & \sum_{g\in\mathcal{G}} y_{g,s,t} + y_{s,t}^0 \geq d_{s,t} \quad \forall s \in \mathcal{S}, t \in \mathcal{T} \\
& P_g^{\min} x_{g,t} \leq y_{s,g,t} \leq P_g^{\max} x_{g,t} \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T} \\
& y_{g,s,t} - y_{g,s,t-1} \leq x_{g,t-1} R_g + (1 - x_{g,t-1}) \bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \\
& y_{g,s,t-1} - y_{g,s,t} \leq x_{g,t} R_g + (1 - x_{g,t}) \bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \\
& x_{g,t} = \sum_{p\in\mathcal{CP}(g)} x_{g,t}^p \lambda_g^p \quad \forall p \in \mathcal{CP}(g), g \in \mathcal{G} \\
& \sum_{p\in\mathcal{CP}(g)} \lambda_g^p = 1 \quad \forall g \in \mathcal{G} \\
& \lambda_g^p \in \{0,1\}, y_{g,s,t}, y_{s,t}^0 \geq 0, \quad \forall p \in \mathcal{CP}(g), g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}.
\end{aligned}
$$

Note that $\boldsymbol{x}$ is a decision variable of the problem while $\boldsymbol{x}^p$ is not, and the integer restriction does not need since it is naturally satisfied by the pattern decision variables $(\lambda_g^p)_{p\in\mathcal{CP}(g),g\in\mathcal{G}}$. Since there are an exponential number of pattern variables, it is hard to solve the model (PC) directly. Therefore, as in Section 3.3.1, we propose a method based on solving its linear relaxation problem, which we denote (PC-LM), by column generation while also obtaining primal solutions. To solve (PC-LM) by column generation method, we state a restricted master problem of (PC-LM), which we denote (PC-RLM), with a pattern subset $\mathcal{CP}'(g) \subseteq \mathcal{CP}(g)$ is given for $g \in \mathcal{G}$.

Then, (PC-RLM) can be formulated as follows.

(PC-RLM)

$$\min \sum_{g\in\mathcal{G}} \sum_{p\in\mathcal{CP}'(g)} c_g^p \lambda_g^p + \sum_{g\in\mathcal{G},s\in\mathcal{S},t\in\mathcal{T}} p_s C_{g,t}^V y_{g,s,t} + \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s K_t y_{s,t}^0$$

$$\text{s.t.} \sum_{g\in\mathcal{G}} y_{g,s,t} + y_{s,t}^0 \geq d_{s,t} \quad \forall s \in \mathcal{S}, t \in \mathcal{T}$$

$$P_g^{\min} x_{g,t} \leq y_{s,g,t} \leq P_g^{\max} x_{g,t} \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}$$

$$y_{g,s,t} - y_{g,s,t-1} \leq x_{g,t-1} R_g + (1 - x_{g,t-1})\bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T},$$

$$y_{g,s,t-1} - y_{g,s,t} \leq x_{g,t} R_g + (1 - x_{g,t})\bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T},$$

$$x_{g,t} = \sum_{p\in\mathcal{CP}'(g)} x_{g,t}^p \lambda_g^p \quad \forall g \in \mathcal{G}, t \in \mathcal{T} \qquad (\xi_{g,t})$$

$$\sum_{p\in\mathcal{CP}'(g)} \lambda_g^p = 1 \quad \forall g \in \mathcal{G} \qquad (\zeta_g)$$

$$\lambda_g^p, y_{g,s,t}, y_{s,t}^0 \geq 0, \quad \forall p \in \mathcal{CP}'(g), g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}.$$

We also let the dual variables $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ next to the corresponding constraints. When the problem (PC-RLM) is solved and the corresponding dual optimal values are $\hat{\boldsymbol{\xi}}$ and $\hat{\boldsymbol{\zeta}}$, we can check whether the current solution is optimal to (PC-LM) by calculating the reduced cost. In other words, the solution is optimal when

$$c_g^p - \sum_{t\in\mathcal{T}} \hat{\xi}_{g,t} x_{g,t}^p \geq \hat{\zeta}_g \quad \forall p \in \mathcal{CP}(g), g \in \mathcal{G}. \tag{3.9}$$

We can obtain the minimum value of the left-hand-side of (3.9) by solving the following subproblem for each generator $g \in \mathcal{G}$, which can be easily shown that it is a self-commitment problem with a slightly different objective on the commitment

variable.

$$z_{SP}^{CD}(g) = \min \quad \sum_{t \in \mathcal{T}} \left(\left(C_{g,t}^{ON} - \hat{\xi}_{g,t} \right) x_t + C_{g,t}^{U} x_t^{U} + C_{g,t}^{D} x_t^{D} \right)$$

$$\text{s.t.} \quad \boldsymbol{x} \text{ satisfies (3.8a)} - \text{(3.8d)}$$

When $z_{SP}^{CD}(g) \geq \hat{\zeta}_g$ for all generators, the current optimal objective value of (PC-RLM) with is optimal for (PC-LM). If not, there exists at least one generator that $z_{SP}^{CD}(g) < \hat{\zeta}_g$, whose optimal solutions become a new pattern $\boldsymbol{x}^p$ that needs to be added to the (PC-RLM). The procedure repeats until no patterns (columns) are added to the (PC-RLM), i.e. $z_{SP}^{CD}(g) \geq \hat{\zeta}_g$ for all generators $g \in \mathcal{G}$.

3.3.3 Comparison of the dual bounds

We have presented two unit decomposition frameworks in the previous subsections, including a total of four solution approaches. The methods use the Lagrangian relaxation or column generation that can derive a dual bound to (EXTS). Here we compare the dual bounds obtained by each method, also with the LP relaxation bound of (EXTS), which we denote by z_{LP}^{EXTS}. Recall that z_{LD}^{SD} and z_{LD}^{CD} are Lagrangian duals obtained from `SD-LR` and `CD-LR`, respectively. We also denote z_{CG}^{SD} and z_{CG}^{CD} by the dual bound obtained from column generation-based methods `SD-CG` and `CD-CG`, which are equal to the optimal objective values of (PS-LM) and (PC-LM), respectively. Then, we can state the hierarchy of bounds as the following proposition.

Proposition 3.2.

$$z_{LP}^{\text{EXTS}} \leq z_{LD}^{CD} = z_{CG}^{CD} \leq z_{LD}^{SD} = z_{CG}^{SD}.$$

Proof. claim 1: $z_{LP}^{\mathrm{EXTS}} \leq z_{LD}^{CD} \leq z_{LD}^{SD}$.

We prove claim 1 by using the well-known theorem that indicates the strength of the Lagrangian dual. First, we define a set $X^{(i)}$ as follows.

$$X^{(i)} := \left\{ (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{B}^{3GT} \times \mathbb{R}_{+}^{GST} : \text{ satisfies constraints } (i) \right\}$$

In addition, we define $conv(X)$ to be a convex hull of the points in a set X. According to Theorem 10.1 in Wolsey (2020), z_{LD}^{SD} and z_{LD}^{CD} can be written as follows.

$$\begin{aligned} z_{LD}^{SD} &= \min \left\{ (3.5\mathrm{a}) : (\boldsymbol{x}, \boldsymbol{y}) \in X^{(3.5\mathrm{b})} \cap conv(X^{(3.5\mathrm{c})-(3.5\mathrm{j})}) \right\} \\ z_{LD}^{CD} &= \min \left\{ (3.5\mathrm{a}) : (\boldsymbol{x}, \boldsymbol{y}) \in X^{(3.5\mathrm{b})-(3.5\mathrm{f})} \cap conv(X^{(3.5\mathrm{g})-(3.5\mathrm{j})}) \right\} \end{aligned}$$

Since the set inclusion is as follows,

$$\begin{aligned} \left(X^{(3.5\mathrm{b})} \cap conv(X^{(3.5\mathrm{c})-(3.5\mathrm{j})}) \right) &\subseteq \left(X^{(3.5\mathrm{b})-(3.5\mathrm{f})} \cap conv(X^{(3.5\mathrm{g})-(3.5\mathrm{j})}) \right) \\ &\subseteq \left\{ (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}_{+}^{3GT} \times \mathbb{R}_{+}^{GST} : \text{ satisfies } (3.5\mathrm{b}) - (3.5\mathrm{i}) \right\}, \end{aligned}$$

minimizing the linear objective function (3.5a) over these three sets gives the following results.

claim 2: $z_{LD}^{SD} = z_{CG}^{SD}$, claim 3 : $z_{LD}^{CD} = z_{CG}^{CD}$

Since the proofs of claims 2 and 3 are similar, here we only write the proof of claim 2 for simplicity. The proof is also based on the well-known equivalence of Lagrangian dual and Dantzig-Wolfe reformulation (e.g., Conforti et al., 2014;

Wolsey, 2020). Recall that the Lagrangian dual z_{LD}^{SD} is

$$\begin{aligned} z_{LD}^{SD} =& \max_{\boldsymbol{\lambda} \geq 0} z_{LR}^{SD}(\boldsymbol{\lambda}) \\ =& \max_{\boldsymbol{\lambda} \geq 0} \left\{ \sum_{s \in \mathcal{S}, t \in \mathcal{T}} \lambda_{s,t} d_{s,t} + \min \left\{ \sum_{s \in \mathcal{S}, t \in \mathcal{T}} \left(p_s K_t - \lambda_{s,t}\right) y_{s,t}^0 + \sum_{g \in \mathcal{G}} z^{SS}(g, \boldsymbol{\lambda}) \right\} \right\}. \end{aligned}$$

To represent $z^{SS}(g, \boldsymbol{\lambda})$, we define a set J_g be a set of extreme points of $conv(X_g)$, where $X_g := \left\{ (\boldsymbol{x}_g, \boldsymbol{y}_g) \in \mathbb{B}^{3T} \times \mathbb{R}_+^{ST} : \text{ satisfies (3.6a)} - \text{(3.6g)} \right\}$. We represent each extreme point with $(x_{g,t}^j, x_{g,t}^{j,U}, x_{g,t}^{j,D}, y_{g,s,t}^j)_{g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}}$ for $j \in J_g$. Then, we can write

$$\begin{aligned} & z^{SS}(g, \boldsymbol{\lambda}) \\ & = \min_{j \in J_g} \left\{ \sum_{t \in \mathcal{T}} \left(C_{g,t}^{ON} x_{g,t}^j + C_{g,t}^U x_{g,t}^{j,U} + C_{g,t}^D x_{g,t}^{j,D} \right) + \sum_{s \in \mathcal{S}, t \in \mathcal{T}} \left(p_s C_{g,t}^V y_{g,s,t}^j - \lambda_{s,t} y_{g,s,t}^j \right) \right\}. \end{aligned}$$

Then, the Lagrangian dual can be written by representing minimum values with additional variables α and $(\sigma_g)_{g \in \mathcal{G}}$,

$$\begin{aligned} z_{LD}^{SD} = \max_{\lambda \geq 0} \quad & \sum_{s \in \mathcal{S}, t \in \mathcal{T}} \lambda_{s,t} d_{s,t} + \alpha + \sum_{g \in \mathcal{G}} \sigma_g \\ \text{s.t.} \quad & \sigma_g \leq \sum_{t \in \mathcal{T}} \left(C_{g,t}^{ON} x_{g,t}^j + C_{g,t}^U x_{g,t}^{j,U} + C_{g,t}^D x_{g,t}^{j,D} \right) \\ & \qquad + \sum_{s \in \mathcal{S}, t \in \mathcal{T}} \left(p_s C_{g,t}^V y_{g,s,t}^j - \lambda_{s,t} y_{g,s,t}^j \right) \quad \forall j \in J_g, g \in \mathcal{G} \\ & \alpha \leq \sum_{s \in \mathcal{S}, t \in \mathcal{T}} p_s K_t y_{s,t}^0 - \sum_{s \in \mathcal{S}, t \in \mathcal{T}} \lambda_{s,t} y_{s,t}^0 \end{aligned}$$

Then, we represent the Lagrangian dual with the minimization problem by using

the LP dual.

$$\min \sum_{g\in\mathcal{G}} \left(\left(C^{ON}_{g,t} x^{j}_{g,t} + C^{U}_{g,t} x^{j,U}_{g,t} + C^{D}_{g,t} x^{j,D}_{g,t} \right) + \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s C^{V}_{g,t} y^{j}_{g,s,t} \right) \pi^{j}_{g} + \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s K_t y^{0}_{s,t} \delta$$

$$\text{s.t.} \sum_{j\in J_g} \pi^{j}_{g} = 1 \quad \forall g \in \mathcal{G}$$

$$\delta = 1$$

$$\sum_{g\in\mathcal{G}} y_{g,s,t} \pi^{j}_{g} + y^{0}_{s,t} \delta \geq d_{s,t} \quad \forall s \in \mathcal{S}, t \in \mathcal{T}$$

When we define the coefficient of π by

$$c^{j}_{g} := \left(C^{ON}_{g,t} x^{j}_{g,t} + C^{U}_{g,t} x^{j,U}_{g,t} + C^{D}_{g,t} x^{j,D}_{g,t} \right) + \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s C^{V}_{g,t} y^{j}_{g,s,t},$$

then, the Lagrangian dual value becomes as follows

$$\begin{aligned} z^{SD}_{LD} = \min & \sum_{g\in\mathcal{G}} \sum_{j\in J_g} c^{j}_{g} \pi^{j}_{g} + \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s K_t y^{0}_{s,t} \\ \text{s.t.} & \sum_{g\in\mathcal{G}} \sum_{j\in J_g} y^{j}_{g,s,t} \pi^{j}_{g} + y^{0}_{s,t} \geq d_{s,t} \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \\ & \sum_{j\in J_g} \pi^{j}_{g} = 1 \quad \forall g \in \mathcal{G} \\ & \pi^{j}_{g}, y^{0}_{s,t} \geq 0 \quad \forall j \in J_g, g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \end{aligned}$$

which is equal to z^{SD}_{CG}. □

From Proposition 3.2, we can observe that the dual bounds from the schedule decomposition are the best among all methods. However, the formulation (PS) has the largest number of decision variables among them, since each decision variable

(pattern) includes the generation amount for all scenarios. Rather, the formulation (PC) has a smaller number of decision variables compared to (PS) since each pattern decision variable only contains commitment states for a generator. Therefore, whether the method is efficient or not needs to be evaluated in a practical sense with computational experiments, which will be discussed in Section 3.5.

3.3.4 Upper bounding

Since both Lagrangian relaxation and column generation methods do not derive primal solutions themselves, we present a framework that derives primal solutions in the solution approaches. This simple and efficient method to derive a feasible solution of (EXTS) for each iteration is to recover dispatch decisions from the commitment decisions. To do this, an economic dispatch problem is solved for given commitment decisions $(\bar{x}_{g,t})_{g\in\mathcal{G},t\in\mathcal{T}}$ and the problem is written as below.

$$
\begin{aligned}
\min \quad & \sum_{s\in\mathcal{S},t\in\mathcal{T}} p_s \left(\sum_{g\in\mathcal{G}} C^V_{g,t} y_{g,s,t} + K_t y^0_{s,t} \right) \\
\text{s.t.} \quad & \sum_{g\in\mathcal{G}} y_{g,s,t} + y^0_{s,t} \geq d_{s,t} \quad \forall s \in \mathcal{S}, t \in \mathcal{T}, \\
& P^{\min}_g \bar{x}_{g,t} \leq y_{g,s,t} \leq P^{\max}_g \bar{x}_{g,t} \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \\
& y_{g,s,t} \leq y_{g,s,t-1} + RU_g \bar{x}_{g,t-1} + (1 - \bar{x}_{g,t-1})\bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \\
& y_{g,s,t} \geq y_{g,s,t-1} - RD_g \bar{x}_{g,t} - (1 - \bar{x}_{g,t})\bar{R}_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}, \\
& y_{g,s,t}, y^0_{s,t} \geq 0 \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, t \in \mathcal{T}.
\end{aligned}
$$

Note that the problem is an LP problem that can be solved in a relatively short time. Then the summation of the first and second stage cost becomes the primal objective

value and we let the primal bound as the lowest primal objective value so far. Now, we describe how to obtain commitment decisions for every iteration. First, it is straightforward for `SD-LR` and `CD-LR` that one can obtain the commitment decision by the end of an iteration since the 1UC problem is solved for every iteration. On the other hand, for `SD-CG` and `CD-CG` that use column generation, there are many patterns that have fractional values when solving the restricted linear master (RLM) problem in each iteration. Here, we propose to choose each pattern in each generator that has the largest value (closest to one) among the generated so far. Then, the commitment decisions of the corresponding pattern are obtained for each generator.

3.3.5 Implementation details

Here, we provide implementation details to apply the proposed solution approaches in a computationally efficient manner. First, in column generation-based methods `SD-CG` and `CD-CG`, a set of initial patterns for the problems (PS-LM) or (PC-LM) be determined as follows. For `CD-CG`, a pattern p that a generator is on for the entire planning horizon is used for each generator, i.e. $x^p_{g,t} = 1$ for all $g \in \mathcal{G}, t \in \mathcal{T}$. For `SD-CG` in addition, a pattern whose status is on and the generation amounts for all scenarios are the minimum generation amounts is used, i.e. $x^p_{g,t} = 1$ and $y^p_{g,s,t} = P^{\min}_g$ for all $g \in \mathcal{G}, t \in \mathcal{T}, s \in \mathcal{S}$. In addition, when implementing the column generation, we use the barrier method to solve RLM in each iteration, which has been shown to be efficient when solving large-scale linear programs. Furthermore, we use a well-known stopping criterion to terminate the column generation procedure in a reasonable amount of time, especially for `SD-CG` which needs a number of iterations to converge in the preliminary experiments. For this, we record the dual bound of

the linear master (LM) problem for each iteration. Let objective values of RLM and the corresponding subproblems be z_n^{RLM} and $\left(z_n^{SP}(g)\right)_{g\in\mathcal{G}}$, respectively at iteration $n \in \mathbb{Z}_+$. Then, $v_n := z_n^{RLM} - \sum_{g\in\mathcal{G}} z_n^{SP}(g)$ becomes a dual bound for LM (see Wolsey, 2020, for details). So, the criterion is to terminate when the objective value of RLM is close enough to the dual bound, to be specific when

$$\frac{z_n^{RLM} - v_n}{z_n^{RLM}} \leq \epsilon,$$

where ϵ is a pre-determined parameter.

In Lagrangian relaxation-based methods `SD-LR` and `CD-LR`, the well-known subgradient method is implemented to obtain Lagrangian dual. In the subgradient method, whether the algorithm converges to the optimal Lagrangian dual or the performance differs from how the stepsize is chosen. Here, we choose the stepsize at the iteration n be $\frac{1}{\sqrt{S}}(0.98)^n$, where an initial Lagrangian multiplier is set to a vector of 1's, which has shown a decent performance in the preliminary tests. Although this method does not guarantee the convergence of the optimal Lagrangian dual, it has been widely used because of faster convergence (e.g., Conforti et al., 2014). Then, we run the algorithm for a maximum of 600 iterations.

3.4 Computational experiments

Through the experiments, we demonstrate the efficiency of the proposed methods on 1UC problems and unit decomposition approaches. Both 10-unit instance (Kazarlis et al., 1996) and 54-unit instance (Zimmerman et al., 2010) are used. We let a ramp-up/down limit be half of the maximum generation limit. Then, at the beginning of

the planning horizon, all generators are assumed to be off for a sufficient amount of time. Test results conducted for 1UC problems, which are stochastic self-scheduling and self-commitment problems are presented in Section 3.4.1. In addition, we show the effectiveness of proposed unit decomposition methods in the UC problems under stochastic net load in Section 3.4.2. All the models and algorithms are implemented with C++, where CPLEX 20.1 is used as a (mixed-integer) linear programming solver. In addition, all the computational experiments were conducted on an Intel i7-8700 3.20 GHz personal computer with 32 gigabytes RAM.

3.4.1 Results on single-unit commitment problems

First, experiments are conducted for two types of 1UC problems. A total of 10 generators from Kazarlis et al. (1996) are used. For the stochastic self-scheduling problem, scenarios of the net profit coefficients which is $(b_{s,t})_{s\in\mathcal{S},t\in\mathcal{T}}$, are generated from a uniform distribution $U[0, 20]$. For the self-commitment problem, the net profit coefficient for when a generator $g \in \mathcal{G}$ is on in period $t \in \mathcal{T}$ be $C_{g,t} := \eta_t C^{ON}_{g,t}$, where η_t is generated from the uniform distribution $U[0.5, 1.5]$. Average computation times among 10 generators are reported in the experiments.

For the stochastic self-scheduling problem, three approaches are compared: `MIP`, `DP+LP`, and `DP+DP`. First, `MIP` is to use the commercial solver to solve the MIP problem. Next, both `DP+LP` and `DP+DP` are the methods that use dynamic programming approaches to obtain commitment decisions (as in Figure 3.1 in Section 3.2.1). However, in `DP+LP` while costs of arcs for all pairs are calculated by solving linear programs with the commercial solver. On the other hand, in `DP+DP`, all arc costs are calculated by another dynamic programming framework with the proposed imple-

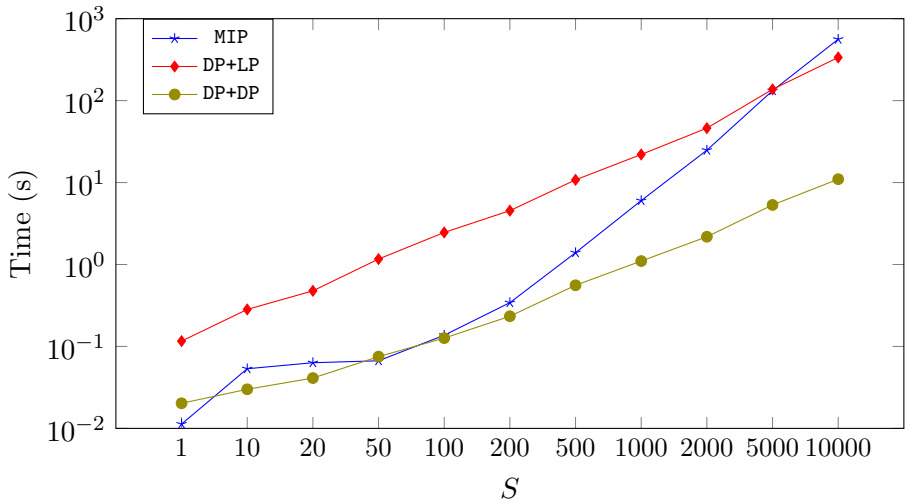


Figure 3.4: Comparison of the computation times for various numbers of scenarios

mentation method (as in Figure 3.2). We reported average computation times for all approaches in Figure 3.4, for various numbers of scenarios from 1 to 10,000. The results demonstrate the efficiency of the proposed solution method `DP+DP`, which has shown the lowest increase in computation times as the number of scenarios increases. It can also be observed that the `MIP` shows less computation time for the small number of scenarios, however, it has shown that the rate of increase regarding the number of scenarios is highest among the three. Alternatively, `DP+LP` exhibits the highest computation times for small numbers of scenarios, the shows comparable performance with the method `MIP` for large numbers of scenarios. Above all, the proposed `DP+DP` shows the best computation times among the three methods for various numbers of scenarios.

Next, we compare four methods for solving the self-commitment problem: `IP`, `DP_C`, `DP_Q`, and `DP_L`. First, `IP` is to use the commercial solver to solve the self-commitment problem which is an integer program. Next, `DP_C` is the widely-used

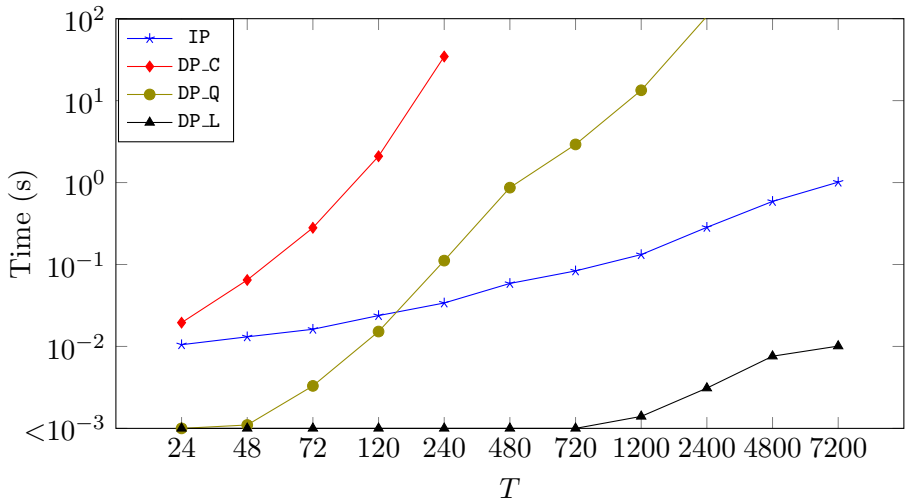


Figure 3.5: Comparison of the computation times for various numbers of periods

dynamic programming algorithm based on state-space graph (e.g., Frangioni and Gentile, 2006b) to solve the problem, where the computational complexity is $\mathcal{O}(T^3)$. In addition, `DP_Q` is the $\mathcal{O}(T^2)$ algorithm which shows advances than `DP_C` by improving computational complexity in the literature (e.g., Frangioni and Gentile, 2015; Guan et al., 2018). Lastly, `DP_L` is the proposed dynamic programming algorithm in Section 3.3.2, whose computational complexity is $\mathcal{O}(T)$. Tests are conducted for various numbers of time periods from 24 to 7,200 and the average computation times are reported in Figure 3.5. The results from the figure emphasize the superiority and scalability of the proposed algorithm `DP_L`, where the problem with $T = 7,200$ can be solved within 10^{-2} seconds. Among the others, the `IP` methods show scalable performance with regard to the increase in the number of time periods. On the other hand, although `DP_C` and `DP_Q` are efficient for small numbers of time periods, they become burdensome when dealing with large numbers of time periods.

3.4.2 Results on unit decomposition methods

Next, we compare various unit decomposition methods for solving the UC problem under stochastic net load. Two sets of instances are used: the 10-unit instance from Kazarlis et al. (1996), and the 54-unit instance from Zimmerman et al. (2010). The nominal net load data is scaled by 0.5 from the given data. Then, the net load scenarios are generated from the normal distribution $N\left(\bar{d}, \left(0.1\bar{d}\right)^2\right)$, where $\bar{d}$ is nominal net load. Various numbers of scenarios are tested, where $S \in \{1, 100, 200, 500, 1{,}000, 2{,}000, 5{,}000\}$ for 10-unit instance and $S \in \{1, 10, 50, 100, 200, 500, 1{,}000\}$ for 54-unit instance.

We implemented and compared four solution approaches based on two unit decomposition frameworks: `SD-LR`, `SD-CG`, `CD-LR`, and `CD-CG`. Recall that the methods `SD-LR` and `SD-CG` are the schedule decomposition, while the methods `CD-LR` and `CD-CG` are the commitment decomposition. In addition, where the Lagrangian relaxation-based approach is used in `SD-LR` and `CD-LR`, and the column generation-based approach is used in `SD-CG` and `CD-CG`. The corresponding subproblems, which are a stochastic self-scheduling problem for `SD-LR` and `SD-CG` and self-commitment problem for `CD-LR` and `CD-CG`, are solved with the proposed algorithms in Section 3.2. For Lagrangian relaxation-based solution approaches, `SD-LR` and `CD-LR`, we used the early termination approach mentioned in Section 3.3.5. For column generation-based solution approaches, we use the early termination framework with $\epsilon = 0.05\%$ only for `SD-CG`, which has shown many iterations to converge in preliminary experiments.

First, we compare four solution approaches for $S = 10$, where solving EXTS is not burdensome when using a commercial solver. For each method, we plot the lower and upper bounds until the algorithm terminates. The results are shown in Figures

3.6 and 3.7. First of all, it can be shown that the lower bounds of SD-LR and SD-CG are better than those of CD-LR and CD-CG. However, it can also be shown that the lower bounds of column generation-based methods (SD-CG and CD-CG) are higher than the Lagrangian relaxation-based methods (SD-LR and CD-LR). It is because it is hard to obtain the exact Lagrangian duals (e.g., Conforti et al., 2014), in addition, we utilized early termination methods to solve the problem in a reasonable amount of time. Further, it can be seen that the number of iterations of SD-CG is much greater than the others, which is greater than 6,000 even with the small-sized instance ($S = 10$). It is because there need a large number of patterns for each generator since each pattern corresponds to the generation amount for each scenario. Next, when comparing the quality of the upper bounds, all of the methods give decent upper bounds, except for SD-LR. In addition, the good upper bounds are shown to be obtained in a relatively small number of iterations.

Next, we compared the lower and upper bounds of three methods except SD-CG, which needs a large number of iterations to converge and thus is hard to be solved in a reasonable amount of time as shown in Figure 3.6. The bounds with various numbers of scenarios are plotted in Figure 3.8. From the figure, it can be shown that although the optimal Lagrangian dual of SD-LR is the highest among the three, it is not always the highest in practice. It is because it is hard to obtain an exact optimal value through the subgradient method as mentioned. On the other hand, lower bounds of column generation methods are obtained when no columns are added, and it is shown that the lower bounds of the method CD-CG are the highest in the experiment. In addition, the method SD-LR has higher lower bounds compared to the

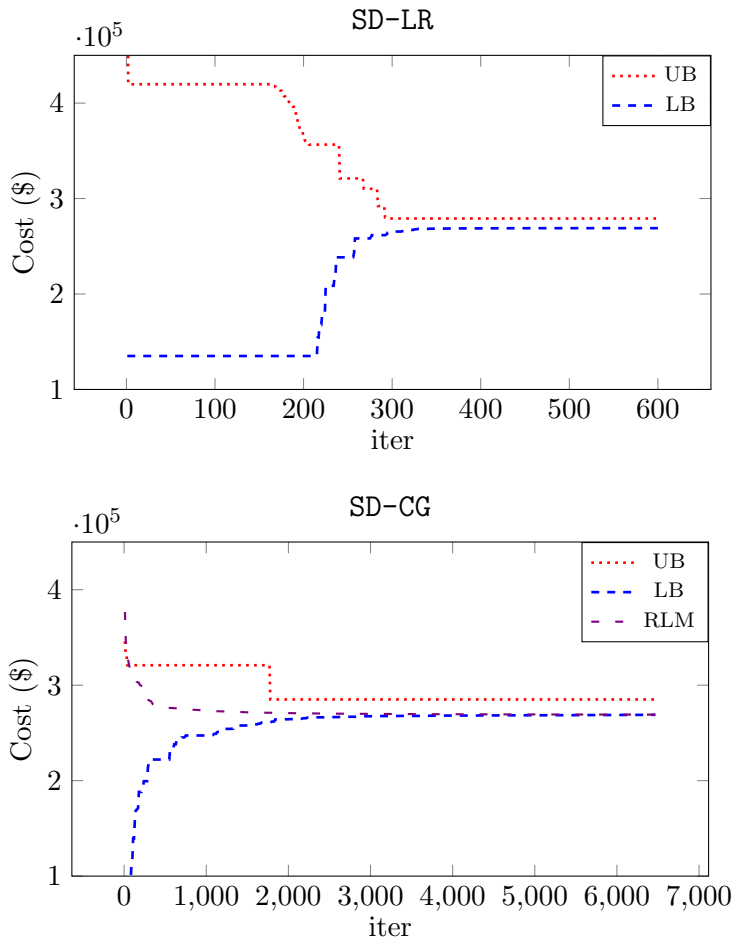


Figure 3.6: Lower and upper bounds trajectories of schedule decomposition for $S = 10$

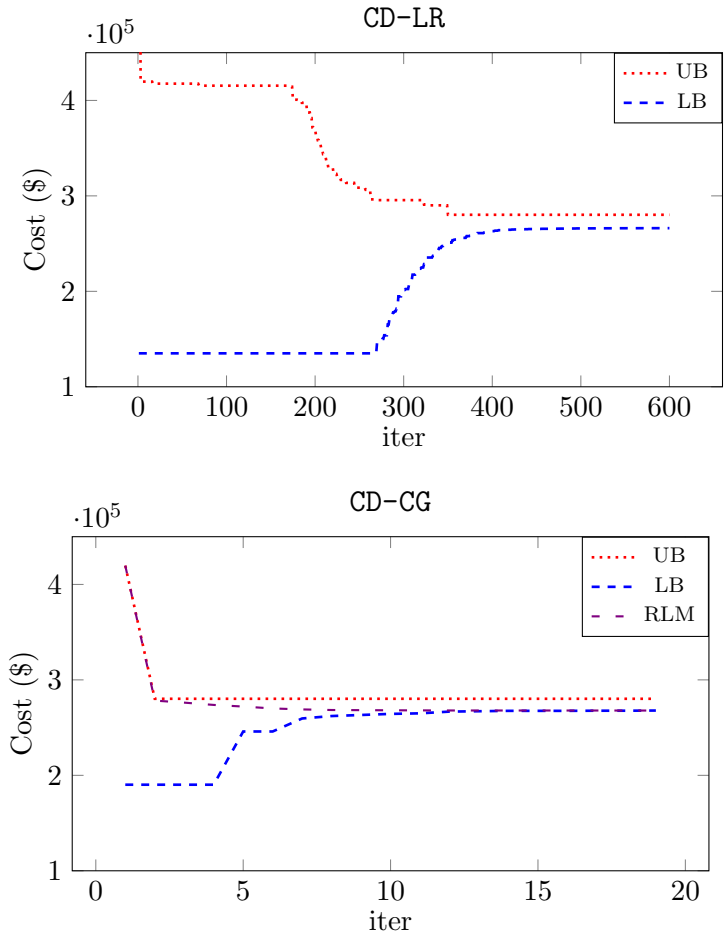


Figure 3.7: Lower and upper bounds trajectories of commitment decomposition for $S = 10$

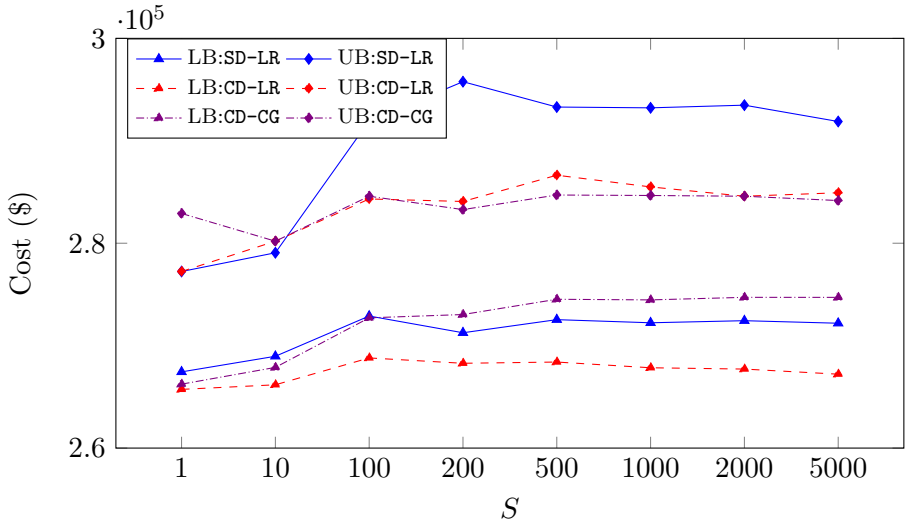


Figure 3.8: Lower and upper bounds of unit decomposition methods with regard to various numbers of scenarios

method `CD-LR`, which is in accordance with the results in Proposition 3.2. Next, when comparing the primal bounds of the three methods, it is observed that the upper bounds of `CD-LR` and `CD-CG` are similar, while those of `SD-LR` are slightly higher. It is because since the proposed upper-bounding methods are based on incumbent commitment solutions, commitment decomposition that focuses on finding good commitment solutions works well in this setting. In addition, the upper bounds of `SD-LR` vary with regard to the number of scenarios. Overall, `CD-CG` has shown the best bounds among the three. Lastly, the computation times of three methods with regard to the number of scenarios are compared in Figure 3.9. From the figure, we can see that `CD-CG` is the fastest, while `SD-LR` is the slowest. To sum up, `CD-CG` shows the best and most stable performance among the three for various numbers of scenarios.

Finally, we compare the solution approach `CD-CG`, which has shown best among

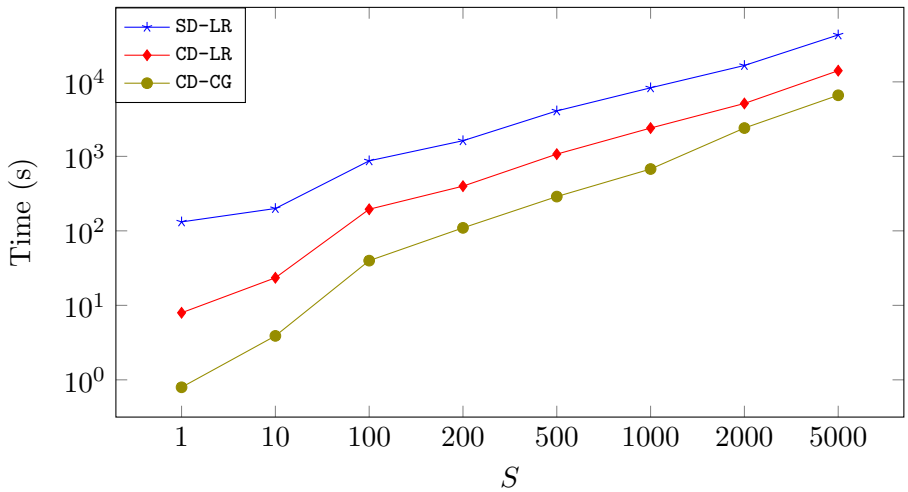


Figure 3.9: Computation times of unit decomposition methods with regard to various numbers of scenarios

the decomposition methods, with the extensive formulation approach `EXTS`, which is to solve the extensive formulation (EXTS) by the solver CPLEX. We conducted tests for 10-unit and 54-unit instances, where the results are in Tables 3.2 and 3.3 respectively. In the experiment, the time limit for each instance is five hours. For `CD-CG`, 'LB' is an optimal objective value of the linear master problem (PC-RLM). In addition, 'UB' is the best upper bound, the value of which is denoted by $\bar{z}_{CG}^{CD}$. For `EXTS`, the 'LP' is linear programming relaxation value (z_{LP}^{EXTS}) and 'MIP' is the objective value (z^{EXTS}) within the time limit. When the problem cannot be solved to optimality within time limits, we report its computation time with an asterisk (*), and the best primal objective value obtained is reported. In addition, in the tables, `cgap` denoting closed gap represents the relative ratio between optimal objective value and linear programming values, and `ogap` denoting optimality gap represents the relative ratio between the objective values of `CD-CG` and `EXTS`. Those values are

calculated as follows.

$$\texttt{cgap} = \frac{z_{CG}^{CD} - z_{LP}^{\text{EXTS}}}{z^{\text{EXTS}} - z_{LP}^{\text{EXTS}}},$$
$$\texttt{ogap} = \frac{\bar{z}_{CG}^{CD} - z^{\text{EXTS}}}{z^{\text{EXTS}}}.$$

From the results, the efficiency and scalability of `CD-CG` are observed. First of all, the linear programming relaxation bounds provided by `CD-CG` are greater than those of `EXTS`, which are closer to the optimal objective value. In addition, the primal objective value obtained from `CD-CG` is less than 1% above on average in a relatively short time compared to `EXTS`. It can also be shown that `EXTS` suffers from large numbers of scenarios, especially when greater than 1,000. However, in those cases, `CD-CG` has shown to be efficient in obtaining good primal and dual bounds. We note that the number of iterations in `CD-CG` is not dependent on the number of scenarios, which emphasizes the scalability of the method when dealing with a large number of scenarios. Next, when we compared `CD-CG` and `EXTS` in a 54-unit instance, the efficiency of the `CD-CG` can also be seen. The number of iterations to converge is reduced compared to the 10-unit instance. Except for the deterministic case ($S = 1$) that `CD-CG` gives poor upper bound, the `CD-CG` gives primal bounds within 0.3% compared to the `EXTS`, with an average 13% of computation time.

3.5 Summary

In this chapter, we proposed dynamic programming algorithms for 1UC problems, which arise when an individual power producer submits its schedule or commitment to the electricity market. First, we presented an efficient dynamic programming

Table 3.2: Performance comparison between CD-CG and EXTS for 10-unit instance

	EXTS				CD-CG					
S	LP	LP time (s)	MIP	MIP time (s)	LB	UB	iter	time (s)	cgap (%)	ogap (%)
1	266,002	0.6	277,240	1.2	266,241	282,905	12	0.8	2.1	2.0
100	272,283	2.6	282,549	139.9	272,716	284,587	18	39.4	4.2	0.7
200	272,698	8.9	282,495	463.7	273,038	283,282	20	110.7	3.5	0.3
500	274,216	41.4	282,934	1,563.9	274,527	284,706	18	288.4	3.6	0.6
1,000	274,229	107.9	282,117	6,281.2	274,467	284,660	19	675.0	3.0	0.9
2,000	274,466	269.8	282,537	*	274,717	284,587	24	2,390.8	3.1	0.7
5,000	274,489	1,128.3	*	*	274,714	284,166	20	6,705.4	-	-

Table 3.3: Performance comparison between CD-CG and EXTS for 54-unit instance

	EXTS				CD-CG					
S	LP	LP time (s)	MIP	MIP time (s)	LB	UB	iter	time (s)	cgap (%)	ogap (%)
1	614,352	4.1	615,464	4.6	614,457	850,383	9	1.2	9.4	38.2
10	610,564	4.6	612,346	21.5	610,869	613,572	9	5.9	17.1	0.2
50	615,028	8.1	617,761	334.4	615,263	619,518	11	40.8	8.6	0.3
100	616,590	14.9	619,427	442.7	616,840	620,722	12	103.5	8.8	0.2
200	616,273	38.9	619,122	2,362.3	616,503	620,415	12	262.1	8.1	0.2
500	617,486	105.5	620,857	*	617,775	623,337	11	783.7	8.6	0.4
1,000	617,338	278.3	*	*	617,624	620,822	11	2,004.2	-	-

approach for the self-scheduling problem under stochastic electricity prices. The proposed algorithm is specifically designed to handle a large number of scenarios and involves proposing an additional dynamic programming implementation for dispatch problems. Second, for the self-commitment problem, which involves making decisions only on the on/off status based on a given profit, we proposed an efficient dynamic programming algorithm with superior computational complexity to other existing methods. Then, two unit decomposition frameworks that can utilize proposed algorithms on 1UC problems are presented to deal with the UC problem with the stochastic net load. The approaches include a novel approach that employs the self-commitment problem and has not been addressed before. We devised Lagrangian relaxation-based and column generation-based solution approaches for each unit decomposition method. Through numerical experiments, the efficiency of

proposed algorithms for 1UC problems is demonstrated for various numbers of scenarios and time periods. Further, the computational performance of a total of four solution approaches based on unit decomposition is demonstrated and the scalability of the novel solution approach, which is a column generation-based method for commitment decomposition, is presented.

Chapter 4

Scalable optimization approaches for microgrid operation under stochastic islanding and net load

4.1 Introduction

A microgrid is a localized electric power system with a low voltage phase consisting of various distributed energy resources that can operate with connection to the centralized power system (main grid) or as an independent system while being disconnected from the main grid (Katiraei et al., 2008). Microgrids have been being considered as a key element of the future energy transition since they can increase the efficiency of distributed energy systems and facilitate the penetration of renewable energy sources (Moretti et al., 2020). A microgrid operator needs to regularly establish operation plans for the efficient and reliable operation of the system, each of which determines the optimal on/off status and dispatch level of each internal generator for each time period in a given planning horizon. To operate a microgrid efficiently and reliably, various factors including islanding events must be carefully considered when establishing the operation plans. Moreover, those factors may be uncertain in that they cannot be accurately estimated in advance, and they can be sequentially realized in each time period in the planning horizon. In this study, we aimed to operate microgrids effectively and efficiently considering two important

uncertain factors in a microgrid environment: islanding and net load. First, the uncertainty of the islanding event, which denotes the time periods when a microgrid disconnects from the main grid, must be carefully considered. Otherwise, the electricity supply of the internal generators cannot be sufficient during the islanding period when receiving supply from the main grid is impossible. Next, the net load, which is referred to as the load obtained by subtracting a negative load (e.g., renewable generation) from a positive one (e.g., load demand), is widely considered in the power system operations because its uncertainty can incorporate those on both sides.

To deal with various uncertain factors in microgrids, most of the previous studies have focused on reactive approaches to operating a microgrid (Mitra and Vallem, 2012; Mohan et al., 2015). That is, operation plans are re-constructed whenever uncertainty is realized, with a carefully determined pre-specified generation capacity, called the reserve. Since then, optimization models for microgrid operation planning have been extended to explicitly incorporate uncertain factors with various modeling frameworks. Examples including stochastic optimization (Farzin et al., 2017; Alvarado-Barrios et al., 2020), robust optimization (Moretti et al., 2020; Gholami et al., 2017), distributionally robust optimization (Yurdakul et al., 2021) models have been proposed, considering uncertainty in renewable generation (Gholami et al., 2016), load demand (Moretti et al., 2020; Cheng et al., 2022), and electricity price (Yurdakul et al., 2021). Among them, islanding events have been considered in the recent literature. Optimization models were proposed in Zacharia et al. (2019) for grid-connected and islanded modes with different objectives. A rolling horizon approach combined with a stochastic optimization model was proposed in Bashir

et al. (2019) to operate a microgrid for a one-year planning horizon. A stochastic optimization model was proposed in Farzin et al. (2017) to minimize the operation costs during unscheduled islanding where the islanding periods are uncertain. However, these studies considered that the grid-connected/islanded modes are given and do not change during the planning horizon, which could be improved by considering the possibility of islanding during the planning horizon. In this regard, a few papers have considered it in an *integrated* manner, with appropriate characterization of possible islanding events in advance. Khodaei (2013) proposed the $T - \tau$ criteria, which considers an islanding event of τ consecutive time periods during the given planning horizon with T time periods, and non-dispatchable generation was additionally considered in Khodaei (2014). The models in both studies aim to minimize power mismatches under worst-case realizations. Lee et al. (2021) proposed a multistage stochastic optimization approach considering the possibility of multiple islanding events in a planning horizon. Gholami et al. (2016) proposed a two-stage stochastic optimization model that considers both contingency-based and normal-operation-based uncertainty. Uncertain factors, including both islanding events and net load, were considered in a two-stage robust optimization framework in Gholami et al. (2017), Guo and Zhao (2018), and Mansouri et al. (2022), where the set of possible islanding events is predetermined.

Although these studies have made significant contributions to operating microgrids under uncertainty, we note a research gap that has not been sufficiently addressed. It is the sequential realization of uncertain factors, that is, one cannot exactly know the future realizations in advance. Specifically, it is difficult to know the exact net load values of the following time periods at a certain time period.

Even for islanding events, considering the possibility of multiple islanding events as mentioned and modeled in Lee et al. (2016) and Ebrahimi et al. (2021), they can occur regardless of the number of times that they have already occurred. Considering this, to make the operation of the microgrids reliable, the decisions in each time period must be *non-anticipative*; in other words, adaptable to such dynamics without anticipating exact values in the following periods. Non-anticipativity needs to be carefully considered; otherwise, the solutions from a model that cannot fully capture the dynamics of the uncertainty could make the operation inexecutable or lead to significant costs, as mentioned in Lorca et al. (2016) and Cho et al. (2019). In this regard, multistage stochastic optimization models, which we call *standard multistage models* hereafter, are widely used in the literature to deal with this issue (e.g., Jiang et al., 2016; Huang et al., 2021). It is because they can naturally address the dynamics and probabilistic nature of uncertainty, where each stage in the models corresponds to the uncertainty realization up to the stage. However, despite the advantages, the standard multistage model is notoriously difficult to solve because of its large number of *scenarios*, where each refers to a combination of samples during the entire planning horizon. In particular, the size of the model depends on the number of scenarios and it is exponential to the number of time periods: $\mathcal{O}(S^T)$, where S is the number of possible realizations for each time period and T is the number of time periods. Therefore, *scalability* should be carefully treated for stochastic optimization models that consider dynamics. In this regard, for models in the literature, dynamics are not fully considered or are not scalable enough to consider all the possible realizations in the planning horizon.

The main contribution of this study is the proposal of scalable stochastic opti-

mization models under stochastic islanding and net load to obtain non-anticipative decisions for period-wise independent realizations. Specifically, we assume that only dispatch decisions are adaptable to the realization of uncertain factors, which is a common assumption in the literature (e.g., Gholami et al., 2017; Lorca et al., 2016). This is because commitment decisions cannot be easily changed in the operational sense owing to physical restrictions regarding generators. To the best of our knowledge, this is the first time that both uncertain factors have been incorporated simultaneously, considering their stochastic nature and dynamics, in a scalable way that practical-sized instances can be solved in a reasonable amount of computation time. Briefly, the integrated models under both islanding and net load uncertainty are presented by combining two scalable optimization approaches proposed for each uncertain factor. First, to make the optimization models scalable for islanding uncertainty, our approach is based on repeatedly solving the models, each of which has a reduced number of possible realizations. Next, to incorporate net load uncertainty in a scalable manner, the proposed modeling framework reduces the number of dispatch decision variables by allowing adequate restrictions on the dispatch level range. Although such decision variables have been used in the literature to increase the feasibility of the two-stage robust unit commitment problem Cho et al. (2019), this is the first time that they have been utilized to improve the scalability of models with stochastic net load.

To summarize, the contribution of the chapter is three-fold:

- We propose optimization models under stochastic islanding and net load, respectively. The proposed models not only can derive solutions that are adaptable to sequential realizations, but have strong advantages in scalability com-

pared to the standard multistage models.

- We propose integrated optimization models that can address both stochastic factors simultaneously. To the best of our knowledge, it is the first time to propose optimization models considering both that are scalable to solve practical-sized instances within a reasonable amount of time. Because the standard multistage model has an exponential number of scenarios with respect to the number of time periods, it can rarely be solved by itself even if only one of the two is uncertain.

- We conducted extensive numerical experiments to demonstrate the efficiency and effectiveness of the proposed optimization models. We demonstrated significantly reduced computation times and effectiveness of the obtained solutions compared to the standard multistage model, which emphasizes the practical usefulness of the proposed models. In addition, we tested the integrated models for practical-sized instances to operate microgrids when both factors are uncertain, demonstrating the effectiveness of the models.

The remainder of this chapter is organized as follows. In Section 4.2, a standard multistage model that can address islanding and net load uncertainty is presented based on the modeling approach widely used in the literature. In Section 4.3, three optimization models are presented. First, optimization models and corresponding replanning procedures are developed to address various islanding events. Next, scalable optimization models under period-wise uncertain net load are proposed and compared to the standard multistage model. Finally, integrated models based on the combination of the two proposed methodologies are presented to operate a mi-

crogrid when both factors are uncertain. Numerical experiments are presented in Section 4.4 to demonstrate the efficiency and effectiveness of the proposed models. Finally, conclusions and future research directions are presented in Section 4.5.

4.2 Standard multistage stochastic optimization model

In this section, we present a standard multistage model for microgrid operation under stochastic islanding and net load. As mentioned, it can naturally address sequential realizations of uncertain factors (Jiang et al., 2016; Huang et al., 2021). First, we formally describe period-wise independent islanding and net load realizations as follows. For an uncertain connection state, two possible realizations are considered for each time period: connected or islanded. We let the set of possible realizations be $S_t^I := \{0, 1\}$, where the value 1 indicates that it is connected and 0 is islanded, and let the probability q_t^I indicate the possibility of islanding for each time period $t \in \mathcal{T}$. Similarly, for an uncertain net load for each time period $t \in \mathcal{T}$, we let the number of possible realizations be σ_t, that is, $S_t^D := \{D_t^1, D_t^2, \ldots, D_t^{\sigma_t}\}$. The probability of each possible realization is q_{st}^D for $s \in S_t^D$, satisfying $\sum_{s=1}^{\sigma_t} q_{st}^D = 1$ for $t \in \mathcal{T}$. To consider both uncertain factors, we define the possible realizations in time period t as a set of pairs $S_t = \{(D_t^1, 0), (D_t^1, 1), (D_t^2, 0), (D_t^2, 1), \ldots, (D_t^{\sigma_t}, 0), (D_t^{\sigma_t}, 1)\}$. Then, considering all the combinations of two realizations, $|S_t| = 2\sigma_t$. The corresponding probability of each realization is defined as q_{st} for $s \in S_t, t \in \mathcal{T}$, where $\sum_{s \in S_t} q_{st} = 1$ for all $t \in \mathcal{T}$.

To present a multistage stochastic optimization model, we construct a node set containing all possible realizations, as illustrated in Figure 4.1, with the following formal description: First, we let $\mathcal{S}_t = S_1 \times \cdots \times S_t$ for $t \in \mathcal{T}$ and each element in

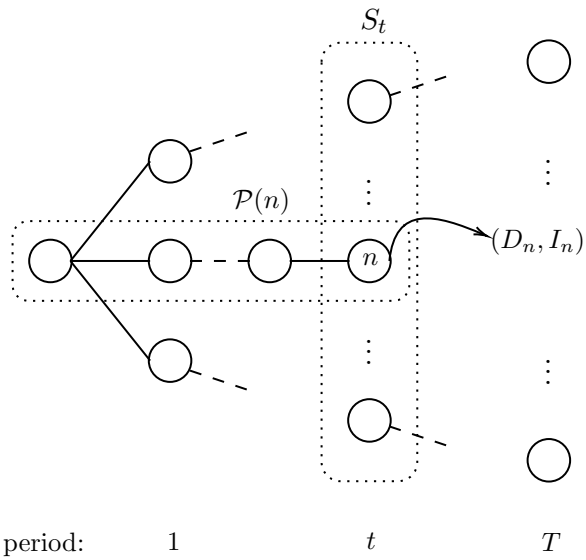


Figure 4.1: A scenario tree based on the node set $\mathcal{N}$

$\cup_{t=0}^{T}\mathcal{S}_t$ be denoted as a *node*, where $\mathcal{S}_0 := \{0\}$ means the root node. In particular, a node in $\mathcal{S}_T$ is called a *scenario*. Node set $\mathcal{N}$, defined as $\mathcal{N} := \cup_{t=0}^{T}\mathcal{S}_t$, represents a set of all nodes in a scenario tree. For each node $n \in \mathcal{N}$, its corresponding time period is denoted by $t(n)$ and its unique parent is denoted by $n^- \in \mathcal{N}$, except for the root node. In addition, the set of all the immediate children of node n is $\mathcal{C}(n)$. For each node, D_n and I_n represent the net load and connection state, respectively, where $(D_n, I_n) \in S_{t(n)}$. Furthermore, the set of nodes on the unique path from the root node to node n, including node n, is denoted as $\mathcal{P}(n)$. The probability of each node p_n is defined as the product of the corresponding probability q_{st} of each node in $\mathcal{P}(n)$, that is, $p_n := \Pi_{m\in\mathcal{P}(n)} q_{m,t(m)}$. Similarly, we additionally define node sets $\mathcal{N}^D$ and $\mathcal{N}^I$ based on realizations $\{S_t^D\}_{t\in\mathcal{T}}$ and $\{S_t^I\}_{t\in\mathcal{T}}$, respectively, to represent node sets when only one of the two factors is uncertain. The probability in each node set, denoted by p_n^D and p_n^I respectively, can also be calculated analogously.

The standard multistage model is based on the aforementioned multistage stochastic optimization framework and operational requirements of a microgrid, as in the literature (Khodaei, 2013; Lee et al., 2021), where a generic microgrid with BESS and thermal units is considered. The notation used for the standard multistage model is as follows.

Notations

Sets and Indices

$\mathcal{T}$	set of time periods, $t \in \mathcal{T} = \{1, 2, \ldots, T\}$
$\mathcal{G}$	set of generators, $g \in \mathcal{G} = \{1, 2, \ldots, G\}$
$\mathcal{E}$	set of battery energy storage systems (BESSs), $e \in \mathcal{E}$

Parameters

c_g^V	variable generation cost of $g \in \mathcal{G}$
c_g^{ST}	start-up cost of $g \in \mathcal{G}$
ρ_t	market price in $t \in \mathcal{T}$
K_t	value of lost load in $t \in \mathcal{T}$
D_n	load demand of node $n \in \mathcal{N}$
I_n	connection state of node $n \in \mathcal{N}$ (1: connected, 0: islanded)
P^M	maximum amount of transaction with main grid
$P_g^{\min}$ ($P_g^{\max}$)	minimum (maximum) generation limit of $g \in \mathcal{G}$
RU_g (RD_g)	ramp-up (ramp-down) limit of $g \in \mathcal{G}$
MU_g (MD_g)	minimum-up (down) time of $g \in \mathcal{G}$
$P_e^{CH,\max}$($P_e^{DCH,\max}$)	maximum charging (discharging) limit of $e \in \mathcal{E}$
η_e	charging/discharging efficiency rate of $e \in \mathcal{E}$
$E_e^{\min}$($E_e^{\max}$)	minimum (maximum) storage level of $e \in \mathcal{E}$

TE_e	target energy level of $e \in \mathcal{E}$ at the end of time period
Binary Variables	
x_{gt}	1 if $g \in \mathcal{G}$ is on in $t \in \mathcal{T}$, 0 otherwise
x_{gt}^{ST}	1 if $g \in \mathcal{G}$ is starting up in $t \in \mathcal{T}$, 0 otherwise
x_{gt}^{SP}	1 if $g \in \mathcal{G}$ is shutting down in $t \in \mathcal{T}$, 0 otherwise
Continuous Variables	
y_{gn}	generation amount of $g \in \mathcal{G}$ of $n \in \mathcal{N}$
y_n^M	transaction amount with main grid of $n \in \mathcal{N}$
y_n^{SH}	load shedding amount of $n \in \mathcal{N}$
y_{en}^{SOC}	energy storage level of $e \in \mathcal{E}$ of $n \in \mathcal{N}$
y_{en}^{CH}	charging amount of $e \in \mathcal{E}$ of $n \in \mathcal{N}$
y_{en}^{DCH}	discharging amount of $e \in \mathcal{E}$ of $n \in \mathcal{N}$

Then, the standard multistage model $\mathtt{B}(\mathcal{N})$ based on the set of nodes $\mathcal{N}$ is given as follows:

$$\min \quad \sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} c_g^{ST} x_{gt}^{ST} + \sum_{n\in\mathcal{N}} p_n \left(\sum_{g\in\mathcal{G}} c_g^V y_{gn} + \rho_{t(n)} y_n^M + K_{t(n)} y_n^{SH} \right) \tag{4.1a}$$

$$\text{s.t.} \quad \sum_{g\in\mathcal{G}} y_{gn} + \sum_{e\in\mathcal{E}} \left(y_{en}^{DCH} - y_{en}^{CH} \right) + y_n^M + y_n^{SH} \geq D_n, \quad \forall n \in \mathcal{N}, \tag{4.1b}$$

$$-I_n P^M \leq y_n^M \leq I_n P^M, \quad \forall n \in \mathcal{N}, \tag{4.1c}$$

$$x_{gt} - x_{g,t-1} \leq x_{gk}, \quad \forall k \in [t+1, \min\{t + MU_g - 1, T\}], g \in \mathcal{G}, t \in \mathcal{T} \setminus \{1\}, \tag{4.1d}$$

$$x_{g,t-1} - x_{gt} \leq 1 - x_{gk}, \quad \forall k \in [t+1, \min\{t + MD_g - 1, T\}], g \in \mathcal{G}, t \in \mathcal{T} \setminus \{1\}, \tag{4.1e}$$

$$x_{gt}^{ST} - x_{gt}^{SP} = x_{gt} - x_{g,t-1}, \quad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{4.1f}$$

$$P_g^{\min} x_{g,t(n)} \leq y_{gn} \leq P_g^{\max} x_{g,t(n)}, \quad \forall g \in \mathcal{G}, n \in \mathcal{N}, \tag{4.1g}$$

$$y_{gn} - y_{g,n^-} \leq RU_g, \quad \forall g \in \mathcal{G}, n \in \mathcal{N}, \tag{4.1h}$$

$$y_{g,n^-} - y_{gn} \leq RD_g, \quad \forall g \in \mathcal{G}, n \in \mathcal{N}, \tag{4.1i}$$

$$y_{en}^{CH} \leq P_e^{CH,\max}, \quad \forall e \in \mathcal{E}, n \in \mathcal{N}, \tag{4.1j}$$

$$y_{en}^{DCH} \leq P_e^{DCH,\max}, \quad \forall e \in \mathcal{E}, n \in \mathcal{N}, \tag{4.1k}$$

$$y_{en}^{SOC} = y_{e,n^-}^{SOC} + \eta_e y_{en}^{CH} - \frac{1}{\eta_e} y_{en}^{DCH}, \quad \forall e \in \mathcal{E}, n \in \mathcal{N}, \tag{4.1l}$$

$$E_e^{\min} \leq y_{en}^{SOC} \leq E_e^{\max}, \quad \forall e \in \mathcal{E}, n \in \mathcal{N}, \tag{4.1m}$$

$$y_{en}^{SOC} = TE_e, \quad \forall e \in \mathcal{E}, n \in \mathcal{N} : t(n) = T, \tag{4.1n}$$

$$x_{gt}, x_{gt}^{ST}, x_{gt}^{SP} \in \{0, 1\}, \quad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{4.1o}$$

$$y_{gn}, y_n^{SH}, y_{en}^{SOC}, y_{en}^{CH}, y_{en}^{DCH} \geq 0, \quad \forall g \in \mathcal{G}, e \in \mathcal{E}, n \in \mathcal{N}. \tag{4.1p}$$

The objective function (4.1a) minimizes the total expected operation cost, consisting of the generation cost, start-up cost, transaction cost, and cost of load shedding. Here, the transaction cost is a multiple of the market price and the transaction amount with the main grid. A positive cost occurs when purchasing power from the main grid, while a negative cost is incurred in the opposite situation. Constraints (4.1b) ensure that the net load is fulfilled with the total generation amount for each time period. Constraints (4.1c) limit the amount of transaction with the main grid, indicating that the transaction is possible only if $I_n = 1$. Constraints (4.1d) (or

(4.1e)) indicate the minimum on (or off) requirements of all generators, imposing that a generator must be on (or off) for certain time periods from it starts up (or shut down). Constraints (4.1f) indicate the logical relationship between the start-up, shut-down, and on decisions. Constraints (4.1g) indicate that the generation amount can only be positive if the generator is on. Constraints (4.1h) and (4.1i) represent the ramp-up and ramp-down limits, respectively. The maximum charging and discharging amount of BESSs are given by (4.1j) and (4.1k), respectively. Constraints (4.1l) represent the transition function of the storage level and constraints (4.1m) limit the possible range of storage level to operate BESSs safely. Constraints (4.1n) ensure that the target storage level is satisfied at the end of the time period.

Note that the commitment decision variables have indices with time periods, whereas the dispatch counterparts are defined for each node. This indicates that the former variables are not adaptable to uncertainty realizations, as assumed and mentioned in Section 4.1. For easier representation in the following sections, we let $\boldsymbol{x} = (x_{gt}, x^{ST}_{gt}, x^{SP}_{gt})_{g\in\mathcal{G},t\in\mathcal{T}}$ and define a set $\mathcal{X}$ that contains all feasible commitment decisions. In other words,

$$\mathcal{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^{3GT} \mid \boldsymbol{x} \text{ satisfies (4.1d), (4.1e), (4.1f), and (4.1o)} \right\}.$$

The total number of scenarios in $\mathcal{N}^D$ and $\mathcal{N}^I$ are $\mathcal{O}(\sigma^T)$ and $\mathcal{O}(2^T)$, respectively, where $\sigma := \max_{t\in\mathcal{T}} \sigma_t$. Thus, the total number of scenarios in $\mathcal{N}$ where both uncertain factors are considered is $\mathcal{O}\left((2\sigma)^T\right)$. Since the number of variables and constraints depend on the number of scenarios, it is notoriously difficult to directly solve the model using a commercial solver, even if only one of the two factors is considered

uncertain. Therefore, in the following sections, we compare the model as a baseline with the proposed models.

4.3 Proposed optimization models

4.3.1 Optimization models and replanning procedure under islanding uncertainty

First, we focus on proposing optimization models under islanding uncertainty, where the net load is regarded as certain for ease of presentation. In other words, the goal of this subsection is to make optimization models scalable compared to the standard multistage model $\mathtt{B}(\mathcal{N}^I)$. Recall that although one can obtain solutions that are adaptable to all possible islanding events by solving the standard multistage model, it is not practical to solve the model because the number of scenarios is exponential to the number of time periods.

The proposed scheme addresses islanding uncertainty using two types of scalable optimization models. The key idea for both types of models is to consider a restricted number of scenarios instead of all possible scenarios simultaneously. Specifically, we denote an *islanding event* as the situation in which islanding occurs for *consecutive* time periods. As shown in Figure 4.2, each islanding scenario may consist of several islanding events. We define a node set $\mathcal{N}^I(k) \subset \mathcal{N}^I$ for each time period $k \in \mathcal{T}$ by the set of nodes with at most one islanding event in the time period $\mathcal{T}_k := \{k, k+1, \ldots, T\}$. Then, $\mathcal{N}^I(T) \subset \mathcal{N}^I(T-1) \subset \cdots \subset \mathcal{N}^I(1) \subset \mathcal{N}^I$ holds. In addition, we let the time period $t(n)$ for each node n in $\mathcal{N}^I(k)$ start from period k. Two sets of nodes, $\mathcal{N}^I$ defined in Section 4.2 and $\mathcal{N}^I(1)$ with $\mathcal{T} = \{1, 2, 3, 4\}$, are compared in Figure 4.3. As can be seen in the figure, in $\mathcal{N}^I(1)$, node n with a connection

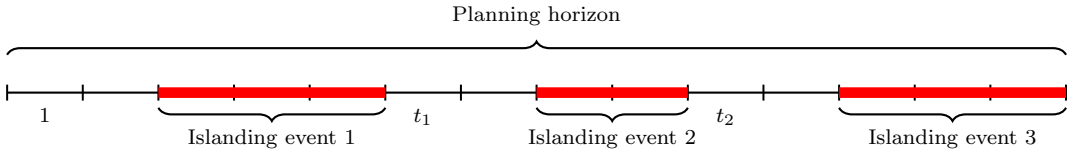
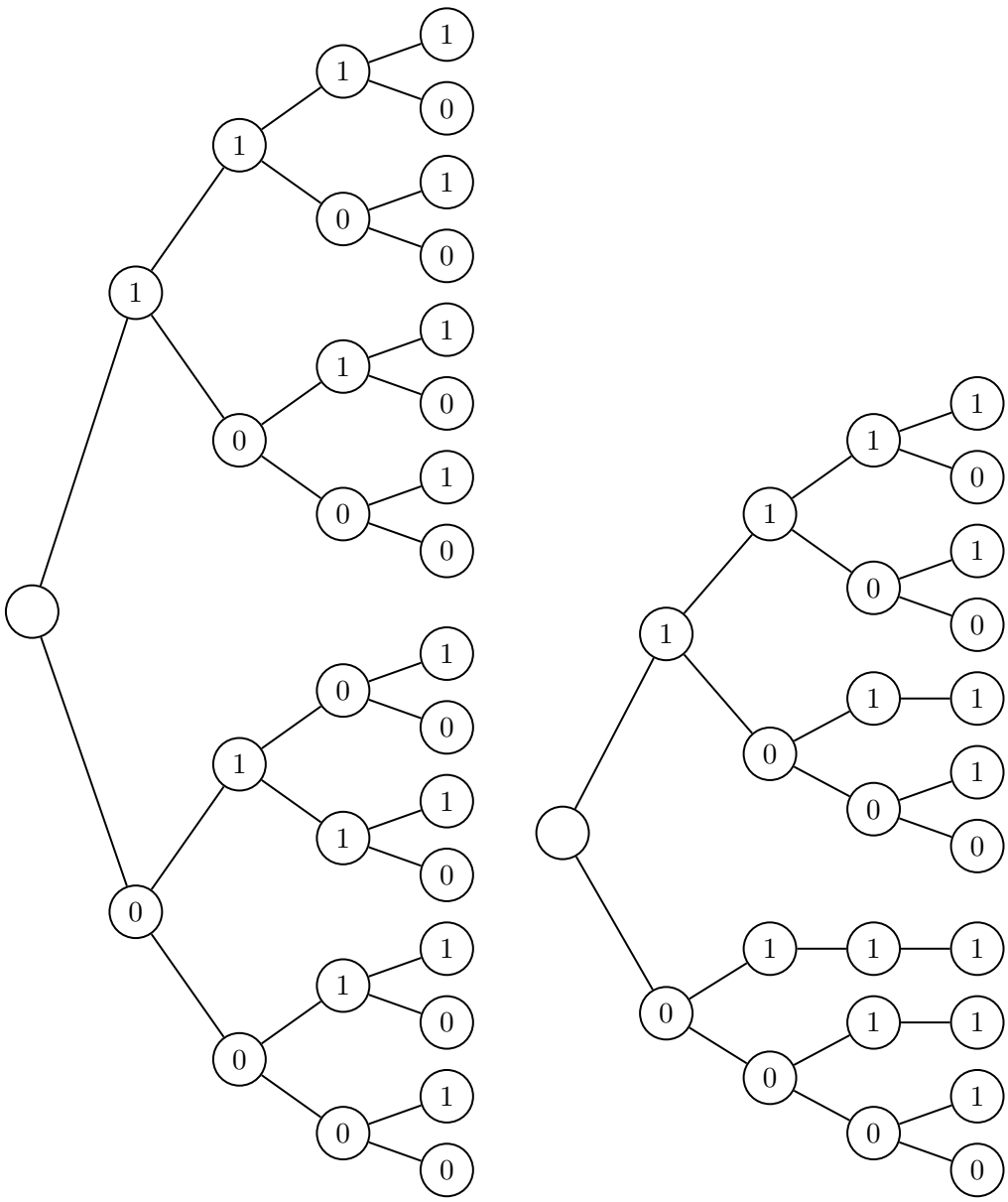


Figure 4.2: Illustration of a scenario with multiple islanding events

state ($I_n = 1$) does not have descendent nodes with islanded states if it already has an islanded state in $\mathcal{P}(n)$. This enables a reduction in the number of scenarios compared to $\mathcal{N}^I$. In particular, since only the start and end times are needed to define an islanding event, the number of scenarios in $\mathcal{N}^I(k)$ is $\mathcal{O}((T-k)^2)$.

Now, we introduce two types of optimization models based on the node set $\mathcal{N}^I(k)$, `BL` and `RP`(k) for $k \in \mathcal{T}$, referred to as *baseline* and *replan*, respectively. First, the baseline model `BL` is similar to the standard multistage model `B`$(\mathcal{N})$ in that the decisions for the entire planning horizon are considered. The only difference is that `BL` is constructed with the node set $\mathcal{N}^I(1)$; recall that islanding scenarios with at most one islanding event in $\mathcal{T}$ are considered. Therefore, the model can be rewritten as `BL`=`B`$(\mathcal{N}^I(1))$, which has a strong advantage over `B`$(\mathcal{N}^I)$ in terms of scalability, where the number of scenarios in the former is $\mathcal{O}(T^2)$ and in the latter is $\mathcal{O}(2^T)$. Thus, the role of model `BL` is to obtain good commitment solutions within a reasonable amount of time.

It should be mentioned that it is not sufficient to deal with all possible islanding scenarios with only `BL`, as the solutions from the model are only adaptable to a restricted number of scenarios and several islanding events are possible in general. This is the reason why the *replan* model `RP` and the corresponding replanning procedure are required. The model `RP`(k) for $k \in \mathcal{T}$ is similar to `BL` in that the node set $\mathcal{N}^I(k)$ is considered but is defined slightly differently with it. The purpose of model `RP`(k)



(a) $\mathcal{N}^I$ (b) $\mathcal{N}^I(1)$

Figure 4.3: Comparison of two sets of nodes with $\mathcal{T} = \{1, 2, 3, 4\}$

is to amend the pre-determined schedule from k onward, to cope with consequent islanding events. In other words, the planning horizon of $\texttt{RP}(k)$ is $\mathcal{T}_k$. In addition, only dispatch decision variables can be determined though RP, where commitment decisions are given by $\bar{\boldsymbol{x}}$. With proper initial dispatch and SOC conditions, $\bar{y}_g$ for $g \in \mathcal{G}$ and $\bar{y}_e^{SOC}$ for $e \in \mathcal{E}$, $\texttt{RP}(k)$ is defined as $\texttt{B}(\mathcal{N}^I(k))$ on $\mathcal{T}_k$ with the following additional constraints

$$
\begin{aligned}
&\boldsymbol{x} = \bar{\boldsymbol{x}}, \\
&y_{e0}^{SOC} = \bar{y}_e^{SOC}, \quad \forall e \in \mathcal{E}, \\
&y_{g0} = \bar{y}_g, \quad \forall g \in \mathcal{G},
\end{aligned}
$$

where node index zero represents the root node. Notably, it is a linear program that can be solved much faster than BL because it considers the given commitment decisions.

We now describe the replanning procedure using scalable optimization models BL and RP to obtain solutions that are adaptable for any possible case for the entire planning horizon. Let $\boldsymbol{y}_n = (y_{gn}, y_n^M, y_n^{SH}, y_{en}^{SOC}, y_{en}^{CH}, y_{en}^{DCH})_{g\in\mathcal{G}, e\in\mathcal{E}}$ for a given node n. First, before the start of the planning horizon, we obtain commitment decision $\bar{\boldsymbol{x}}$ and dispatch decisions $\bar{\boldsymbol{y}}_n$ for all node $n \in \mathcal{N}^I(1)$ by solving BL. With the assumption that only dispatch decisions are adaptable to the realization, the commitment decision $\bar{\boldsymbol{x}}$ will be used throughout the planning horizon. Subsequently, for each time period $k \in \mathcal{T}$, the connection state $i_k \in \{0, 1\}$ is observed, where a microgrid is connected for $i_k = 1$ and islanded for $i_k = 0$. The key time period of the replanning procedure is immediately after the end of an islanding event, that is,

Algorithm 4.1 Replanning procedure for a given planning horizon $\mathcal{T} = \{1, \ldots, T\}$

$l \leftarrow 1$ and obtain $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}_n$ for all $n \in \mathcal{N}^I(l)$ by solving BL
for $k = 1$ to T **do**
    observe $i_k \in \{0, 1\}$
    if $k = l$ **then**
        $n_k \leftarrow n \in \mathcal{N}^I(l)$ such that $t(n) = k$ and $I_n = i_k$
    else
        $n_k \leftarrow n \in \mathcal{C}(n_{k-1})$ such that $I_n = i_k$
    end if
    if $k \leq T - 1$ and $i_k - i_{k-1} = 1$ **then**
        $l \leftarrow k + 1$ and obtain $\bar{\boldsymbol{y}}_n$ for all $n \in \mathcal{N}^I(l)$ by solving RP(l)
    end if
    return decision at period k as ($\bar{\boldsymbol{x}}_k$,$\bar{\boldsymbol{y}}_{n_k}$)
end for

$i_{k-1} = 0$ and $i_k = 1$. During this period, the *new* node set $\mathcal{N}^I(k+1)$ is constructed to update the dispatch decisions to appropriately deal with another islanding event that may occur. Then, model RP($k + 1$) is solved to update the dispatch decisions (y_n for $n \in \mathcal{N}^I(k+1)$) from period $k + 1$ onward.

The dispatch decisions for each time period k are then decided by choosing node n_k appropriately as follows: Node n_k is selected from one of the immediate children of n_{k-1} that satisfies $I_{n_k} = i_k$. However, period k is the first period when an optimization model is solved, and the node is chosen from the new node set with $t(n) = k$ and $I_n = i_k$. Then, the obtained $\bar{\boldsymbol{y}}_{n_k}$ becomes a decision at the time period k. This procedure is repeated until the end of the planning horizon. The entire procedure is presented in Algorithm 4.1.

For example, in the case illustrated in Figure 4.2, with a scenario where three islanding events can occur, three optimization models are solved to deal with the events. First, the model BL is solved before the start of the planning horizon to determine the commitment decisions $\bar{\boldsymbol{x}}$ for the entire planning horizon and the dispatch

decisions until the end of the first islanding event. Because the first islanding event ends in period t_1, the replanning model $\texttt{RP}(t_1+1)$ is solved in the period to update dispatch decisions from t_1+1 onward. Similarly, the dispatch decisions from period t_2+1 are obtained by solving $\texttt{RP}(t_2+1)$ for period t_2.

4.3.2 Scalable optimization models under net load uncertainty

Next, we focus on making the optimization models scalable with regard to the net load uncertainty, especially compared to $\texttt{B}(\mathcal{N}^D)$, as in the previous subsection. For simplicity, the connection states are provided in advance, that is, the maximum transaction amount for each time period $t \in \mathcal{T}$ is given as

$$\bar{P}_t^M = \begin{cases} 0, & \text{if islanded in period } t, \\ P_M, & \text{if connected in period } t. \end{cases}$$

On the whole, the key idea is to make the dispatch decisions determined period-wise independently by introducing new decision variables that decide their ranges to which all of them reside. It enables the redefinition of the dispatch decision variables for each realization in each period, which significantly reduces the number of variables and constraints while keeping the solutions from the model still adaptable. The main consideration in designing such variables is to make the dispatch decision variables feasible in terms of inter-temporal constraints. We note that a similar concept was proposed in Cho et al. (2019), where the idea was used to enhance feasibility in terms of the two-stage robust optimization framework. The new variable $z_{gt}^{\min}(z_{gt}^{\max})$ is defined as the minimum (maximum) generation amount for $g \in \mathcal{G}$ and $t \in \mathcal{T}$. In addition, the new variables $z_{et}^{CH,\min}(z_{et}^{CH,\max})$ and $z_{et}^{DCH,\min}(z_{et}^{DCH,\max})$ are defined similarly. Then, when we let $\boldsymbol{z} = (z_{gt}^{\min}, z_{et}^{CH,\min}, z_{et}^{DCH,\min}, z_{gt}^{\max}, z_{et}^{CH,\max}$

$, z_{et}^{DCH,\max})_{g\in\mathcal{G},e\in\mathcal{E},t\in\mathcal{T}}$ for brevity, we design a set of constraints that $\boldsymbol{z}$ satisfies, as follows:

$$P_g^{\min} x_{gt} \le z_{gt}^{\min} \le z_{gt}^{\max} \le P_g^{\max} x_{gt}, \quad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{4.2a}$$

$$z_{gt}^{\max} - z_{g,t-1}^{\min} \le RU_g, \quad \forall g \in \mathcal{G}, t \in \mathcal{T} \setminus \{1\}, \tag{4.2b}$$

$$z_{g,t-1}^{\max} - z_{gt}^{\min} \le RD_g, \quad \forall g \in \mathcal{G}, t \in \mathcal{T} \setminus \{1\}, \tag{4.2c}$$

$$0 \le z_{et}^{CH,\min} \le z_{et}^{CH,\max} \le P_e^{CH,\max}, \quad \forall e \in \mathcal{E}, t \in \mathcal{T}, \tag{4.2d}$$

$$0 \le z_{et}^{DCH,\min} \le z_{et}^{DCH,\max} \le P_e^{DCH,\max}, \quad \forall e \in \mathcal{E}, t \in \mathcal{T}, \tag{4.2e}$$

$$E_e^{\min} \le \sum_{k=1}^{t} \left(\eta_e z_{ek}^{CH,\min} - \frac{1}{\eta_e} z_{ek}^{DCH,\max} \right), \quad \forall e \in \mathcal{E}, t \in \mathcal{T}, \tag{4.2f}$$

$$\sum_{k=1}^{t} \left(\eta_e z_{ek}^{CH,\max} - \frac{1}{\eta_e} z_{ek}^{DCH,\min} \right) \le E_e^{\max}, \quad \forall e \in \mathcal{E}, t \in \mathcal{T}, \tag{4.2g}$$

$$\sum_{k=1}^{T} \left(\eta_e z_{ek}^{CH,\max} - \frac{1}{\eta_e} z_{ek}^{DCH,\min} \right) = \sum_{k=1}^{T} \left(\eta_e z_{ek}^{CH,\min} - \frac{1}{\eta_e} z_{ek}^{DCH,\max} \right) = TE_e, \quad \forall e \in \mathcal{E}. \tag{4.2h}$$

First, constraints (4.2a) require the relationship between $\boldsymbol{x}$ and $\boldsymbol{z}$ variables, ensuring that $\boldsymbol{z}$ has values between the minimum and maximum generation limits when the generator is on, or zero otherwise. Next, constraints (4.2b) and (4.2c) indicate ramping requirements on $\boldsymbol{z}$ that the maximum possible difference in generation level cannot exceed the ramping requirement. Similarly, constraints (4.2d) and (4.2e) impose the ranges of the charging and discharging variables, respectively. Constraints (4.2f)-(4.2h) require the $\boldsymbol{z}$ variables to satisfy the SOC ranges. In particular, constraints (4.2f) mean that the minimum possible SOC level for any time period t is not less than the given value $E_e^{\min}$, and constraints (4.2g) are defined similarly. In addition, the target SOC level for the last time period is satisfied by (4.2h). Note that all $\boldsymbol{z}$ values have non-negative values, as naturally satisfied in (4.2a), (4.2d),

and (4.2e). For simplicity, we now define a set $\mathcal{Z}(\boldsymbol{x})$ containing feasible $\boldsymbol{z}$ values satisfying (4.2a)-(4.2h). Note that the set depends on the commitment variable $\boldsymbol{x}$.

As mentioned, the key advantage of introducing variable $\boldsymbol{z}$ is that one can re-define the dispatch decisions, which are adaptable to sequential realizations, independently of realizations for each time period, without having dispatch decision variables correspond to every node in $\mathcal{N}^D$. More specifically, for $g \in \mathcal{G}$, instead of using y_{gn} that indicates the dispatch amount for node $n \in \mathcal{N}^D$, y_{gst} is defined to indicate the dispatch amount for realization $s \in S_t^D$, $t \in \mathcal{T}$. The remaining dispatch decisions y_{st}^M, y_{st}^{SH}, y_{est}^{CH}, and y_{est}^{DCH} can be defined in a similar manner and are summarized as follows:

Continuous Variables for `R`

y_{gst}	generation amount of $g \in \mathcal{G}$ of $s \in S_t^D, t \in \mathcal{T}$
y_{st}^M	transaction amount with main grid of $s \in S_t^D, t \in \mathcal{T}$
y_{st}^{SH}	load shedding amount of $s \in S_t^D, t \in \mathcal{T}$
y_{est}^{CH}	charging amount of $e \in \mathcal{E}$ of $s \in S_t^D, t \in \mathcal{T}$
y_{est}^{DCH}	discharging amount of $e \in \mathcal{E}$ of $s \in S_t^D, t \in \mathcal{T}$
$z_{gt}^{\min}(z_{gt}^{\max})$	minimum (maximum) generation amount for $g \in \mathcal{G}, t \in \mathcal{T}$
$z_{et}^{CH,\min}(z_{et}^{CH,\max})$	minimum (maximum) charging amount for $e \in \mathcal{E}, t \in \mathcal{T}$
$z_{et}^{DCH,\min}(z_{et}^{DCH,\max})$	minimum (maximum) discharging amount for $e \in \mathcal{E}, t \in \mathcal{T}$

Then, the proposed model `R`, based on the new variable $\boldsymbol{z}$ and re-defined variable $\boldsymbol{y}$, is represented as follows:

$$\texttt{R}: \quad \min \quad \sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} c_g^{ST} x_{gt}^{ST} + \sum_{t\in\mathcal{T}}\sum_{s\in S_t^D} q_{st}^D \left(\sum_{g\in\mathcal{G}} c_g^V y_{gst} + \rho_t y_{st}^M + K_t y_{st}^{SH} \right) \tag{4.3a}$$

s.t. $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{z} \in \mathcal{Z}(\boldsymbol{x})$,

$$\sum_{g\in\mathcal{G}} y_{gst} + \sum_{e\in\mathcal{E}} (y^{DCH}_{est} - y^{CH}_{est}) + y^{M}_{st} + y^{SH}_{st} \geq D^{s}_{t}, \quad \forall s \in S^{D}_{t}, t \in \mathcal{T}, \tag{4.3b}$$

$$-\bar{P}^{M}_{t} \leq y^{M}_{st} \leq \bar{P}^{M}_{t}, \quad \forall s \in S^{D}_{t}, t \in \mathcal{T}, \tag{4.3c}$$

$$z^{\min}_{gt} \leq y_{gst} \leq z^{\max}_{gt}, \quad \forall g \in \mathcal{G}, s \in S^{D}_{t}, t \in \mathcal{T}, \tag{4.3d}$$

$$z^{CH,\min}_{et} \leq y^{CH}_{est} \leq z^{CH,\max}_{et}, \quad \forall e \in \mathcal{E}, s \in S^{D}_{t}, t \in \mathcal{T}, \tag{4.3e}$$

$$z^{DCH,\min}_{et} \leq y^{DCH}_{est} \leq z^{DCH,\max}_{et}, \quad \forall e \in \mathcal{E}, \forall s \in S^{D}_{t}, t \in \mathcal{T}, \tag{4.3f}$$

$$y_{gst}, y^{SH}_{st}, y^{CH}_{est}, y^{DCH}_{est} \geq 0, \quad \forall g \in \mathcal{G}, e \in \mathcal{E}, s \in S^{D}_{t}, t \in \mathcal{T}. \tag{4.3g}$$

In the model, the objective function (4.3a) and constraints (4.3b)-(4.3c) are obtained by replacing the node index $n \in \mathcal{N}^D$ with $s \in S^D_t$ and $t \in \mathcal{T}$ in (4.1a)-(4.1c) of $\texttt{B}(\mathcal{N}^D)$. In particular, the probability of each net load realization q^D_{st} is used instead of the node probability p^D_n. Similarly, the net load realization D^s_t is used instead of D_n. Constraints (4.3d)-(4.3f) utilize the range variable $\boldsymbol{z}$, indicating that the dispatch variables have values within such ranges.

We note that the number of decision variables is significantly reduced to $\mathcal{O}(\sigma T)$, compared to $\mathcal{O}(\sigma^T)$ where all the nodes are considered. In the following proposition, we show the relationship between the two models, $\texttt{B}(\mathcal{N}^D)$ and R.

Proposition 4.1. *Let Z_B and Z_R be the optimal value of $\texttt{B}(\mathcal{N}^D)$ and R, respectively. Then, $Z_B \leq Z_R$.*

Proof. First, we define an artificial problem R' by following,

$$Z'_R = \min \sum_{t\in\mathcal{T}} \sum_{g\in\mathcal{G}} c^{ST}_{g} x^{ST}_{gt} + \sum_{n\in\mathcal{N}^D} p^{D}_{n} \left(\sum_{g\in\mathcal{G}} c^{V}_{g} y_{gn} + \rho_{t(n)} y^{M}_{n} + K_{t(n)} y^{SH}_{n} \right) \tag{4.4a}$$

$$\text{s.t. } \boldsymbol{x} \in \mathcal{X}, \boldsymbol{z} \in \mathcal{Z}(\boldsymbol{x}), \tag{4.4b}$$

$$\sum_{g\in\mathcal{G}} y_{gn} + \sum_{e\in\mathcal{E}} \left(y_{en}^{DCH} - y_{en}^{CH}\right) + y_n^M + y_n^{SH} \geq D_n, \ \ \forall n \in \mathcal{N}^D, \tag{4.4c}$$

$$-\bar{P}_{t(n)}^M \leq y_n^M \leq \bar{P}_{t(n)}^M, \quad \forall n \in \mathcal{N}^D, \tag{4.4d}$$

$$z_{g,t(n)}^{\min} \leq y_{gn} \leq z_{g,t(n)}^{\max}, \quad \forall g \in \mathcal{G}, n \in \mathcal{N}^D, \tag{4.4e}$$

$$z_{e,t(n)}^{CH,\min} \leq y_{en}^{CH} \leq z_{e,t(n)}^{CH,\max}, \quad \forall e \in \mathcal{E}, n \in \mathcal{N}^D, \tag{4.4f}$$

$$z_{e,t(n)}^{DCH,\min} \leq y_{en}^{DCH} \leq z_{e,t(n)}^{DCH,\max}, \quad \forall e \in \mathcal{E}, n \in \mathcal{N}^D, \tag{4.4g}$$

$$y_{gn}, y_n^{SH}, y_{en}^{CH}, y_{en}^{DCH} \geq 0, \quad \forall g \in \mathcal{G}, e \in \mathcal{E}, n \in \mathcal{N}^D, \tag{4.4h}$$

which is a formulation that introduces the range variables $\boldsymbol{z}$ but still uses dispatch variables corresponding to every node in $\mathcal{N}^D$. The purpose of the artificial model $\mathtt{R}'$ is to demonstrate the relationship between $\mathtt{B}(\mathcal{N}^D)$ and $\mathtt{R}$ in a simpler manner. Then, we prove that $Z_B \leq Z_R$ by showing $Z_B \leq Z_R'$ and $Z_R' \leq Z_R$.

First, we prove that $Z_B \leq Z_R'$ by demonstrating that any feasible solution of $\mathtt{R}'$ can be transformed into a feasible solution of $\mathtt{B}(\mathcal{N}^D)$. With a slight abuse of the notation, let $\boldsymbol{y}^{node} = (y_{gn}, y_n^M, y_n^{SH}, y_{en}^{SOC}, y_{en}^{CH}, y_{en}^{DCH})_{g\in\mathcal{G},e\in\mathcal{E},n\in\mathcal{N}^D}$. Then, a feasible solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}^{node}, \bar{\boldsymbol{z}})$ is constructed with $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}})$ satisfying (4.4b), $(\bar{y}_{gn}, \bar{y}_n^M, \bar{y}_n^{SH}, \bar{y}_{en}^{CH}, \bar{y}_{en}^{DCH})_{g\in\mathcal{G},e\in\mathcal{E},n\in\mathcal{N}^D}$ satisfying (4.4c)-(4.4h), and $\bar{y}_{en}^{SOC}$ derived from $\bar{y}_{en}^{CH}$ and $\bar{y}_{en}^{DCH}$ as follows:

$$\bar{y}_{en}^{SOC} = \sum_{m\in\mathcal{P}(n)} \left(\eta_e \bar{y}_{em}^{CH} - \frac{1}{\eta_e} \bar{y}_{em}^{DCH}\right)$$

for $e \in \mathcal{E}, n \in \mathcal{N}^D$. We demonstrate that $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}^{node}, \bar{\boldsymbol{z}})$ is feasible for $\mathtt{B}(\mathcal{N}^D)$. For

$g \in \mathcal{G}$ and $n \in \mathcal{N}^D$, since $\bar{z}^{\min}_{g,t(n)} \le \bar{y}_{gn} \le \bar{z}^{\max}_{g,t(n)}$,

$$
\begin{aligned}
&P^{\min}_g \bar{x}_{g,t(n)} \le \bar{z}^{\min}_{g,t(n)} \le \bar{y}_{gn} \le \bar{z}^{\max}_{g,t(n)} \le P^{\max}_g \bar{x}_{g,t(n)},\\
&\bar{y}_{gn} - \bar{y}_{g,n^-} \le \bar{z}^{\max}_{g,t(n)} - \bar{z}^{\min}_{g,t(n^-)} \le RU_g,\\
&\bar{y}_{g,n^-} - \bar{y}_{gn} \le \bar{z}^{\max}_{g,t(n^-)} - \bar{z}^{\min}_{g,t(n)} \le RD_g.
\end{aligned}
$$

For $e \in \mathcal{E}$ and $n \in \mathcal{N}^D$, we know that $\bar{z}^{CH,\min}_{e,t(n)} \le \bar{y}^{CH}_{en} \le \bar{z}^{CH,\max}_{e,t(n)}$ and $\bar{z}^{DCH,\min}_{e,t(n)} \le \bar{y}^{DCH}_{en} \le \bar{z}^{DCH,\max}_{e,t(n)}$. Then,

$$
\begin{aligned}
&\bar{z}^{CH,\min}_{e,t(n)} \le \bar{y}^{CH}_{en} \le \bar{z}^{CH,\max}_{e,t(n)} \le P^{CH,\max}_e,\\
&\bar{z}^{DCH,\min}_{e,t(n)} \le \bar{y}^{DCH}_{en} \le \bar{z}^{DCH,\max}_{e,t(n)} \le P^{DCH,\max}_e.
\end{aligned}
$$

Also,

$$
\begin{aligned}
\bar{y}^{SOC}_{en} &= \sum_{m \in \mathcal{P}(n)} \left(\eta_e \bar{y}^{CH}_{em} - \frac{1}{\eta_e} \bar{y}^{DCH}_{em} \right) \le \sum_{m \in \mathcal{P}(n)} \left(\eta_e \bar{z}^{CH,\max}_{e,t(m)} - \frac{1}{\eta_e} \bar{z}^{DCH,\min}_{e,t(m)} \right) \le E^{\max}_e,\\
\bar{y}^{SOC}_{en} &= \sum_{m \in \mathcal{P}(n)} \left(\eta_e \bar{y}^{CH}_{em} - \frac{1}{\eta_e} \bar{y}^{DCH}_{em} \right) \ge \sum_{m \in \mathcal{P}(n)} \left(\eta_e \bar{z}^{CH,\min}_{e,t(m)} - \frac{1}{\eta_e} \bar{z}^{DCH,\max}_{e,t(m)} \right) \ge E^{\min}_e,
\end{aligned}
$$

and

$$
\begin{aligned}
&\sum_{k=1}^{T} \left(\eta_e \bar{z}^{CH,\min}_{ek} - \frac{1}{\eta_e} \bar{z}^{DCH,\max}_{ek} \right) \le \bar{y}^{SOC}_{en}, \quad \forall n \in \mathcal{N}^D : t(n) = T,\\
&\bar{y}^{SOC}_{en} \le \sum_{k=1}^{T} \left(\eta_e \bar{z}^{CH,\max}_{ek} - \frac{1}{\eta_e} \bar{z}^{DCH,\min}_{ek} \right), \quad \forall n \in \mathcal{N}^D : t(n) = T.
\end{aligned}
$$

Therefore, $\bar{y}^{SOC}_{en} = TE_e$ when $t(n) = T$. Because the two objective functions are equal, we obtain $Z_B \le Z'_R$.

Second, we demonstrate that $Z'_R \leq Z_R$ by transforming any feasible solution in $\mathtt{R}$ into a feasible solution in $\mathtt{R}'$ with the same objective value. Let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}^{branch}, \hat{\boldsymbol{z}})$ be a feasible solution to $\mathtt{R}$, where $\boldsymbol{y}^{branch} = (y_{gst}, y^{SH}_{st}, y^{CH}_{est}, y^{DCH}_{est})_{g\in\mathcal{G}, e\in\mathcal{E}, s\in S^D_t, t\in\mathcal{T}}$. Subsequently, $\hat{\boldsymbol{x}} \in \mathcal{X}$ and $\hat{\boldsymbol{z}} \in \mathcal{Z}(\hat{\boldsymbol{x}})$. For a given $s \in S^D_t$ and $t \in \mathcal{T}$, define $N(k, s, t) := \{n \in \mathcal{N}^D \mid t(n) = t \text{ and } D_n = D^s_t\}$ to represent a set of nodes in $\mathcal{N}^D$ that corresponds to realization s at time period t. Then, for $g \in \mathcal{G}, e \in \mathcal{E}, s \in S^D_t, t \in \mathcal{T}$, a solution corresponding to each node in $\mathtt{B}(\mathcal{N}^D)$ can be constructed by letting $\hat{y}_{gn} = \hat{y}_{gst}, \hat{y}^{SH}_n = \hat{y}^{SH}_{st}, \hat{y}^{CH}_{en} = \hat{y}^{CH}_{est}, \hat{y}^{DCH}_{en} = \hat{y}^{DCH}_{est}$ that satisfies $n \in N(k, s, t)$. Because $(\hat{\boldsymbol{y}}^{branch}, \hat{\boldsymbol{z}})$ satisfies (4.3d)-(4.3g) and $\hat{\boldsymbol{y}}^{branch}$ satisfies (4.3b) and (4.3c), we can easily check that $(\hat{\boldsymbol{y}}^{node}, \hat{\boldsymbol{z}})$ satisfies (4.4e)-(4.4h) and $\hat{\boldsymbol{y}}^{node}$ satisfies (4.4c) and (4.4d). Therefore, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}^{node}, \hat{\boldsymbol{z}})$ is feasible for $\mathtt{R}'$. Furthermore, as $q^D_{st} = \sum_{n\in N(k,s,t)} p^D_n$ holds because the probability for each realization is independent of the time period, the following holds:

$$
\begin{aligned}
&\sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} c^{ST}_g x^{ST}_{gt} + \sum_{t\in\mathcal{T}}\sum_{s\in S^D_t} q^D_{st}\left(\sum_{g\in\mathcal{G}} c^V_g y_{gst} + \rho_t y^M_{st} + K_t y^{SH}_{st}\right) \\
=&\sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} c^{ST}_g x^{ST}_{gt} + \sum_{t\in\mathcal{T}}\sum_{s\in S^D_t}\sum_{n\in N(k,s,t)} p^D_n\left(\sum_{g\in\mathcal{G}} c^V_g y_{gn} + \rho_t y^M_n + K_t y^{SH}_n\right) \\
=&\sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} c^{ST}_g x^{ST}_{gt} + \sum_{n\in\mathcal{N}^D} p^D_n\left(\sum_{g\in\mathcal{G}} c^V_g y_{gn} + \rho_t y^M_n + K_t y^{SH}_n\right).
\end{aligned}
$$

□

We note the important implications of Proposition 4.1. First, the results demonstrate the conversion of an optimal solution of $\mathtt{R}$ to a feasible solution of $\mathtt{B}(\mathcal{N}^D)$, which directly indicates that one can obtain an adaptable solution to the uncertainty

that is sequentially realized by solving R instead of B($\mathcal{N}^D$). Since the number of decision variables in the former is $\mathcal{O}(\sigma T)$, compared to $\mathcal{O}(\sigma^T)$ in the latter, model R significantly reduces the size of the problem. In addition, the difference in objective values indicates that the optimal solution of R has relatively restricted flexibility compared to B($\mathcal{N}^D$) because the constraints (4.3d)-(4.3f) in R require a feasible region in which the adaptable decision variables should reside. However, this does not mean deterioration of the obtained solutions; alternatively, robustness and enhanced feasibility can be obtained in that the feasible operational ranges can guide the decision of the dispatch level, which needs to be analyzed in the computational sense.

4.3.3 Integrated optimization models for both islanding and net load uncertainty

The scalable optimization approaches proposed in Sections 4.3.1 and 4.3.2 can be used separately when only one of the two factors is uncertain. However, the main goal of this study is to operate a microgrid in a scalable manner, even when both islanding and net load are uncertain. Therefore, we propose integrated optimization approaches to operate a microgrid under stochastic islanding and net load, which are based on combining the two frameworks in Sections 4.3.1 and 4.3.2.

Overall, the key is to extend the optimization models BL and RP(k) for $k \in \mathcal{T}$ to incorporate the net load uncertainty. To do this, the decision variables and constraints of model R are first extended to incorporate all the nodes in $\mathcal{N}^I(k)$. For each node $m \in \mathcal{N}^I(k)$, the corresponding connection state is given as $I_m \in \{0, 1\}$, and the probability is given as $p'_{stm} = p^I_m q^D_{st}$ for $s \in S^D_t, t \in \mathcal{T}_k$, considering the

independence of the two uncertain factors. In addition, decision variables $\boldsymbol{y}^{branch}$ and $\boldsymbol{z}$ are extended with the new component $m \in \mathcal{N}^I(k)$. We define set $\mathcal{Z}_m(\boldsymbol{x})$ as the extension of $\mathcal{Z}(\boldsymbol{x})$ with $m \in \mathcal{N}(k)$. In $\mathcal{Z}_m(\boldsymbol{x})$, all constraints (4.2a)-(4.2h) of $\mathcal{Z}(\boldsymbol{x})$ are defined for a given $m \in \mathcal{N}^I(k)$ with the extended variable $\boldsymbol{z}$. Then, the integrated model $\mathtt{I}(k)$ for $k \in \mathcal{T}$, which is used for both the baseline and replan models, is as follows:

$\mathtt{I}(k)$:

$$\min \sum_{t\in\mathcal{T}_k}\sum_{g\in\mathcal{G}} c_g^{ST} x_{gt}^{ST} + \sum_{m\in\mathcal{N}^I(k)}\sum_{t\in\mathcal{T}_k}\sum_{s\in S_t^D} p'_{stm}\left(\sum_{g\in\mathcal{G}} c_g^V y_{gstm} + \rho_t y_{stm}^M + K_t y_{stm}^{SH}\right)$$

$$\text{s.t. } \boldsymbol{x} \in \mathcal{X},$$

$$\boldsymbol{z} \in \mathcal{Z}_m(\boldsymbol{x}), \quad \forall m \in \mathcal{N}^I(k),$$

$$\sum_{g\in\mathcal{G}} y_{gstm} + \sum_{e\in\mathcal{E}}(y_{estm}^{DCH} - y_{estm}^{CH}) + y_{stm}^M + y_{stm}^{SH} \geq D_t^s, \ \ \forall s \in S_t^D, t \in \mathcal{T}_k, m \in \mathcal{N}^I(k),$$

$$-I_m P_M \leq y_{stm}^M \leq I_m P_M, \quad \forall s \in S_t^D, t \in \mathcal{T}_k, m \in \mathcal{N}^I(k),$$

$$z_{gtm}^{\min} \leq y_{gstm} \leq z_{gtm}^{\max}, \quad \forall g \in \mathcal{G}, s \in S_t^D, t \in \mathcal{T}_k, m \in \mathcal{N}^I(k),$$

$$z_{etm}^{CH,\min} \leq y_{estm}^{CH} \leq z_{etm}^{CH,\max}, \quad \forall e \in \mathcal{E}, s \in S_t^D, t \in \mathcal{T}_k, m \in \mathcal{N}^I(k),$$

$$z_{etm}^{DCH,\min} \leq y_{estm}^{DCH} \leq z_{etm}^{DCH,\max}, \quad \forall e \in \mathcal{E}, \forall s \in S_t^D, t \in \mathcal{T}_k, m \in \mathcal{N}^I(k),$$

$$y_{gstm}, y_{stm}^{SH}, y_{estm}^{CH}, y_{estm}^{DCH} \geq 0, \quad \forall g \in \mathcal{G}, e \in \mathcal{E}, s \in S_t^D, t \in \mathcal{T}_k, m \in \mathcal{N}^I(k).$$

We now demonstrate how one can operate a microgrid using models based on $\mathtt{I}(k)$ for $k \in \mathcal{T}$, which is similar to those presented in Section 4.3.1. First, model $\mathtt{I}(1)$ is solved before the start of the planning horizon. When islanding events occur no more than once, the range decision $\boldsymbol{z}$ remains unchanged over the entire planning horizon.

In addition, regardless of islanding events, a single-period dispatch problem is solved to derive the exact dispatch decisions under any realized net load $d_t \in \mathbb{R}_+$ for each period $t \in \mathcal{T}$. Note that it is a simple linear programming problem to determine the dispatch amount in a given range that satisfies the net load, which is written as follows:

$$
\begin{aligned}
\min \quad & \sum_{g \in \mathcal{G}} c_g^V y_g + \rho_t y^M + K_t y^{SH} \\
\text{s.t.} \quad & \sum_{g \in \mathcal{G}} y_g + \sum_{e \in \mathcal{E}} \left(y_e^{DCH} - y_e^{CH} \right) + y^M + y^{SH} \geq d_t, \\
& -\bar{I}_t P^M \leq y^M \leq \bar{I}_t P^M, \\
& z_{gt}^{\min} \leq y_g \leq z_{gt}^{\max}, \quad \forall g \in \mathcal{G}, \\
& z_{et}^{CH,\min} \leq y_e^{CH} \leq z_{et}^{CH,\max}, \quad \forall e \in \mathcal{E}, \\
& z_{et}^{DCH,\min} \leq y_e^{DCH} \leq z_{et}^{DCH,\max}, \quad \forall e \in \mathcal{E}, \\
& y_g, y^{SH}, y_e^{SOC}, y_e^{CH}, y_e^{DCH} \geq 0, \quad \forall g \in \mathcal{G}, e \in \mathcal{E},
\end{aligned}
$$

where $\bar{I}_t$ is a realized connection state for period t. Note that the problem is always feasible because the given range variables are determined in advance to satisfy the constraints linked among periods. When the first islanding event ends at a certain time period $k \in \mathcal{T}$, the range decisions from period $k+1$ are updated to address the possibility of the next islanding event. This can be achieved by solving a replanning model, which is based on model $\mathtt{I}(k+1)$ with initial conditions (i.e. decisions at period k and on/off decisions for the entire planning horizon), which is a linear programming problem.

Finally, the number of variables and constraints of the integrated model is

$\mathcal{O}\left(\sigma \cdot T^3\right)$, which is much smaller compared to $\mathcal{O}\left((2\sigma)^T\right)$ for the standard multistage model $\mathtt{B}(\mathcal{N})$ in Section 4.2. This makes the proposed model of practical size that can be solved in a reasonable amount of computation time.

4.4 Computational experiments

Computational experiments were conducted to demonstrate the effectiveness and efficiency of the proposed models. First, the scalability and quality of the solutions from the models were evaluated and compared with those of the standard multistage model. In addition, solutions from the integrated model, considering the uncertainty of both factors, were evaluated for practical-sized instances. A microgrid with four thermal generators and one BESS was tested, where the configuration, nominal net load data, and electricity price data for 24 time periods were from Lee et al. (2021), and the details of generators and BESSs are presented in Tables 4.3 and 4.4. In addition, the value of lost load (K_t) was set to 5,000 \$/MWh for every time period and the maximum amount of transaction with the main grid (P^M) was set to 10 MW.

In the experiments, uncertain islanding events were regarded to follow a binomial distribution, with *islanding probability* p for each time period, where $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. In addition, to represent the uncertain net load, we define parameters σ and δ to describe each period's realization in a scenario tree. The *branch* parameter σ is defined as the number of realizations, and the *deviation* parameter δ indicates how the realizations differ from each other. For simplicity, we assume that the number of realizations is time-invariant; in other words, $\sigma_1 = \cdots = \sigma_T = \sigma$ and $S_t^D = \{D_t^1, D_t^2, \ldots, D_t^\sigma\}$ for $t \in \mathcal{T}$. Then, we assign each net load value in S_t

Table 4.3: Characteristics of thermal generators

Generators	G1	G2	G3	G4
Unit Cost [$/MWh]	27.7	39.1	61.3	65.6
Min-Max Capacity [MW]	2-10	1-5	1-5	0.8-3
Min Up/Down Time [h]	3	3	3	1
Ramp Up/Down Rate [MW/h]	4	3	3	2.5
Start-up Cost [$]	50	20	20	5
Initial State [h]	5	3	-3	-1

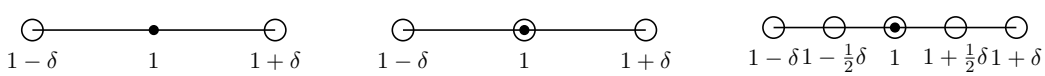
Table 4.4: Characteristics of BESS

BESS	Capacity [MWh]	Max Charging/ Discharging Power [MW]	SOC Operation Range [%]	Initial-Target SOC [%]	Charging/ Discharging Efficiency [%]
E1	10	5	10-90	50	90

by $D_t^i = \gamma_i \bar{D}_t$, where $\bar{D}_t$ is the nominal net load value and the set of coefficients $\{\gamma_i\}_{i\in\{1,\ldots,\sigma\}}$ is affected by the parameter δ. In the experiment, the uncertain net load is assumed to follow a uniform distribution, specifically, $\gamma \sim U[1-\delta, 1+\delta]$. For each given σ and δ combination, specific coefficients are generated as follows and shown in Figure 4.4.

- $\sigma = 2 \rightarrow \gamma_1 = 1-\delta$, $\gamma_2 = 1+\delta$
- $\sigma = 2k+1$ for $k \in \mathbb{Z}_+ \rightarrow \gamma_i = 1 + (i-k-1)\delta/k$ for all $i = 1, \ldots, 2k+1$

The probability of each realization is assigned equal to each other for all realizations.



(a) $\sigma = 2$ (b) $\sigma = 3$ (c) $\sigma = 5$

Figure 4.4: Illustration of parameters σ and δ on each period's net load realizations

All models were implemented with C++ using a PC with 32GB RAM, where CPLEX 20.1 with its default setting was used to solve mixed-integer optimization problems. We set the time limit for solving each optimization model to 600 s. In the remaining section, the effects of the proposed optimization models under islanding events are evaluated in Section 4.4.1, and those under net load uncertainty are demonstrated in Section 4.4.2. The effectiveness of the integrated models under both uncertain factors is tested in Section 4.4.3.

4.4.1 Effectiveness of proposed model with replanning procedure under islanding uncertainty

In this subsection, to evaluate the proposed models under islanding uncertainty, we assume that the net load is certain and is given as a nominal value for the planning horizon. First, the computation times of the proposed and standard multistage models were analyzed to demonstrate the scalability of the proposed approach. Specifically, the proposed baseline model (`BL` in Section 4.3.1) and standard multistage model ($\texttt{B}(\mathcal{N}^I)$ in Section 4.2) were tested. The average computation times among the five p values for $T \in \{6, \ldots, 24\}$ are demonstrated in Figure 4.5. It can be observed from the figure that the computational time exponentially increases as T increases for the standard multistage model because the number of scenarios is $\mathcal{O}(2^T)$. Consequently, it was impossible to solve the model when $T \geq 17$ because of the out-of-memory status in the commercial solver. On the contrary, the computation time of model `BL` gradually increases as T increases, and it can be easily solved even when $T = 24$ because the size of the model is $\mathcal{O}(T^2)$.

Next, the solutions of the entire planning horizon from the two types of opti-

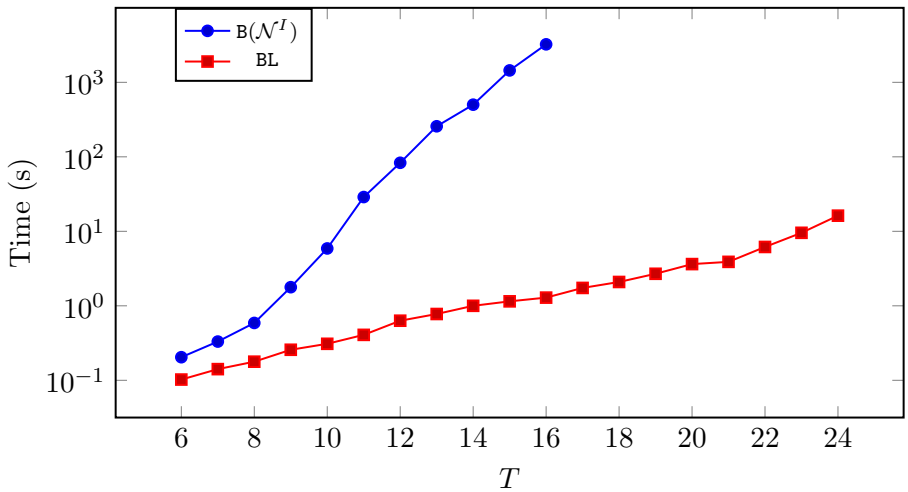


Figure 4.5: Comparison of the computation times for different numbers of time periods

Table 4.5: Cost statistics ($) for all possible islanding cases for $T = 12$

	Standard method					Proposed method				
p	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
exp	6,334.7	6,522.2	6,675.2	6,837.3	6,987.7	6,370.2	6,569.0	6,711.5	6,925.0	7,046.8
stdev	550.4	424.8	471.4	292.0	224.2	732.7	924.1	507.8	429.2	273.1

mization models were evaluated and compared. Recall that the solutions from the proposed baseline model BL itself are not sufficient to deal with all possible islanding cases because at most one islanding event is considered in the model. Thus, we can obtain solutions adaptable to all islanding cases by implementing the replanning procedure mentioned in Algorithm 4.1, which we call the *proposed method*. In contrast, in the *standard method*, decisions adaptable for all possible islanding cases are obtained simultaneously by solving B($\mathcal{N}^I$).

The operational costs of the two methods were compared by evaluating the costs for all possible cases, where the number is 2^T, for $T = 12$, because the standard multistage model cannot be solved for large T values, as shown in Figure 4.5. The

expected costs (exp) and standard deviations (stdev) for all islanding cases are reported in Table 4.5. From the table, it can be seen that when p increases, the expected costs increase in both models. This is because when the risk of islanding increases, more generation is required to meet its net load; otherwise, load shedding is required. On the contrary, the variances in costs tend to decrease as p increases, showing the robustness of the model BL as p increases. In addition, the results demonstrate that although the expected costs of the proposed method are slightly higher than those of the standard multistage model, the differences are small, with an average gap of 0.8% for all p values.

4.4.2 Effectiveness of proposed model under net load uncertainty

Here, we describe the examination of the proposed scalable optimization models under net load uncertainty, where the connection states are assumed to be given in advance. First, we demonstrated the efficiency of the proposed models for various numbers of time periods and branches, by comparing the computation times of the proposed model R to the standard multistage model B($\mathcal{N}^D$). Tests were conducted for $T \in \{4, \ldots, 24\}$ and $\sigma \in \{2, 3, 5\}$, of which the average computation times for $\delta \in \{0.1, 0.2, 0.3\}$ are reported here. The results in Figure 4.6 emphasize the scalability of the proposed model R. It can be seen that R obtained the optimal solution in a substantially short amount of time, even when $T = 24$. Here, less than 1 second was required to solve the models. However, it is not practical to use the standard multistage model for real-sized instances, because the computation time exponentially increases when T increases. In addition, it is more difficult to solve using a standard multistage model when the number of branches increases. For the

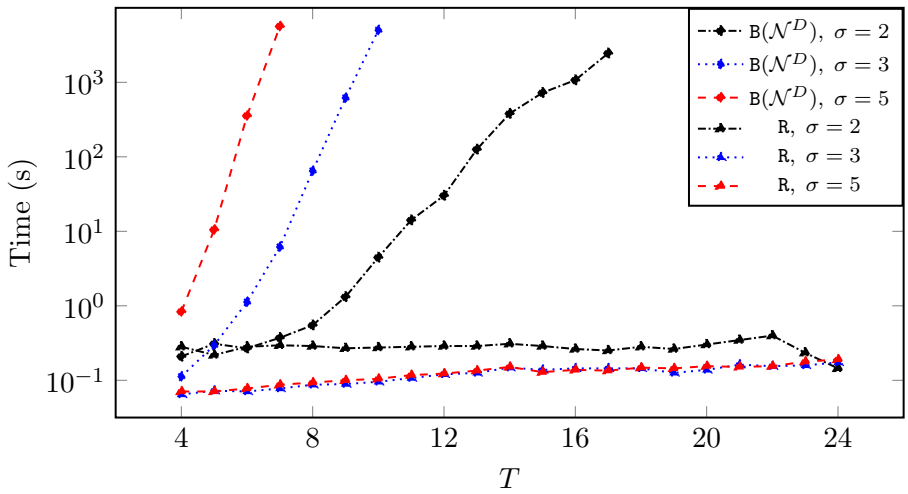


Figure 4.6: Comparison of the computation times for various numbers of branches

standard multistage model, the times can only be plotted for a restricted number of time periods; otherwise, the model cannot be completed because of time or memory limits.

Next, we compared the two objective values from the two models, R and B($\mathcal{N}^D$), by calculating the relative objective value of the former compared to the latter. As shown in Section 4.3.2, the objective value of R is not less than that of B($\mathcal{N}^D$) because the dispatch decisions are more flexible for the latter. Figure 4.7 shows the relative difference for $\sigma = 2$, $\delta \in \{0.1, 0.2, 0.3\}$, where T is within the values at which B($\mathcal{N}^D$) can be solved. In the figure, each value is calculated as

$$\text{Difference}(\%) = 100 \times \frac{z_{\mathtt{R}} - z_{\mathtt{B}(\mathcal{N}^D)}}{z_{\mathtt{B}(\mathcal{N}^D)}},$$

where z_M is the optimal objective value of model M. We show that the differences are larger when the deviation increases. In addition, although there is a difference between the two, it is quite small (less than 1% for $\delta \leq 0.2$). Combining the results

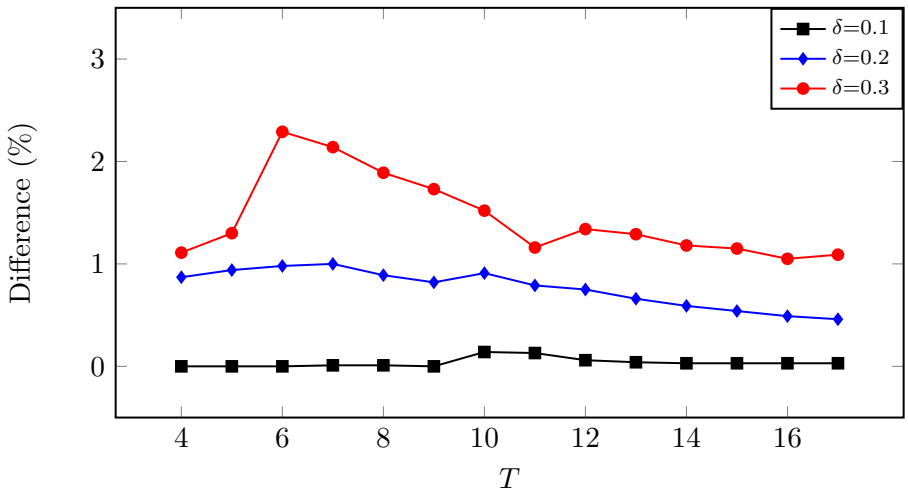


Figure 4.7: Relative objective values of the proposed model for $\sigma = 2$

in Figure 4.6, the proposed models can also yield similar solutions compared with the standard multistage model, with a superior computation time.

4.4.3 Effectiveness of integrated model under both uncertain factors

Finally, we demonstrate how an integrated model combining both methodologies performs when both factors are uncertain for $T = 24$. First, computational aspects of the proposed models were evaluated for various p, δ, and σ values. Recall that the standard multistage model, which has an exponential number of variables and constraints, cannot be solved for $T = 24$ even when only one of the two factors is uncertain. However, the proposed models can be solved in a reasonable amount of time, as shown in Figure 4.8. In the figure, the computation times for solving the baseline model $\mathtt{I}(1)$ are shown. From the figure, the effects on σ can be observed, resulting in a higher computational burden as σ increases. On the contrary, the

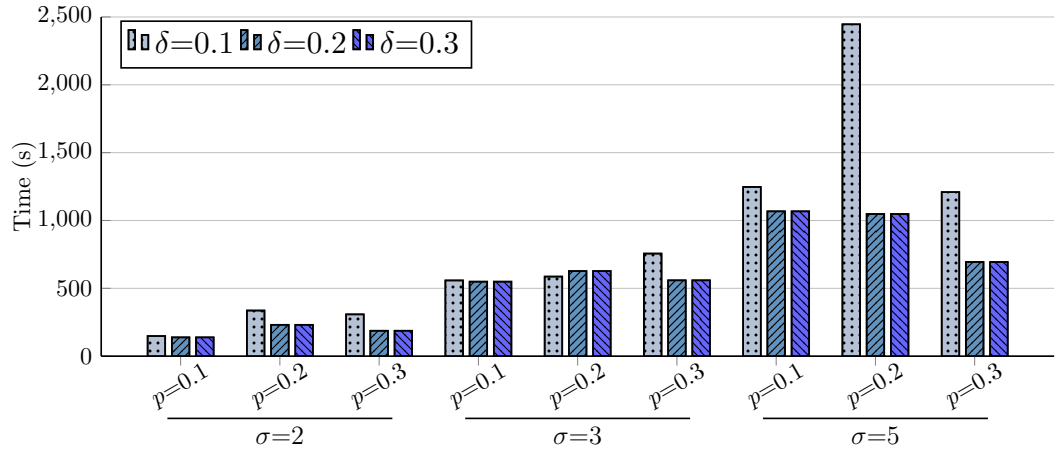


Figure 4.8: Computation times of the proposed baseline models for both uncertainty for $T = 24$

impact of parameters p and δ is not significant; only slight reductions in computation times are observed as δ increases. Finally, we emphasize that the models can be solved within one hour for any combination, which highlights the practical usefulness of the proposed approaches.

Further, out-of-sample tests were conducted to show how the solutions from the proposed optimization models are effective over unexpected realizations. For the islanding uncertainty, test scenarios were obtained from a total 2^{24} number of possible cases. For the net load uncertainty, test scenarios were obtained from the uniform distribution, which is $U[1-\delta, 1+\delta]$ for $\delta = 0.1, 0.2$, and 0.3. Overall, 1,000 test scenarios were sampled from the joint sample space of both. For each sample, the operation cost over the planning horizon was calculated using the methods based on models $\texttt{I}(k)$ for $k \in \mathcal{T}$ in Section 4.3.3. Two types of costs, `base` and `exp`, were calculated for the out-of-sample evaluations as follows: First, `base` refers to the cost when no islanding and no net load deviation occurs, and is obtained by solving the

baseline model. Then, `exp` is the expected operation costs of 1,000 test scenarios. The results in Figure 4.9 demonstrate the two costs under various islanding probabilities and deviation parameters when $\sigma = 2$. First, the expected operation costs increase as p or δ increases, which indicates that higher penetration of uncertainty increases the operation costs. In particular, the expected costs become much higher when both factors are uncertain. On the contrary, the baseline costs are less affected by the parameters p and δ, showing only slight increases when δ increases. Next, cost statistics were obtained for various σ values to determine the impact of σ on the operation costs. The lower quartile (`Q1`), median (`Q2`), and upper quartile (`Q3`) of operation costs among 1,000 scenarios are shown in Table 4.6. From the table, it can be seen that the operation costs, especially median costs, slightly decrease as σ increases. This implies enhanced accuracy of the distribution estimation as the number of samples increases. However, for a small σ, the cost differences between upper and lower quartile values are small, which indicates the robustness of the on/off decisions. In general, it can also be shown that the overall costs increase as p or δ increases, incurring significant cost damage when both parameter values are large.

4.5 Summary

In this chapter, scalable optimization models are developed for microgrid operation under stochastic islanding events and net load. The main purpose of this study is to significantly reduce the computational burden of solving practical-sized instances with both factors. For uncertain islanding events, we developed optimization models along with the replanning procedure based on node sets having a reduced number of

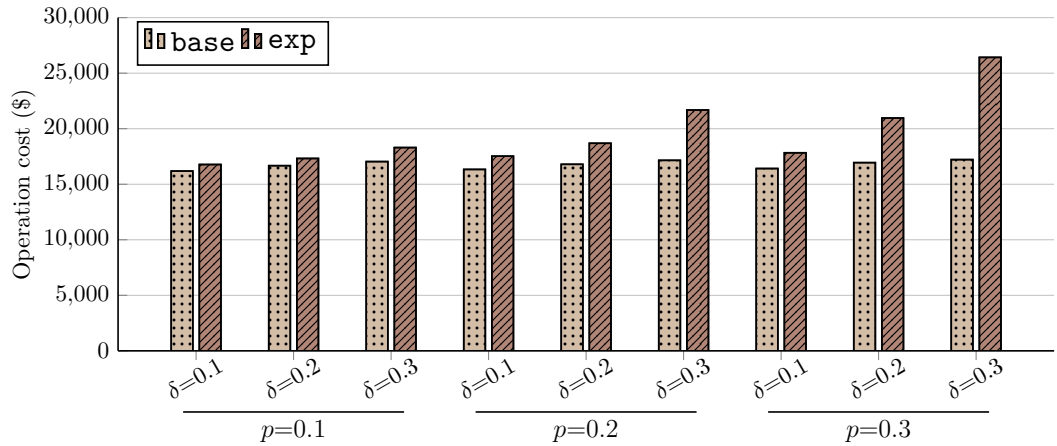


Figure 4.9: Baseline and expected operation costs for $T = 24$ and $\sigma = 2$

Table 4.6: Operation cost statistics ($) under both uncertain factors for $T = 24$

p	δ	σ=2			σ=3			σ=5		
		Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
0.1	0.1	16,109.0	16,375.3	16,668.3	16,028.0	16,301.2	16,590.4	16,038.6	16,313.0	16,609.3
	0.2	16,403.4	16,872.7	17,442.9	16,288.1	16,782.8	17,325.6	16,197.0	16,701.2	17,271.9
	0.3	16,621.4	17,383.6	18,278.8	16,525.2	17,289.6	18,207.8	16,475.6	17,231.9	18,144.5
0.2	0.1	16,455.5	16,769.2	17,117.3	16,408.6	16,726.0	17,088.0	16,402.1	16,722.3	17,074.7
	0.2	16,816.1	17,389.4	18,079.3	16,805.1	17,380.0	18,088.3	16750.6	17,320.7	18,068.8
	0.3	17,048.1	18,089.2	21,329.0	17,029.8	18,045.6	20,996.4	16,993.9	18,036.8	21,236.2
0.3	0.1	16,759.6	17,089.8	17,509.6	16,718.3	17,049.8	17,478.5	16,710.5	17,050.0	17,502.4
	0.2	17,124.7	17,849.3	20,140.3	17,121.8	17,837.6	20,231.6	17,089.9	17,795.0	20,185.8
	0.3	17,576.3	19,281.5	32,265.9	17,526.3	19,184.4	32,205.1	17,497.7	19,201.8	32,285.1

nodes, which is quadratically dependent on the number of time periods. In addition, we developed scalable optimization models under stochastic net load based on the introduction of range variables to redefine dispatch variables for each period. Lastly, we proposed the integrated models to operate a microgrid when both islanding and net load are uncertain. Through various numerical experiments, we demonstrated the effectiveness of the solutions and the strong advantage in the scalability of the proposed models, compared to the standard multistage model. The practical usefulness of the integrated models was also demonstrated.

Chapter 5

Conclusion

5.1 Summary and contributions

In this dissertation, we investigated various optimization problems for power system operation under uncertainty. Since it is computationally demanding when various uncertain factors are incorporated into the optimization models, we focused on efficient optimization models and solution approaches to reduce the computational burden of optimization models. The proposed solution approaches are based on decomposition to deal with large-scale optimization model with usually comes from the number of possible realizations. We also presented efficient optimization models which can be used stand-alone or efficient solution approaches that can easily be applied. We summarize the contributions of each chapter.

In Chapter 2, we focus on a generic two-stage stochastic program with finite support in the literature, that is also widely used in power system operation. Although it is widely used, the large number of scenarios makes it hard to solve the model, and there also are potential risks of inaccurate estimation of the underlying distribution. To mitigate the drawbacks, we proposed a novel model, which is a partition-based risk-averse two-stage stochastic program based on a partition and risk function. Various risk-averse two-stage models can be represented with the proposed model, and

various levels of risks can be controlled by changing the partition. We propose an efficient and generic solution approach based on column-and-constraint generation to solve the model with a given partition. We also devised partitioning algorithms for constructing a partition to enable the risk of the model closest to the pre-defined target. We also presented partitioning schemes to incorporate the proposed partitioning algorithm into the proposed column-and-constraint generation. Numerical experiments demonstrate that the models with the proposed partitioning methods induce significantly low deviation from the target, compared to the other methods. Further, the proposed column-and-constraint generation outperforms the commercial solver or the algorithm based on the literature, and a controllable parameter Γ is also useful for reducing computation times while obtaining a significantly small optimality gap. Lastly, the computational aspects of the adaptive partitioning scheme show its potential utility over an *a priori* partitioning scheme.

In Chapter 3, we investigate 1UC problems that arise when an individual power producer submits its schedule or commitment state to the electricity market. Specifically, we devised efficient dynamic programming algorithms for two types of 1UC problems: self-scheduling and self-commitment problems. First, to deal with the self-scheduling problem with stochastic electricity prices, we proposed algorithms that focus on the dispatch subproblem to efficiently handle a number of scenarios. Second, for the self-commitment problem, which decides only the on/off status of a generator to maximize profit, we propose an algorithm with reduced computational complexity to other existing methods. Then, we presented two unit decomposition approaches to deal with the general UC problem with stochastic net load, which are decomposition methods that use 1UC problems as subproblems. The approaches include a

novel approach, which employs the self-commitment problem and has not been addressed before. We presented Lagrangian relaxation and column generation methods to implement each unit decomposition method. Through numerical experiments, the efficiency of the proposed algorithms for 1UC problems is demonstrated for various numbers of scenarios and time periods. In addition, we compared the efficiency of various unit decomposition methods to solve the UC problem under stochastic net load, and we emphasize the scalability of the novel unit decomposition methods when the number of scenarios increases.

In Chapter 4, we study operating microgrids under stochastic islanding events and net load. Considering those uncertain factors are sequentially realized in the planning horizon, a standard multistage stochastic optimization model is widely used. However, since the size of the model is exponential with the number of possible realizations in a period, we proposed scalable optimization models to solve practical-sized instances. For uncertain islanding events, we developed optimization models along with the replanning procedure based on node sets having a reduced number of nodes, which is quadratically dependent on the number of time periods. In addition, we developed scalable optimization models under stochastic net load based on the introduction of range variables to redefine dispatch variables for each period. Lastly, we proposed the integrated models to operate a microgrid when both islanding and net load are uncertain. Through various numerical experiments, we first emphasized the strong advantage in the scalability of the proposed models to solve practical-sized instances, compared to the standard multistage model which cannot. We also demonstrated the effectiveness of the models in the environment that uncertain factors sequentially realize.

The optimization models and solution approaches proposed for the two-stage stochastic program in Chapter 2 are generic and thus have various applications. The proposed methodology can be used to solve other optimization problems in power systems, such as capacity expansion planning, and nuclear outage planning problems. In addition, it can be used to solve problems in other fields, such as supply chain management, transportation system, and so on. The additional computational results in Appendix A, which are done for the facility location problem, indicate the applicability of the proposed methodology.

On the other hand, the optimization models and solution approaches in Chapters 3 and 4 are proposed considering specific characteristics in power systems. However, it can also be applied to solve other optimization problems that have a similar structure. For example, since the unit decomposition methods in Chapter 3 are based on relaxing the system-wise constraints, it can be related to the production systems where a set of machines are coordinated to meet the customer demands of items. In addition, since production ramping constraints can also exist in machines (see Damcı-Kurt et al., 2016), the scalable optimization models that mitigate the computational burden by exploring ramping requirements can be applied in the lot-sizing and scheduling problem context.

5.2 Future research directions

Although various optimization models and decomposition approaches are proposed in the dissertation, there is room for further research. We close the dissertation by presenting some points of limitation and extension, which is worthwhile for the follow-up studies.

First, for the proposed novel optimization model which we call PSP in Chapter 2, various risk functions other than the CVaR function can be considered. As mentioned, using the relationship between risk-averse stochastic programming and DRO, the various risk functions in PSP correspond to the various ambiguity sets in DRO. Whether the proposed column-and-constraint generation approach can be applied in those contexts and is efficient compared to other existing solution approaches is worthwhile to investigate. In addition, the proposed partitioning criteria and problems can be further investigated. There can be various criteria on how to make a partition, and the clarity of partitioning criteria and ease of the resulting partitioning problem are also important. Finally, the proposed partitioning methods could be elaborated for effective embedding in a multi-stage stochastic program, where they would need to be adapted to settings in which scenarios have complex relationships.

Next, the single-generator system can be further investigated, which is worthwhile to explore not only to solve 1UC problems efficiently but also to enhance the solvability of UC problems. For deterministic 1UC problems, various classes of strong valid inequalities are developed and compact extended formulations that can describe the convex hull of the feasible solution in higher dimensions are also proposed in the literature. However, the classes of valid inequalities that can sufficiently characterize the convex hull in the original space have not been proposed, to the best of our knowledge. If it is possible with efficient separation, it can be utilized to reduce the computational burden of UC problems. Next, algorithms for the self-scheduling problem can be further explored. Since the single-generator dispatch problem that arises in the self-scheduling problem is a linear program with a

relatively simple structure, there may be another efficient algorithm for it. Lastly, 1UC problems with various uncertainty modeling frameworks can be investigated, and whether the corresponding unit decomposition framework to the UC problems can be efficient also needs to be analyzed.

Further, models and solution approaches for optimization under the sequential realization of uncertainty can be further investigated. In this case, scalability should be carefully considered to mitigate the so-called curse of dimensionality. The optimization models in Chapters 2 and 3, which are based on a two-stage setting, can be extended to the multistage setting that the uncertain factors such as net load sequentially realize. The scalable optimization models in microgrid operation in Chapter 4 is one efficient method to deal with the environment that needs sequential decision-making. Although it has shown superior scalability compared to the standard multistage model, further acceleration on the models can be considered, because the computational burden increases as the number of time periods increases. In addition, it can be compared to other solution approaches or frameworks that can deal with such situations. Since the proposed optimization framework under stochastic net load is based on efficiently restricting the space of recourse actions, it can be interesting to relate it to the policy (decision-rule) approximation widely used in the multistage model. In addition, the proposed framework can be compared to other well-known frameworks that can deal with sequential decision-making such as approximate dynamic programming (ADP) or stochastic dual dynamic programming. The good upper and lower bounds in a relatively short time in Chapter 3 can be helpful when obtaining good quality solutions in the ADP framework.

Lastly, various characteristics or considerations in the power systems can be

further considered. For example, although thermal generation is one of the widely used generators, various types of generators, such as pumped-hydro generators, and combined-cycle generators can be considered. Especially in microgrid environments, various distributed energy resources such as electric vehicles and battery energy storage units are widely utilized. Since each type of generation resource has distinguished characteristics, the studies need to be extended to incorporate them. Further, various considerations of the system can be considered. For example, operational requirements regarding the reliability and security of transmission lines, are important when dealing with power systems with electricity networks. One representative consideration is to impose upper bounds that limit the amount of power flow in a certain transmission line, where the flow depends on the generation amount of the generator related to the line. Furthermore, we note that one can consider discrete recourse actions to cope with the realization of uncertainty, such as amending the on/off status of generators. In stochastic programming, it is related to stochastic programming with integer recourse, which is much more challenging than continuous recourse. Therefore, the solution approaches need to be further elaborated to deal with such cases. Finally, we have mentioned the applicability of the proposed methodologies in the previous subsection that it can be applied to various optimization problems other than power system operation. It is worthwhile to investigate their effectiveness for the problems and develop them by exploring characteristics of them.

Bibliography

Ackooij, Wim van, Welington de Oliveira, and Yongjia Song (2018). "Adaptive partition-based level decomposition methods for solving two-stage stochastic programs with fixed recourse". *INFORMS Journal on Computing* 30 (1), pp. 57–70.

Akrami, Alireza, Meysam Doostizadeh, and Farrokh Aminifar (2019). "Power system flexibility: an overview of emergence to evolution". *Journal of Modern Power Systems and Clean Energy* 7 (5), pp. 987–1007.

Alvarado-Barrios, Lázaro, Alvaro Rodrıguez del Nozal, Juan Boza Valerino, Ignacio Garcıa Vera, and Jose L Martınez-Ramos (2020). "Stochastic unit commitment in microgrids: Influence of the load forecasting error and the availability of energy storage". *Renewable Energy* 146, pp. 2060–2069.

An, Yu and Bo Zeng (2015). "Exploring the modeling capacity of two-stage robust optimization: Variants of robust unit commitment model". *IEEE Transactions on Power Systems* 30 (1), pp. 109–122.

Artzner, Philippe, Freddy Delbaen, Jean-Marc Eber, and David Heath (1999). "Coherent measures of risk". *Mathematical Finance* 9 (3), pp. 203–228.

Bansal, Manish, Kuo-Ling Huang, and Sanjay Mehrotra (2018). "Decomposition algorithms for two-stage distributionally robust mixed binary programs". *SIAM Journal on Optimization* 28 (3), pp. 2360–2383.

Basciftci, Beste, Shabbir Ahmed, and Siqian Shen (2021). "Distributionally robust facility location problem under decision-dependent stochastic demand". *European Journal of Operational Research* 292 (2), pp. 548–561.

Bashir, Arslan Ahmad, Mahdi Pourakbari-Kasmaei, Javier Contreras, and Matti Lehtonen (2019). "A novel energy scheduling framework for reliable and economic operation of islanded and grid-connected microgrids". *Electric Power Systems Research* 171, pp. 85–96.

Benders, J.F. (1962). "Partitioning procedures for solving mixed-variables programming problems". *Numerische Mathematik* 4 (1), pp. 238–252.

Bendotti, Pascale, Pierre Fouilhoux, and Cécile Rottner (2018). "The min-up/min-down unit commitment polytope". *Journal of Combinatorial Optimization* 36, pp. 1024–1058.

— (2019). "On the complexity of the unit commitment problem". *Annals of Operations Research* 274 (1-2), pp. 119–130.

Bertsimas, Dimitris and Melvyn Sim (2004). "The price of robustness". *Operations Research* 52 (1), pp. 35–53.

Bertsimas, Dimitris, Eugene Litvinov, Xu Andy Sun, Jinye Zhao, and Tongxin Zheng (2013). "Adaptive robust optimization for the security constrained unit commitment problem". *IEEE Transactions on Power Systems* 28 (1), pp. 52–63.

Birge, John R (1982). "The value of the stochastic solution in stochastic linear programs with fixed recourse". *Mathematical Programming* 24 (1), pp. 314–325.

Birge, John R and Francois Louveaux (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Blanco, Ignacio and Juan M Morales (2017). "An efficient robust solution to the two-stage stochastic unit commitment problem". *IEEE Transactions on Power Systems* 32 (6), pp. 4477–4488.

Cain, Mary B, Richard P O'neill, Anya Castillo, et al. (2012). "History of optimal power flow and formulations". *Federal Energy Regulatory Commission* 1, pp. 1–36.

Carøe, Claus C and Rüdiger Schultz (1999). "Dual decomposition in stochastic integer programming". *Operations Research Letters* 24 (1-2), pp. 37–45.

Carpentier, Pierre, Guy Gohen, J-C Culioli, and Arnaud Renaud (1996). "Stochastic optimization of unit commitment: a new decomposition framework". *IEEE Transactions on Power Systems* 11 (2), pp. 1067–1073.

Cheng, Zhiping, Dongqiang Jia, Zhongwen Li, Shuai Xu, Changyi Zhi, and Long Wu (2022). "Multi-time scale energy management of microgrid considering the uncertainties in both supply and demand". *Energy Reports* 8, pp. 10372–10384.

Cho, Youngchae, Takayuki Ishizaki, Nacim Ramdani, and Jun-ichi Imura (2019). "Box-based temporal decomposition of multi-period economic dispatch for two-stage robust unit commitment". *IEEE Transactions on Power Systems* 34 (4), pp. 3109–3118.

Conforti, Michele, Gérard Cornuéjols, and Giacomo Zambelli (2014). *Integer programming*. Springer.

Crainic, Teodor Gabriel, Mike Hewitt, and Walter Rei (2014). "Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design". *Computers & Operations Research* 43, pp. 90–99.

Damcı-Kurt, Pelin, Simge Küçükyavuz, Deepak Rajan, and Alper Atamtürk (2016). "A polyhedral study of production ramping". *Mathematical Programming* 158, pp. 175–205.

Delage, Erick and Yinyu Ye (2010). "Distributionally robust optimization under moment uncertainty with application to data-driven problems". *Operations Research* 58 (3), pp. 595–612.

Delfino, Federico, Renato Procopio, Mansueto Rossi, Massimo Brignone, Michela Robba, and Stefano Bracco (2018). *Microgrid design and operation: toward smart energy in cities*. Artech House.

Deng, Yan, Huiwen Jia, Shabbir Ahmed, Jon Lee, and Siqian Shen (2020). "Scenario grouping and decomposition algorithms for chance-constrained programs". *INFORMS Journal on Computing*.

Duan, Chao, Lin Jiang, Wanliang Fang, and Jun Liu (2017). "Data-driven affinely adjustable distributionally robust unit commitment". *IEEE Transactions on Power Systems* 33 (2), pp. 1385–1398.

Ebrahimi, Mohammad Reza, Nima Amjady, and Nikos D Hatziargyriou (2021). "Microgrid Operation Optimization Considering Transient Stability Constraints: A New Bidirectional Stochastic Adaptive Robust Approach". *IEEE Systems Journal*.

Escudero, Laureano F, M Araceli Garın, Gloria Pérez, and Aitziber Unzueta (2013). "Scenario cluster decomposition of the Lagrangian dual in two-stage stochastic mixed 0-1 optimization". *Computers & Operations Research* 40 (1), pp. 362–377.

Fan, Wei, Xiaohong Guan, and Qiaozhu Zhai (2002). "A new method for unit commitment with ramping constraints". *Electric Power Systems Research* 62 (3), pp. 215–224.

Farzin, Hossein, Mahmud Fotuhi-Firuzabad, and Moein Moeini-Aghtaie (2017). "Stochastic energy management of microgrids during unscheduled islanding period". *IEEE Transactions on Industrial Informatics* 13 (3), pp. 1079–1087.

Frangioni, Antonio and Claudio Gentile (2006a). "Perspective cuts for a class of convex 0–1 mixed integer programs". *Mathematical Programming* 106, pp. 225–236.

— (2006b). "Solving nonlinear single-unit commitment problems with ramping constraints". *Operations Research* 54 (4), pp. 767–775.

— (2015). "New MIP Formulations for the Single-unit Commitment Problems with Ramping Constraints". *IASI Research Report*.

Frangioni, Antonio, Claudio Gentile, and Fabrizio Lacalandra (2008). "Solving unit commitment problems with general ramp constraints". *International Journal of Electrical Power & Energy Systems* 30 (5), pp. 316–326.

Gamboa, Carlos Andrés, Davi Michel Valladão, Alexandre Street, and Tito Homem-de-Mello (2021). "Decomposition methods for Wasserstein-based data-driven distributionally robust problems". *Operations Research Letters* 49 (5), pp. 696–702.

Gholami, Amin, Tohid Shekari, Farrokh Aminifar, and Mohammad Shahidehpour (2016). "Microgrid scheduling with uncertainty: The quest for resilience". *IEEE Transactions on Smart Grid* 7 (6), pp. 2849–2858.

Gholami, Amin, Tohid Shekari, and Santiago Grijalva (2017). "Proactive management of microgrids for resiliency enhancement: An adaptive robust approach". *IEEE Transactions on Sustainable Energy* 10 (1), pp. 470–480.

Griset, Rodolphe, Pascale Bendotti, Boris Detienne, Marc Porcheron, Halil Şen, and François Vanderbeck (2022). "Combining Dantzig-Wolfe and Benders decompositions to solve a large-scale nuclear outage planning problem". *European Journal of Operational Research* 298 (3), pp. 1067–1083.

Guan, Yongpei, Kai Pan, and Kezhuo Zhou (2018). "Polynomial time algorithms and extended formulations for unit commitment problems". *IISE Transactions* 50 (8), pp. 735–751.

Guo, Yuanxiong and Chaoyue Zhao (2018). "Islanding-aware robust energy management for microgrids". *IEEE Transactions on Smart Grid* 9 (2), pp. 1301–1309.

Huang, Kai and Shabbir Ahmed (2009). "The value of multistage stochastic programming in capacity planning under uncertainty". *Operations Research* 57 (4), pp. 893–904.

Huang, Jianqiu, Kai Pan, and Yongpei Guan (2021). "Multistage stochastic power generation scheduling co-optimizing energy and ancillary services". *INFORMS Journal on Computing* 33 (1), pp. 352–369.

Jiang, Ruiwei and Yongpei Guan (2018). "Risk-averse two-stage stochastic program with distributional ambiguity". *Operations Research* 66 (5), pp. 1390–1405.

Jiang, Ruiwei, Yongpei Guan, and Jean-Paul Watson (2016). "Cutting planes for the multistage stochastic unit commitment problem". *Mathematical Programming* 157 (1), pp. 121–151.

Katiraei, Farid, Reza Iravani, Nikos Hatziargyriou, and Aris Dimeas (2008). "Microgrids management". *IEEE Power and Energy Magazine* 6 (3), pp. 54–65.

Kaufman, Leonard and Peter J Rousseeuw (1990). "Partitioning around medoids (program pam)". *Finding groups in data: an introduction to cluster analysis* 344, pp. 68–125.

Kazarlis, Spyros A, AG Bakirtzis, and Vassilios Petridis (1996). "A genetic algorithm solution to the unit commitment problem". *IEEE Transactions on Power Systems* 11 (1), pp. 83–92.

Khodaei, Amin (2013). "Microgrid optimal scheduling with multi-period islanding constraints". *IEEE Transactions on Power Systems* 29 (3), pp. 1383–1392.

— (2014). "Resiliency-oriented microgrid optimal scheduling". *IEEE Transactions on Smart Grid* 5 (4), pp. 1584–1591.

Kim, Kibaek, Audun Botterud, and Feng Qiu (2018). "Temporal decomposition for improved unit commitment in power system production cost modeling". *IEEE Transactions on Power Systems* 33 (5), pp. 5276–5287.

Kim, Kibaek and Sanjay Mehrotra (2015). "A two-stage stochastic integer programming approach to integrated staffing and scheduling with application to nurse management". *Operations Research* 63 (6), pp. 1431–1451.

Kleywegt, Anton J, Alexander Shapiro, and Tito Homem-de-Mello (2002). "The sample average approximation method for stochastic discrete optimization". *SIAM Journal on Optimization* 12 (2), pp. 479–502.

Knueven, Ben, Jim Ostrowski, and Jianhui Wang (2018). "The ramping polytope and cut generation for the unit commitment problem". *INFORMS Journal on Computing* 30 (4), pp. 739–749.

Knueven, Bernard, James Ostrowski, and Jean-Paul Watson (2020). "On mixed-integer programming formulations for the unit commitment problem". *INFORMS Journal on Computing* 32 (4), pp. 857–876.

Künzi-Bay, Alexandra and János Mayer (2006). "Computational aspects of minimizing conditional value-at-risk". *Computational Management Science* 3 (1), pp. 3–27.

Lee, Jongheon, Siyoung Lee, and Kyungsik Lee (2021). "Multistage stochastic optimization for microgrid operation under islanding uncertainty". *IEEE Transactions on Smart Grid* 12 (1), pp. 56–66.

Lee, Si Young, Young Gyu Jin, and Yong Tae Yoon (2016). "Determining the optimal reserve capacity in a microgrid with islanded operation". *IEEE Transactions on Power Systems* 31 (2), pp. 1369–1376.

Lee, Jon, Janny Leung, and François Margot (2004). "Min-up/min-down polytopes". *Discrete Optimization* 1 (1), pp. 77–85.

Lorca, Álvaro, X Andy Sun, Eugene Litvinov, and Tongxin Zheng (2016). "Multistage adaptive robust optimization for the unit commitment problem". *Operations Research* 64 (1), pp. 32–51.

Lowery, PG (1966). "Generating unit commitment by dynamic programming". *IEEE Transactions on Power Apparatus and Systems* (5), pp. 422–426.

Lu, Yiruo, Tong Zhang, and Yongpei Guan (2022). "An extended formulation for two-stage stochastic unit commitment with reserves". *Operations Research Letters* 50 (3), pp. 235–240.

Mansouri, Milad, Mohsen Eskandari, Yousef Asadi, Pierluigi Siano, and Hassan Haes Alhelou (2022). "Pre-perturbation operational strategy scheduling in microgrids by two-stage adjustable robust optimization". *IEEE Access* 10, pp. 74655–74670.

MISO (2022). "Energy and operating reserve markets". *MISO, BPM-002-r23 Sep-30-2022*.

Mitra, Joydeep and Mallikarjuna R Vallem (2012). "Determination of storage required to meet reliability guarantees on island-capable microgrids with intermittent sources". *IEEE Transactions on Power Systems* 27 (4), pp. 2360–2367.

Mınguez, R, W van Ackooij, and R Garcıa-Bertrand (2021). "Constraint generation for risk averse two-stage stochastic programs". *European Journal of Operational Research* 288 (1), pp. 194–206.

Mohajerin Esfahani, Peyman and Daniel Kuhn (2018). "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations". *Mathematical Programming* 171 (1-2), pp. 115–166.

Mohan, Vivek, Jai Govind Singh, and Weerakorn Ongsakul (2015). "An efficient two stage stochastic optimal energy and reserve management in a microgrid". *Applied Energy* 160, pp. 28–38.

Morales-España, Germán, Claudio Gentile, and Andres Ramos (2015). "Tight MIP formulations of the power-based unit commitment problem". *OR Spectrum* 37 (4), pp. 929–950.

Moretti, Luca, Emanuele Martelli, and Giampaolo Manzolini (2020). "An efficient robust optimization model for the unit commitment and dispatch of multi-energy systems and microgrids". *Applied Energy* 261, p. 113859.

Muckstadt, John A and Sherri A Koenig (1977). "An application of Lagrangian relaxation to scheduling in power-generation systems". *Operations Research* 25 (3), pp. 387–403.

Nowak, Matthias P and Werner Römisch (2000). "Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty". *Annals of Operations Research* 100, pp. 251–272.

O'Neill, RP (2017). "Computational issues in ISO market models". *Workshop on energy systems and optimization*. Vol. 2.

Ostrowski, James, Miguel F Anjos, and Anthony Vannelli (2011). "Tight mixed integer linear programming formulations for the unit commitment problem". *IEEE Transactions on Power Systems* 27 (1), pp. 39–46.

Pan, Kai and Yongpei Guan (2016). "Strong formulations for multistage stochastic self-scheduling unit commitment". *Operations Research* 64 (6), pp. 1482–1498.

— (2017). "Data-driven risk-averse stochastic self-scheduling for combined-cycle units". *IEEE Transactions on Industrial Informatics* 13 (6), pp. 3058–3069.

Papavasiliou, Anthony, Yi He, and Alva Svoboda (2014). "Self-commitment of combined cycle units under electricity price uncertainty". *IEEE Transactions on Power Systems* 30 (4), pp. 1690–1701.

Papavasiliou, Anthony and Shmuel S Oren (2013). "Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network". *Operations Research* 61 (3), pp. 578–592.

Park, Kyungchul and Kyungsik Lee (2017). "Distribution-robust loss-averse optimization". *Optimization Letters* 11, pp. 153–163.

Pereira, Mario VF and Leontina MVG Pinto (1991). "Multi-stage stochastic optimization applied to energy planning". *Mathematical Programming* 52, pp. 359–375.

Qu, Ming, Tao Ding, Yuge Sun, Chenggang Mu, Kai Pan, and Mohammad Shahidehpour (2022). "Convex hull model for a single-unit commitment problem with pumped hydro storage unit". *IEEE Transactions on Power Systems*.

Rahimian, Hamed and Sanjay Mehrotra (2019). "Distributionally robust optimization: A review". *arXiv preprint arXiv:1908.05659*.

Rahmaniani, Ragheb, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei (2017). "The Benders decomposition algorithm: A literature review". *European Journal of Operational Research* 259 (3), pp. 801–817.

Rajan, Deepak and Samer Takriti (2005). "Minimum up/down polytopes of the unit commitment problem with start-up costs". *IBM Res. Rep* 23628, pp. 1–14.

Rockafellar, R Tyrrell and Stanislav Uryasev (2000). "Optimization of conditional value-at-risk". *Journal of Risk* 2, pp. 21–42.

Rockafellar, R Tyrrell and Roger J-B Wets (1991). "Scenarios and policy aggregation in optimization under uncertainty". *Mathematics of Operations Research* 16 (1), pp. 119–147.

Ryan, Kevin, Shabbir Ahmed, Santanu S Dey, Deepak Rajan, Amelia Musselman, and Jean-Paul Watson (2020). "Optimization-driven scenario grouping". *INFORMS Journal on Computing* 32 (3), pp. 805–821.

Sandikçi, Burhaneddin and Osman Y Özaltin (2017). "A scalable bounding method for multistage stochastic programs". *SIAM Journal on Optimization* 27 (3), pp. 1772–1800.

Sandıkçı, Burhaneddin, Nan Kong, and Andrew J Schaefer (2013). "A hierarchy of bounds for stochastic mixed-integer programs". *Mathematical Programming* 138 (1), pp. 253–272.

Santoso, Tjendera, Shabbir Ahmed, Marc Goetschalckx, and Alexander Shapiro (2005). "A stochastic programming approach for supply chain network design under uncertainty". *European Journal of Operational Research* 167 (1), pp. 96–115.

Shiina, Takayuki and John R Birge (2004). "Stochastic unit commitment problem". *International Transactions in Operational Research* 11 (1), pp. 19–32.

Song, Yongjia and James Luedtke (2015). "An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse". *SIAM Journal on Optimization* 25 (3), pp. 1344–1367.

Street, Alexandre, Fabrıcio Oliveira, and José M Arroyo (2011). "Contingency-constrained unit commitment with $n - K$ security criterion: A robust optimization approach". *IEEE Transactions on Power Systems* 26 (3), pp. 1581–1590.

Tahanan, Milad, Wim van Ackooij, Antonio Frangioni, and Fabrizio Lacalandra (2015). "Large-scale unit commitment under uncertainty". *4OR* 13, pp. 115–171.

Takriti, Samer, John R Birge, and Erik Long (1996). "A stochastic model for the unit commitment problem". *IEEE Transactions on Power Systems* 11 (3), pp. 1497–1508.

Takriti, Samer, Benedikt Krasenbrink, and Lilian S-Y Wu (2000). "Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem". *Operations Research* 48 (2), pp. 268–280.

Tseng, Chung-Li (1996). *On power system generation unit commitment problems* [Doctoral dissertation]. University of California, Berkeley.

Tuohy, Aidan, Peter Meibom, Eleanor Denny, and Mark O'Malley (2009). "Unit commitment for systems with significant wind penetration". *IEEE Transactions on Power Systems* 24 (2), pp. 592–601.

Uçkun, Canan, Audun Botterud, and John R Birge (2015). "An improved stochastic unit commitment formulation to accommodate wind uncertainty". *IEEE Transactions on Power Systems* 31 (4), pp. 2507–2517.

Van Ackooij, Wim, Irene Danti Lopez, Antonio Frangioni, Fabrizio Lacalandra, and Milad Tahanan (2018). "Large-scale unit commitment under uncertainty: an updated literature survey". *Annals of Operations Research* 271 (1), pp. 11–85.

Wang, Qianfan, Jean-Paul Watson, and Yongpei Guan (2013). "Two-stage robust optimization for $N-k$ contingency-constrained unit commitment". *IEEE Transactions on Power Systems* 28 (3), pp. 2366–2375.

Wang, Jiadong, Jianhui Wang, Cong Liu, and Juan P Ruiz (2013). "Stochastic unit commitment with sub-hourly dispatch constraints". *Applied Energy* 105, pp. 418–422.

Wolsey, Laurence A (2020). *Integer programming*. John Wiley & Sons.

Wu, Lei, Mohammad Shahidehpour, and Tao Li (2007). "Stochastic security-constrained unit commitment". *IEEE Transactions on Power Systems* 22 (2), pp. 800–811.

Wuijts, Rogier Hans, Marjan van den Akker, and Machteld van den Broek (2021). "An improved algorithm for single-unit commitment with ramping limits". *Electric Power Systems Research* 190, p. 106720.

Xavier, Álinson S, Feng Qiu, and Shabbir Ahmed (2021). "Learning to solve large-scale security-constrained unit commitment problems". *INFORMS Journal on Computing* 33 (2), pp. 739–756.

Xiong, Peng, Panida Jirutitijaroen, and Chanan Singh (2016). "A distributionally robust optimization model for unit commitment considering uncertain wind power generation". *IEEE Transactions on Power Systems* 32 (1), pp. 39–49.

Yan, Bing, Peter B Luh, Tongxin Zheng, Dane A Schiro, Mikhail A Bragin, Feng Zhao, Jinye Zhao, and Izudin Lelic (2020). "A systematic formulation tightening approach for unit commitment problems". *IEEE Transactions on Power Systems* 35 (1), pp. 782–794.

Yurdakul, Ogun, Fikret Sivrikaya, and Sahin Albayrak (2021). "A distributionally robust optimization approach to unit commitment in microgrids". *2021 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, pp. 1–5.

Zacharia, L, L Tziovani, M Savva, L Hadjidemetriou, E Kyriakides, AD Bintoudi, A Tsolakis, D Tzovaras, JL Martinez-Ramos, A Marano, et al. (2019). "Optimal energy management and scheduling of a microgrid in grid-connected and islanded modes". *2019 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, pp. 1–6.

Zeng, Bo and Long Zhao (2013). "Solving two-stage robust optimization problems using a column-and-constraint generation method". *Operations Research Letters* 41 (5), pp. 457–461.

Zhao, Chaoyue and Yongpei Guan (2016). "Data-driven stochastic unit commitment for integrating wind generation". *IEEE Transactions on Power Systems* 31 (4), pp. 2587–2596.

Zhao, Long and Bo Zeng (2012). "Robust unit commitment problem with demand response and wind energy". *2012 IEEE Power and Energy Society General Meeting*. IEEE, pp. 1–8.

Zheng, Qipeng P, Jianhui Wang, and Andrew L Liu (2015). "Stochastic optimization for unit commitment—A review". *IEEE Transactions on Power Systems* 30 (4), pp. 1913–1924.

Zheng, Qipeng P, Jianhui Wang, Panos M Pardalos, and Yongpei Guan (2013). "A decomposition approach to the two-stage stochastic unit commitment problem". *Annals of Operations Research* 210, pp. 387–410.

Zhu, Jizhong (2015). *Optimization of power system operation*. John Wiley & Sons.

Zimmerman, Ray Daniel, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas (2010). "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education". *IEEE Transactions on Power Systems* 26 (1), pp. 12–19.

Zou, Jikai, Shabbir Ahmed, and Xu Andy Sun (2019). "Stochastic dual dynamic integer programming". *Mathematical Programming* 175 (1), pp. 461–502.

Appendix A

Additional test results in Chapter 2

Appendix A addresses additional numerical experiments in Chapter 2 for the two-stage facility location problem. In the problem, a set of facilities is selected to open in order to meet the demands of customer sites. We first describe a mathematical formulation of a classical facility location problem with stochastic demand. A set I denotes a set of possible facilities, and a set J indicates a set of customer sites. In addition, x_i is a binary decision variable that indicates whether a facility $i \in I$ is open or not, and y_{ij}^s indicates the amount of demand of $j \in J$ assigned by facility $i \in I$ for each scenario $s \in \mathcal{S}$. Further, for each customer site $j \in J$, u_j^s indicates the amount of demand that is not satisfied for each scenario $s \in \mathcal{S}$. To represent the risk level for second-stage, η_k for $k \in \mathcal{K}$ and v_s for $s \in \mathcal{S}$ are also introduced. The mathematical formulation of the two-stage risk-averse facility location problem is presented below:

$$\min \quad \sum_{i \in I} f_i x_i + \sum_{k \in \mathcal{K}} \tilde{p}_k \eta_k + \frac{1}{1-\alpha} \sum_{s \in \mathcal{S}} p_s v_s \tag{A.1a}$$

$$\text{s.t.} \quad v_s + \eta_k \geq \sum_{i \in I, j \in J} c_{ij} y_{ij}^s + \sum_{j \in J} p_j u_j^s \quad \forall s \in S_k, k \in \mathcal{K}, \tag{A.1b}$$

$$\sum_{j \in J} y_{ij}^s \leq C_i x_i \quad \forall i \in I, \forall s \in \mathcal{S}, \tag{A.1c}$$

$$\sum_{i \in I} y_{ij}^s + u_j^s = d_j^s \quad \forall j \in J, \forall s \in \mathcal{S}, \tag{A.1d}$$

$$x_i \in \{0,1\}, u_j^s, y_{ij}^s, v_s \geq 0 \quad \forall i \in I, j \in J, s \in \mathcal{S}. \tag{A.1e}$$

In the formulation, the objective function (A.1a) minimizes the total cost consisting of fixed opening costs and second-stage costs. Constraints (A.1b) represent the relationship between second-stage risk and cost for each scenario, where the right-hand side consists of transportation costs and penalty costs for unmet demand. Constraints (A.1c) restrict the maximum transportation amount for each facility and constraints (A.1d) represent the demand balance for each customer site. Constraints (A.1e) indicates the domain for each decision variable.

We let $|I| = 50$ and $|J| = 100$ and instances are generated as follows, similar to Basciftci et al. (2021). Each facility $i \in I$ and customer site $j \in J$ is located in a two-dimensional space where the $x-$ and $y-$ coordinates of each follow $U(0, 100)$. The transportation cost of each facility to each customer site, c_{ij} for $i \in I$ and $j \in J$, is set equal to the Euclidean distance of i and j. Fixed opening cost f_i follows $U(5,000, 10,000)$ and the capacity is $C_i = 0.002 \times f_i$ for $i \in I$. The unit penalty cost for unmet demand is set to $p_j = 100$, and the nominal demand of each customer site $\bar{d}_j$ follows $U(90, 180)$ for $j \in J$.

A total of 100 scenarios are generated from a normal distribution for the problem. Various number of groups (K) in $\{1, 5, 10, 20\}$ and α values in $\{0.5, 0.75, \texttt{max}\}$ are tested. The specific value of $\alpha = \texttt{max}$ is a minimum value that corresponding PSP($\mathcal{K}, \alpha$) is the same as the RO($\mathcal{K}$) ((2.2) in Section 2.1), recalling that the latter corresponds to the model PSP whose risk function is a maximum function. For this

case, where the probabilities of a scenario are equal to each other and those of a group also are equal to each other, $\alpha = \texttt{max}$ becomes $\alpha = 1 - K/|\mathcal{S}|$. The quantile parameter for target risk function β is set to 0 or 0.5.

We first analyze the effectiveness of the proposed partitioning methods in an *a priori* partitioning scheme. The methods, with two sets of $\bar{X}$s, are compared with two other methods: `dagg` and `rand`, which are the same as defined in Section 2.5.1. For the proposed methods, the only difference is in selecting $\bar{X}$: one is to let $\bar{X} = X^{EV}$ and the other is to let $\bar{X} = X^{MS}$, as mentioned in Section 2.4.2. Three measures are recorded for the proposed partitioning algorithm: `relobj`, `time`, and `iter`. Among them, `relobj` indicates how the algorithm can reduce the maximum deviation ($D_{\mathcal{K}}$) from the initial value. We finally note the actual deviation (`dev`) for all of the partitioning methods to show how well a partition is constructed according to the criteria.

Table A.1 presents the performances of the various partitioning methods, including the statistics on the partitioning algorithm, for various (α, K) combinations with $\beta = 0$ or 0.5. As can be seen, the partitioning algorithm shows sufficiently small `relobj` for most of the α and K combinations, which means that deviations are much reduced relative to the initial value. There are some exceptions, for example, $\alpha = \texttt{max}$ and $K = 5$, because α is so high that a partition cannot move sufficiently toward the risk-neutral position ($\beta = 0$) by the algorithm. For the rest case, where $K \neq 5$, the values are close to zero, which means that one can find a partition close enough to the target for the given first-stage solutions in $\bar{X}$. It is shown that the algorithm works more efficiently for $\beta = 0.5$ than for $\beta = 0$, where the value is sufficiently close to zero except for $\alpha = \texttt{max}$ and $K = 5$. It is also demonstrated

Table A.1: Partitioning statistics for four partitioning methods for the facility location problem

β	α	K	Proposed partitioning methods								dagg	rand
			$\bar{X} = X^{EV}$				$\bar{X} = X^{MS}$					
			relobj (%)	time (s)	iter	dev (%)	relobj (%)	time (s)	iter	dev (%)	dev (%)	dev (%)
0	0.5	5	30.0	2.0	145.4	0.91	30.3	176.8	138.6	0.92	1.26	3.05
		10	16.7	1.1	186.0	0.50	18.2	164.3	178.6	0.52	1.00	2.93
		20	9.6	0.8	205.6	0.26	12.7	197.2	224.6	0.29	0.82	2.64
	0.75	5	29.0	1.8	133.0	1.44	29.3	211.3	142.2	1.44	2.07	4.98
		10	16.6	1.1	200.0	0.77	18.1	183.5	191.6	0.80	1.65	4.64
		20	10.2	1.0	233.4	0.43	13.0	225.2	251.6	0.48	1.31	4.08
	max	5	91.4	2.9	110.0	2.28	91.5	285.4	109.2	2.29	3.41	7.57
		10	17.2	2.3	163.2	1.08	19.3	435.3	189.8	1.13	2.32	6.14
		20	10.4	1.3	219.8	0.50	13.1	301.5	240.4	0.54	1.49	4.60
	avg		25.7	1.6	177.4	0.91	27.3	242.3	185.2	0.93	1.70	4.51
0.5	0.75	5	<0.1	0.6	58.2	<0.01	1.3	116.9	65.6	<0.01	1.04	1.78
		10	<0.1	0.3	54.6	0.01	1.0	155.7	69.8	<0.01	1.43	1.45
		20	<0.1	0.2	34.0	0.02	1.6	94.1	55.0	<0.01	1.77	0.92
	max	5	91.1	2.9	110.0	0.84	91.2	299.1	110.6	0.82	0.27	4.30
		10	<0.1	0.7	59.4	0.02	1.3	138.7	67.0	0.02	0.78	2.92
		20	<0.1	0.3	39.8	<0.01	1.4	96.7	49.4	0.01	1.59	1.42
	avg		15.2	0.8	59.3	0.15	16.3	150.2	69.6	0.14	1.15	2.13

that the actual deviation (`dev`) is much less than expected (`relobj`), which is close enough to zero, even when the partitioning algorithm does not reduce the deviation very much. When comparing the proposed method with the two sets, the deviation is similar for both $\bar{X} = X^{MS}$ and $\bar{X} = X^{EV}$, albeit the computational time is significantly larger for the former, because a larger number of first-stage solutions in $\bar{X}$ leads to a computational burden. Compared with the other two methods, the actual deviations of the proposed partitioning methods are the lowest for all cases, and the deviation of `dagg` is lower than that of `rand`.

We next demonstrate the efficiency of the proposed solution approach, which is *primal implementation* of the column-and-constraint generation method detailed in Section 2.3.1. The computational aspects of the proposed solution approach are compared with those of two other methods, one of which is *dual implementation*

mentioned in Section 2.3.2, and the other being solving extensive formulation of the master problem (MP=RMP$_{\mathcal{K}}(\mathcal{S})$) by commercial solver CPLEX directly with the default setting ('CPLEX'). We note that, for the purposes of a fair comparison, no acceleration scheme is implemented to speed up the solving (restricted) master problem. In this and the following subsection, we use an *a priori partitioning scheme* with $\bar{X} = X^{EV}$ and $\beta = 0$. Also, for the proposed solution approach, we let Γ=1 to implement column-and-constraint generation, while the analysis of the effectiveness of various Γ values is presented in the following subsection. We recorded the computational time (`time`) for each method and we additionally recorded two more measures for the column-and-constraint-generation methods, where `iter` means the number of iterations to converge, and `totscn` means the number of totally added scenarios through the algorithm.

The computational performances of the three solution approaches are provided in Table A.2. First of all, we describe the computational aspects of column-and-constraint generation. The computational time of the algorithm is proportional to both the number of iterations and the total number of scenarios added. The total number of scenarios added is not much different from the minimum number of scenarios, which can be calculated analytically, i.e. $K\lceil(1-\alpha)|\mathcal{S}|/K\rceil$, as mentioned in Section 2.3.1. The algorithm generally is faster for higher α, since a smaller number of scenarios are needed to solve the model exactly. For the two implementations, primal implementation has shown less computational time than dual implementation. This is due to the fact that when the lower bound of the former is larger than that of the latter, as shown in Proposition 2.4, fewer iterations are needed for convergence. We lastly note that both of the column-and-constraint-generation methods

Table A.2: Computational performance of three solution approaches for the facility location problem

α	K	column-and-constraint generation primal implementation			column-and-constraint generation dual implementation			CPLEX
		`iter`	`totscn`	`time` (s)	`iter`	`totscn`	`time` (s)	`time` (s)
0.5	1	3.0	55.2	449.0	3.0	55.2	317.5	674.4
	5	3.2	64.6	694.0	3.8	65.4	855.9	1,167.3
	10	3.4	70.8	980.4	3.8	71	1,012.7	1,239.0
	20	3.6	84.6	1,445.0	4.2	85.8	2,268.4	1,482.7
avg		3.3	68.8	892.1	3.7	69.4	1,113.6	1,140.8
0.75	1	3.0	28.6	100.6	3.2	28.6	100.7	694.1
	5	3.4	38.8	288.0	3.8	39.8	359.9	1,235.6
	10	3.8	48.6	563.1	4.0	49.4	596.1	1,219.7
	20	3.6	66.2	1,043.4	4.4	68.6	1,387.4	1,566.5
avg		3.5	45.6	498.8	3.9	46.6	611.0	1,179.0
`max`	1	2.2	1.2	5.7	2.2	1.2	4.5	264.7
	5	3.8	10.8	38.0	3.8	10.8	33.5	1,266.0
	10	4.2	21	114.7	4.2	21	108.5	1,242.7
	20	4.2	40.6	453.7	4.2	40.6	472.7	1,449.5
avg		3.6	18.4	153.0	3.6	18.4	154.8	1,055.7

outperform CPLEX for nearly all α and K combinations in terms of computation time. Especially, the difference is much greater for higher α, where that of column-and-constraint-generation is much faster and that of CPLEX remains nearly the same.

We investigate the computational performance of the proposed solution approach for various Γ values, where one can manage a parameter Γ to control the number of scenarios generated in one iteration. In the experiment, $\alpha \in \{0.5, 0.75\}$ is tested for various Γ values in $\{0.6, 0.8, 1.0, 1.2, 1.4\}$. We also report `iter`, `scn`, and `time`, whose meanings are the same as defined in Section 2.5.3. In addition, `gap` is recorded

for $\Gamma \in \{0.6, 0.8\}$, to evaluate the objective values that might not be optimal.

Table A.3 presents the computational performance of the proposed solution approach with various Γ values. The results in the table demonstrate a trade-off between the number of iterations and the number of scenarios. Since more scenarios are added for higher Γ, it is shown that the total number of generated scenarios is larger as Γ increases. On the other hand, the number of iterations declines as Γ increases. For $\Gamma \geq 1$, it is shown that more iterations do not significantly reduce the number of iterations, and that thus, the computational time increases as Γ increases. To the contrary, when $\Gamma < 1$, although the number of iterations increases given the lesser number of scenarios added compared with $\Gamma = 1$, the `gap` is significantly low, which means that the algorithm yields an almost optimal solution. In other words, although the solution cannot be proved to be optimal (because the lower and upper bounds do not converge to the same value), the quality is good for $\Gamma < 1$, which had shown nearly a 0.5% gap for $\Gamma = 0.6$ or a 0.05% gap for $\Gamma = 0.8$. Since the computation time significantly declines compared to the cases where $\Gamma \geq 1$, it can be used as an efficient heuristic to solve PSP$(\mathcal{K}, \alpha)$, where the large number of scenarios can still be burdensome.

We lastly examine the effectiveness of the adaptive partitioning scheme by comparing its performance with the *a priori* partitioning scheme. We choose $\bar{X} = X^{EV}$ when constructing a partition in the *a priori* partitioning scheme. We report the number of iterations (`iter`), the total number of scenarios added (`totscn`), computation time (`time`) and deviation (`dev`) for each scheme, each having been defined in previous subsections. For *a priori* partitioning, we separate computation time for constructing a partition (`ptime`) and solving corresponding model (`stime`). We also

Table A.3: Computational performance of proposed solution approach for various Γ values for the facility location problem

α	K	Γ=0.6				Γ=0.8				Γ=1.0			Γ=1.2			Γ=1.4		
		iter	totscn	gap (%)	time (s)	iter	totscn	gap (%)	time (s)	iter	totscn	time (s)	iter	totscn	time (s)	iter	totscn	time (s)
0.5	1	3.6	33.6	2.051	98.9	3.2	44.4	0.195	185.3	3.0	55.2	452.9	2.8	63.0	573.0	2.4	72.0	690.8
	5	4.0	46.2	0.133	353.8	3.8	57.6	0.002	602.1	3.2	64.6	702.4	3.0	73.0	906.6	3.0	82.6	1,300.7
	10	4.0	52.6	0.050	366.3	4.4	65.0	0.001	928.2	3.4	70.8	992.9	3.2	81.0	1,212.5	3.2	88.4	1,664.1
	20	4.4	70.8	0.005	1,050.1	4.4	70.8	0.005	1,055.5	3.6	84.6	1,463.5	3.6	84.6	1,469.5	3.0	95.4	1,880.2
avg		4.0	50.8	0.566	467.3	4.0	59.5	0.051	692.8	3.3	68.8	902.9	3.2	75.4	1,040.4	2.9	84.6	1,383.9
0.75	1	3.0	17.8	1.554	39.5	3.0	24.2	0.024	57.5	3.0	28.6	102.0	3.0	33.6	153.5	2.6	37.2	180.0
	5	4.8	29.0	0.036	209.8	4.4	34.8	0.001	326.4	3.4	38.8	292.8	3.4	44.8	420.8	3.4	49.8	538.2
	10	4.2	37.6	0.003	273.5	4.2	37.6	0.003	277.4	3.8	48.6	567.6	3.8	48.6	569.2	3.6	62.2	806.7
	20	4.6	41.8	0.027	431.8	4.6	41.8	0.027	436.3	3.6	66.2	1,063.4	3.6	66.2	1,064.0	3.6	66.2	1,063.4
avg		4.2	31.6	0.413	238.7	4.1	34.6	0.014	274.4	3.5	45.6	506.5	3.5	48.3	551.9	3.3	53.9	647.1

report the objective value of the model in each scheme as `objval`.

Table A.4 provides the performance statistics of the two partitioning schemes. It can be seen that the objective values of the adaptive partitioning scheme are slightly larger than those of its *a priori* counterpart, which implies that a partition from adaptive partitioning scheme is slightly more risk-averse sense. It has shown similar performance in terms of `dev` and it has better performance when $\beta = 0.5$, where the deviation is closer to zero. Although similar characteristics between the two schemes are shown, the computational effort for adaptive partitioning is much lower than that of *a priori*, and the trend is especially more obvious for $\beta = 0$. The reason is that the total number of iterations is lower than with the *a priori* partitioning scheme, which can be explained by the fact that a partition is constructed in a relatively small number of iterations. Especially with the additional advantage that there is no need to choose $\bar{X}$ in advance, the results show the practical usefulness of the adaptive partitioning scheme.

Table A.4: Overall statistics of two partitioning schemes for the facility location problem

β	α	K	adaptive partitioning					a priori partitioning					
			objval	iter	totscn	time (s)	dev (%)	objval	iter	totscn	ptime (s)	stime (s)	dev (%)
0	0.5	5	543,239.4	3.6	64.8	901.2	1.26	542,030.6	3.2	64.6	2.0	740.4	0.91
		10	542,337.0	3.2	68.2	795.9	1.00	540,623.4	3.4	70.8	1.1	1,104.6	0.50
		20	541,375.8	3.2	83.6	1,218.5	0.72	539,839.0	3.6	84.6	0.8	1,846.7	0.26
	0.75	5	545,947.8	3.4	37.4	258.8	2.07	543,840.4	3.4	38.8	1.8	303.6	1.44
		10	544,495.4	3.4	46.0	404.2	1.63	541,555.8	3.8	48.6	1.1	617.1	0.77
		20	542,866.6	3.2	62.0	633.0	1.14	540,417.2	3.6	66.2	1.0	1,180.0	0.43
	max	5	550,609.8	3.4	8.8	30.8	3.41	546,694.6	3.8	12.4	2.9	88.6	2.28
		10	546,715.6	3.4	17.6	55.2	2.27	542,633.2	4.2	21.0	2.3	131.3	1.08
		20	543,366.0	4.0	37.6	307.5	1.29	540,639.2	4.2	40.6	2.3	487.9	0.50
	avg		544,550.4	3.4	47.3	511.7	1.64	542,030.4	3.7	49.7	1.6	722.3	0.91
0.5	0.75	5	549,589.4	3.0	35.2	159.4	<0.01	549591.4	3.0	31.6	0.6	148.6	<0.01
		10	549,590.6	3.0	38.6	198.8	<0.01	549,559.6	3.0	36.6	0.3	202.6	0.01
		20	549,591.2	3.0	48.6	318.8	<0.01	549,537.2	3.0	45.8	0.2	352.6	0.02
	max	5	550,609.8	3.4	8.8	30.7	0.27	546,694.6	3.8	12.4	2.9	88.7	0.84
		10	549,599.6	4.2	18.8	98.3	<0.01	549,646.8	3.4	16.6	0.7	54.8	0.02
		20	549,590.0	3.2	29.0	109.1	<0.01	549,615.2	3.2	28.2	0.3	116.2	<0.01
	avg		549,761.8	3.3	29.8	152.5	0.05	549,107.5	3.2	28.5	0.8	160.6	0.15

국문초록

전력시스템은 전력의 생산으로부터 최종 소비자에 이르기까지의 발전, 송전, 배전 단계로 구성된 시스템이다. 전력시스템의 다양한 의사결정 중, 발전계획 문제로 대표되는 운영계획 단계에서는 매 시구간별 시스템의 전력 수요를 충족시키며 전체 운영 비용을 최소로 하는 발전원들의 발전상태와 발전량을 결정한다. 또한, 운영계획 수립 시 신재생 발전량, 부하 수요, 위험 상황 등 점차 다변화되는 불확실한 요소를 사전에 고려해야 실시간 운영을 안정적으로 수행할 수 있다. 그러나 불확실성을 반영한 최적화 모형은 계산 부담이 증가하기 때문에, 이를 완화하기 위한 효과적인 최적화 모형과 효율적인 해법이 필요하다. 본 논문은 불확실성이 존재하는 전력시스템의 효과적이고 안정적인 운영을 위한 최적화 모형과 분해기법을 제안한다.

우선, 전력시스템의 최적화 문제에도 널리 활용되는 불확실성이 시나리오로 표현되는 일반적인 2단계 확률적 최적화 모형을 탐구한다. 전통적인 모형은 시나리오 수가 증가함에 따라 계산 부담이 비례하여 증가하며 시나리오 확률을 잘못 추정할 위험이 존재한다. 이를 완화하기 위하여 시나리오의 집합을 여러 그룹으로 분할하고 목적함수를 모든 그룹에 대한 위험 반영 비용의 기댓값으로 표현하는 새로운 최적화 모형을 제안한다. 해당 모형을 효과적으로 해결하기 위하여 변수와 제약식을 점진적으로 생성하는 해법을 제안하며, 모형의 위험 수준이 의사결정자의 의도에 따라 결정되도록 시나리오 분할 방법을 고안하고 이를 제안한 해법과 결합한다. 제안한 모형과 해법을 2단계 확률적 발전계획 문제에 적용하여, 그 효과성과 효율성을 확인한다.

다음으로, 하나의 발전원을 갖는 발전사업자가 전력 시장에 자신의 발전 계획을 제안하는 환경에서 발생하는 단일발전원 계획문제를 탐구한다. 확률적인 시장 전력가격 하에서 기대 수익을 최대화하는 확률적 단일발전원 계획문제를 해결하기 위해 확정적

인 환경에서 제안된 해법을 확장한 동적계획 알고리듬을 제안한다. 또한, 이 해법을 활용하는 수요의 불확실성 하에서의 확률적인 발전계획 문제에 대한 두 종류의 발전원 분해기법을 제안한다. 각 분해기법은 라그랑지안 완화 기법 혹은 열 생성을 통하여 하한을 도출하고 매 반복단계에서 상한을 도출하는 최적화 해법이다. 수치 실험을 통하여 확률적 단일발전원 계획문제에 대한 동적계획 알고리듬 및 확률적 발전계획 문제를 위한 발전원 분해기법이 다수의 시나리오 하에서 효율적임을 확인한다.

마지막으로, 마이크로그리드 운영자 관점에서 순차적으로 발현되는 불확실성 하에서의 운영을 위한 최적화 모형들을 탐구한다. 마이크로그리드는 배전 단계에서 독립운전이 가능한, 다양한 분산 전원들로 이루어진 소규모 전력시스템이다. 기존에 널리 고려된 전력 수요의 불확실성에 더불어 마이크로그리드가 중앙의 시스템과 한시적으로 분리되는 독립운전의 불확실성을 반영하며 이들이 계획기간 내 순차적으로 발현될 수 있음을 상정한다. 해당 상황을 표현하는 일반적인 다단계 확률적 최적화 모형은 필요한 시나리오의 수가 시구간의 수에 지수적이기 때문에 현실 크기의 문제의 해결이 어렵다. 따라서 이를 적은 계산 부담으로 다룰 수 있는 최적화 모형과 방법론을 제안한다. 독립운전의 불확실성에 대응하기 위해 순차적 재계획 방법을 동반한 최적화 모형을 제안하며, 수요의 불확실성을 다루기 위해 발전 가능량의 범위를 발전 상태와 더불어 결정하는 줄어든 규모의 모형을 제안한다. 다단계 확률적 최적화 모형과 비교하는 계산 실험을 통하여 제안한 모형의 줄어든 계산 부담과 실제 운영에서의 효과성을 확인한다.

주요어: 전력시스템 운영, 발전계획, 마이크로그리드, 분해기법, 불확실성 하의 최적화, 확률적 최적화

학번: 2019-36357