Université catholique de Louvain
École Polytechnique de Louvain

Center for Operations Research and Econometrics

Optimization Under Uncertainty with Power System Applications

Jehum Cho

Thesis submitted in partial fulfillment of the requirements for the degree of
Docteur en sciences de l'ingénieur et technologie

Supervisor:
Anthony Papavasiliou
(NTUA, Greece)
Daniele Catanzaro
(UCLouvain, Belgium)

Jury:
Philippe Chevalier (UCLouvain, Belgium)
Bert Willems (UCLouvain, Belgium)
Andy Philpott (University of Auckland, New Zealand)
Daniel Kuhn (EPFL, Swiss)

Chair:
Philippe Chevalier (UCLouvain, Belgium)

PhD Organization

Jehum Cho
UCLouvain
École Polytechnique de Louvain
Center for Operations Research and Econometrics

Thesis Supervisor

Anthony Papavasiliou
Assistant Professor, National Technical University of Athens
Department of Electrical and Computer Engineering

Daniele Catanzaro
Professor, UCLouvain
École Polytechnique de Louvain
Center for Operations Research and Econometrics

Supervisory Committee

Mehdi Madani
Senior Consultant, N-SIDE

Dane Schiro
Senior Analyst, ISO New England

Abstract

The transition of the energy sector towards decarbonization involves the integration of more renewable sources, which introduces unpredictability from solar and wind energy. This necessitates sophisticated decision-making models to manage such variability effectively. Emphasizing the importance of innovative modeling approaches, this thesis highlights how effective modeling serves as a critical strategy to utilize these mathematical techniques in practical scenarios. By presenting examples, including a novel perspective on uncertainty in multi-interval real-time markets and reformulation strategies for multi-area reserve sizing problems, the dissertation demonstrates significant computational efficiency gains and the potential for solving large-scale, practical power system challenges.

The first part of this dissertation explores the dynamics of multi-interval real-time markets, where the unique characteristics of rolling implementation pose significant challenges for both optimal dispatch decisions and pricing models. Through theoretical and empirical analyses, this chapter uncovers the difference between these two models and introduces a method that leverages the stochastic gradient algorithm. This innovative approach circumvents the complexities of multi-stage stochastic programming, yielding near-optimal solutions swiftly for large-scale problems and highlighting the significance of advanced modeling in reducing opportunity costs.

The second part of this dissertation considers the multi-area reserve dimensioning problem, aiming to optimize reserve allocations within the constraints of system reliability. Beginning with a foundational two-stage chance-constrained programming model, this chapter evaluates three distinct reformulations. The final approach is particularly notable for the development of an efficient solution method that not only solves real-world problems optimally but also has been adopted by a Nordic Transmission System Operator for a number of planning functions. This example underlines the impact of modeling, especially in handling integer variables.

Collectively, the dissertation underscores the role of modeling as a foundational element for optimization under uncertainty in power systems, presenting modeling not just as a theoretical endeavor, but as a practical tool that addresses specific and complex requirements, thereby bridging the gap between computational capabilities and practical needs in the power industry.

Acknowledgements

A long life lesson, this entire journey has been. It has taught me perseverance, relationships and adaptability as well as academic knowledge. Across three continents, four movements unfolded. The pandemic has passed. Without the assistance of others, this would have been an impossibility. I would like to express my gratitude to all those who made a significant contribution to making this journey possible.

Anthony, I admire the supportive, parent-like role you play for all your students, including me. From our very first Skype meeting, all my worries dissipated, and your warm heart opened mine wide. That brief moment was more than enough to convince me move to Belgium with my family. Four year later, I can confidently say that you have met all my expectations and demonstrated that you did not exaggerate. Your consistent support, especially during challenging times, your near-constant availability, your willingness to acknowledge even during heated conflicts, and your guidance tailored to each stage of my journey have repeatedly reminded me of how fortunate I am to have you as my supervisor.

Mathieu, I still vividly recall our hours-long discussion about mixing inequalities and their extended formulations. You have been a great source of inspiration, and your input was invaluable in completing the final chapter of this thesis.

Daniele, thank you for accepting to be my co-supervisor even in one of the busiest time serving as the president of CORE. I remember our brief discussion about integer programming when you had just moved your office next to mine. Your sharp insights and extensive knowledge have been truly inspirational.

Medhi, as a committee member, your guidance has been incredibly helpful to me. Besides your deep knowledge of optimization and European electricity markets, which greatly assisted me in the technical aspects of my Ph.D., your professional approach in dealing with companies, stakeholders, and colleagues is truly impressive.

Dane, I appreciate your acceptance for being a committee member. Even though we have never had the opportunity to meet face-to-face, your practical feedback has been great assistance to me. I would also like to acknowledge your prompt responses despite the time difference between the US and Europe.

To the Jury members, Prof. Philippe Chevalier, Prof. Bert Willems, Prof.

Andy Philpott, and Prof. Daniel Kuhn, I would like to express my deepest gratitude for taking the time out of your busy schedules to participate in this defense.

In addition to those involved in research, I must acknowledge a diverse group of individuals whose support was essential in navigating the challenges of recent years and sharing in times of joy.

A special thank you goes to the Energy Group— Yuting, Céline, Gilles, Ilyes, Quentin, Daniel, Jacques, and Nicolas. Sharing lunchtime conversations, coffee breaks, and outings to bars has been a delightful part of my journey.

Lastly, to my beloved friends and family members. Without you my life would be meaningless. Sharing happy moments and facing difficult ones together, you have been the foremost help in finishing this journey. Thank you, and I love you all deeply.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Context and motivation

Starting from the Paris Agreement in 2015, there is an increasing global consensus to achieve the carbon net-zero goal by 2050. Through several political initiatives (the European Green Deal [Com19], the long-term strategy of the US [oStUSEOotP21]), each country sets its own concrete plans to meet this target [Nat].

The production and use of energy account for more than 75% of the EU's greenhouse gas emissions. Decarbonizing the EU's energy system is therefore critical to reach our 2030 climate objectives and the EU's long-term strategy of achieving carbon neutrality by 2050 [Com].

Renewable energy plays a fundamental role in delivering the European Green Deal and in achieving climate neutrality. In the EU, the share of renewable energy sources (RES) in primary energy production has surpassed 40%. In addition to the policy momentum initiated by the carbon net-zero objective, the recent global energy crisis is driving rapid growth in renewable power. Not only the western countries that are originally considered leaders of the carbon net-zero initiative but also other countries responsible for a considerable share of carbon emissions, such as China and India, are adding to global renewable capacity at an unprecedented rate. For example, China's share of global annual renewable capacity deployment has exceeded 50% [Age23].

Renewable energy sources, particularly solar and wind power, are variable and unpredictable. This limited predictability raises a series of questions about the reliability and cost-effectiveness of our energy systems when supporting high shares of renewable energy [OEC14]. Moreover, recent crises, such as the 2021 Texas power crisis and the energy crisis associated with the Russian-Ukraine war, have shown the fragility of our energy systems in responding to uncertainty. To mitigate the impact of these new challenges, it is inevitable to explicitly consider uncertainty in models for decision-makers.

"Power systems," "electricity grids," or "electrical power systems" are networks of interconnected electrical components, including power generation plants, transmission lines, substations, and distribution networks. An energy system encompasses a broader scope and includes all forms of energy, not just electricity, but also natural gas, oil, coal, and so on. Regarded as one of the biggest and most complicated systems that humankind has ever created, power systems are the core of the entire energy system. In an electricity grid, supply and demand must be constantly balanced in real time under the complex physical laws of electricity. Moreover, as of now, there is no efficient technology for large-scale electricity storage to address the issue of increasing uncertainty. These distinct characteristics are what make power systems special as one of the core research subjects in the energy sector. Given the current global changes, decision-making under uncertainty for power systems is a necessary and timely research topic.

In light of this, an increasing effort from academia has been observed, focusing on optimization under uncertainty within power system operations, especially in the past couple of decades. One of the most common approaches to modeling uncertainty is so-called "stochastic programming" [BL11]. In stochastic programming, scenarios for future uncertainty are identified, and the goal is to optimize the expectation of the objective function. Since the size of the problem grows as the number of scenarios increases, and this is further multiplied by the number of time periods in the case of multi-stage settings, it is significantly more demanding than assuming that all the future information is certain and deterministic. Early examples date back to the 20th century such as [Blo82] (generation expansion planning), [PP91] (long-term hydrothermal scheduling), and [TBL96] (stochastic unit commitment). These seminal works established the foundations for future research, and the core models are still in use today. However, due to computational limitations, they could only address relatively small instances compared to a realistic scale, or target long-term operational problems (e.g. units of each period amounting to several years). It is only relatively recent that researchers have demonstrated the ability to solve large enough real-world problems. Large-scale stochastic unit commitment [POR14], and real-world capacity generation expansion problems [GAC14] are some such examples. The scope of research has also expanded to tackle short-term operational problems, such as trading decisions within hydro systems [LWM13] and real-time storage dispatch [PMCS17]. Through the use of parallel and distributed computing, the authors of [ÁPL21], [ÁPJE23] have made further developments in solving large-scale problems more efficiently. Nevertheless, there are still many challenges to face. In dynamic real-time operations, for example, a large-scale problem should be solved every 5 to 15 minutes, as it is ideal to consider the updated available information for the future every time a decision is made. Moreover, to capture inter-temporal dynamics, a multi-stage approach should be considered even for real-time operations. However, even with recent algorithmic and computational developments, the unit of solving time for real-world multi-stage stochastic programming problems is several hours rather than minutes [ÁPL21].

Another common approach to modeling uncertainty is to employ chance constraints. In deterministic optimization problems, the goal is to find the solutions that optimize the objective function while remaining feasible within the set of constraints. It is indispensable to satisfy all the constraints. This concept is similar in stochastic programming; the optimal solutions must be feasible under all possible uncertain scenarios. Chance constraints, on the other hand, allow for violations of these constraints with uncertain parameters. Instead of treating these parameters as fixed values (e.g., one deterministic value or several scenarios), chance constraints specify the acceptable level of risk or probability associated with violating these constraints. In chance-constrained programming, the objective is to find the optimal solution to an optimization problem while ensuring that certain constraints are satisfied with a specified probability or chance.

Although chance-constrained programming was first suggested at the end of the 1950s [CC59], it was not until the 2000s that this approach was utilized in power system applications. At first, it was suggested as an alternative method for stochastic unit commitment [OMN04], [WGW11]. Later, it was employed to address various applications, including the transmission network expansion problem [YCWZ09] and optimal power flow under uncertainty [ZL11]. The main challenge in tackling chance-constrained programming is that it is intrinsically non-convex. When making assumptions about the underlying distributions for uncertain parameters, such as assuming a Gaussian distribution or assuming independence among the parameters, it is possible to reformulate the original problem into tractable one. However, without such assumptions or when there are several jointly related chance constraints, solving real-world problems remains difficult. Instead, a convex approximation of chance-constrained programming is often suggested, such as CVaR (Conditional Value-at-Risk). To solve this more general form of chance-constrained programming problem closer to its optimality, a scenario based method called "sample approximation" [LAN10] is used. The issue is that the resulting optimization problem becomes a mixed-integer linear programming (MILP) problem, which is also generally challenging. There are several layers of integer programming techniques available based on [ANS00], [GP01] that help accelerate the computational performance of the specific form of MILP that results from sample approximation. However, it remains a significant challenge when dealing with complex underlying systems, such as considering multiple areas simultaneously with network constraints.

Modeling as a Strategy

This dissertation proposes the use of modeling as a strategy to make significant advancements beyond the current state. It is an undeniable fact that algorithmic developments play a major role in improving the optimization or operations research field. The invention of the simplex algorithm for linear programming by G. Dantzig [Mur83] is one of the most prominent examples. For convex optimization, all the variants of gradient descent-based algorithms and the interior point algorithm [Kar84] represent legacies and evidence of successful efforts. In stochastic programming, the L-shaped method [VSW69] established the basis for future variations that are widely used to this day. Exact methods for chance-constrained programming are primarily supported by the integer programming literature through the sample approximation link [LAN10].

However, modeling is another essential pillar for solving practical real-world problems using these algorithms. There are often gaps between the standardized forms of problems that currently available algorithms can efficiently solve and what we need to solve in practice. Modeling is an art that bridges these gaps and finds a balance between theory and practice. For example, how should we model uncertainty? What are the uncertain parameters, and which ones are important and why? What do we want to achieve with uncertainty modeling? Is the goal to maximize expected profit or guarantee a certain level of system reliability? Is the aim to find a "robust" solution that is feasible in any possible scenario? These questions have led to the aforementioned divisions of optimization under uncertainty: stochastic programming, chance-constrained programming, and robust optimization [BTGN09]. Algorithmic developments have occurred in each of these divisions separately and in combination. Starting from a specific problem, there can be various types of modeling depending on the focus and practical usage. For instance, [Bru16] presents possible variations and their results when it comes to modeling uncertainty in the unit commitment problem. By exploring the universe of these possibilities, one can sometimes find a good match that marries practical needs with theoretical foundations. As a contribution of this dissertation, one such example is introduced in chapter 2. The particular issue addressed is how to price under uncertainty in the context of multi-interval real-time markets. By proposing a new perspective on modeling uncertainty, this example identifies an ideal point to bridge the practical needs (meeting the requirements of establishing good prices) and theory (minimizing the proposed objective function using a stochastic gradient descent algorithm).

Another facet of the value of modeling lies in its effectiveness when dealing with optimization problems that include integer or binary variables. Except for some special cases, as of now, Branch-and-Bound-type algorithms or their variants are used to find optimal solutions for integer programming problems. The performance of the Branch-and-Bound algorithm depends on the tightness of the continuous relaxation problem. In other words, it depends on the amount on the gap between the optimal objective function values in the original problem and the relaxed problem where integer variables are replaced by continuous ones. This leads to an interesting phenomenon; two equivalent models (formulations) can have completely different performance. Being equivalent means they are fundamentally identical in terms of their solutions and feasible regions in the space of integer variables. However, when we relax these integer variables to continuous ones, the resulting relaxation problems can have different feasible regions and solution sets. If one relaxation is closer to the convex hull of the original problem than the other in terms of optimality gap, then the computation time for solving can be much faster. It is important to note that this difference can be dramatic. Remember that integer programming problems are NP-hard in general. That is why seemingly small changes can have a significant impact on performance differences.

The traveling salesman problem (TSP) [Lap10] is a classic example where this phenomenon can be observed. One type of modeling can obtain the optimal solution for the instance with 85,900 nodes, while slightly modified modeling cannot even achieve optimal solutions for instances with 100 nodes. In addition to this, there are myriads of examples in different applications, including the vehicle routing problem [ZGYT21] and the cutting stock problem [GG61], [Van00], where a new type of modeling (reformulation) has dramatically improved computational time and enabled the solution of previously unsolved instances. As the second contribution of this dissertation, in chapter 3, a power system application problem is presented that has been similarly improved through different modeling. The multi-area reserve sizing problem is first formulated as a basic two-stage chance-constrained programming problem, followed by several reformulation strategies. The last one utilizes a minimal projection formulation to transform the original problem into a single-stage chance-constrained programming problem, for which more powerful algorithms are available. This work also serves as an example of bridging the gap between the practice (addressing a more complex problem closer to what is encountered in practice) and theory (utilizing integer programming techniques applicable to a specific format).

The upcoming sections of this chapter provide relevant background knowledge for the following chapters. The organization of the chapter is as follows. Section 1.2 presents a brief overview of electricity markets relevant to the subsequent chapters. Section 1.3 focuses on stochastic programming. Two different modeling strategies are first introduced for multi-stage stochastic programming. Later, for unconstrained stochastic programming, the stochastic gradient descent algorithm is introduced. In section 1.4, an approach to tackle chance-constrained programming mainly used in chapter 3, is introduced. As mentioned earlier, this dissertation focuses on exact methods based on sample approximation rather than other types of convex approximation. Therefore, after introducing general chance-constrained programming, the sample approximation method is presented, followed by corresponding integer programming techniques to tighten the general types of chance-constrained programming.

1.2 Electricity Market

In the early ages of electric power systems, dating back to the late 19th century, electricity markets were composed of numerous small, local players who often operated as monopolies within their respective areas. As the industry grew, there was a trend towards consolidation. Some entities were nationalized, while others remained as large, vertically integrated companies that handled generation, transmission, and distribution. By the late 20th century, some markets had gradually opened to competition, and regulatory reforms were introduced to separate the generation, transmission, and distribution of electricity. Examples include the Energy Policy Act of 1992 in the US and the Electricity Directive 96/92/EC in 1996 in the EU. Depending on the historical context and policy goals, each electricity market in the world has developed its own distinct characteristics. One of the major differences is the distinction between self-scheduling, which is a dominant paradigm in Europe, and centralized market clearing, which is a dominant paradigm in the US.

The centralized market clearing paradigm aims to imitate vertically integrated operations. Its primary goal is to find optimal coordinated schedules that minimize the cost of serving demand (or maximize the total welfare of supply and demand), subject to both system constraints and each participant's operational constraints [Wil02]. A system operator serves as the central planner who operates the market with submitted bids representing the true cost of generation, including startup cost, minimum running cost, and marginal costs. This often requires extra market power mitigation efforts, where bids are checked for consistency. The countries that have adopted this paradigm have established regulatory authorities empowered by laws or comparable measures to ensure that bids reflect actual costs. One downside is that this paradigm relies on side payments due to the underlying non-convexity of the market clearing models. It is known that these side payments can be gamed without careful surveillance of the system [FER13].

Self-scheduling, on the other hand, reduces the role of system operators in scheduling individual assets. Market participants have more freedom in bidding, and it is up to them to schedule their generation units according to their accepted bids. Although this approach is more flexible, it is also more challenging for the bidders since they are required to internalize complex factors such as inter-temporal constraints (e.g. ramp constraints). The low coordination of markets for energy, transmission and reserves results in efficiency loss for system operation. Proponents of this paradigm argue that the incentive effects overshadow the deficiency in efficient coordination. This paradigm is often associated with portfolio-based bidding systems where the markets clear aggregate positions and the owners of portfolios self-schedule resources in order to deliver the aggregate market position. This is the case in the majority of the EU electricity market.

The increase in renewable energy resources has made the inter-temporal linkage among different market intervals tighter. Energy storage resources require explicit inter-temporal coordination. In the self-scheduling paradigm, market participants are expected to predict the effects of inter-temporal constraints and internalize the opportunity costs into their bids. However, estimating such opportunity costs is a challenging task, as observed in [WP18]. The difficulty intensifies with the increasing amount of uncertainty. In contrast, some system operators in the US, such as New York ISO, California ISO, and PJM, have adopted multi-period market clearing models for their real-time market operations to explicitly capture these inter-temporal constraints in their models. This is due to the fact that the US system operators have closer access to unit-based technical information. In chapter 2, we investigate and analyze such multi-interval real-time markets, starting from an assumption of deterministic systems, and then moving to the case of uncertainty.

The difference in paradigms also affects the differences in reserve market (or balancing capacity market) design. The forward reserve markets in Europe are often separated from energy markets, whereas US market clearing models co-optimize energy and reserves simultaneously. The centralized approach allows them to directly account for security constraints. US models incorporate security constraints in their market clearing for both day-ahead and real-time [CAI13]. By contrast, European system operators rather rely on probabilistic criteria, by law, throughout the EU [Com17a]. The collection of detailed data on a unit-level basis is challenging in EU electricity markets, where resources are bid as portfolios [DVSD+19]. Probabilistic criteria are used to dimension the requirements for reserves to meet a certain reliability level. In the US, a so-called bottom-up approach is used where the decision is endogenously made considering the technical details of units.

One remarkable feature in Europe is the encouragement of coordination across different countries within the EU market for balancing capacity. The reserve products are relatively more uniform than in the US, due to the target model which aims for a common European market where reserve services can be traded between zones. This motivates the project in chapter 3, where reserve dimensioning is conducted across several zones with consideration of the underlying transmission network.

1.3 Stochastic Programming

The first paper that suggested stochastic programming is regarded as [Dan55], published in 1955 by George B. Dantzig. He begins his paper as follows.

"The essential character of the general models under consideration is that activities are divided into two or more stages. The quantities of activities in the first stage are the only ones that are required to be determined; those in the second (or later) stages can not be determined in advance since they depend on the earlier stages and the random or uncertain demands which occur on or before the latter stage."

As written in the text, it is common to consider multiple stages of decision-making in stochastic programming. When a decision is made in a certain stage, randomness exists for the later stages, and the previous decision has an impact on the subsequent decisions. The goal of stochastic programming is to minimize (or maximize) the expectation of an objective function across all possible scenarios while satisfying the problem's constraints. There are two different ways to model stochastic programming with multiple stages where

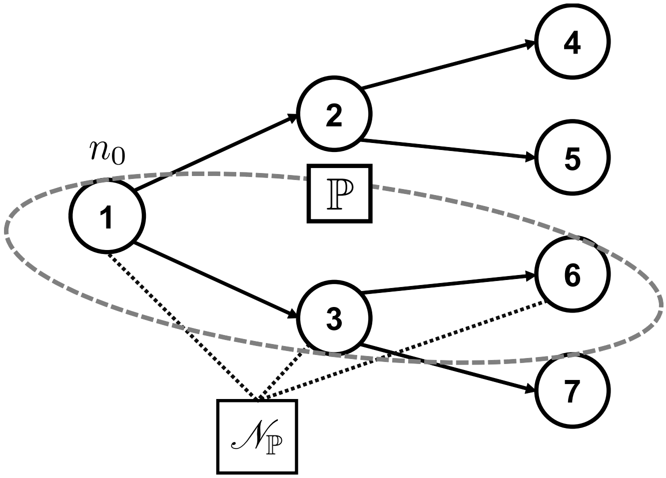

Figure 1.1: An example of a scenario tree with the root node n_0. One sample path $\mathbb{P}$ is shown with the set of nodes $\mathcal{N}_{\mathbb{P}}$ in the sample path $\mathbb{P}$.

there is a dependency between previous decisions and later decisions. This section is dedicated to introducing the difference between these approaches to modeling stochastic programming. Before delving into the models, a tool for analyzing uncertain sequential decision-making processes is introduced in the following subsection.

1.3.1 Scenario Tree

A scenario tree is a graphical representation for visualizing different possible outcomes and decisions that can be made at various stages of a complex process. An example of a scenario tree is illustrated in Figure 1.1. A tree is made up of nodes and branches. In a scenario tree, nodes represent decision points and branches the possible paths of events.

The notation regarding scenario trees for the rest of the dissertation is as follows: for a scenario tree $\mathcal{G}$, let $n \in \mathcal{N}$ denote a node of the scenario tree, where $\mathcal{N}$ is the entire set of nodes of $\mathcal{G}$. We call n_0 the root node of a scenario tree, $n-$ denotes the parent node of n. A sample path is denoted as $\mathbb{P}$, and the set of nodes in the sample path $\mathbb{P}$ as $\mathcal{N}_{\mathbb{P}}$.

The root node n_0 represents the first stage in sequential decision making. In this stage, the outcomes of uncertainties in subsequent stages remain uncertain. For example, in Figure 1.1, at node 1, which is the root node n_0, the decision should be made without knowing which paths (e.g. $1 \rightarrow 2 \rightarrow 5$ or $1 \rightarrow 3 \rightarrow 7$) it will follow on the scenario tree. In the next stage, if the uncertainty is revealed as 3, then the decision should be made given only the previous information 1 $\rightarrow$ 3 without knowing that the next node will be either 6 or 7 in the following stage.

Figure 1.1 illustrates a three-stage example in which each node presents two possibilities for future outcomes. Conceptually, this can be extended to infinite-dimensional cases, where there is no termination stage, and at every node, there are infinite possibilities. In this dissertation, a finite number of stages is considered, even though an infinite number of outcomes is assumed in some cases. In a finite length scenario tree, a path from the root node to a leaf node is called a sample path. A sample path represents a scenario of a full sequence of uncertain outcomes throughout all the stages.

Note that every sequence on a scenario tree shares at least one common node, which is the root node. This is why it is impossible to assess each scenario separately and consider the expectation on all the scenarios independently. It is necessary to devise an approach to model this dependency. Two widely used major examples are introduced in the following subsections. One way is node based modeling, and another way is sample path based modeling.

1.3.2 Node-Based Modeling

Node-based modeling is the formulation suggested in the paper [Dan55]. In node-based modeling, each node on a scenario tree corresponds to one set of decision variables. Let us denote the set of variables for node n as $x(n)$. The uncertain feasible region is denoted as $\mathcal{X}$, and given the stage and uncertain realization (i.e. given the node) the feasible region for $x(n)$ is defined as $\mathcal{X}(n)$. $\sigma(n)$ denotes the probability that the scenario of node n occurs from the perspective of the root node n_0. For example, $\sigma(n_0)$ is equal to 1, and the sum of $\sigma(n)$ for each stage is also equal to 1. Each branch represents an inter-temporal relationship between the parent and the child node in the form of constraints. For each branch with parent node $n-$ and child node n, let $h(x(n-), x(n)) \leq 0$ be the set of inter-temporal constraints between these two nodes. Consequently, for the branches that share the same parent node, the corresponding constraints also have the same common decision variables. Let the objective function for decision variable $x(n)$ be denoted as $f(x(n))$. It is worth noting that in general, the functions f and h can have different forms according to different nodes and branches, similarly to the case of the feasible region $\mathcal{X}(n)$. In this dissertation, this fact is implied, and explicit notation for the dependency on n is omitted for brevity. The resulting stochastic programming formulation is as follows:

$$\begin{aligned} \min_{x} \quad & \sum_{n \in \mathcal{N}} \sigma(n) \cdot f(x(n)) \\ s.t. \quad & h(x(n-), x(n)) \leq 0, \quad n \in \mathcal{N} \\ & x(n) \in \mathcal{X}(n) \qquad\quad n \in \mathcal{N}. \end{aligned} \tag{1.1}$$

In the first row of (1.1), the objective function is minimized over all the decision variables $x = \{x(n), \forall n \in \mathcal{N}\}$. The objective function represents the expected cost across all stages, as the sum of $\sigma(n)$ for every stage should be equal to 1.

Example 1.1 (Ramp Constraint). *Let x represent the power generation level in each stage for a generator. This generator has a limit on the rate of change in power generation. Between two consecutive stages, the generator can change its level by up to R in both directions. Additionally, the generator has a maximum capacity limit, denoted as X_{max}. The cost function is linear with a marginal cost C. The generator's objective is to maximize its expected profit in the face of uncertain prices in the electricity market. It is assumed that although information about future prices is uncertain, the (finite) probability distribution of prices $p(n)$ with probability $\sigma(n), \forall n \in \mathcal{N}$ is known.*

Through node-based modeling, as shown in (1.1), the expected profit maximization problem can be formulated as follows:

$$\begin{aligned} \underset{x}{max} \quad & \sum_{n \in \mathcal{N}} \sigma(n) \cdot [p(n) - C] \cdot x(n) \\ s.t. \quad & |x(n-) - x(n))| \leq R, \quad n \in \mathcal{N} \\ & 0 \leq x(n) \leq X_{max} \qquad n \in \mathcal{N}. \end{aligned} \tag{1.2}$$

Let us consider a more explicit example with only two stages and three nodes, $n = \{1, 2, 3\}$. Node 1 is the root node and nodes 2 and 3 are its children, each with an equally distributed probability of $\sigma(2) = \sigma(3) = 1/2$.

$$\begin{aligned} \underset{x}{max} \quad & [p(1) - C] \cdot x(1) + 1/2 \cdot [p(2) - C] \cdot x(2) + 1/2 \cdot [p(3) - C] \cdot x(3) \\ s.t. \quad & |x(1) - x(2)| \leq R, \\ & |x(1) - x(3)| \leq R, \\ & 0 \leq x(n) \leq X_{max} \qquad n \in \{1, 2, 3\}. \end{aligned} \tag{1.3}$$

In this example, it is clearer that the objective function represents the expected profit, and two inter-temporal constraints share the same decision variable $x(1)$ from their common parent node.

Solution Methods

One of the challenges in solving multi-stage stochastic programming problems is the rapid increase in problem size as the number of stages and branches per node grows. Sometimes, solving the entire problem at once in the form of (1.1), known as the extensive form, becomes impossible. A common strategy to address this issue is to decompose the extensive form into smaller problems.

Let us consider a two-stage stochastic programming problem. Given the first-stage solution $\hat{x}(1)$ and an uncertain realization $\omega \in \Omega$, the function $\mathcal{V}(\hat{x}(1), \omega)$ is denoted as follows:

$$\begin{aligned} \mathcal{V}(\hat{x}(1), \omega) = \min_{x(\omega)} \quad & f(x(\omega)) \\ s.t. \quad & h(\hat{x}(1), x(\omega)) \leq 0 \quad : \lambda(\omega) \\ & x(\omega) \in \mathcal{X}(\omega). \end{aligned} \tag{1.4}$$

With this function, the two-stage stochastic programming problem can be represented as follows:

$$\begin{aligned} \min_{x} \quad & f(x(1)) + \mathbb{E}_\omega[\mathcal{V}(x(1), \omega)] \\ s.t. \quad & x(1) \in \mathcal{X}(1). \end{aligned} \tag{1.5}$$

If f and h are linear functions and $\mathcal{X}(\omega)$ is a convex set $\forall \omega \in \Omega$, then the dual multipliers $\lambda(\omega)$ become subgradients for $\mathcal{V}(x(1), \omega)$. When ω is a finite discrete random variable, $\mathbb{E}_\omega[\mathcal{V}(x(1), \omega)]$ is a piece-wise linear function that is convex in $x(1)$ [BL11]. Therefore, $\mathbb{E}_\omega[\mathcal{V}(x(1), \omega)]$ can be underestimated by the supporting hyperplanes generated by subgradients $\lambda(\hat{\omega})$.

This fact motivates an iterative algorithm called "L-shaped algorithm", introduced in [VSW69]. This algorithm is a variation of Benders' Decomposition in [Ben62]. Fixing the first-stage variable ($x(1)$ in this case) allows us to decompose the problem into first-stage variables and second-stage variables. In the L-shaped algorithm, the second-stage variables can be further decomposed into the form of (1.4) for each $\omega \in \Omega$. Starting from a master problem with zero cuts (or supporting hyperplanes), cuts are added from the sub-problem consisting of the second-stage variables in each iteration. Similarly to Benders' Decomposition, the L-shaped algorithm converges after finitely many iterations when the functions f and h are linear [PG08].

For multi-stage stochastic programming problems, there exists an extension of the L-shaped algorithm called the nested L-shaped algorithm, which was introduced in [BL11]. This method requires scanning all possible paths and solving corresponding sub-problems in every iteration of the algorithm. To alleviate this issue, a sampling-based approach is used. Instead of scanning all possible paths, Monte Carlo sampling is employed, making the algorithm much more scalable. This algorithm has a specific name, stochastic dual dynamic programming (SDDP). First introduced in [PP91], SDDP is widely used in industrial applications [FBP10, DMPFG10, PBM13, LWM13, LW20, DPMD19].

1.3.3 Sample Path Based Modeling

An alternative way of modeling a multi-stage stochastic programming problem is through sample path-based modeling. This approach was first introduced in [RW91]. As the name of the method suggests, this modeling strategy focuses on sample paths rather than nodes in a scenario tree. Since a sample path on a scenario tree corresponds to a fully sequenced scenario, this method is often regarded as a scenario-based formulation. As a reminder, in Figure 1.1, a sample path is denoted as $\mathbb{P}$ and the set of nodes in $\mathbb{P}$ is defined as $\mathcal{N}_\mathbb{P}$. The set of all the sample paths is $\mathcal{P}$. The probability of each sample path is denoted as $Pr(\mathbb{P})$. According to the definitions, the sum over all sample paths, denoted as $\sum_{\mathbb{P} \in \mathcal{P}} Pr(\mathbb{P})$, is equal to 1.

In sample path-based modeling, a certain node n can have several sets of decision variables. For instance, every sample path contains the root node n_0.

Consequently, at the root node, the number of sets of decision variables equals the total number of sample paths. For a given sample path $\mathbb{P}$, the corresponding set of decision variables is denoted as $x^{\mathbb{P}}$ and is further decomposed into the node levels $x^{\mathbb{P}}(n), \forall n \in \mathcal{N}_{\mathbb{P}}$. Due to the presence of multiple decision variables for each node, inter-temporal constraints do not share decision variables. However, an additional set of constraints is introduced to represent the dependency of each decision. These constraints impose that the decisions made at the same node are the same. They are referred to as "non-anticipativity constraints". The resulting formulation is as follows:

$$
\begin{aligned}
\min_{x,\hat{x}} \quad & \sum_{\mathbb{P}\in\mathcal{P}} Pr(\mathbb{P}) \sum_{n\in\mathcal{N}_{\mathbb{P}}} f(x^{\mathbb{P}}(n)) \\
s.t. \quad & h(x^{\mathbb{P}}(n-), x^{\mathbb{P}}(n)) \leq 0, \quad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P} \\
& x^{\mathbb{P}}(n) \in \mathcal{X}(n) \qquad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P} \\
& x^{\mathbb{P}}(n) - \hat{x}(n) = 0, \qquad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P}.
\end{aligned} \tag{1.6}
$$

The first row of (1.6) minimizes the objective function, which represents the expected cost function under all possible scenarios or sample paths. This is because $\sum_{n\in\mathcal{N}_{\mathbb{P}}} f(x^{\mathbb{P}}(n))$ represents the cost function for a given sample path $\mathbb{P}$. When summed over all sample paths and with weights $Pr(\mathbb{P})$, it becomes the expected profit across all sample paths. The last row of (1.6) contains the non-anticipativity constraints, where $\hat{x}(n)$ is an auxiliary variable ensuring that the decisions at the same node must be identical, i.e. $x^{\mathbb{P}_1}(n) = x^{\mathbb{P}_2}(n) = \cdots = \hat{x}(n)$.

Example 1.2 (continued)**.** *Through sample path-based modeling as in* (1.6)*, the expected profit maximization problem can be formulated as follows:*

$$
\begin{aligned}
\mathit{max}_{x,\hat{x}} \quad & \sum_{\mathbb{P}\in\mathcal{P}} Pr(\mathbb{P}) \sum_{n\in\mathcal{N}_{\mathbb{P}}} \sigma(n) \cdot [p(n) - C] \cdot x^{\mathbb{P}}(n) \\
s.t. \quad & |x^{\mathbb{P}}(n-) - x^{\mathbb{P}}(n))| \leq R, \quad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P} \\
& 0 \leq x^{\mathbb{P}}(n) \leq X_{max}, \qquad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P} \\
& x^{\mathbb{P}}(n) - \hat{x}(n) = 0, \qquad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P}.
\end{aligned} \tag{1.7}
$$

Let us consider a more explicit example with only two stages and three nodes, $n = \{1, 2, 3\}$*. Node 1 is the root node and nodes 2 and 3 are its children, each having an equally distributed probability of* $\sigma(2) = \sigma(3) = 1/2$*. There are two sample paths (*$[1, 2]$ *and* $[1, 3]$*), with their probabilities being also equally distributed;* $Pr([1, 2]) = Pr([1, 3])$*.*

$$\begin{aligned}
\underset{x,\hat{x}}{max} \quad & 1/2 \cdot \{[p(1) - C] \cdot x^{[1,2]}(1) + [p(2) - C] \cdot x^{[1,2]}(2)\} \\
& + 1/2 \cdot \{[p(1) - C] \cdot x^{[1,3]}(1) + [p(3) - C] \cdot x^{[1,3]}(3)\} \\
s.t. \quad & |x^{[1,2]}(1) - x^{[1,2]}(2)| \leq R, \\
& |x^{[1,3]}(1) - x^{[1,3]}(3)| \leq R, \\
& 0 \leq x^{[1,2]}(n) \leq X_{max} \qquad n \in \{1, 2\} \\
& 0 \leq x^{[1,3]}(n) \leq X_{max} \qquad n \in \{1, 3\} \\
& x^{\mathbb{P}}(1) - \hat{x}(1) = 0 \qquad \mathbb{P} \in \{[1, 2], [1, 3]\}.
\end{aligned} \tag{1.8}$$

In this example, it is clearer that the objective function represents the expected profit. Unlike the node-based model, the two inter-temporal constraints do not share the same decision variable. They have separate sets of variables $x^{[1,2]}(1)$ *and* $x^{[1,3]}(1)$*. However, these two variables, which correspond to the same node (node 1), must have equal values, as enforced by the non-anticipativity constraints in the last row.*

Solution Methods

One of the reasons for using different modeling methods is to apply various solution techniques. The most common algorithms designed for sample path-based model is called "progressive hedging", introduced in [RW91]. Similar to the case of node-based modeling, the primary challenge in stochastic programming problems is their size. Therefore, a method to decompose the original large problem into smaller ones is needed. In sample path-based modeling, the separation between scenarios is more explicit than in node-based modeling. The only set of constraints representing dependencies among scenarios is the non-anticipativity constraints. The progressive hedging algorithm relaxes these constraints to enable decomposition.

Let us consider a two-stage stochastic programming problem. Given a first-stage solution $\bar{x}(1)$ and an uncertain realization $\omega \in \Omega$, the progressive hedging algorithm decomposes the original problem into the following sub-problems.

$$\begin{aligned}
\min_{x(1),x(\omega)} \quad & f(x(1)) + f(x(\omega)) + \pi(\omega)^T(x(1) - \bar{x}(1)) + \frac{\rho}{2} \cdot ||x(1) - \bar{x}(1)||^2 \\
s.t. \quad & h(x(1), x(\omega)) \leq 0 \\
& x(1) \in \mathcal{X}(1), x(\omega) \in \mathcal{X}(\omega).
\end{aligned} \tag{1.9}$$

In the objective function of (1.9), the second-to-last term represents the relaxation of the non-anticipativity constraints with corresponding dual multipliers $\pi(\omega)$. Additionally, a quadratic regularization term is added as the last term of the objective function. Starting from initial values for the dual multipliers $\pi(\omega)$, let us denote the optimal solutions of the sub-problems as $x^*_\omega(1)$ and $x^*_\omega(\omega)$. Then, for the next iteration $\pi(\omega)$ is updated as follows:

$$\pi(\omega) \leftarrow \pi(\omega) + \rho \cdot (x^*_\omega(1) - \bar{x}(1)), \tag{1.10}$$

and $\bar{x}(1)$ is set as $\mathbb{E}_\omega[x^*_\omega(1)]$.

The performance of the progressive hedging algorithm depends crucially on the choice of the tuning parameter ρ. Determining the appropriate level of tuning for a given problem is a critical consideration, and this can vary depending on the specific problem characteristics, as discussed in [BSdG+22].

1.3.4 Stochastic Gradient Descent

In this subsection, an algorithm is introduced that is widespread and can be applied to a specific type of stochastic programming problem. Unlike the previous models targeted at a general form of constrained optimization problem, the goal of this algorithm is to tackle an unconstrained optimization problem. In 1951, a predecessor method of the stochastic gradient descent algorithm was introduced by Herbert Robbins and Sutton Monro under the name 'stochastic approximation method' [RM51]. A year later, the first optimization algorithm that used the stochastic approximation method was published [KW52], and this algorithm is regarded as the first form of the stochastic gradient descent algorithm. Since the algorithm is used for solving the perceptron model [Ros58], which is a precursor to neural networks, the stochastic gradient descent algorithm or its variations are widely used, especially in the field of machine learning. Nevertheless, it is less frequently used for tackling stochastic programming problems due to the presence of constraints.

The stochastic gradient descent algorithm is a stochastic approximation version of gradient descent.

$$F(x) = \frac{1}{n} \sum_{i=1}^{n} F_i(x) \tag{1.11}$$

It is mostly used if the objective function $F(x)$ that is minimized has the form of (1.11). A gradient descent algorithm proceeds with the following iterations

$$x \leftarrow x - \gamma \cdot \nabla F(x), \tag{1.12}$$

where γ is a step size. The full gradient information can be obtained through the following equation.

$$\nabla F(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla F_i(x) \tag{1.13}$$

For a stochastic gradient descent, however, a randomly chosen subset of $\{1, \cdots, n\}$ is used for each iteration instead of the full gradient. As a simple example, consider a stochastic gradient descent algorithm with a single sample. If the chosen sample is i, then the iteration is as follows:

$$x \leftarrow x - \gamma \cdot \nabla F_i(x). \tag{1.14}$$

The step size γ can be dynamically chosen along with the number of iterations. At the t-th iteration, let the corresponding step size be γ_t. For the algorithm to converge, it usually requires a diminishing sequence of step sizes that satisfy $\sum_t \gamma_t^2 < \infty$ and $\sum_t \gamma_t = \infty$. With mild assumptions [Bot98] and appropriate diminishing rates of γ, the stochastic gradient descent algorithm converges almost surely to its local (or global) minimum depending on the convexity of the function $F(x)$ [Bot12]. There are many variations in terms of how to choose step size rules or utilize momentum. A partial list of variants can be found in [Bot12]. When multiple samples are chosen per iteration, it is called 'mini-batching', and it often exhibits smoother convergence than the single-sample version.

The form of function $F(x)$ as in (1.11) often arises in the context of minimizing the expectation of an uncertain objective function. In this case, n can be viewed as the size of a dataset or observations, while $F_i(x)$ corresponds to the objective function assuming the i-th data point or observation. If the set of uncertain outcomes is finite, the expectation can be directly replaced by the average of all possibilities. Otherwise, when dealing with infinite outcomes, the expectation is approximated by the average of available data or observations, a method known as sample average approximation [SHdM00, KSHdM02].

When the size n is too large, calculating the full gradient becomes computationally burdensome. By utilizing a small subset of the dataset, the stochastic gradient descent algorithm significantly reduces the time required for each iteration. While it may have a slower per-iteration convergence rate in theory compared to its full gradient descent counterpart, empirical evidence demonstrates its excellent performance in various applications.

1.4 Chance-Constrained Programming

Let us consider the following form of an optimization problem:

$$\begin{aligned} \min_x \quad & f(x) \\ s.t. \quad & Pr\{G(x,\xi) \le 0\} \ge 1-\epsilon \\ & x \in X, \end{aligned} \tag{1.15}$$

where f is the objective function, X is a deterministic feasible region, $\xi \in \Xi$ represents uncertain parameters, G is a constraint mapping with a reliability parameter ϵ. In general, the probabilistic constraint imposes all rows of the constraints jointly instead of considering them separately. Due to the explicit representation of probabilistic constraints, this form is sometimes referred to as probabilistic programming or chance-constrained programming.

Chance-constrained programming was first suggested by Charnes and Cooper [CC59], and they published several papers on this topic, including [CC63]. Even in their early works, mixed-integer programming (MIP) was often used to tackle chance-constrained programming problems. The choice is particularly natural when the support of uncertain parameters Ξ is a finite set, as binary variables could model each realization of uncertain parameters. However, due to the NP-hardness of MIP, this approach had limitations in terms of scalability.

Therefore, many early works on chance-constrained programming focused on identifying conditions under which problems could be considered convex.

1.4.1 Convex Cases

In general, chance-constrained programming problems are non-convex. However, there are some special cases where the deterministic equivalent problem of (1.15) is convex. Let us consider the case where $G(x, \xi)$ is a linear mapping as follows:

$$G(x, \xi) = A(\xi)^T x - h(\xi), \tag{1.16}$$

where $A(\xi)$ is a matrix consisting of the coefficients of linear inequalities, and $h(\xi)$ is the corresponding right-hand-side vector.

Let us start from the case where $A(\xi)$ is fixed to a deterministic matrix A. When G represents a single linear inequality, it is straightforward to see that

$$\begin{aligned} & Pr\{A^T x \leq h(\xi)\} \geq 1 - \epsilon \\ \Leftrightarrow \quad & A^T x \leq F^{-1}(\epsilon), \end{aligned} \tag{1.17}$$

where F is the probability distribution function of $h(\xi)$. For a given ϵ, $F^{-1}(\epsilon)$ is a fixed scalar, making it convex in this case.

When G represents multiple linear inequalities, there is a seminal result by Prekopa [Pre03], indicating that if the distribution function of $h(\xi)$ is log-concave, the resulting joint chance-constrained programming problem is convex.

In the case where $A(\xi)$ is also random, along with $h(\xi)$, and if the joint distribution function of every row follows a normal distribution with a common covariance structure, then the chance-constrained programming problem is convex [Pre74]. As an example, consider once again the case where G corresponds to a single inequality. Assume that h is fixed, and the coefficients of $A(\xi)$ are normally distributed with a mean vector μ and covariance matrix Σ. In this scenario, $A(\xi)^T x$ follows a one-dimensional normal distribution with mean $\mu^T x$ and variance $x^T \Sigma x$. Then,

$$\begin{aligned} & Pr\{A(\xi)^T x \leq h\} \geq 1 - \epsilon \\ \Leftrightarrow \quad & \mu^T x + \Phi^{-1}(1 - \epsilon) \cdot [x^T \Sigma x]^{1/2} - h \leq 0, \end{aligned} \tag{1.18}$$

where Φ is the probability distribution function of the standard normal distribution. By introducing a new variable v this can be reformulated as follows:

$$\begin{aligned} & \mu^T x + \Phi^{-1}(1 - \epsilon) \cdot v - h \leq 0 \\ & x^T \Sigma x \leq v^2 \\ & v \geq 0, \end{aligned} \tag{1.19}$$

a second-order cone programming problem, which is convex.

1.4.2 Sample Approximation

Except for some special cases, the general form of joint chance-constrained programming is non-convex and NP-hard. One way to address this issue is through sample approximation. This approach was first introduced in [LA08]. As the name suggests, it approximates a probabilistic constraint by sampling several scenarios for uncertain parameters. By doing so, it allows us to avoid the need to assume a specific type of underlying uncertainty distribution, such as Gaussian. This method yields an exact optimal solution with probability approaching one exponentially fast with the sample size N, under mild assumptions [LA08]. Since it is often the case that we do not know the underlying distribution for the uncertain parameters but do have access to historical data, this approach is widely used in many applications.

Let $\xi^1, \cdots, \xi^n$ be independent Monte Carlo samples of ξ. Then, the sample approximation of (1.15) is defined as follows:

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(G(x,\xi^i) \le 0) \ge 1-\epsilon \\ & x \in X, \end{aligned} \tag{1.20}$$

where $\mathbb{I}(\cdot)$ is an indicator function that takes value one when its argument is true and zero otherwise. Although it remains a challenging problem, there are some cases where it can be efficiently solved at a large scale with the assistance of integer programming techniques.

Let G be a linear mapping with only right-hand-side uncertainty as follows:

$$G(x,\xi) = h(\xi) - Ax. \tag{1.21}$$

Let $X \subseteq \mathbb{R}_+^d$ be a polyhedron, where d is the dimension of x. Then, the sample approximation (1.20) can be reformulated with additional variables y and $u_i, \forall i \in \{1, \cdots N\}$ as follows:

$$\begin{aligned} \min_{x,y,u} \quad & f(x) \\ s.t. \quad & Ax - y = 0 \\ & y + h(\xi^i)u_i \ge h(\xi^i) \\ & \frac{1}{N}\sum_{i=1}^{N} u_i \le \epsilon \\ & x \in X, u \in \{0,1\}^N, \end{aligned} \tag{1.22}$$

where u_i is a binary variable that represents the indicator function $\mathbb{I}(\cdot)$ in the opposite way. For all $i \in \{1, \cdots, N\}$, u_i takes value one when $\cdot$ is false for the i-th scenario ξ^i and zero otherwise. Observe that in order for the problem to be feasible, $Ax \ge 0$.

1.4.3 Mixing Set and Mixing Inequalities

Given N scalars h_i for $i \in [N]$, a mixing set is defined as

$$P = \{(y, u) \in \mathbb{R}_+ \times \{0,1\}^N : y + h_i u_i \geq h_i, i \in [N]\}. \tag{1.23}$$

Mixing inequalities, also known as star inequalities, were developed by Atamturk et. al [ANS00] and Gunluk and Pochet [GP01]. Assuming, without loss of generality, that $h_1 \geq h_2 \geq \cdots \geq h_N$, the mixing inequalities for the mixing set (1.23) are defined as

$$y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) u_{t_j} \geq h_{t_1}, \forall \{t_1, \ldots, t_l\} \subset [N], \tag{1.24}$$

where $t_1 < \cdots < t_l$ and $h_{t_{l+1}} := 0$. It is known that the mixing inequalities are valid and sufficient for defining the convex hull of the mixing set (1.23). Defining a convex hull with a certain set of inequalities implies that the LP relaxation gap is zero. In other words, the integer programming problem can be solved by solving a linear programming problem with these convex hull defining inequalities. This fact can be used even for more complex problems in which we may not be able to fully characterize the convex hull of the problem. In such cases, while we can no longer guarantee a zero LP relaxation gap, these mixing inequalities remain valid and significantly help in reducing the LP relaxation gap, resulting in better performance when solving the original integer programming problem with binary variables u.

Furthermore, for the following set with a cardinality constraint induced by a reliability criterion ϵ, where we define $q = \lfloor \epsilon N \rfloor$,

$$G = \{(y, u) \in \mathbb{R}_+ \times \{0,1\}^N : \sum_{i=1}^{N} u_i \leq q, y + h_i u_i \geq h_i, i \in [N]\}, \tag{1.25}$$

the strengthened mixing inequalities

$$y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) u_{t_j} \geq h_{t_1}, \forall \{t_1, \ldots, t_l\} \subset [q] \tag{1.26}$$

with $t_1 < \cdots < t_l$ and $h_{t_{l+1}} := h_{t_{q+1}}$ are facet-defining for $conv(G)$ if and only if $t_1 = 1$ [LAN10]. For a polyhedron and a valid inequality for the polyhedron, a face of the polyhedron is defined as the intersection of the polyhedron and the corresponding hyperplane of the valid inequality. When the dimension of a face is equal to the dimension of the polyhedron minus one, we call it a facet of the polyhedron. Facets are important since they exist in any description of the polyhedron including its minimal description. If we know all the facets of a polyhedron, e.g. $conv(G)$, we can then describe the polyhedron in its minimal form. Therefore, a facet-defining valid inequality is crucial for defining the convex hull. It implies that these inequalities are expected to be the most effective for reducing the LP relaxation gap among similar types of inequalities, since the convex hull is the smallest convex set containing all the feasible solutions.

In this thesis, we assume that all the scenarios have equal probabilities. There are similar techniques for knapsack constraints for unequal probabilities. In this case, the inequalities are less tight. However, there is no significant loss of generality since we can approximate the problem with equal probabilities by re-sampling from the original distribution. The readers who are interested are referred to [LAN10].

1.4.4 A Strong Extended Formulation

Although these inequalities are useful for tightening the LP relaxation gap, there are additional steps required to use them. One might first consider completely enumerating all such inequalities. However, since there are exponentially many such inequalities, this will not be a practical approach. Alternatively, there are ways to add a subset of the inequalities on-the-fly, while we are running the Branch-and-Bound algorithm. This is a potential way forward; however, it is not straightforward to implement such algorithms. In this subsection, a method is introduced that is very easy to implement and achieves excellent performance.

The essence of this method is to use a very compact formulation that has the same effect as when we add all the inequalities (1.26) by introducing a new set of variables. This new formulation is called a strong extended formulation and is first introduced in [LAN10]. The name 'extended' comes from the fact that there are additional variables to the original formulation, and 'strong' refers to the fact that this new formulation is as strong as adding the entire exponential family of valid inequalities (1.26) to the original one.

Formally, the extended formulation of G in Eq. (1.25) is defined as follows:

$$EG := \{(y,u,w) \in \mathbb{R}_+ \times \{0,1\}^{N+q} : \sum_{i=1}^{N} u_i \leq q, (1.28a) - (1.28c)\}, \tag{1.27}$$

where

$$y + \sum_{i=1}^{q} (h_i - h_{i+1}) w_i \geq h_1 \tag{1.28a}$$

$$w_i - w_{i+1} \geq 0, \qquad \forall i \in [q-1] \tag{1.28b}$$

$$u_i - w_i \geq 0, \qquad \forall i \in [q]. \tag{1.28c}$$

Theorem 1.1. *(Theorem 6 from [LAN10]) $Proj_{(y,u)}(EG) = G$. Moreover, the projection of the linear relaxation of EG is the linear relaxation of G with all the inequalities* (1.26) *added.*

As Theorem 1.1 shows us, by simply introducing q binary variables w, we can tighten the LP relaxation gap. Notice that Eqs. (1.28) are easily implemented in a commercial solver.

1.5 Organization of Dissertation

This dissertation is organized into two main chapters. Each chapter corresponds to an independent problem related to power system applications with a consideration of uncertainty. The first problem utilizes stochastic programming for modeling uncertainty, while the second problem is focused on chance-constrained programming.

Chapter 2. Pricing Under Uncertainty in Multi-Interval Real-Time Markets

Recent research has demonstrated that real-time auctions can generate the need for side payments, even if the market clearing models are convex, because of the rolling nature of real-time market clearing. This observation has inspired proposals for modifying the real-time market-clearing model in order to account for binding past decisions. This analysis is extended in order to account for uncertainty by proposing a real-time market clearing model with look-ahead and an endogenous representation of uncertainty. Two different types of expected lost opportunity cost are defined as performance metrics. The novel market-clearing model provides the price signal minimizing one of these metrics using the Stochastic Gradient Descent algorithm. Computational results are presented from a case study of the ISO New England system under a scenario of significant renewable energy penetration while accounting for ramp rates, storage, and transmission constraints. The results of this chapter have been published in the following work:

- ◇ Cho, Jehum, and Anthony Papavasiliou, Pricing Under Uncertainty in Multi-Interval Real-Time Markets, *Operations Research*, (2022).

Chapter 3. Multi-Area Reserve Dimensioning Problem

Multiple modeling approaches for the multi-area reserve dimensioning problem are presented in this chapter. The problem can be formulated as a two-stage stochastic mixed integer linear program using sample approximation. Due to its intricate structure, existing methods often rely on heuristics to find feasible solutions. However, leveraging integer programming techniques allows us to reformulate the problem into a more solvable form. In this chapter, two such possibilities are explored. First, a Benders' Decomposition-based reformulation is employed. It enables the direct application of a certain type of mixed-integer algorithms to address the two-stage stochastic programming problem. Subsequently, a distinct modeling approach is adopted. It utilizes a minimal description of the projection of our problem onto the space of the first-stage variables. This enables us to directly apply more general integer programming techniques for handling mixing sets, which commonly appear in chance-constrained programming problems. By combining the advantages of the minimal projection and the strengthening reformulation from IP techniques, this innovative method can tackle real-world problems effectively. This result is demonstrated with a case study of the 10-zone Nordic network with 100,000 scenarios, where the optimal solution can be found in approximately 5 minutes.

The results of this chapter have been published in the following works:

◇ Cho, Jehum, and Anthony Papavasiliou, A Branch-and-cut Algorithm for Chance-constrained Multiarea Reserve Sizing, *No. UCL-Université Catholique de Louvain*, CORE, (2022).

◇ Cho, Jehum, and Anthony Papavasiliou, Exact Mixed-Integer Programming Approach for Chance-Constrained Multi-Area Reserve Sizing, *IEEE Transactions on Power Systems*, (2023).

Part I

Pricing Under Uncertainty in Multi-Interval Real-Time Markets

2 Pricing Under Uncertainty in Real-Time Markets

2.1 Introduction

2.1.1 Motivation

In a regime of large-scale renewable resource integration, the multi-period and uncertainty effects of renewable resources are becoming increasingly important. Flexibility, in the sense of the ability of resources to respond rapidly to real-time conditions [Sch17], is becoming a valuable resource for system operators. Two important challenges that system operators face in real time are to arrive to efficient dispatch decisions, but also prices that provide an incentive to flexibility providers to offer their resources voluntarily to the market. Concretely, the real-time market is operated at a time step of 5-15 minutes in US and EU markets, and determines the dispatch of resources such as storage, pumped hydro plants, combined cycle units and demand response, that can respond rapidly to the significant and often unpredictable variations of renewable supply. Look-ahead matters in this respect, because these resources have inter-temporal constraints such as ramp limits, state of charge limits, startup/shutdown costs, and so on. An increasingly important challenge in real-time market operations is to account for these inter-temporal effects since ramp episodes induced by renewable resources are placing increasing stress on the system. And prices need to match the increasingly complex schedules, otherwise system operators are facing the threat that flexibility owners may "take matters in their own hands" by self-committing or self-scheduling their resources at a time when these flexibility resources are needed most. It is therefore imperative that the price signal that accompanies the central dispatch decision match the profit-maximizing objectives of flexible resource owners. This challenge has placed multi-period pricing in real-time markets at the spotlight of stakeholders and academics in recent years.

2.1.2 Market Clearing Proposals in the Literature

In what follows we revisit the existing literature on multi-period pricing. Although this literature has been cast in the context of price consistency, our approach is rather to connect the literature to a related but different notion, that of equilibrium. An equilibrium is a pair of prices and quantities such that the market clears and the prices support the dispatch for profit-maximizing agents. We comment along the way on its relation to price consistency and other metrics that are often monitored and deemed important in market operations.

Single-Period Market Clearing.

For each agent $k \in K$, an abstract formulation of single-period market clearing models can be written as follows:

$$\begin{aligned} \min_{x} \quad & \sum_{k \in K} f_k(x_k) \\ \text{s.t.} \quad & \sum_{k \in K} x_k = y : p \\ & h_k(x_k^{\#}, x_k) \leq 0, \quad k \in K \\ & x_k \in X_k, \quad k \in K \end{aligned} \tag{2.1}$$

where x_k denotes the amount of power generation, f_k denotes the cost function, h_k denotes inter-temporal constraints such as ramp constraints with $x_k^{\#}$ being a given initial condition (the amount of power generation in the previous time step), X_k denotes constraints for each k that do not depend on time such as generation output limits, and p denotes the dual multiplier of the power balance constraint. Observe the generality of this formulation. For example, if an agent k is a network owner and y is a vector representing net demand for each node in the network, then the transmission network can be incorporated in this formulation with x_k being the vector of the power flows of each line and X_k containing optimal power flow constraints and transmission capacity constraints. In the remainder of this chapter, we assume that the functions f_k, h_k are convex.

Basic convex optimization arguments establish that the solution of the fully coordinated problem provides the optimal price and quantity. The optimal price and quantity pair (p^*, x^*) forms an equilibrium since p^* supports x_k^* for profit-maximizing agents. In other words, for each agent $k \in K$, x_k^* is an optimal profit-maximizing solution under the price p^*. This is the approach adopted in real-time market clearing in ISO-NE, MISO, PJM, SPP [Sch17] and future integrated EU balancing platforms [Com17b].

Multi-Period Deterministic Market Clearing.

In a multi-period deterministic setting, the notion of an equilibrium can be extended if the quantities clear for every period, and if the dispatch is profit-maximizing over the full horizon. Consider an economic dispatch problem over

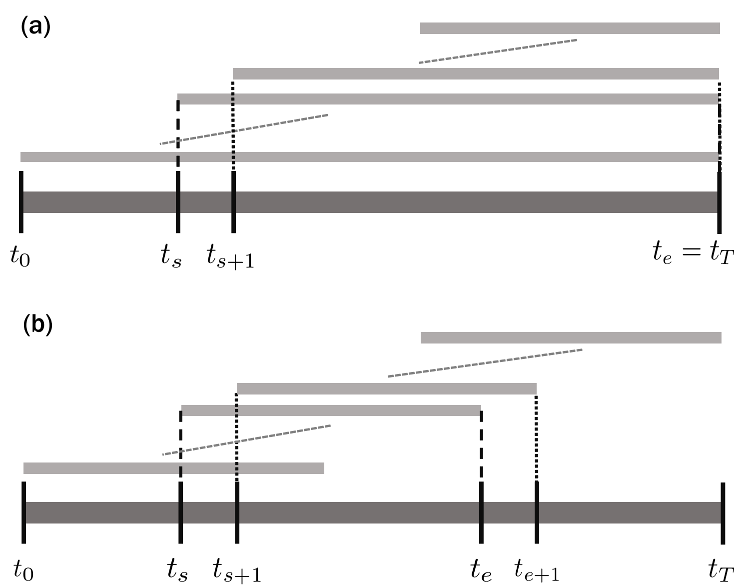

Figure 2.1: Rolling multi-period implementations with (a) a fixed horizon and (b) a moving horizon. The grey bars represent the time steps that a look-ahead market clearing model covers.

the time interval $[t_s, t_e]$. Let us define the following optimization problem as $\mathbf{LAD}(t_s, t_e)$, an abbreviation of Look Ahead Dispatch Model (LAD). This term is used in [HSZ+19].

$$
\begin{aligned}
\mathbf{LAD}(t_s, t_e): \quad \min_{x} \quad & \sum_{k \in K} \sum_{t \in [t_s, t_e]} f_{k,t}(x_{k,t}) \\
\text{s.t.} \quad & \sum_{k \in K} x_{k,t} = y_t, \quad t \in [t_s, t_e] : p_t \\
& h_k(x^{\#}_{k,t_s-1}, x_{k,t_s}) \leq 0, \quad k \in K \\
& h_k(x_{k,t-1}, x_{k,t}) \leq 0, \quad k \in K, t \in [t_s + 1, t_e] \\
& x_{k,t} \in X_k, \quad k \in K, t \in [t_s, t_e]
\end{aligned}
\tag{2.2}
$$

where the inter-temporal constraints are divided into two parts in order to clearly show the treatment of initial conditions ($x^{\#}_{k,t_s-1}$). For the sake of brevity, the initial condition is implied and the set of constraints will be written as $h_k(x_{k,t-1}, x_{k,t}) \leq 0, k \in K, t \in [t_s, t_e]$ in the remainder of this chapter.

In practice, multi-period deterministic models can be categorized as either static, rolling with a fixed horizon, or rolling with a moving horizon. In a one-shot multi-period market clearing model we run the dispatch and pricing model once and clear prices and quantities for the entire horizon at the beginning of the horizon. The same convex optimization arguments as in the static case guarantee that a centralized optimization problem yields the equilibrium result [Bal06]. The static setting is easy to study but unrealistic. The most realistic assumption is a rolling multi-period market clearing model with a moving horizon as in Figure 2.1(b), where the look-ahead length is fixed and we run the model multiple times for clearing [Hog20], [Mic15]. As time moves forward, we fix the decisions for the current time step (e.g. t_s) and at the next time step we solve another optimization problem including new information for the future demand (e.g. $\mathbf{LAD}(t_s+1, t_e+1)$). This is the approach currently adopted in CAISO, NYISO [Sch17] and under consideration in Texas [Mic15]. However, this setting is very difficult to study. Thus, most, if not all, previous research has focused on the case of rolling with fixed horizon [Hog16], [HSZ$^+$19], [ZZL19], [GCT21], [GCT20], [BH22], [Hog20]. In this setting, we still run market clearing models multiple times but we know when the end of the horizon is and even from the beginning we can access the demand information of the end of the horizon. Notice that in Figure 2.1(a), which corresponds to a multi-period model with a fixed horizon, all the bars (representing the time coverage of a market clearing model) have the same ending at t_T.

Metrics for Deterministic Multi-Period Market Clearing. Two metrics that are often employed in practice for assessing the quality of market clearing solutions are lost opportunity cost (LOC) and make-whole payments (MWP). Given a pair of price-quantity time series, lost opportunity cost (LOC) refers to the difference between the maximum profit that could have been ensured by an agent that is reacting freely to prices and the profit of an agent that follows the dispatch schedule of the system operator. Zero LOC is equivalent to an equilibrium. For price-quantity pairs that do not constitute an equilibrium, make-whole payments (MWP) are non-zero whenever the cost of a deployed resource exceeds the revenue that is obtained by following the dispatch instructions of the system operator.

Rolling Multi-Period with a Fixed Horizon. For a rolling multi-period planning with a fixed horizon, it would be tempting to argue that one should solve the dispatch problem at every time stage (as in model predictive control), and keep current-period dispatch decisions and prices as the market clearing quantities and prices. The resulting sequence of prices and quantities actually turns out not to carry guarantees of being an equilibrium price-quantity pair. This is due to dual degeneracy where dual optimal solutions are not unique [BH22].

One way to mitigate this issue is to utilize the dual multipliers from the past dispatch problems. [Hog16] uses power balance constraints and [HSZ$^+$19] use inter-temporal constraints for the dual multipliers. In this chapter, we introduce the method of [Hog16] that uses power balance constraints, and we generalize it in the next section to the case of uncertainty. Consider a time interval $[t_s, t_e]$. Assume that the current time step is t_c, where $t_s < t_c < t_e$, and the past decisions $x_0^\# = \{x_{t_s}^\#, \ldots, x_{t_c-1}^\#\}$, $p_0^\# = \{p_{t_s}^\#, \ldots, p_{t_c-1}^\#\}$ are available. Let us define $\mathbf{PMP}(t_s, t_c, t_e)$, an abbreviation of Price-preserving Multi-interval Pricing Model (PMP), as follows:

$$
\begin{aligned}
\mathbf{PMP}(t_s, t_c, t_e): \quad \min_{x} \quad & \sum_{k\in K}\sum_{t\in[t_s,t_e]} f_{k,t}(x_{k,t}) + \sum_{t\in[t_s,t_c-1]} p_t^\#(-\sum_{k\in K} x_{k,t} + y_t) \\
s.t. \quad & \sum_{k\in K} x_{k,t} = y_t, \quad t \in [t_c, t_e] : p_t \\
& h_k(x_{k,t-1}, x_{k,t}) \leq 0, \quad k \in K, t \in [t_s, t_e] \\
& x_{k,t} \in X_k \quad t \in [t_s, t_e], k \in K
\end{aligned} \tag{2.3}
$$

This pricing model (PMP) is first introduced by [Hog16] and formalized by [HSZ+19].

[HSZ+19] show that for every time step t_c if $\mathbf{PMP}(t_0, t_c, t_T)$ is used instead of $\mathbf{LAD}(t_c, t_T)$ for pricing models with rolling implementation, the resulting prices coincide with the prices from the one-shot multi-period optimization problem $\mathbf{LAD}(t_0, t_T)$. This is closely related to the concept of "price consistency," defined in slide 35 of [Hog20] as the property that, "given perfect foresight, where actual conditions equal the forecast conditions, the methodology produces the same set of prices."

Rolling Multi-Period with a Moving Horizon. In a more practical setting, we can no longer assume that a horizon is fixed. For a rolling multi-period model with a moving horizon, even in the case of strongly convex market clearing models, the application of the PMP approach can produce a price-quantity pair that does not satisfy an equilibrium for an entire horizon, i.e. in the sense of perfect hindsight, and with a horizon which spans $\{1, \ldots, T\}$. This is shown in section VI-B of [HSZ+19]. This deviation of equilibrium is different from the dual degeneracy pointed out by [BH22].

Nevertheless, empirically PMP achieves better performance than LAD with respect to LOC and MWP in this setting. In this chapter, we provide an explanation by introducing an additional characteristic of PMP. PMP not only guarantees price consistency for a rolling multi-period planning with a fixed horizon, but also minimizes LOC for an entire horizon including past time steps given past prices. PMP balances the past decisions and future decisions in a way that LOC is minimized; hence the better performance. This is further discussed in section 2.2.3. We focus on this property of PMP and extend it to the setting under uncertainty in section 2.3.3.

Multi-Period Market Clearing under Uncertainty.

Our interest in this chapter is an extension of the analysis on multi-period pricing in the context of uncertainty. When extending the basic setting to incorporate uncertainty, the standard definition of an equilibrium is a pair of price-quantity *stochastic processes*, such that the market clears at every stage for every possible sample path of uncertainty, and the dispatch instructions of agents are maximizing risked profit given the prevailing price process. Risked profit is commonly quantified using risk measures [RS15], [PFW16], [PF21], and is an *ex-ante* (i.e. without the benefit of hindsight) measure of how well an agent can do given the attitude of the agent towards risk and given the underlying price process.

The computation of the underlying equilibrium relies on posing the problem in question (the look-ahead economic dispatch, in the case of our application) as a multi-stage stochastic program. We will concentrate our discussion to the risk-neutral setting, therefore time consistency [Sha12] is automatically satisfied in our setting. An equilibrium can be computed in this setting as the generalization of the multi-period deterministic case, i.e. by retrieving the dispatch decisions of the stochastic program and the dual multipliers of the market clearing constraints for every stage and every sample path. As a natural extension for LOC in the multi-period deterministic setting, it is possible to define a metric of performance, ex-ante expected LOC, i.e. the difference between the maximum expected profit and the expected profit of an agent following the dispatch decision by the system operator. A stochastic equilibrium is equivalent to this metric being equal to zero. However, this definition stumbles upon a number of implementation challenges in a practical setting. These challenges include (i) the definition of the scenarios that constitute the stochastic program that needs to be solved, (ii) an underlying assumption that all agents in the market share the same views about the distribution of uncertainty (i.e. the same set of scenarios, and the same probability for all scenarios), (iii) an assumption that the system operator can correctly identify the risk attitude of the least risk-averse agent in the market, and (iv) the need to solve a large-scale stochastic program under the tight run times that are imposed by real-time operations. Consequently, this pricing method has not directly seen its way into practical implementation.

In the present work, and inspired by the spirit of the discussion in the literature on multi-period pricing, we rather pose the question of finding a price that (i) is non-anticipative (i.e. can be computed at a given time stage given the information that is available up to that moment in time), and (ii) delivers a stochastic process of real-time prices that minimize the expected lost opportunity cost defined in an *ex-post* sense, i.e. with lost opportunity cost being defined in a hindsight fashion when all uncertainty in the market (renewable forecast errors, etc.) has been revealed. An important property of the price that is obtained is that this price minimizes ex-post expected LOC not only for the optimal system dispatch, but for any dispatch that satisfies the aggregate uncertain demand in the market. This is further discussed in the following sections.

Our motivation for the first requirement (non-anticipativity) is that real-time market clearing is intrinsically a process that resembles model predictive control, in the sense that it is executed in a rolling fashion. Concretely, we contrast this to a situation (not applied in practice) where prices are computed after the fact, i.e. at the end of the horizon in question, and with the benefit of hindsight when we can observe the realized uncertainty for the *full* horizon.

Our motivation for the second requirement is to propose a computationally viable proxy of the ideal stochastic equilibrium benchmark, but one that can be computed in the realistic time frames of real-time market operations with minimal assumptions related to stochastic models / scenario selection. Note that both definitions of expected LOC coincide with the one in the multi-period deterministic setting. Note also, as a consequence, that price consistency is satisfied automatically for prices that minimize ex-post expected LOC in the multi-period deterministic setting.

This chapter outlines a computational procedure that can be applied for computing the prices with the requisite properties. The procedure amounts to executing a separate pricing procedure as in [HSZ$^+$19] and [Hog16]. In contrast to the case of computing a stochastic equilibrium, this minimization is essentially minimizing an expectation (as opposed to a multi-stage stochastic program in [PFW16] and [PF21]) which can be implemented with a straightforward algorithmic procedure. Moreover, the procedure can be applied to a continuous model of uncertainty without requiring scenario selection in order to restore computational tractability.

It is important to point out that the line of work pursued here is distinct from the literature on stochastic market clearing such as [BGC05], [PZP10], [ZKAB17], and [MZPP14]. Whereas the latter is concentrated on day-ahead auctions without considerations of consistency in a rolling market clearing (since day-ahead auctions are non-overlapping), the interest of this chapter is on real-time market clearing in a rolling fashion. The discussion is also distinct from that of flexible ramp products [WH15] and their associated implementation challenges [Sch17]. Flexible ramp products amount to an ancillary service that is priced in addition to energy. Instead, our focus here is on the pricing of energy in real time.

2.2 Multi-Period Deterministic Setting

In this section, we formally define what lost opportunity cost (LOC) is in the multi-period sense. This definition is extended to the case of uncertainty in the next section. In a simple fixed horizon setting, we show that the pair of the optimal dispatch solution and the dual price from the economic dispatch problem minimizes LOC, indeed makes it zero. In a more practical setting, i.e. the rolling multi-period optimization with a moving horizon, this is no longer possible. The simple look-ahead model cannot be guaranteed to achieve either a zero LOC or even a low LOC. We show that the PMP procedure proposed by [Hog16] relieves this issue by minimizing LOC for the horizon including the past time steps given the prices for the past time periods.

2.2.1 Lost Opportunity Cost in a Deterministic Setting

First, let us define an individual profit maximization problem for each agent $k \in K$. Let $t \in \mathcal{T} = \{t_0, \dots, t_T\}$ be a time step in the entire horizon $\mathcal{T}$. From now on, tilde is used (e.g. $\tilde{p}, \tilde{x}$) to indicate that these variables represent decisions made by system operators. On the other hand, the notations without accents or superscripts (e.g. p, x) correspond to the variables that each agent uses for optimizing their individual profit maximization problem. Given price signals $\tilde{p} = \{\tilde{p}_t : \forall t \in \mathcal{T}\}$, the maximized profit in the time interval $[t_0, t_T]$ is defined as follows:

$$\begin{aligned} v_k^{[t_0,t_T]}(\tilde{p}) = \max_x \quad & \sum_{t\in\mathcal{T}} \tilde{p}_t x_{k,t} - f_{k,t}(x_{k,t}) \\ \text{s.t.} \quad & h_k(x_{k,t-1}, x_{k,t}) \leq 0, \quad t \in [t_0, t_T] \\ & x_{k,t} \in X_k, \quad t \in \mathcal{T} \end{aligned} \tag{2.4}$$

The lost opportunity cost (LOC) for each agent k in the period $[t_0, t_T]$ is defined as the difference between the maximized profit and the profit of an agent following the dispatch decision by the system operator ($\tilde{x}_k$).

$$LOC_k^{[t_0,t_T]}(\tilde{p}, \tilde{x}_k) = v_k^{[t_0,t_T]}(\tilde{p}) - \sum_{t\in\mathcal{T}} (\tilde{p}_t \tilde{x}_{k,t} - f_{k,t}(\tilde{x}_{k,t})) \tag{2.5}$$

By definition, LOC is a nonnegative value. The vector $\tilde{x}_k$ is a feasible solution for the optimization problem (2.4) and since it is a maximization problem the first term for (2.5) is greater than and equal to the second term.

Another frequently used performance measure is make-whole payments (MWP), which is the amount of costs exceeding revenue for each agent k. Formally, we define it as follows:

$$MWP_k^{[t_0,t_T]}(\tilde{p}, \tilde{x}_k) = \max\{0, -\sum_{t\in\mathcal{T}} (\tilde{p}_t \tilde{x}_{k,t} - f_{k,t}(\tilde{x}_{k,t}))\} \tag{2.6}$$

The concepts of LOC and MWP are suggested in [SZZL16] and [HSZ+19]. [SZZL16] refers to them as deviation incentives mostly in the context of dealing with non-convexity. However, [HSZ+19] extends the usage of the metrics to the convex case in the rolling multi-period optimization with a moving horizon where we can no longer guarantee to be able to have a zero LOC.

In this chapter, the focus of our analysis is on LOC. Note that LOC is an upper bound of MWP if v_k is nonnegative. Even though v_k can be negative in certain cases, we can expect it to be nonnegative in most of the cases where the length of the interval $[t_0, t_T]$ is large enough. Practically, there will be no agents who would continue their business if their maximum profit is below zero. Consequently, by minimizing LOC, a low level of MWP can be obtained as a by-product. The opposite argument is not valid: the most straightforward way of guaranteeing zero MWP is clearing the market with high prices $\tilde{p}$, which would however result in a high LOC.

For the rest of this chapter, we use the notation for the aggregation of LOC or MWP as $LOC^{[t_0,t_T]}(\tilde{p},\tilde{x}) = \sum_{k\in K} LOC_k^{[t_0,t_T]}(\tilde{p},\tilde{x_k})$ or $MWP^{[t_0,t_T]}(\tilde{p},\tilde{x}) = \sum_{k\in K} MWP_k^{[t_0,t_T]}(\tilde{p},\tilde{x_k})$.

2.2.2 Simple Look-Ahead Model

When the pricing horizon is fixed as $\mathcal{T}$, and if we assume that all the future demand information is available, it is possible to obtain an equilibrium price-quantity pair by solving a one-shot multi-period optimization problem. It can be easily shown by inspection of the KKT conditions that the primal and dual pair of optimal solutions for $\mathbf{LAD}(t_0, t_T)$, $(p^*, x^*) = \{(p_t^*, x_t^*) : t \in \mathcal{T}\}$, results in a zero LOC, which is equivalent to an equilibrium. In other words, the price-quantity pair (p^*, x^*) minimizes $LOC^{[t_0,t_T]}(\tilde{p},\tilde{x})$.

Nonetheless, in reality, the entire horizon is not fixed. A common practical setting is one with a moving horizon. With a fixed look-ahead time length, it is natural to solve **LAD** at every time stage (e.g. solve $\mathbf{LAD}(t_s, t_e)$ at t_s, solve $\mathbf{LAD}(t_{s+1}, t_{e+1})$ at t_{s+1}, etc), and to keep the current period dispatch decisions and prices. The resulting (p_t^*, x_t^*) pair sequence is no longer an equilibrium for a longer period. One of the main issues is that the look-ahead model ignores the decisions from the previous stages as the look-ahead horizon moves forward. The existence of inter-temporal constraints often results in certain agents suffering losses in certain time steps that are expected to be made up for in the next stages. When the look-ahead horizon moves forward, this loss is regarded as sunk costs by the look-ahead model. Mathematically, the solutions from the $\mathbf{LAD}(t_1, t_2)$ minimizing $LOC^{[t_1,t_2]}$ do not guarantee to be a part of a solution minimizing $LOC^{[t_0,t_T]}$, when $t_0 < t_1$ and $t_2 < t_T$. A simple illustrative three-stage example where the look-ahead length is two time steps is provided by [HSZ+19].

2.2.3 Binding Past Prices

In this subsection, we analyze the effect of using a pricing model that takes into account past price decisions $p_0^{\#}$ to mitigate the price inconsistency issue. Let us define $LOC^{[t_s,t_e]}(\tilde{p},\tilde{x}|p_0^{\#},x_0^{\#})$ as the LOC defined on the interval $[t_s, t_e]$ where t_s is a time step in the past and t_e is a time step in the future. Note that at the current time step t_c, $p_0^{\#}$ and $x_0^{\#}$ represent decisions already made in the past, specifically during the time steps from t_s to t_{c-1}. The notation explicitly indicates that they are given. $\tilde{p}$ and $\tilde{x}$ represent the decisions yet to be made for the interval $[t_c, t_e]$.

Theorem 2.1. *A primal optimal solution $x^* = \{x_t^* : t \in [t_c, t_e]\}$ of $\boldsymbol{LAD}(t_c, t_e)$ and an optimal dual multiplier $p^* = \{p_t^* : t \in [t_c, t_e]\}$ of $\boldsymbol{PMP}(t_s, t_c, t_e)$ are minimizers of $LOC^{[t_s,t_e]}(\tilde{p},\tilde{x}|p_0^{\#},x_0^{\#})$.*

Proof. Redistribute the terms of $LOC^{[t_s,t_e]}(\tilde{p},\tilde{x}|p_0^{\#},x_0^{\#})$:

$$\min_{\tilde{p},\tilde{x}} LOC^{[t_s,t_e]}(\tilde{p},\tilde{x}|p_0^{\#},x_0^{\#}) = \sum_{k\in K}\sum_{t\in[t_s,t_{c-1}]} f_{k,t}(x_{k,t}^{\#}) + \min_{\tilde{x}} \sum_{k\in K}\sum_{t\in[t_c,t_e]} f_{k,t}(\tilde{x}_{k,t}) - Z, \quad (2.7)$$

$$\begin{aligned} \text{where } Z = \max_{\tilde{p}} \min_{x} \quad & \sum_{k\in K}\sum_{t\in[t_s,t_e]} f_{k,t}(x_{k,t}) + \sum_{t\in[t_s,t_c-1]} p_t^{\#}(-\sum_{k\in K} x_{k,t} + y_t) \\ & + \sum_{t\in[t_c,t_e]} \tilde{p}_t(-\sum_{k\in K} x_{k,t} + y_t) \\ s.t. \quad & h_k(x_{k,t-1},x_{k,t}) \le 0, \quad k\in K, t\in[t_s,t_e] \\ & x_{k,t}\in X_k \quad t\in[t_s,t_e], k\in K \end{aligned} \quad (2.8)$$

In the redistribution process, the following fact is utilized: $\sum_{k\in K}\tilde{x}_{k,t} = y_t$, and likewise $\sum_{k\in K} x_{k,t}^{\#} = y_t$. This is based on the assumption that $\tilde{x}$ (or $x^{\#}$), a dispatch decision made by a system operator, should be feasible. Therefore, $Z = \max_{\tilde{p}} [-\sum_{k\in K} v_k^{[t_s,t_e]}(\tilde{p}|p_0^{\#}) + \sum_{t\in[t_s,t_c-1]}(p_t^{\#}\cdot\sum_{k\in K}x_{k,t}^{\#}) + \sum_{t\in[t_c,t_e]}(\tilde{p}_t\cdot\sum_{k\in K}\tilde{x}_{k,t})]$ takes the form of equation (2.8) after substituting $\sum_{k\in K}\tilde{x}_{k,t}$ and $\sum_{k\in K}x_{k,t}^{\#}$ with y_t. Note that the considered time interval is on $[t_s,t_e]$ for both LOC and v, resulting in the same time interval for the objective function of equation (2.8). Notice that, in equation (2.7), the first term is a constant, x^* is the minimizer of the second term, and p^* is the maximizer of the third term (Z), since Z is the dual problem of $\mathbf{PMP}(t_s,t_c,t_e)$. Q.E.D.

Corollary 2.2. *If $x_0^{\#},p_0^{\#}$ are a part of the primal and dual optimal solutions of $\boldsymbol{LAD}(t_s,t_e)$, then with x^* of $\boldsymbol{LAD}(t_c,t_e)$ and p^* of $\boldsymbol{PMP}(t_s,t_c,t_e)$, $(x_0^{\#},x^*),(p_0^{\#},p^*)$ become a primal and dual optimal solution of $\boldsymbol{LAD}(t_s,t_e)$.*

Theorem 2.1 implies that the price from $\mathbf{PMP}(t_s,t_c,t_e)$ minimizes $LOC^{[t_s,t_e]}$ incorporating not only future but also past time periods given the past decisions $x_0^{\#},p_0^{\#}$. Intuitively, as the look-ahead horizon moves forward, PMP takes into account past decisions and balances in a way that LOC is minimized. This new property of PMP explains the empirically better performance measured by LOC or MWP than simple look-ahead models such as LAD. What Corollary 2.2 guarantees is the notion of price consistency as defined in [Hog20].

Another point worth commenting is that Theorem 2.1 suggests to separate the pricing model (PMP) from the dispatch model (LAD). In equation (2.7), observe that LOC is divided into the dispatch related term and the price related term. In order to minimize LOC, each of the terms is minimized from the solution of different models. We focus on these properties of PMP and extend the analysis to the setting under uncertainty.

2.3 Multi-Period Market Clearing Under Uncertainty

In this section, we extend the previous theory for the deterministic case to the setting under uncertainty. We use a scenario tree to visualize the uncertainty as in Figure 1.1. A scenario tree in the setting under uncertainty is analogous to a time interval in the deterministic setting. In other words, in the same way that we define performance metrics over intervals in the deterministic case, we define performance metrics over scenario trees in the stochastic case.

First, we provide the definitions of expected lost opportunity cost from two different aspects, i.e. ex-ante (AEL) and ex-post (PEL), and we compare their characteristics. Then, we discuss two different methods for minimizing each of the definitions of expected lost opportunity cost respectively, when a scenario tree (time horizon) is fixed. Finally, we extend one of the methods to an algorithm analogous to PMP, which can deal with the more realistic setting of a moving horizon.

2.3.1 Two Definitions of Expected Lost Opportunity Cost Under Uncertainty

Care must be taken, when we extend the definition of lost opportunity cost to the setting under uncertainty. Depending on the perspective of the individual profit maximization problem, the expected lost opportunity cost can be defined in two different ways.

One perspective is that each agent would solve an optimization problem in order to maximize its expected profits. We assume that the future price and dispatch distribution for all the scenarios is available and all agents share the same information. We define *ex-ante* expected lost opportunity cost (AEL) as the difference between the maximized expected profit and the expected profit that an agent can obtain if it follows the dispatch decisions by system operators. Formally, AEL for each agent k under the scenario tree $\mathcal{G}$ is defined as follows:

$$AEL_k^{\mathcal{G}}(\tilde{p}, \tilde{x}_k) = W_k^{\mathcal{G}}(\tilde{p}) - \sum_{n \in \mathcal{N}} \sigma(n)(\tilde{p}(n)\tilde{x}_k(n) - f_k(\tilde{x}_k(n))), \tag{2.9}$$

$$\begin{aligned} \text{where } W_k^{\mathcal{G}}(\tilde{p}) = \max_x \quad & \sum_{n \in \mathcal{N}} \sigma(n)[\tilde{p}(n)x(n) - f_k(x(n))] \\ s.t. \quad & h(x(n-), x(n)) \leq 0, \quad n \in \mathcal{N} \\ & x(n) \in X(n) \qquad n \in \mathcal{N}, \end{aligned} \tag{2.10}$$

with a slight abuse of the notation for variables $(\tilde{p}, \tilde{x}_k)$, e.g. $\tilde{p} = \{\tilde{p}(n) : \forall n \in \mathcal{N}\}$.

Another perspective is that each agent solves a *deterministic* optimization problem maximizing its profit in the ex-post fashion once all the uncertainty is revealed (i.e. under a chosen sample path). We define *ex-post* expected lost opportunity cost as the expectation of the difference between the maximized

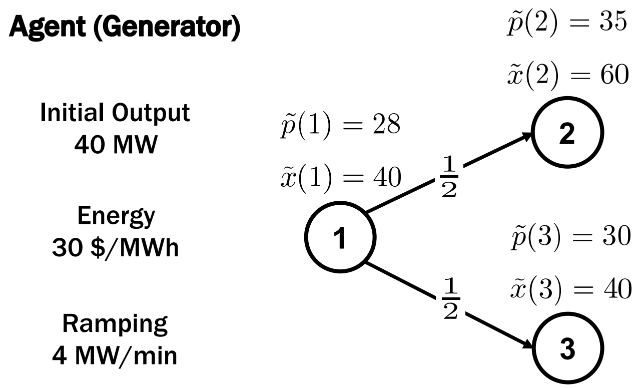

Figure 2.2: A two-stage example for comparing the calculation of AEL and PEL. Two possible scenarios for the second stage (5 minutes from the first one) and the prices and the dispatch decisions from the system operator are given.

profit that an agent could have earned if it had known the uncertain information and the actual profit that the agent obtains following the dispatch decisions by system operators. Mathematically, PEL for each agent k under the scenario tree $\mathcal{G}$ is defined as follows:

$$PEL_k^{\mathcal{G}}(\tilde{p}, \tilde{x}_k) = \mathbb{E}_{\mathbb{P}}[w_k^{\mathbb{P}}(\tilde{p}) - \sum_{n \in \mathcal{N}_{\mathbb{P}}} (\tilde{p}(n)\tilde{x}_k(n) - f_k(\tilde{x}_k(n))], \tag{2.11}$$

$$\begin{aligned} \text{where } w_k^{\mathbb{P}}(\tilde{p}) = &\max_x \quad \sum_{n \in \mathcal{N}_{\mathbb{P}}} \tilde{p}(n)x(n) - f_k(x(n)) \\ &s.t. \quad h(x(n-), x(n)) \leq 0, \quad n \in \mathcal{N}_{\mathbb{P}} \\ &\qquad\; x(n) \in X(n) \qquad n \in \mathcal{N}_{\mathbb{P}}. \end{aligned} \tag{2.12}$$

The interpretation of $w_k^{\mathbb{P}}(\tilde{p})$ is the profit that the agent k can achieve with perfect foresight of path $\mathbb{P}$ given prices $\tilde{p}$. In the remainder of this chapter, we define $AEL^{\mathcal{G}}(\tilde{p}, \tilde{x}) = \sum_{k \in K} AEL_k^{\mathcal{G}}(\tilde{p}, \tilde{x}_k)$ and $PEL^{\mathcal{G}}(\tilde{p}, \tilde{x}) = \sum_{k \in K} PEL_k^{\mathcal{G}}(\tilde{p}, \tilde{x}_k)$.

The ex-ante expected lost opportunity cost (AEL) is closely related to stochastic equilibrium as zero AEL is equivalent to a stochastic equilibrium. As a special case, zero LOC is equivalent to an equilibrium in the deterministic setting. However, using AEL directly in practice seems to be unrealistic as it relies on strong assumptions and requires solving a large-scale multi-period stochastic program (this is discussed further in section 2.1.2).

Example 2.1. *Here, we provide a simple two-stage example for the comparison between AEL and PEL in Figure 2.2, where there are two scenarios with equal probability. The parameters for this agent are given in the left side of the figure, and the prices and dispatch decisions from the system operator* $(\tilde{p}, \tilde{x})$ *are also given for each possible scenario (node). Even though in order to compute the* *precise profit we should re-scale the price or cost according to the size of a time step, for this example we present the results without re-scaling to avoid fractional numbers.*

For the calculation of AEL, let us start from calculating $W_k^{\mathcal{G}}(\tilde{p})$, the individual expected profit maximization problem. Since the expected gain for the second stage $(32.5 - 30 = 2.5)$ *is higher than the loss at the first stage* $(30 - 28 = 2)$*, generating the maximum possible output at the first stage would make the best expected profit, thus $x^*(1) = 60$. Next, at node 2, since the gain is positive* $(35 - 30 = 5)$*, the agent should generate the maximum possible output given $x^*(1)$, thus $x^*(2) = 80$. For node 3, since the marginal cost is equal to the price any feasible solution would produce the same result, here let us pick $x^*(3) = 60$. Then, we can calculate that $W_k^{\mathcal{G}}(\tilde{p})$ becomes 80, and since the actual expected profit that this agent can achieve by following $(\tilde{p}, \tilde{x})$ is 70, $AEL_k^{\mathcal{G}}(\tilde{p}, \tilde{x}_k) = 10$.*

On the other hand, for the calculation of PEL, for each sample path $\mathbb{P}$, the deterministic individual profit maximization problem for $\mathbb{P}$, $w_k^{\mathbb{P}}(\tilde{p})$ should be obtained. In this example, there are two possible sample paths: node 1 to 2, and node 1 to 3. For the sample path that shifts from node 1 to 2, since the gain for the second stage $(35 - 30 = 5)$ *is higher than the loss at the first stage* $(30-28 = 2)$*, the unit should generate the maximum possible output for both the first and the second stage, thus $x^*(1) = 60, x^*(2) = 80$, which makes $w_k^{(1,2)}(\tilde{p})$ equal to 280. The sample path that transitions from node 1 to 3 corresponds to the opposite. The unit should generate the minimum possible output for the first stage since for the second stage the marginal cost is equal to the price so any feasible solution would be optimal. Thus, let us pick $x^*(1) = 20, x^*(3) = 20$, which makes $w_k^{(1,3)}(\tilde{p})$ equal to -40. Considering that the actual profit for sample path (1,2) is 220 and that for sample path (1,3) is -80, we can see that $PEL_k^{\mathcal{G}}(\tilde{p}, \tilde{x}_k) = 50$.*

Theorem 2.3. *For any $\mathcal{G}$, $AEL_k^{\mathcal{G}}(\tilde{p}, \tilde{x_k}) \le PEL_k^{\mathcal{G}}(\tilde{p}, \tilde{x_k})$.*

Proof. First, we show that $W_k^{\mathcal{G}}(\tilde{p}) \le \mathbb{E}_{\mathbb{P}}[w_k^{\mathbb{P}}(\tilde{p})]$. There is another equivalent formulation for a multi-period stochastic program other than (2.10). This scenario based formulation is introduced in [RW91]. The formulation uses separate variables $x^{\mathbb{P}}$ for each sample path $\mathbb{P} \in \mathcal{P}$. Let $Pr(\mathbb{P})$ denotes the probability for a sample path $\mathbb{P}$. The formulation with our notation is as follows:

$$\begin{aligned} W_k^{\mathcal{G}}(\tilde{p}) = \max_{x,\hat{x}} \quad & \sum_{\mathbb{P}\in\mathcal{P}} Pr(\mathbb{P}) \sum_{n\in\mathcal{N}_{\mathbb{P}}} [\tilde{p}(n)x^{\mathbb{P}}(n) - f_k(x^{\mathbb{P}}(n))] \\ s.t. \quad & h(x^{\mathbb{P}}(n-), x^{\mathbb{P}}(n)) \le 0, \quad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P} \\ & x^{\mathbb{P}}(n) \in X(n), \qquad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P} \\ & x^{\mathbb{P}}(n) - \hat{x}(n) = 0, \qquad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P}. \end{aligned} \tag{2.13}$$

The last set of constraints is called the set of non-anticipativity constraints. These constraints impose that variables in the same node n have the same value. Notice that if we relax these non-anticipativity constraints, formulation (2.13) becomes $\mathbb{E}_{\mathbb{P}}[w_k^{\mathbb{P}}(\tilde{p})]$; hence $W_k^{\mathcal{G}}(\tilde{p}) \leq \mathbb{E}_{\mathbb{P}}[w_k^{\mathbb{P}}(\tilde{p})]$.

Second, we show that

$$\sum_{n\in\mathcal{N}} \sigma(n)(\tilde{p}(n)\tilde{x_k}(n) - f_k(\tilde{x_k}(n)) = \mathbb{E}_{\mathbb{P}}[\sum_{n\in\mathcal{N}_{\mathbb{P}}} (\tilde{p}(n)\tilde{x_k}(n) - f_k(\tilde{x_k}(n))].$$

This comes from the fact that in a scenario tree,

$$\sigma(n) = \sum_{\mathbb{P}\in\mathcal{P}:n\in\mathcal{N}_{\mathbb{P}}} Pr(\mathbb{P}), \forall n \in \mathcal{N}.$$

Q.E.D.

As a metric of economic behavior, the ex-post expected lost opportunity cost (PEL), on the other hand, has some practical advantages relative to AEL. It is easier to calculate, and it is possible to estimate PEL when the underlying uncertainty model is continuous. By Theorem 2.3, PEL is an upper bound of AEL. In a similar fashion as in the deterministic case, PEL is also an upper bound of the expected MWP under a mild condition ($\mathbb{E}_{\mathbb{P}}[w_k^{\mathbb{P}}(\tilde{p})]$ is nonnegative). Note that the definition of MWP is independent from the two different perspectives of solving an individual profit maximization problem mentioned above in the description of AEL and PEL, hence the expected MWP is identical under both of the perspectives unlike LOC. Thus, by minimizing PEL, we can regulate AEL and the expected MWP to a low level.

It is true that the economical implication of PEL diverges from conventional stochastic equilibrium theory. Even though AEL and PEL coincide (both become LOC) in the deterministic case, zero PEL is not equivalent to a stochastic equilibrium anymore, as is the case with AEL. It is even impossible to have zero PEL unless the setting is deterministic. PEL becomes zero only when an agent enjoys perfect foresight and acts optimally under this perfect foresight, i.e. applies the solution of LAD over the entire horizon. Instead, PEL can be interpreted as a metric that measures how far the decisions of system operators are compared to the optimal case under the perfect foresight assumption. When PEL is minimized in the case under uncertainty (e.g. under a scenario tree), the minimum value, which is strictly positive now, represents the inevitable LOC (calculated ex-post) caused by the underlying uncertainty.

2.3.2 Look-Ahead Models Under Uncertainty

AEL Minimizing Look-Ahead Model.

Consider a stochastic economic dispatch problem under a scenario tree $\mathcal{G}$. Let us refer to the following optimization problem as **SLAD**($\mathcal{G}$), an abbreviation for Stochastic Look Ahead Dispatch Model (SLAD).

$$\begin{aligned}
\mathbf{SLAD}(\mathcal{G}): \quad \min_{x} \quad & \sum_{n\in\mathcal{N}} \sigma(n) \sum_{k\in K} f_k(x_k(n)) \\
s.t. \quad & \sum_{k\in K} x_k(n) = y(n), \quad n \in \mathcal{N} : \lambda(n) \\
& h(x_k(n-), x_k(n)) \leq 0, \quad k \in K, n \in \mathcal{N} \\
& x(n) \in X(n) \qquad k \in K, n \in \mathcal{N}.
\end{aligned} \tag{2.14}$$

Let $p^*(n) = \lambda^*(n)/\sigma(n), \forall n \in \mathcal{N}$, where $\lambda^*(n)$ are the nodal balance optimal dual multipliers of $\mathbf{SLAD}(\mathcal{G})$. Similar to the deterministic case in section 2.1.2, it can be shown using KKT conditions that $(p^*, x^*) = \{(p^*(n), x^*(n)) : n \in \mathcal{N}\}$ minimizes $AEL^{\mathcal{G}}(\tilde{p}, \tilde{x_k})$ to zero. This implies that (p^*, x^*) forms the (risk neutral) stochastic equilibrium, defined as "a stochastic process of prices and a corresponding collection of actions for each agent with the property that the actions are individual expected profit maximizing solutions for each agent," see e.g. [PFW16], [PF21] for further information.

PEL Minimizing Look-Ahead Model.

For the pricing model minimizing PEL, we use the notation for the scenario based formulation introduced in the proof of Theorem 2.3 as follows:

$$\begin{aligned}
\min_{x} \quad & \sum_{\mathbb{P}\in\mathcal{P}} Pr(\mathbb{P}) \sum_{n\in\mathcal{N}_{\mathbb{P}}} \sum_{k\in K} f_k(x_k^{\mathbb{P}}(n)) \\
s.t. \quad & \sum_{\mathbb{P}\in\mathcal{P}:n\in\mathcal{N}_{\mathbb{P}}} Pr(\mathbb{P}) \sum_{k\in K} x_k^{\mathbb{P}}(n) = \sum_{\mathbb{P}\in\mathcal{P}:n\in\mathcal{N}_{\mathbb{P}}} Pr(\mathbb{P}) y(n), \quad n \in \mathcal{N} : p(n) \\
& h(x_k^{\mathbb{P}}(n-), x_k^{\mathbb{P}}(n)) \leq 0, \quad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P} \\
& x^{\mathbb{P}}(n) \in X(n), \qquad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P}.
\end{aligned} \tag{2.15}$$

The equivalent way of writing the first set of constraints of (2.15) is

$$\mathop{\mathbb{E}_{\mathbb{P}}}_{\mathbb{P}\in\mathcal{P}:n\in\mathcal{N}_{\mathbb{P}}} [\sum_{k\in K} x_k^{\mathbb{P}}(n)] = \sigma(n) y(n).$$

As an example, when $n = n_0$, the constraint can be written succinctly as $\mathbb{E}_{\mathbb{P}}[\sum_{k\in K} x_k^{\mathbb{P}}(n_0)] = y(n_0)$.

Theorem 2.4. *For a scenario tree $\mathcal{G}$, a primal optimal solution $x^* = \{x^*(n) : n \in \mathcal{N}\}$ of $\boldsymbol{SLAD}(\mathcal{G})$ and a dual optimal multiplier $p^* = \{p^*(n) : n \in \mathcal{N}\}$ of formulation (2.15) under $\mathcal{G}$ are minimizers of $PEL^{\mathcal{G}}(\tilde{p}, \tilde{x})$.*

Proof. Redistribute the terms of $PEL^{\mathcal{G}}(\tilde{p}, \tilde{x})$:

$$\min_{\tilde{p},\tilde{x}} PEL^{\mathcal{G}}(\tilde{p}, \tilde{x}) = \min_{\tilde{x}} \mathbb{E}_{\mathbb{P}}[\sum_{n\in\mathcal{N}_{\mathbb{P}}} \sum_{k\in K} f_k(\tilde{x}_k(n))] - Z^{\mathcal{G}}, \tag{2.16}$$

$$\begin{aligned}
\text{where } Z^{\mathcal{G}} = \max_{\tilde{p}} \mathbb{E}_{\mathbb{P}}[\min_{x} \quad & \sum_{n\in\mathcal{N}_{\mathbb{P}}} \sum_{k\in K} f_k(x_k(n)) + \tilde{p}(n)(-\sum_{k\in K} x_k(n) + y(n))] \\
s.t. \quad & h(x_k(n-), x_k(n)) \leq 0, \quad n \in \mathcal{N}_{\mathbb{P}} \\
& x(n) \in X(n), \qquad n \in \mathcal{N}_{\mathbb{P}}.
\end{aligned} \tag{2.17}$$

Since

$$\min_{\tilde{x}} \mathbb{E}_{\mathbb{P}}[\sum_{n\in\mathcal{N}_{\mathbb{P}}}\sum_{k\in K} f_k(\tilde{x_k}(n))] = \min_{\tilde{x}} \sum_{n\in\mathcal{N}}\sigma(n)\sum_{k\in K} f_k(\tilde{x_k}(n)), \tag{2.18}$$

$x^* = \{x^*(n) : n \in \mathcal{N}\}$ of $\mathbf{SLAD}(\mathcal{G})$ minimizes the first term. From (16), $Z^{\mathcal{G}}$ can be expressed as follows:

$$\begin{aligned} Z^{\mathcal{G}} = \max_{\tilde{p}} \min_{x} \quad & \sum_{\mathbb{P}\in\mathcal{P}} Pr(\mathbb{P}) \sum_{n\in\mathcal{N}_{\mathbb{P}}}\sum_{k\in K} f_k(x_k^{\mathbb{P}}(n)) + \sum_{\mathbb{P}\in\mathcal{P}} Pr(\mathbb{P})\tilde{p}(n)(-\sum_{k\in K} x_k^{\mathbb{P}}(n) + y(n)) \\ s.t. \quad & h(x_k^{\mathbb{P}}(n-), x_k^{\mathbb{P}}(n)) \le 0, \quad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P} \\ & x^{\mathbb{P}}(n) \in X(n), \qquad n \in \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P}. \end{aligned} \tag{2.19}$$

Notice that (2.19) is the dual problem of the formulation (2.15); hence p^* maximizes the second term ($Z^{\mathcal{G}}$). Q.E.D.

Theorem 2.4 shows that we can minimize PEL with the dispatch decision from $\mathbf{SLAD}(\mathcal{G})$ and the price signal from formulation (2.15). Notice that even when the dispatch decision is not optimal, the price obtained from formulation (2.15) is still optimal since PEL is divided into a dispatch related term and a price related term (see equation (2.16)). This allows us to treat the dispatch and price model independently. In practice, it is not realistic to assume that we would always be able to find optimal dispatch decisions in the sense of solving SLAD. Let $x^{\#}$ be sub-optimal dispatch decisions that we would encounter in practice, then we can still guarantee that p^* from the formulation (2.15) minimizes $PEL^{\mathcal{G}}(\tilde{p}, x^{\#})$.

Notwithstanding, we note that formulation (2.15) is not practical for being used in real-time pricing because it requires too many variables and the first set of constraints prevents the formulation from being separable. Here, we show a slightly modified version of (2.15) in order to make it more workable. The key is in the way to approach (2.17). Now, instead of changing (2.17) to (2.19), we redualize the dualized power balance constraints back to the constraints except for the one for the root node as follows:

$$\begin{aligned} Z^{\mathcal{G}} = \max_{\tilde{p}(n_0)} \mathbb{E}_{\mathbb{P}}[\min_{x\in X^{\mathbb{P}}} \quad & \sum_{n\in\mathcal{N}_{\mathbb{P}}}\sum_{k\in K} f_k(x_k(n)) + \tilde{p}(n_0)(-\sum_{k\in K} x_k(n_0) + y(n_0))] \\ s.t. \quad & \sum_{k\in K} x_k(n) = y(n), \quad n \in \mathcal{N}_{\mathbb{P}} \setminus n_0, \end{aligned} \tag{2.20}$$

where we express the set of inter-temporal constraints and the set of independent constraints as $X^{\mathbb{P}}$. Let $F_{\mathbb{P}}(\tilde{p}(n_0))$ be the inner optimization problem of (2.20). Then we can write $Z^{\mathcal{G}}$ more concisely as follows:

$$Z^{\mathcal{G}} = \max_{\tilde{p}(n_0)} \mathbb{E}_{\mathbb{P}}[F_{\mathbb{P}}(\tilde{p}(n_0))]. \tag{2.21}$$

Notice that we can compute the gradient of $F_{\mathbb{P}}(\tilde{p}(n_0))$ by Danskin's Theorem [Dan67]:

$$\nabla F_{\mathbb{P}}(\tilde{p}(n_0)) = -\sum_{k\in K} \bar{x}_k(n_0) + y(n_0), \tag{2.22}$$

where $\bar{x}_k(n_0)$ is the optimal solution for the inner optimization problem of (2.20).

Observe that it is possible to utilize the Stochastic Gradient Descent algorithm in order to find the maximizer $p^*(n_0)$ for (2.21). This modification allows us to deal with continuous uncertainty models by incorporating the principle of stochastic approximation methods in [RM51]. All we need is a model which samples sample paths. Instead of knowing the whole set of scenarios for the future, if we can somehow sample future sample paths, we can apply our approach to find $p^*(n_0)$. Practically, we can either directly use historical data as sample paths, or build an uncertainty model such as an auto-regressive model in order to predict the distribution of future information. Although this does not guarantee having an exhaustive uncertainty set which represents reality perfectly, at least this approach liberates us from discretization for building a scenario tree.

This modification can be further developed by binding cleared past prices as PMP does for the deterministic setting in section 2.2.3. In the next section, we introduce a PMP-style version of the PEL minimizing look-ahead model.

2.3.3 Binding Past Prices

Before we propose a formulation for coping with binding past prices under uncertainty, we define some additional notation. Referring to Figure 2.3, it is necessary to extend a scenario tree with its past path in order to link a scenario tree $\mathcal{G}$ starting from the current time step n_0 to the past. The extension is indeed a sub-tree of a larger scenario tree $\mathcal{H}$ whose root node is m_0. We denote the past path we have followed right before the current time step as $\mathbb{Q}$, and the node set of the path as $\mathcal{N}_{\mathbb{Q}}$. Naturally, we denote the extension of $\mathcal{G}$ as $\mathcal{G} \cup \mathbb{Q}$. For a future sample path including the current time step, we use $\mathbb{P}$ as in the previous sections. To avoid ambiguity, let the node set of $\mathcal{G}$ be $\mathcal{N}^{\mathcal{G}}$, and that of $\mathcal{H}$ be $\mathcal{N}^{\mathcal{H}}$.

Let the cleared past prices be $p_0^{\#} = \{p^{\#}(m) : \forall m \in \mathcal{N}_{\mathbb{Q}}\}$, and the past dispatch decisions be $x_0^{\#} = \{x^{\#}(m) : \forall m \in \mathcal{N}_{\mathbb{Q}}\}$. Let us denote the following optimization problem as **SPMP**$(\mathcal{H}, \mathcal{G})$, an abbreviation of Stochastic Price-preserving Multi-interval Pricing Model (SPMP):

$$\begin{aligned}
\min_{x} \quad & \sum_{\mathbb{P}\in\mathcal{P}} Pr(\mathbb{P}) \sum_{n\in\mathcal{N}_{\mathbb{Q}}\cup\mathcal{N}_{\mathbb{P}}} \sum_{k\in K} f_k(x_k^{\mathbb{P}}(n)) + \sum_{m\in\mathcal{N}_{\mathbb{Q}}} p^{\#}(m)\Big(-\sum_{k\in K} x_k^{\mathbb{P}}(m) + y(m)\Big) \\
s.t. \quad & \sum_{\mathbb{P}\in\mathcal{P}:n\in\mathcal{N}_{\mathbb{P}}} Pr(\mathbb{P}) \sum_{k\in K} x_k^{\mathbb{P}}(n) = \sum_{\mathbb{P}\in\mathcal{P}:n\in\mathcal{N}_{\mathbb{P}}} Pr(\mathbb{P}) y(n), \quad n \in \mathcal{N}^{\mathcal{G}} : p(n) \\
& h(x_k^{\mathbb{P}}(n-), x_k^{\mathbb{P}}(n)) \leq 0, \quad n \in \mathcal{N}_{\mathbb{Q}} \cup \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P} \\
& x^{\mathbb{P}}(n) \in X(n), \qquad n \in \mathcal{N}_{\mathbb{Q}} \cup \mathcal{N}_{\mathbb{P}}, \mathbb{P} \in \mathcal{P}.
\end{aligned} \tag{2.23}$$

Theorem 2.5. *(p^*, x^*) is a part of the minimizer for $PEL^{\mathcal{H}}(\tilde{p}, \tilde{x} | p_0^{\#}, x_0^{\#})$, where x^* is an optimal solution for **SLAD**$(\mathcal{G})$ and p^* is an optimal dual multiplier for **SPMP**$(\mathcal{H}, \mathcal{G})$.*

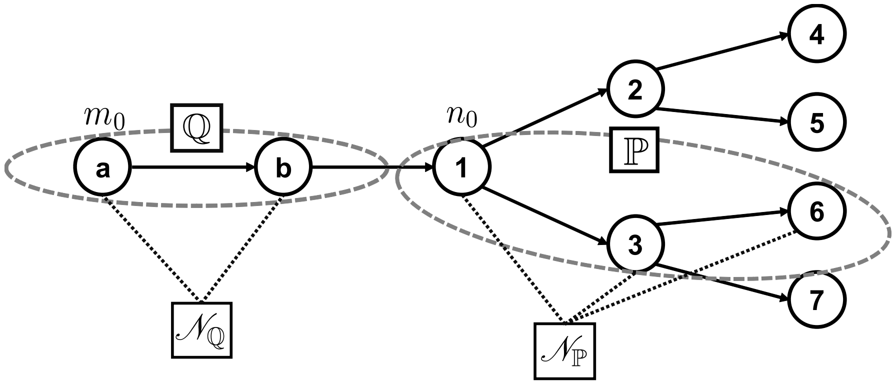

Figure 2.3: An example of a sub-tree of a scenario tree with the root node m_0 which incorporates another scenario tree with the root node n_0. Here, n_0 denotes the current time step. $\mathbb{Q}$ is the past path that starts from m_0 until right before the current time step, and $\mathcal{N}_{\mathbb{Q}}$ is the set of nodes in the past path $\mathbb{Q}$. One future sample path $\mathbb{P}$ including the current time step is shown with the set of nodes $\mathcal{N}_{\mathbb{P}}$ in the sample path $\mathbb{P}$.

Proof. The proof is a combination of the proofs of Theorem 2.1 and Theorem 2.4. Q.E.D.

Corollary 2.6. *If $(p_0^{\#}, x_0^{\#})$ is part of a minimizer for $PEL^{\mathcal{H}}(\tilde{p}, \tilde{x})$, then (p^*, x^*) in Theorem 2.5 is also part of a minimizer for $PEL^{\mathcal{H}}(\tilde{p}, \tilde{x})$.*

Theorem 2.5 and Corollary 2.6 are analogous to Theorem 2.1 and Corollary 2.2 respectively in the deterministic case. The price from SPMP minimizes PEL incorporating not only future but also past time periods given past decisions. It no longer treats past losses as sunk costs. However, notice the subtle difference between the two cases (deterministic / under uncertainty). In the set $\mathcal{N}^{\mathcal{H}} \setminus \mathcal{N}^{\mathcal{G}}$, there are nodes that are not in $\mathcal{N}_{\mathbb{Q}}$. Since the support for (p^*, x^*) is $\mathcal{N}^{\mathcal{G}}$, it is only a part of the minimizer for $PEL^{\mathcal{H}}(\tilde{p}, \tilde{x}|p_0^{\#}, x_0^{\#})$ in Theorem 2.5, unlike in the deterministic case where it is a minimizer for $LOC^{[t_s, t_e]}(\tilde{p}, \tilde{x}|p_0^{\#}, x_0^{\#})$ in Theorem 2.1. Notice that the nodes in $\mathcal{N}^{\mathcal{H}} \setminus (\mathcal{N}^{\mathcal{G}} \cup \mathcal{N}_{\mathbb{Q}})$ are meaningless given the current time step when the uncertainty of the past has been revealed. Mathematically speaking, when $(p_0^{\#}, x_0^{\#})$ is given for $\mathcal{N}_{\mathbb{Q}}$, the terms related to $\mathcal{N}^{\mathcal{G}} \cup \mathcal{N}_{\mathbb{Q}}$ in $PEL^{\mathcal{H}}(\tilde{p}, \tilde{x}|p_0^{\#}, x_0^{\#})$ become completely separable from the others; hence, it can be shown that (p^*, x^*) is a part of the minimizer for $PEL^{\mathcal{H}}(\tilde{p}, \tilde{x}|p_0^{\#}, x_0^{\#})$ without knowing the information for $\mathcal{N}^{\mathcal{H}} \setminus (\mathcal{N}^{\mathcal{G}} \cup \mathcal{N}_{\mathbb{Q}})$. The same argument can be applied to Corollary 2.6.

Now, we are in a position to propose a pricing method for rolling market clearing under uncertainty given past prices. Let us modify (2.23) as we have seen in the previous section (from (2.15) to (2.20) via (2.17)), so that we can apply the Stochastic Gradient Descent Algorithm for $\mathbf{SPMP}(\mathcal{H}, \mathcal{G})$ as follows:

$$\begin{aligned}
\max_{\tilde{p}(n_0)} \mathbb{E}_{\mathbb{P}}[\min_{x \in X^{\mathbb{Q}\cup\mathbb{P}}} \quad & \sum_{n \in \mathcal{N}_{\mathbb{Q}} \cup \mathcal{N}_{\mathbb{P}}} \sum_{k \in K} f_k(x_k(n)) + \sum_{m \in \mathcal{N}_{\mathbb{Q}}} p^{\#}(m)(-\sum_{k \in K} x_k(m) + y(m)) \\
& + \tilde{p}(n_0)(-\sum_{k \in K} x_k(n_0) + y(n_0))] \\
s.t. \quad & \sum_{k \in K} x_k(n) = y(n), \quad n \in \mathcal{N}_{\mathbb{P}} \setminus n_0,
\end{aligned} \tag{2.24}$$

where we express the set of inter-temporal constraints and the set of independent constraints for an extended sample path $\mathbb{Q} \cup \mathbb{P}$ as $X^{\mathbb{Q}\cup\mathbb{P}}$.

Let $F_{\mathbb{Q}\cup\mathbb{P}}(\tilde{p}(n_0))$ be the inner optimization problem of (2.24). Then (2.24) can be expressed as follows:

$$\max_{\tilde{p}(n_0)} \mathbb{E}_{\mathbb{P}}[F_{\mathbb{Q}\cup\mathbb{P}}(\tilde{p}(n_0))]. \tag{2.25}$$

Notice that we can compute the gradient of $F_{\mathbb{Q}\cup\mathbb{P}}(\tilde{p}(n_0))$ by Danskin's Theorem [Dan67]:

$$\nabla F_{\mathbb{Q}\cup\mathbb{P}}(\tilde{p}(n_0)) = -\sum_{k \in K} \bar{x}_k(n_0) + y(n_0), \tag{2.26}$$

where $\bar{x}_k(n_0)$ is the optimal solution for the inner optimization problem of (2.24).

Now SGD can be applied to find the maximizer $p^*(n_0)$ for (2.25). Note that (2.24) is very similar to (2.20). The main differences are twofold. First, x is defined under an extended sample path $\mathbb{Q} \cup \mathbb{P}$ instead of $\mathbb{P}$. Second, the inner optimization is equivalent to solving $\mathbf{PMP}(t_{m_0}, t_{n_0}, t_e)$ instead of solving $\mathbf{LAD}(t_{n_0}, t_e)$, where t_{m_0}, t_{n_0} denote the time steps of the nodes m_0, n_0 respectively, and t_e the time step of the leaf nodes in the scenario tree $\mathcal{H}$.

Thanks to Corollary 2.6, this sequential implementation (clear only the current time step sequentially as time passes) of $\mathbf{SPMP}(\mathcal{H}, \mathcal{G})$ using (2.24)-(2.26) results in the same solution as solving (2.15) under $\mathcal{H}$ for clearing prices for all possible scenarios at once. This property enables us to use $\mathbf{SPMP}(\mathcal{H}, \mathcal{G})$ in a more practical setting. In the next section, we briefly formalize the SGD algorithm for (2.24)-(2.26), and analyze practical details related to initialization and step size rules.

2.3.4 Stochastic Gradient Descent Algorithm for SPMP

We propose the following algorithm for computing prices in a rolling market clearing under uncertainty with binding past prices.

Table 2.1: Unit Parameters

Unit	Energy \$/MWh	Min,Max Output MW	Ramping MW/min
1	28	0,100	3
2	30	0,100	4
3	40	0,100	5

Algorithm 1: SGD for SPMP

Result: $p^I(n_0)$

$i \leftarrow 0$; Initialize $p^0(n_0)$;

while $i < I$ **do**

  Sample a sample path $\mathbb{P}$;

  Obtain $\overline{x}^i(n_0)$ by solving the inner optimization problem of (2.24);

  $\nabla F^i_{\mathbb{Q}\cup\mathbb{P}}(p^i(n_0)) \leftarrow (-\sum_{k\in K} \overline{x}^i_k(n_0) + y(n_0))$;

  $p^{i+1}(n_0) \leftarrow p^i(n_0) + \gamma_i \nabla F^i_{\mathbb{Q}\cup\mathbb{P}}(p^i(n_0))$;

  $i \leftarrow i + 1$;

end

For initializing $p^0(n_0)$, we use the dual multiplier of PMP with future expected demand as input data. In practice, using a deterministic model with expected demand data is a commonly used way to clear market. We use it as an initial value and update it in our algorithm.

For the step sizes $\{\gamma_i : i \in \{0, \dots, I\}\}$, there can be many variations. For updating step sizes, we consult mainly [Bot12]. Note that there is another variation of SGD often referred to as Averaged SGD based on [PJ92]. Detailed rules are introduced in section 5.3 of [Bot12]. While there exist many variations for selecting the initial step size, it is common to use the ratio of the upper bounds of the norm of the argument over the norm of the gradient as a fixed learning rate or dynamic learning rate with some diminishing rules [NJLS09]. In our experiment, we use the maximum cost over the current net load. The argument $p(n_0)$ cannot be greater than the maximum cost, and the upper bound of the gradient is bounded by $y(n_0)$. We have experimented with variations of step size rules according to changes in parameters regarding the initial step size and the rate of diminishing step size. The reader is referred to the details in Appendix 2.A.

2.4 Computational Results

2.4.1 An Illustrative Example

Let us first examine an illustrative three-stage example where two scenarios are possible for each node with equal probability as in Figure 2.4. We have

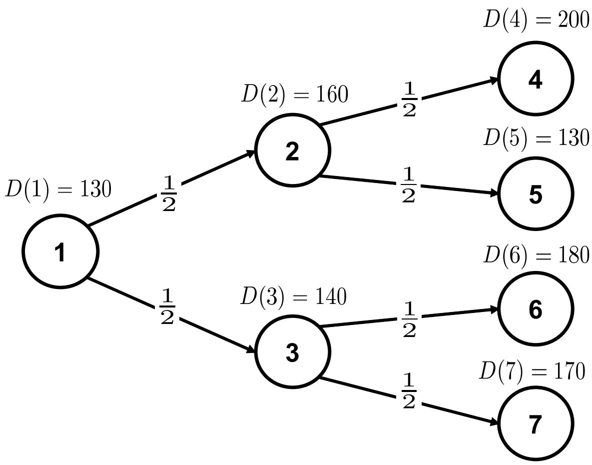

Figure 2.4: A scenario tree with demand for each scenario and transition probabilities between nodes.

Table 2.2: Market Clearing Solutions from Stochastic Model

Node (n)	$x_1^*(n)$ MW	SLAD $x_2^*(n)$ MW	$x_3^*(n)$ MW	SLAD $p^*(n)$ \$/MWh	SPMP $p^*(n)$ \$/MWh
1	90	40	0	28	28
2	100	60	0	30	32
3	85	55	0	25	28
4	100	80	20	40	40
5	90	40	0	28	30
6	100	75	5	40	34
7	100	70	0	30	34

three different units for generating power with ramp constraints as in Table 2.1. Each time step corresponds to 5 minutes. This example is an extension of Example 1 in [HSZ$^+$19] for the case under uncertainty. We show the results of different market clearing models in Table 2.2 and Table 2.3. For the stochastic model, the dispatch solution is solved by SLAD, and the two price distributions are obtained from SLAD and SPMP. For the deterministic model, the dispatch solution is obtained from LAD and the prices are obtained from LAD and PMP. Notice that future expected demand is used for the deterministic models, and as time moves forward (with new demand information updated) truncated problems are solved in a rolling fashion, whereas for the stochastic models the result is obtained from one optimization problem.

Performance metrics are compared in Tables 2.4 and 2.5 with the solutions

Table 2.3: Market Clearing Solutions from Deterministic Model

Node (n)	$x_1^*(n)$ MW	LAD $x_2^*(n)$ MW	$x_3^*(n)$ MW	LAD $p^*(n)$ \$/MWh	PMP $p^*(n)$ \$/MWh
1	100	30	0	30	30
2	100	50	10	40	30
3	90	50	0	28	28
4	100	70	30	40	40
5	100	30	0	28	30
6	100	70	10	40	32
7	100	70	0	40	32

Table 2.4: Comparison of Metrics with Dispatch Solutions from SLAD

Unit	SLAD			SPMP		
	AEL	PEL	MWP	AEL	PEL	MWP
1	0	5	3.4375	5	5	0
2	0	161.25	32.1875	47.5	47.5	0
3	0	0	0	7.5	7.5	1.875
SUM	0	166.25	43.125	60	60	1.875

that are presented in Table 2.2 and Table 2.3. We note that different dispatch solutions are used for Table 2.4 and Table 2.5, SLAD and LAD respectively; therefore, the values for SPMP are different in the two tables. In Table 2.4, observe that (i) for SLAD, AEL is 0 as the results of section 2.3.2, (ii) for SPMP, AEL $\leq$ PEL as foreseen by Theorem 2.3, and expected MWP $\leq$ PEL as discussed in section 2.3.1, (iii) most importantly, SPMP achieves smaller PEL and expected MWP than SLAD because SPMP produces prices that minimize PEL. In Table 2.5, we compare SPMP with deterministic pricing models (LAD and PMP). Observe that PMP achieves better performance than LAD since it accounts for binding past prices, and that SPMP attains better results than PMP since it accounts for the underlying uncertainty distribution. With the optimal dispatch solutions obtained from SLAD, the results can be further reduced to the values shown in Table 2.4. The reason why LAD has zero MWP in this example (it happens coincidentally, it is not a property of LAD) is because the price from this model is very high compared to other models in Table 2.2 and Table 2.3. This example shows that we can achieve zero MWP by clearing prices at high values but then LOC (AEL or PEL under uncertainty) increases significantly as shown in Table 2.5.

It is further worth noting that a stochastic equilibrium can perform poorly in terms of certain metrics. In Table 2.4, SLAD indeed achieves zero AEL, which is equivalent to a stochastic equilibrium; however, it exhibits rather high levels of PEL (the value that each agent would perceive when they calculate

Table 2.5: Comparison of Metrics with Dispatch Solutions from LAD

Unit	LAD			PMP			SPMP		
	AEL	PEL	MWP	AEL	PEL	MWP	AEL	PEL	MWP
1	0	0	0	0	0	0	0	0	0
2	275	275	0	62.5	62.5	0	62.5	62.5	0
3	0	0	0	70	70	17.5	55	55	13.75
SUM	275	275	0	132.5	132.5	17.5	117.5	117.5	13.75

LOC in an ex-post fashion) and expected MWP (simply losses). SPMP, on the other hand, by minimizing PEL directly, can regulate the level of AEL and expected MWP as by-products, but it does not constitute a stochastic equilibrium. In the next section, we present an experiment with realistic data.

2.4.2 Simulation with Realistic Data

In this section, we illustrate our proposal for pricing under uncertainty in a case study of the ISO New England (ISO-NE) system.

Case Study Description.

We consult [KLT16] for the grid, generator and load data based on ISO New England. The model includes 8 zones with 76 generators. The original source of data is hourly. In our case study, we are interested in five-minute time resolution, since certain binding operating constraints that are driven by the random variations of renewable supply are only observable at this shorter time frame. We assume linear cost functions in our analysis, hence we use only the first-order cost terms from [KLT16]. The linear cost terms and the ramp rates of the generators are re-scaled in order to account for the five-minute resolution of our model. Since there is no congestion due to sufficient transmission line capacity for the network in the original data, we adjusted the capacity (to 1500 MW for each line) to induce congestion resulting in the difference in prices for different zones, while making sure that there is no load-shedding because of the lack of capacity.

Minimum generation levels for the units are not specified in the original data. Instead, we assume that nuclear units have a technical minimum which is equal to 80% of their nominal output, 60% for coal-fired units, and 0% for the remaining technologies. We add pumped-hydro reservoirs to the model, in order to introduce an interesting interplay between storage and renewable supply. The pumped-hydro storage data is sourced from [PS17].

In order to introduce uncertainty to the model, we consider a scenario of large-scale wind power penetration. In terms of modeling wind power production, we follow the approach that is introduced by [PO13]. Concretely, we model wind speed using a time series model, and use a power curve in order to transform random fluctuations in wind speed to a resulting wind power stochastic process.

We use wind speed data with one-minute resolution from January 2018 to October 2019 from the Royal Meteorological Institute of Belgium (RMI). We source data with five-minute resolution and we use a cumulative empirical distribution for transforming the data. We remove monthly seasonal effects, and use an auto-regressive model with a 10-order lag (AR(10) model). We control the amount of uncertainty in the wind power production model with a tuning parameter, **WindRate**. When its value is one, the level of wind penetration corresponds to average wind production equal to 17% of annual ISO-NE energy demand and the highest possible penetration rate that we consider in our model amounts to 40% of annual ISO-NE energy demand.

Comparison of Metrics with Different Pricing Models.

The full horizon of our simulation consists of 312 intervals (26 hours), where an interval length is equal to five minutes. We ignore the first and last 12 intervals (one hour each) in our analysis in order to mitigate boundary effects. Two types of inter-temporal constraints exist in our model: ramping constraints and the constraints that represent the dynamics of pumped hydro storage.

Since we use a continuous stochastic uncertainty model (AR model), we can no longer compute AEL. Here, the focus is on comparing deterministic pricing models (LAD and PMP) with our method (SPMP). For the deterministic models, we use expected future demand. We use the solution from the LAD model as the dispatch decision for all the models, and our goal is to compare the effect of the different pricing models. We add one more model as a benchmark named PMP_PF (PMP with Perfect Foresight assumption), where we provide the actual sample path for future demand. For each model, we implement a moving horizon with a look-ahead length of 12 intervals (one hour). For the models which account for binding past prices (PMP, PMP_PF and SPMP), we add the information of past prices over the 12 most recent intervals (one hour). Notice that even the benchmark model PMP_PF can have positive value of PEL, because the look-ahead length is limited (12 intervals), whereas the full horizon length for calculating LOC is much longer (288 intervals). PMP_PF has the information of the actual demand for the future, however it does not solve a one-shot optimization with a full horizon length but instead a rolling implementation, as other models do. The goal of this comparison is to quantify how much the perfect foresight assumption can change the result ceteris paribus (including the look-ahead length).

Figure 2.5 shows the results of PEL and the expected make-whole-payments (MWP) for different models with increasing levels of uncertainty controlled by **WindRate**. The bars correspond to the average result of 500 experiments, and the middle lines show the sample standard deviation of the experiments. For our method, the iteration count for the SGD algorithm I is one hundred for SPMP. First, we can observe that the binding past prices version (PMP)

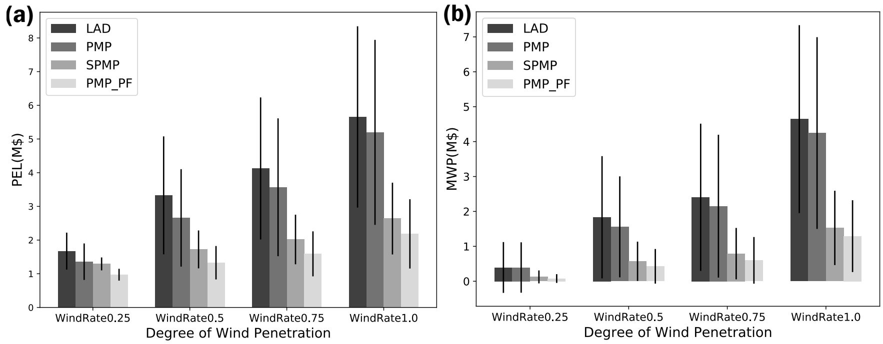

Figure 2.5: Ex-post expected lost opportunity cost for different models as the degree of wind penetration increases.

performs better than the simple look-ahead model (LAD). By exploiting the information of the distribution of the uncertain future, our method (SPMP) achieves significantly lower PEL and variance of the LOC than those models that use the expectation of future demand. Furthermore, SPMP achieves similar performance to the case with perfect foresight (PMP_PF). In Figure 2.5 (b), we can observe the same pattern as (a), but with even more noticeable differences. SPMP achieves lower expected value and variance for MWP than other models. Additionally, it also achieves a comparable result to the case with perfect foresight. The readers who are interested in the difference of prices resulting from different models are referred to Appendix 2.B.

Computation Time.

For the computing time of our method, 500 iterations of SGD require approximately 30 seconds in the experiment on a personal computer with 2.5-GHz dual-core CPU and 8GB of RAM. The results and discussion about the convergence of the SGD algorithm are available in Appendix 2.A.

2.5 Conclusion

In this chapter, we introduce two different definitions of expected lost opportunity cost, and we propose and analyze a pricing method for multi-interval real-time markets that operates under uncertainty. The proposed method minimizes one type of expected lost opportunity cost (PEL). We perform experimental results that demonstrate that our pricing approach results in lower PEL and expected make-whole payments with smaller variance, than alternative pricing methods that have been proposed in the recent literature. We further observe that the gap between our method and other methods that do not exploit

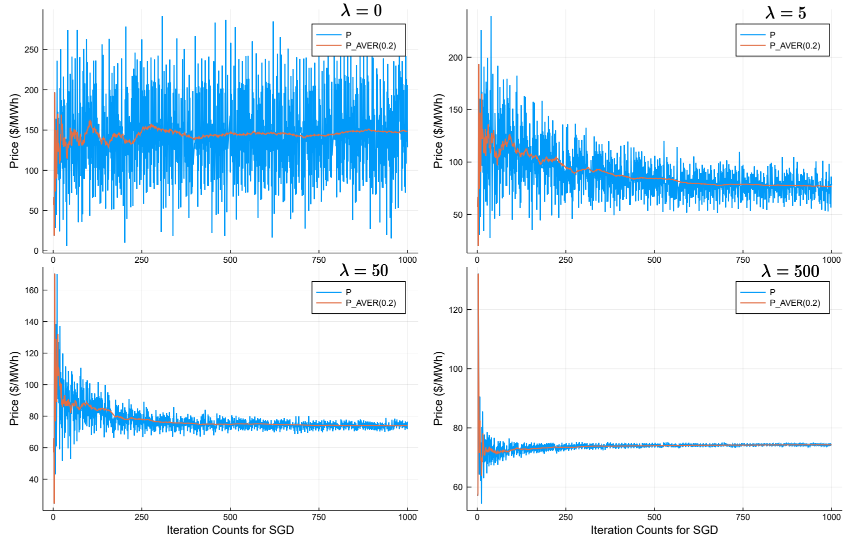

Figure 2.6: Convergence behavior of the SGD algorithm for varying levels of λ.

distribution information increases as the level of uncertainty increases. This indicates that our pricing proposal is especially suitable for future renewable integration scenarios, where the role of uncertainty is expected to become increasingly important, and where the accuracy of price signals will be a crucial element in preventing asset owners from "taking matters in their own hands" through self-commitment or self-dispatching. Our experimental results suggest that near-optimal prices can be obtained with a modest number of iterations of the Stochastic Gradient algorithm. This observation provides encouraging support to the claim that the method proposed in this chapter can be implemented within operationally acceptable time frames for real-time market clearing.

2.A Convergence of the SGD Algorithm According to Changes in Parameters

In this section, we focus on a type of step size rule which is provided by equation (2.27). The general form of this rule is from [Bot12].

$$\gamma_t = \gamma_0(1 + \lambda t)^{-3/4} \tag{2.27}$$

Notice that when $\lambda = 0$, this rule reduces to a constant step size. The parameter γ_0 is the initial step size, and λ controls the rate at which the step size diminishes.

First, let us observe the behavior of the SGD algorithm for different levels of λ when the initial step size γ_0 is fixed to 1/500. In Figure 2.6, the blue line corresponds to $p^t(n_0)$ in Algorithm 1 of section 2.3.4. and the orange line represents the average of the 20% of the last iterations; namely, $\sum_{i=\lceil 0.8t \rceil}^{t} p^i(n_0)/(t - \lceil 0.8t \rceil + 1)$. When $\lambda = 0$, as in the upper left figure, this is equivalent to a constant step size rule. In this case, the variance of the blue line does not decrease as the iteration count increases. Even though the average of the blue line (i.e. the orange line) appears to be converging, notice that the point of convergence is different from that of other figures (the orange line stabilizes around 140, whereas the other figures converge around 75). This might be surprising when one considers the behavior of deterministic (sub)gradient-based algorithms. Unlike deterministic gradient-based algorithms, however, a stochastic gradient descent algorithm does not necessarily generate a full gradient for each iteration, which should decrease as it converges to the (local) minimum point. Typically, what a stochastic gradient descent algorithm obtains at every iteration is a partial gradient whose expectation is the same as a full gradient. Especially for our problem, this partial gradient does not converge, even when the algorithm is very close to the optimal solution. Thus, it is necessary to use a diminishing step size rule ($\lambda > 0$ in our case).

Now, observe the behavior when $\lambda = 500$, at the other side of the range of values. In this case, it is possible that the rate of reduction of the step size is so high that the algorithm could not converge even after 1000 iterations. Out of the four different choices of λ, the lower left figure ($\lambda = 50$) appears to achieve the best performance. However, this may not be the case for different instances. Even though it is possible to find an optimal parameter for each instance, as long as λ is within a certain range (10 - 300 for this instance), the algorithm converges relatively robustly within 500 iterations.

In Figure 2.7, we fix λ to 50, and present the results for varying levels of the initial step size γ_0. When γ_0 is too low, as in the lower right figure ($\gamma_0 = 1/2500$), the speed of convergence can be too slow. Nevertheless, observe that the algorithm converges in all four cases, and the range of the level of this parameter that shows a good performance is wide.

In this chapter, we used $\lambda = 50$, $\gamma_0 = 1/500$ for all 8 zones of the ISO-NE case study. Notice that these values can be further optimized and it is also possible to use different parameters for different zones.

2.B Price Graphs Over Time for Different Models in Various Scenarios

In Figure 2.8, we present the price graphs over time for different models under several scenarios. The horizontal axis corresponds to time steps of 5 minutes each. The graphs present the prices for 288 time steps (24 hours). Except for case (d), we can observe that the model PMP_PF produces more spikes than

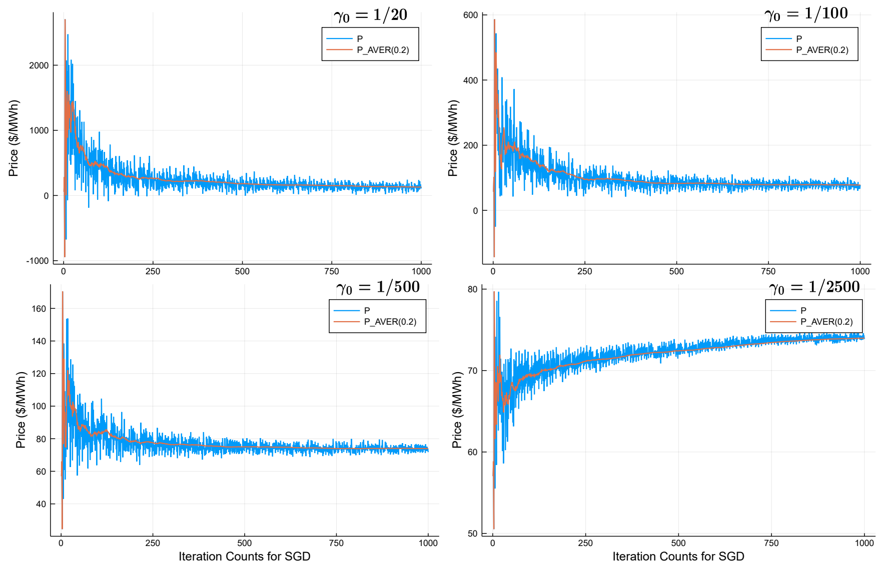

Figure 2.7: Convergence behavior of the SGD algorithm when varying the level of γ_0.

Table 2.6: Average prices from different models under various scenarios.

Scenario	Average Price over 24 hours ($/MWh)			
	LAD	PMP	SPMP	PMP_PF
(a)	83.3	83.7	95.2	98.7
(b)	100.3	93.9	100.7	104.6
(c)	113.7	112.2	112.1	115.4
(d)	132.9	127.6	126.1	127.3

the other models. This can be explained by the fact that the deterministic models (LAD, PMP) use an expected value for the future forecast, whereas the actual trajectory of the future net demand is used in PMP_PF. Especially when the variance of the forecast is large, the expectation of net demand fails to reflect the actual volatility of the net demand process, resulting in fewer spikes than the model with perfect foresight. SPMP tends to follow the trajectory of the price of PMP_PF rather than that of the deterministic models, but exhibits less volatility since SPMP considers all possible future scenarios and minimizes the expectation of LOC (PEL) whereas PMP_PF minimizes the LOC for a certain scenario. Among the two deterministic models, PMP is generally less volatile than LAD, and this is rather expected considering that PMP tends to smooth out the price trajectory by balancing the future and the past prices.

The average prices are presented in Table 2.6. There is a tendency for PMP_PF to produce higher values under different scenarios because of its more

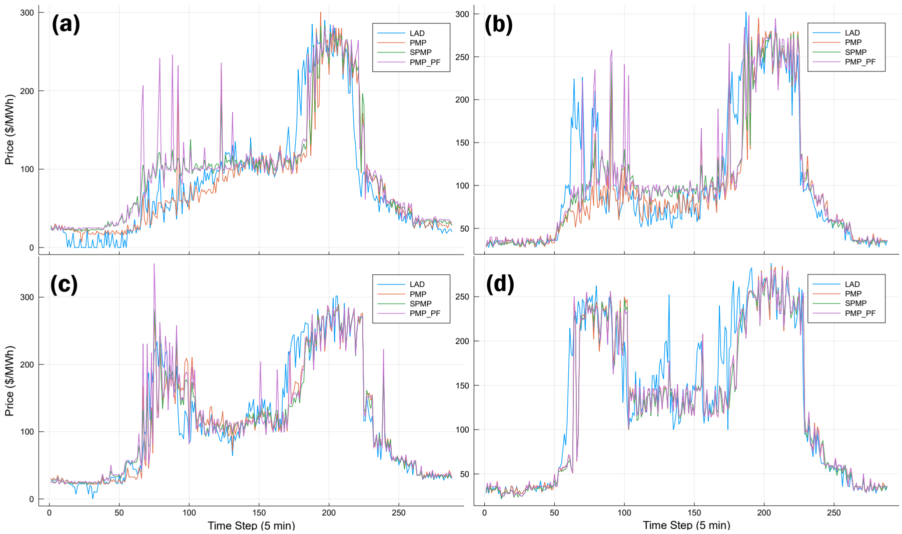

Figure 2.8: Various scenarios of prices over time for different pricing models.

frequent spikes. For a similar reason, PMP tends to exhibit a slightly lower value than LAD. SPMP is clearly lower than PMP_PF for most of the cases, if not all. Compared to the prices of the deterministic models, SPMP is higher for some scenarios and lower for other scenarios.

Part II

Chance-Constrained Multi-Area Reserve Dimensioning

3 Multi-Area Reserve Dimensioning Problem

3.1 Introduction

In Europe, transmission system operators (TSOs) are increasingly coordinating their system operations in response to the pan-European coupling of electricity markets [Com17b]. One of the objectives of this coupling is to organize a system that encompasses multiple areas for dispatching balancing energy from frequency restoration reserves in real time or close to real time (the MARI and PICASSO platforms)[1]. An important problem of interest that is emerging as a result of cross-zonal coordination in balancing is to allocate the right quantities of reserves in the right locations of the network while accounting for possible congestion in the transmission network. This problem is referred to as reserve sizing or reserve dimensioning, with the associated challenge of reserve deliverability [ZL08], [CGG13], depending on the context.

Article 157 of the System Operation Guideline (SOGL) of the European Union [Com17a] explicitly specifies probabilistic requirements for reserve sizing. The Nordic System Operation Agreement (SOA) [TSO19] is an example of an effort for the coordinated operation of frequency reserves among the Nordic countries in response to the SOGL. A recent ENTSO-E report [Ene22] by Danish TSO Energinet demonstrates the continual pursuit in the direction of multi-area reserve sizing in accordance with article 157 of the SOGL.

3.1.1 Literature Review

There exist a number of papers that attempt to address the multi-area reserve sizing problem in the literature. However, much of this literature [LHZ13, HGKK15, SJM18, PZBT20, WC21] focuses on either chance constraints or transmission constraints without treating these aspects jointly. Other literature [VMLA13, RMKA16] considers these two aspects simultaneously; nonetheless,

[1] MARI stands for “Manually Activated Reserves Initiative”, PICASSO stands for “Platform for the International Coordination of Automated Frequency Restoration and Stable System Operation”.

the underlying probabilistic distributions are assumed to belong to a specific class. Recent literature [PBA+21, CP22], which has been proposed by the authors of this paper, accounts for both of these characteristics. In [PBA+21], the authors define a chance-constrained formulation for the problem, but suggest a heuristic method that is not guaranteed to furnish the optimal solution. Subsequently, in [CP22], the authors attempt to solve this two-stage mixed-integer programming to optimality by applying integer programming techniques. However, the method proposed by the authors is not scalable to the size of realistic problems.

3.2 Problem Formulation

For a network $\mathcal{G}(Z,E)$, let r_z^+ [resp. r_z^-] denote the size of upward [resp. downward] balancing capacity of reserve for each zone z. Our goal is to minimize the sum of r_z^+ and r_z^- for all the zones in $\mathcal{G}(Z,E)$. Note that the objective function can be extended straightforwardly to the case where total procurement costs are considered through balancing capacity offers. In this case, the coefficients of $r_z^{+/-}$ would be different values from $+1$, but the method in this section can manage this type of adjustment. F^+ [resp. F^-] denotes the feasible region for r^+ [resp. r^-] representing the region where the capacity of reserve can cover imbalances δ_z for each zone z in the network $\mathcal{G}(Z,E)$. Formally, $F^{+/-}$ are defined as (3.1) and (3.2).

$$
\begin{aligned}
F^+ = \{r^+ \in \mathbb{R}_+^{|Z|} : \exists (p,f) \text{ s.t.} & \\
p_z + \delta_z = \sum_{e=(z,\cdot)\in E} f_e - \sum_{e=(\cdot,z)\in E} f_e, \quad & \forall z \in Z \\
p_z \le r_z^+, \quad & \forall z \in Z \\
-T_e^- \le f_e \le T_e^+, \quad & \forall e \in E\}
\end{aligned}
\tag{3.1}
$$

$$
\begin{aligned}
F^- = \{r^- \in \mathbb{R}_+^{|Z|} : \exists (p,f) \text{ s.t.} & \\
p_z + \delta_z = \sum_{e=(z,\cdot)\in E} f_e - \sum_{e=(\cdot,z)\in E} f_e, \quad & \forall z \in Z \\
-r_z^- \le p_z, \quad & \forall z \in Z \\
-T_e^- \le f_e \le T_e^+, \quad & \forall e \in E\}
\end{aligned}
\tag{3.2}
$$

Here, p_z and f_e are the amounts of balancing energy activated at zone z and the flow from z_1 to z_2, where $e = (z_1, z_2)$, given that link e has capacity limits T_e^+ and T_e^- in the reference and opposite direction respectively. The equations in the first lines denote the power balance equations for each zone z. The inequalities in the second line impose that the activation of balancing energy cannot exceed the amount of available reserve. Flow limits are imposed in the last inequalities. Note that the values of δ_z, $T_e^{+/-}$ can vary under different scenarios, as we discuss in the sequel.

Notice that, we assume that power flow constraints are approximated using a transportation network model. This assumption is aligned with the fact that, within MARI, the platform for the activation of manual frequency restoration reserve, the network will be approximated using an ATC (Available Transfer Capacity) transportation-based model [ACE20, PBDS20] at the launch of the platform.

Given reliability targets for upward/downward reserves $(1-\epsilon^{+/-})$, our problem can be written with probabilistic constraints (3.3b) as follows.

$$\min \sum_{z\in Z}(r_z^+ + r_z^-) \tag{3.3a}$$

$$\text{s.t. } \Pr\{r^{+/-} \in F^{+/-}\} \geq 1-\epsilon^{+/-} \tag{3.3b}$$

$$r^{+/-} \geq 0 \tag{3.3c}$$

This is a two-stage chance-constrained formulation where the first-stage variables are r^+, r^- and the second-stage variables are p_z and f_e.

3.2.1 Sample Approximation

Given a positive integer n, let us denote $[n]$ as the set $\{1,\dots,n\}$. Let us start from declaring $F_i^{+/-}$, i.e., the feasible set of $r^{+/-}$ when the uncertain parameters $(\delta_z, T_e^{+/-})$ are replaced by their realizations for scenario i.

$$\begin{aligned} F_i^+ = \{r^+ \in \mathbb{R}_+^{|Z|} : \exists (p,f) \text{ s.t.} \\ & p_z + \delta_{iz} = \sum_{e=(z,\cdot)\in E} f_e - \sum_{e=(\cdot,z)\in E} f_e, && \forall z \in Z : \lambda_z \\ & p_z \leq r_z^+, && \forall z \in Z : \pi_z \\ & f_e \geq -T_{ei}^-, && \forall e \in E : \mu_e^- \\ & f_e \leq T_{ei}^+, && \forall e \in E : \mu_e^+\} \end{aligned} \tag{3.4}$$

$$\begin{aligned} F_i^- = \{r^- \in \mathbb{R}_+^{|Z|} : \exists (p,f) \text{ s.t.} \\ & p_z + \delta_{iz} = \sum_{e=(z,\cdot)\in E} f_e - \sum_{e=(\cdot,z)\in E} f_e, && \forall z \in Z : \lambda_z \\ & -r_z^- \leq p_z, && \forall z \in Z : \pi_z \\ & f_e \geq -T_{ei}^-, && \forall e \in E : \mu_e^- \\ & f_e \leq T_{ei}^+, && \forall e \in E : \mu_e^+\} \end{aligned} \tag{3.5}$$

By introducing new binary variables $u_i^{+/-}$ for each scenario i, representing whether the probabilistic constraint is violated or not, our problem can be reformulated with logical constraints (3.6b) as follows.

$$\min \sum_{z\in Z}(r_z^+ + r_z^-) \tag{3.6a}$$

$$\text{s.t. } u_i^{+/-} = 0 \implies r^{+/-} \in F_i^{+/-}, \quad \forall i \in [N] \tag{3.6b}$$

$$\sum_{i \in N} u_i^{+/-} \leq \left\lfloor \epsilon^{+/-} N \right\rfloor \tag{3.6c}$$

$$r^{+/-} \geq 0, u^{+/-} \in \{0,1\}^N \tag{3.6d}$$

In this section, we show three different directions for addressing the constraints (3.6b). First, in section 3.3, by using the so-called "Big-M" method, we reformulate (3.6) into a big Linear Programming problem with both first- and second-stage variables. Although it is theoretically possible to obtain optimal solutions with this formulation, it is known that this method is not able to deal with big instances. Instead, a heuristic method derived from this formulation is presented. Next, in section 3.4, we introduce a method based on Benders' Decomposition. Lastly, in section 3.5, we use a projection method to represent the feasible regions of the first-stage variables r^+, r^- explicitly in the space of the first-stage variables. In the following sections, all of the methods are compared with one another.

3.3 Heuristic Method

First, we introduce another way to reformulate the problem using the so-called "Big-M" method. Even though this approach is not scalable in practice, it is also a basis of a heuristic method introduced in [PBA+21]. One of the key characteristics of this approach is to use the second-stage variables p and f directly and use a Big-M method to represent the logical expression Eq. (3.6b).

$$\begin{aligned}
\min\ & \sum_{z \in Z} (r_z^+ + r_z^-) \\
\text{s.t. } & p_{zi} + l_{zi}^+ - l_{zi}^- + \delta_{zi} = \sum_{e=(z,\cdot)\in E} f_{ei} - \sum_{e=(\cdot,z)\in E} f_{ei}, \quad \forall z \in Z, i \in [N] \\
& -r_z^- \leq p_{zi} \leq r_z^+, \quad \forall z \in Z, i \in [N] \\
& l_{zi}^+ \leq \max\{0, -\delta_{zi}\} \cdot u_i^+, \quad \forall z \in Z, i \in [N] \\
& l_{zi}^- \leq \max\{0, \delta_{zi}\} \cdot u_i^-, \quad \forall z \in Z, i \in [N] \\
& -T_{ei}^- \leq f_{ei} \leq T_{ei}^+, \quad \forall e \in E, i \in [N] \\
& \sum_{i \in N} u_i^{+/-} \leq \left\lfloor \epsilon^{+/-} N \right\rfloor \\
& r^{+/-} \geq 0, l^{+/-} \geq 0, u^{+/-} \in \{0,1\}^N
\end{aligned} \tag{3.7}$$

This is achieved by introducing slack variables $l^{+/-}$. Slack variables $l_{zi}^{+/-}$ are non-zero only when $u_i^{+/-} = 1$, enabling the power balance constraints to be violated. Notice that $l_{zi}^{+/-}$ is bounded by $\max\{0, -\delta_{zi}\}$ or $\max\{0, \delta_{zi}\}$. They are the upper and lower bounds for p_{zi} in absolute value. This is the reason why this approach is called the Big-M method since big-sized bounds such as $\max\{0, -\delta_{zi}\}$ or $\max\{0, \delta_{zi}\}$ are used to reformulate logical statements. Thanks to this large bound for $l_{zi}^{+/-}$, when $u_i^{+/-} = 1$, there exists a feasible solution with $p_i = 0, f_i = 0$, which allows scenario i to not be accounted for when calculating the size of reserves $r^{+/-}$.

Unfortunately, it is well known that formulations using the Big-M method are not practical for solving problems to optimality due to large LP relaxation gaps. However, the model of Eq. (3.7) can be used for developing a heuristic method in order to find a feasible solution. In [PBA+21], for example, the authors first solve the LP relaxation of Eq. (3.7) and fix $u_i^{+/-}$ to 1 for the indices in which the optimal solutions for the LP relaxation ${u^*}_i^{+/-}$ are in the sets of the $\lfloor \epsilon^{+/-} N \rfloor$ largest values when the solutions ${u^*}_i^{+/-}$ are sorted in descending order. The formal description of the algorithm is as follows in Algorithm 2.

Algorithm 2: LP-based Heuristic Method

STEP1: Solve the linear relaxation of the problem (3.7);
STEP2: Sort the optimal solutions for the LP relaxation ${u^*}^{+/-}$ in descending order;
STEP3: Define $U^{+/-}$ as the sets of scenarios corresponding to $\lfloor \epsilon^{+/-} N \rfloor$ highest ranked values of ${u^*}^{+/-}$;
STEP4: Solve the problem (3.7) with additional constraints $u_i^{+/-} = 1, \forall i \in U^{+/-}$, and $u_i^{+/-} = 0, \forall i \in (U^{+/-})^c$;

The biggest advantage of this algorithm is that it is comprised of solving two LPs (STEP1 and STEP4), instead of the original form of large-scale mixed integer programming (3.7). Therefore, it is scalable to the size of realistic instances. However, notice that this method does not guarantee to find optimal solutions for the original problem although its proposed solution is always feasible. In the later sections, we introduce exact methods that are designed to solve the original mixed integer programming problem to optimality. There are two variations; one with an approach based on Benders Decomposition, the other one with an approach based on minimal projection.

3.4 Exact Method Using Benders Decomposition

3.4.1 Benders Decomposition

Standard Benders Cuts

Introduced by Benders in [Ben62], the standard Benders Cuts are used in order to decompose a large problem into smaller problems by dividing variables into different groups. So, it is often used for solving stochastic programming with recourse where variables can be easily grouped by stages. In this way, the second stage variables can be decomposed into even smaller problems for each scenario. The method is based on two types of cuts: Benders Optimality Cuts and Benders Feasibility Cuts. The former connect the effects of the second stage variables to the objective function with the first stage variables, and the latter are used for representing the feasible region of the first stage variables depending on second stage variables. Since the second stage variables (p, f) in our problem do not show up in the objective function (3.6a), and what we need is to check $r^{+/-} \in F_i^{+/-}$, we only utilize Benders Feasibility Cuts throughout the paper.

For our problem, Benders Feasibility Cuts can be obtained by solving (3.8) and (3.9), which are the dual problems of the feasibility checking problems for $F_i^{+/-}$ when $\hat{r}^{+/-}$ is given.

$$\begin{aligned} v_i(\hat{r}^+) = \max - &\sum_{z\in Z}(\hat{r}_z^+\pi_z + \delta_{iz}\lambda_z) - \sum_{e\in E}(T_{ie}^+\mu_e^+ + T_{ie}^-\mu_e^-) \\ \text{s.t. } &\pi_z - \lambda_z = 0, \qquad \forall z \in Z \\ &\lambda_{z_t} - \lambda_{z_s} - \mu_e^+ + \mu_e^- = 0, \quad \forall e = (z_s, z_t) \in E \\ &\pi, \lambda \in [0,1]^{|Z|}, \mu^+, \mu^- \in [0,1]^{|E|} \end{aligned} \tag{3.8}$$

$$\begin{aligned} v_i(\hat{r}^-) = \max - &\sum_{z\in Z}(\hat{r}_z^-\pi_z + \delta_{iz}\lambda_z) - \sum_{e\in E}(T_{ie}^+\mu_e^+ + T_{ie}^-\mu_e^-) \\ \text{s.t. } &\pi_z + \lambda_z = 0, \qquad \forall z \in Z \\ &\lambda_{z_t} - \lambda_{z_s} - \mu_e^+ + \mu_e^- = 0, \quad \forall e = (z_s, z_t) \in E \\ &\pi, \lambda \in [0,1]^{|Z|}, \mu^+, \mu^- \in [0,1]^{|E|} \end{aligned} \tag{3.9}$$

Observe that the dual variables $\pi, \lambda, \mu^+, \mu^-$ correspond to certain constraints of (3.4) or (3.5). If $v_i(\hat{r}^{+/-}) > 0$, then $\hat{r}^{+/-} \notin F_i^{+/-}$. Let $(\hat{\pi}, \hat{\lambda}, \hat{\mu}^+, \hat{\mu}^-)$ be an optimal extreme point solution of (3.8). Then, for

$$\begin{aligned} \alpha &= \hat{\pi}, \\ \beta &= -\delta_i\hat{\lambda} - T_i^+\hat{\mu}^+ - T_i^-\hat{\mu}^-, \end{aligned} \tag{3.10}$$

$\alpha\hat{r}^+ < \beta$ and $\alpha r^+ \geq \beta$, for all $r^+ \in F_i^+$. Likewise, let $(\hat{\pi}, \hat{\lambda}, \hat{\mu}^+, \hat{\mu}^-)$ be an optimal extreme point solution of (3.9). Then, $\alpha\hat{r}^- < \beta$ and $\alpha r^- \geq \beta$, for all $r^- \in F_i^-$. We call this $\alpha r^{+/-} \geq \beta$ a Benders Feasibility Cut. For the rest of this subsection, we slightly abuse notation for the sake of simplicity in describing the procedure further. Instead of differentiating directions of reserves (positive or negative) as superscripts $(+/-)$, (r, F_i) are used to represent either of them.

Modified Benders Cuts

The standard Benders Cuts are not directly applicable to our problem. The Benders Feasibility Cuts represent $r \in F_i$, but what we need is the cuts which represent the logical expression in (3.6b). For that, we need the following optimization problem for each scenario[2]:

$$h_i(\alpha) := \min\{\alpha r | r \in F_i\}. \tag{3.11}$$

Here, $\alpha r + h_i(\alpha)u_i \geq h_i(\alpha)$ is a valid inequality for (3.6b), because if $u_i = 0, r \in F_i$, then $\alpha r \geq h_i(\alpha)$ by definition of $h_i(\alpha)$. Then, we can utilize the mixing set Eq. (1.23) and the mixing inequalities Eq. (1.24) introduced in section 1.4.3. Let us define the mixing set with these valid inequalities as follows:

$$P(\alpha) := \{(r, u) \in \mathbb{R}_+^{|Z|} \times \{0, 1\}^N : \alpha r + h_i(\alpha)u_i \geq h_i(\alpha), \forall i \in [N]\}. \tag{3.12}$$

Let us denote Π as the set of all the possible α, which are extreme points of (3.8). Notice that for our problem F_i for all $i \in [N]$ shares the common constraint set of (3.8), resulting in the same set of extreme points; hence the same Π for $F_i, \forall i \in [N]$.

Theorem 3.1. $\bigcap\limits_{\alpha \in \Pi} P(\alpha)$ *is equivalent to* $\{(r, u) : (3.6b), (3.6d)\}$.

Proof. Since all the inequalities in the set $P(\alpha)$ are valid for (3.6b), $\{(r, u) : (3.6b), (3.6d)\} \subseteq P(\alpha)$ for all $\alpha \in \Pi$. For $(\hat{r}, \hat{u})$ such that $\hat{u}_k = 0$ and $\hat{r} \notin F_k$, $\exists(\alpha, \beta)$ s.t. $\alpha\hat{r} < \beta$ but $\alpha\hat{r} \geq \beta$ is valid for F_k. By definition of $h_k(\alpha)$, for all $r \in F_k, \alpha r \geq h_k(\alpha) \geq \beta > \alpha\hat{r}$. So, $(\hat{r}, \hat{u})$ is violated by $\alpha r + h_k(\alpha)u_k \geq h_k(\alpha)$. Thus, $\bigcap\limits_{\alpha \in \Pi} P(\alpha) \subseteq \{(r, u) : (3.6b), (3.6d)\}$. Q.E.D.

Theorem 3.1 implies that the logical expression (3.6b) can be replaced by the modified Benders Cuts. These cuts can be strengthened by including the cardinality constraint (3.6c). For the simplicity of notation, let us denote $q = \lfloor \epsilon N \rfloor$, and for all $\alpha \in \Pi$, σ^α is a permutation of N integers such that

$$h_{\sigma_1^\alpha}(\alpha) \geq h_{\sigma_2^\alpha}(\alpha) \geq \cdots \geq h_{\sigma_N^\alpha}(\alpha).$$

Then, $\alpha r + (h_{\sigma_i^\alpha}(\alpha) - h_{\sigma_{q+1}^\alpha}(\alpha))u_{\sigma_i^\alpha} \geq h_{\sigma_i^\alpha}(\alpha)$ for all $i \in [q]$ is valid for $\{(r, u) : (3.6b), (3.6c), (3.6d)\}$. This is because for at least one of the $q+1$ largest values of $h_i(\alpha)$, $u_i = 0$ and this implies that $\alpha r \geq h_{\sigma_{q+1}^\alpha}(\alpha)$ is always valid. This argument is from Lemma 1 of [Lue14]. Let us define the mixing set with these new valid inequalities as follows:

$$\begin{aligned} P'(\alpha) := \{(r, u) \in \mathbb{R}_+^{|Z|} \times \{0, 1\}^N : (3.6c), \\ \alpha r + (h_{\sigma_i^\alpha}(\alpha) - h_{\sigma_{q+1}^\alpha}(\alpha))u_{\sigma_i^\alpha} \geq h_{\sigma_i^\alpha}(\alpha), \forall i \in [q]\}. \end{aligned} \tag{3.13}$$

Theorem 3.2. $\bigcap\limits_{\alpha \in \Pi} P'(\alpha)$ *is equivalent to* $\{(r, u) : (3.6b), (3.6c), (3.6d)\}$.

[2]Observe that F_i is nonempty. The recession cone of F_i is the positive orthant, whose dual cone is also the positive orthant. Since $\alpha = \hat{\pi}$ is a vector in the positive orthant as well, $h_i(\alpha)$ exists and is finite.

Proof. It is easy to see that $\{(r,u) : (3.6b), (3.6c), (3.6d)\} \subseteq P'(\alpha)$ for all $\alpha \in \Pi$, as in the proof of Theorem 1. Similarly, for $(\hat{r}, \hat{u})$ such that $\hat{u}_k = 0$ and $\hat{r} \notin F_k$, $\exists(\alpha, \beta)$ s.t. $\alpha\hat{r} < \beta$ but $\alpha\hat{r} \geq \beta$ is valid for F_k. Here, we need to consider two cases. If $h_{\sigma^\alpha_{q+1}}(\alpha) \geq h_k(\alpha)$, since $h_{\sigma^\alpha_i}(\alpha) - (h_{\sigma^\alpha_i}(\alpha) - h_{\sigma^\alpha_{q+1}}(\alpha))\hat{u}_{\sigma^\alpha_i} \geq h_{\sigma^\alpha_{q+1}}(\alpha) \geq h_k(\alpha) \geq \beta > \alpha\hat{r}$ for all $i \in [q]$. If $h_{\sigma^\alpha_{q+1}}(\alpha) < h_k(\alpha)$, then $k = \sigma^\alpha_i$ for some $i \in [q]$, and $h_{\sigma^\alpha_k}(\alpha) - (h_{\sigma^\alpha_k}(\alpha) - h_{\sigma^\alpha_{q+1}}(\alpha))\hat{u}_{\sigma^\alpha_i} \geq h_{\sigma^\alpha_{q+1}}(\alpha) \geq h_k(\alpha) \geq \beta > \alpha\hat{r}$. So, there exist some inequalities in (3.13) which violate $(\hat{r}, \hat{u})$. Thus, $\bigcap_{\alpha\in\Pi} P'(\alpha) \subseteq \{(r,u) : (3.6b), (3.6c), (3.6d)\}$. Q.E.D.

Theorem 3.2 shows that we can use these stronger valid inequalities to represent the logical expression (3.6*b*) thanks to the cardinality constraint (3.6*c*). Notice the difference between (3.12) and (3.13). The former has N inequalities whereas the latter has q inequalities, and generally $q << N$ since we consider $\epsilon \leq 0.01$. Now, we apply the star inequalities of [ANS00], or equivalently the mixing inequalities of [GP01] as in [Lue14].

Theorem 3.3. *([Lue14],[ANS00],[GP01]) Let* $T = \{t_1, t_2, \ldots, t_l\} \subseteq \{\sigma^\alpha_1, \ldots, \sigma^\alpha_q\}$ *be such that* $h_{t_i}(\alpha) \geq h_{t_{i+1}}(\alpha)$ *for* $i \in [l]$*, where* $h_{t_{l+1}}(\alpha) = h_{\sigma^\alpha_{q+1}}(\alpha)$*. Then the inequality*

$$\alpha r + \sum_{i=1}^{l}(h_{t_i}(\alpha) - h_{t_{t+1}}(\alpha))u_{t_i} \geq h_{t_1}(\alpha) \tag{3.14}$$

if valid for $P'(\alpha)$*.*

Clearly, (3.14) dominate the inequalities in (3.13), and they are known for their strength because they define the convex hull of the mixing set $P'(\alpha)$ without a cardinality constraint (3.6*c*). See ([Lue14],[ANS00],[GP01]) for more details.

Also, the inequalities in (3.13) are special cases of (3.14) when T is a singleton. So, $Q(\alpha)$ is equivalent to $P'(\alpha)$, where

$$Q(\alpha) := \{(r,u) \in \mathbb{R}^{|Z|}_+ \times \{0,1\}^N : (3.6c), (3.14)\}. \tag{3.15}$$

These mixing sets and new valid inequalities are analogous to the ones in section 1.4.3. $Q(\alpha)$ is similar to Eq. (1.25) and Eq. (3.14) to Eq. (1.26).

Even though the number of constraints in (3.14) grows exponentially with respect to q, for a given $(\hat{r}, \hat{u})$ the separation of these inequalities can be accomplished efficiently ([ANS00],[GP01]). Therefore, in [Lue14], they used these inequalities for their Branch-and-Cut algorithm with the separation problem. In this section, we move a bit further forward by using an extended formulation from [LAN10], introduced in section 1.4.4.

Extended Formulation

Let us first define the extended formulation of Q with q additional binary variables w^α as follows:

$$EQ(\alpha) := \{(r,u,w) \in \mathbb{R}^{|Z|}_+ \times \{0,1\}^{N+q} : (3.6c), (3.17) - (3.19)\}, \tag{3.16}$$

where

$$\alpha r + \sum_{i=1}^{q} (h_{\sigma_i^\alpha}(\alpha) - h_{\sigma_{i+1}^\alpha}(\alpha)) w_i^\alpha \geq h_{\sigma_1^\alpha} \tag{3.17}$$

$$w_i^\alpha - w_{i+1}^\alpha \geq 0, \qquad \forall i \in [q-1] \tag{3.18}$$

$$u_{\sigma_i^\alpha} - w_i^\alpha \geq 0, \qquad \forall i \in [q]. \tag{3.19}$$

Here the set $EQ(\alpha)$ is analogous to the set EG (Eq. (1.27)) in section 1.4.4. Therefore, according to Theorem 1.1, we can show that $EQ(\alpha)$ is also equivalent to $Q(\alpha)$. So, instead of solving the separation problem for (3.14), by utilizing the extended formulation $EQ(\alpha)$, we can have the same effect as adding the whole exponential family of valid inequalities (3.14).

3.4.2 Branch-and-Cut Algorithm

So far, we have explored ways to represent $\{(r, u) : (3.6b), (3.6c), (3.6d)\}$ by modifying the standard Benders Cuts for F_i. Since the size of Π is too big to handle with a one-shot optimization problem, to solve our problem to optimality, a procedure updating α which cuts off incumbent solutions is necessary. This procedure should be combined with the Branch-and-Bound algorithm to get an integer solution for u. Traditionally, this type of combination is called a Branch-and-Cut algorithm because in the course of the Branch-and-Bound algorithm, certain cuts (in our paper Benders Cuts) are added. One of the Branch-and-Cut algorithms using (3.14) for a general two-stage stochastic problem is well documented in [Lue14]. Here, we provide a diagram to describe our Branch-and-Cut algorithm specialized for our problem which is a slightly modified version using $EQ(\alpha)$.

Outline

Figure 1 shows the diagram illustrating the procedures of the Branch-and-Cut algorithm for our problem. In this subsection, we briefly explain the notation and the general flow of the algorithm. Formal descriptions for the Master Problem and the Separation Problem can be found in the following subsections.

The Branch-and-Bound algorithm is basically a tree-search method where we utilize the information of upper bounds and lower bounds of incumbent problems (denoted as Master Problem in the diagram) in order to narrow down the searching space. In Figure 1, $OPEN$ denotes the set of nodes that we still need to explore. At each node of the searching tree (l), we fix binary variables to a certain value (0 or 1); $N_0(l)$ is the set of variables fixed to 0 at node l, $N_1(l)$ the set of variables fixed to 1 at node l. The linear relaxation of the incumbent problem is solved at each node, which provides a lower bound (denoted as lb) for the original optimization problem. At the leaf of the tree or in the middle of the tree by coincidence, sometimes we get an (mixed) integer solution which gives an upper bound for the underlying optimization problem (denoted as U in our diagram) if that solution is feasible to F_i for all $i \in [N]$. Notice that when

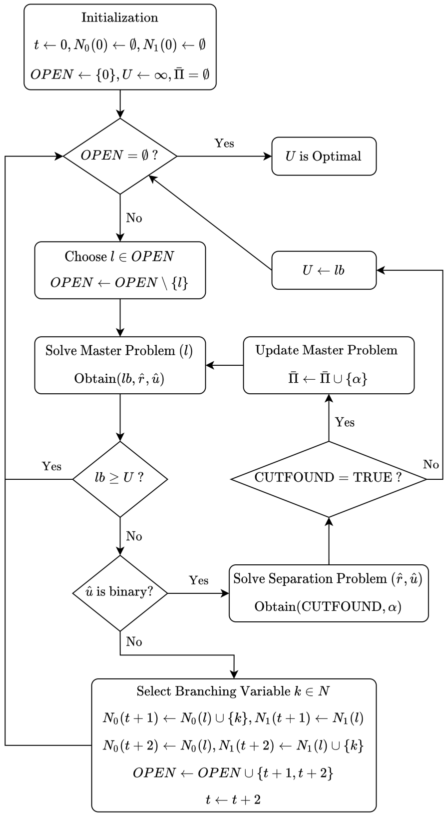

Figure 3.1: A Diagram for the Branch-and-Cut Algorithm

the lower bound for a certain node is higher than the current upper bound (U), it implies that we no longer need to explore that node. In this manner, the searching space gets narrowed down. This is what the rhombus ($lb \geq U$?) does in the diagram.

When the lower bound of the incumbent optimization problem is lower than U, it is still worth exploring that node. So, unless it happens to find an (mixed) integer solution, we branch the node by selecting a non-binary solution $\hat{u}_k$ and adding two new nodes to $OPEN$, where u_k is either fixed to 0 or 1 (see the bottom rectangle of the diagram).

If $\hat{u}$ is binary, the current solution is a candidate for being feasible to the original optimization problem. But, we still need to check if the current solution is feasible to F_i for all $i \in [N]$. That is what the Separation Problem does in the diagram. It checks if there is any $k \in [N]$ such that $\hat{r} \notin F_k$ and $u_k = 0$. CUTFOUND is a boolean whose value is TRUE if there exists such k and FALSE if there is none. When CUTFOUND = TRUE, Separation Problem returns α which cuts off the current solution. This α is added to $\bar{\Pi}$ to update the Master Problem. When CUTFOUND = FALSE, it means that the current solution is feasible to the original optimization problem, so we update U with the optimal objective function value for the incumbent problem Master Problem (l), which is lb. This process continues until there are no unexplored nodes left in the set $OPEN$. Then, the value U when the algorithm terminates is the optimal objective function value for our problem, and it is possible to store the optimal solution when we update U.

Master Problem

Here, we formalize what we solve as Master Problem. Master Problem (l) solves as follows:

$$\begin{aligned} \min \; & \sum_{z\in Z} r_z \\ \text{s.t.} \; & \sum_{i\in N} u_i \leq q \\ & (3.17)-(3.19), \quad \forall \alpha \in \bar{\Pi} \\ & u_k = 0, k \in N_0(l), u_k = 1, k \in N_1(l) \\ & r \geq 0, u \in [0,1]^N, w^\alpha \in [0,1]^q, \forall \alpha \in \bar{\Pi}. \end{aligned} \tag{3.20}$$

For the initial Master Problem (0), $\bar{\Pi}, N_0(l), N_1(l) = \emptyset$. So the optimal solution is $r_z = 0, \forall z \in Z$. During the course of the Branch-and-Cut Algorithm, the set $\bar{\Pi}$ is continually updated, so that (3.20) becomes closer to the original optimization problem until it finds the optimal solution.

After we solve (3.20), when it is infeasible Master Problem (l) returns $lb = \infty$, which results in that the optimal U becomes ∞. This implies that our original problem is infeasible. When (3.20) finds a feasible optimal solution, it returns the the optimal objective function value as lb, and corresponding optimal solution $(\hat{r}, \hat{u})$. Notice that the extended variable w is only used for tightening (3.20), and it is no longer used in the remaining procedure.

Separation Problem

As briefly stated in the outline, the Separation Problem $(\hat{r}, \hat{u})$ checks if the current solution is feasible to the original optimization problem, and if not it returns α which cuts off this solution.

This is done by solving $(3.8), \forall i \in [N]$ such that $\hat{u}_i = 0$. When $\hat{u}_i = 1$, it implies that $\hat{r}$ does not have to be feasible to F_i for that certain solution. If $v_i(\hat{r}) > 0$, then $\hat{r} \notin F_i$ and $\alpha = \hat{\pi}$ for (3.8). Notice that this procedure can be parallelized for each i. If Separation Problem finds k such that $v_k(\hat{r}) > 0$, then it returns a boolean CUTFOUND as TRUE with corresponding α, otherwise it returns CUTFOUND as FALSE.

When CUTFOUND = TURE, α is added to $\bar{\Pi}$. Here, it is possible to find just one single such α or instead screen all $i \in [N]$ such that $\hat{u}_i = 0$ and add all

α for all k such that $v_k(\hat{r}) > 0$). Notice that when we update Master Problem, we need to calculate $h_{\sigma_i^\alpha}, \forall i \in [N]$ and this can be also parallelized.

3.4.3 Computational Results

We compare the formulation of (3.7) and our Branch-and-Cut algorithm. For the underlying Branch-and-Bound algorithm, we used a commercial solver Gurobi 9.1. Here, in order to compare the power of different formulations, we did not use parallelization for Separation Problem. As simulation data, four different networks in Figure 3.2 are used for the computational experiments. For each network, we tested three levels of sample size : 3000, 4000, and 5000. For each sample $i \in [N]$, δ_{iz} is sampled from a normal distribution with mean = 0, standard deviation = 100 for each zone $z \in Z$. For capacity constraints[3], $[T^+_{e_1}, T^+_{e_2}, T^+_{e_3}] = [50, 80, 20], [T^-_{e_1}, T^-_{e_2}, T^-_{e_3}] = [40, 30, 90]$. For each setting, we test 100 instances and check how many instances are solved to optimality within the time limit (1800 seconds).

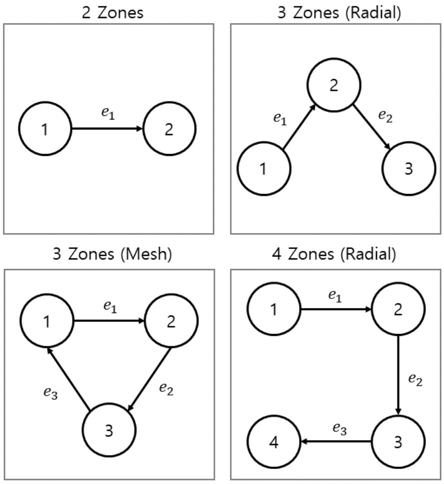

Figure 3.2: Networks for Simulation

[3]The capacity data could have been chosen to be random, but for the simplicity of presentation we chose fixed numbers for our simulation.

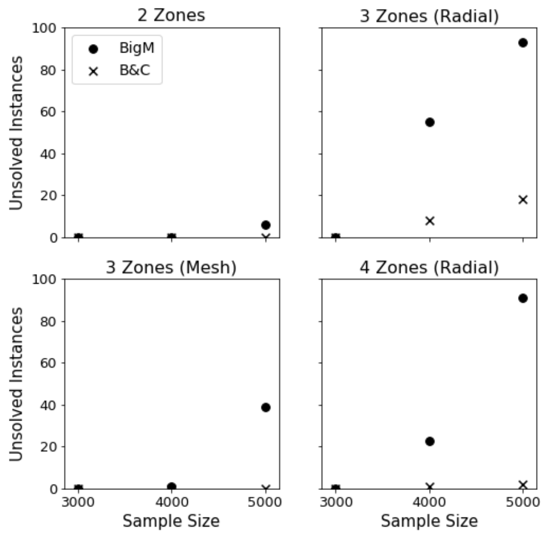

Figure 3.3: Comparison between Big-M Formulation and Branch-and-Cut Algorithm for Different Networks

Figure 3.3 shows the result of the computational experiments. It shows the number of unsolved instances among 100 test instances by two different methods (BigM: Eqs. (3.7), B&C: Branch-and-Cut Algorithm) for different networks in Figure 3.2. Observe that in every setting, the number of unsolved instances by B&C is lower than that by BigM. Even when BigM cannot solve more than 90% of the instances, B&C solves more than 80% of the instances. Notice that even though the effect of stronger valid inequalities (tight formulation of (3.16)) enhanced the performance of Branch-and-Bound drastically, the bottleneck of the B&C is the Separation Problem. If we use parallelization for calculation $v_i(\hat{r})$ or $h_{\sigma_{\hat{i}}^{\alpha}}$ in the Separation Problem, this performance can be further improved.

3.5 Exact Method Using Projection

3.5.1 Minimal Projection Formulation

In this section, we introduce a way to reformulate the constraints (3.3b) so that they represent $r^{+/-} \in F^{+/-}$ explicitly in the space of $r^{+/-}$. Intuitively, one might surmise that imposing that the reserves should be greater than or equal to the lowest possible aggregate imbalances corresponding to each element of the power set[4] of the vertex set would suffice to represent $F^{+/-}$ explicitly with

[4]Recall that the power set of a set is the set of all subsets of the set.

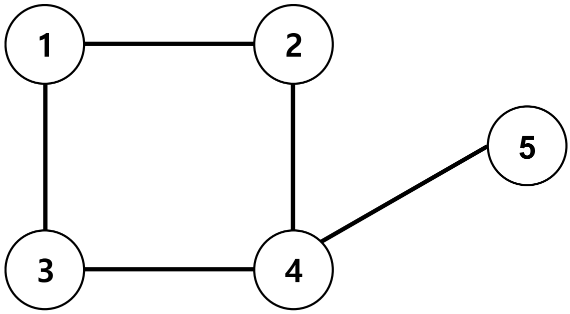

Figure 3.4: A graph with 5 zones for illustrating the definition of a connected vertex set.

$r^{+/-}$. Although this is not trivial, it happens to be correct. However, this representation is not minimal, since there are inequalities that can be represented by a linear combination of other inequalities. In this section, we present a compact representation that *is* minimal. It turns out that using only the subsets of the vertex set whose elements are "connected" on the network is enough to obtain a minimal projection. We, therefore, define this set as a connected vertex set in order to develop our analysis. For each element of the connected vertex set, imposing that the sum of the reserve capacities of the zones included in the element is greater than equal to the size of the total imbalances in the element minus the maximum input (or output) flow of the element, is sufficient to represent $F^{+/-}$. This statement can be formally stated as Theorem 3.4, and we prove it in the sequel. We commence by formally defining the following objects: connected vertex set, and maximum input/output flow.

Definition 3.1 (Connected Vertex Set)**.** *For a graph $\mathcal{G}(V, E)$, the connected vertex set $\mathcal{W}(\mathcal{G})$ is defined as follows:*

$$\mathcal{W}(\mathcal{G}) = \{S \subseteq V : \forall v, w \in S, \exists \text{ a path } P \text{ on } \mathcal{G} \text{ s.t. } v, w \in V(P) \subseteq S\}, \quad (3.21)$$

where $V(P)$ denotes the set of vertices in the path P.

Example 3.1 (Connected Vertex Set)**.** *For the graph in Fig. 3.4, $\{1, 2, 3\}$ is an element of a connected vertex set, whereas $\{1, 4\}$ is not. For a vertex set $\{1, 2, 3\}$, as an example, when $v = 2, w = 3$ there exists a path $3 \to 1 \to 2$ such that $v, w \in V(3 \to 1 \to 2) = \{1, 2, 3\} \subset \{1, 2, 3\}$. This is true for all possible combinations of $v, w \in \{1, 2, 3\}$, so this is an element of the connected vertex set. On the other hand, for a vertex set $\{1, 4\}$, when $v = 1, w = 4$, there is no such path whose vertex sets are subsets of $\{1, 4\}$, since all the paths between $v = 1$ and $w = 4$ contain either 2 or 3, which are not elements of $\{1, 4\}$.*

Intuitively, if all the elements in a vertex set have an edge that connects these vertices to any of the other elements in the vertex set, such vertices are

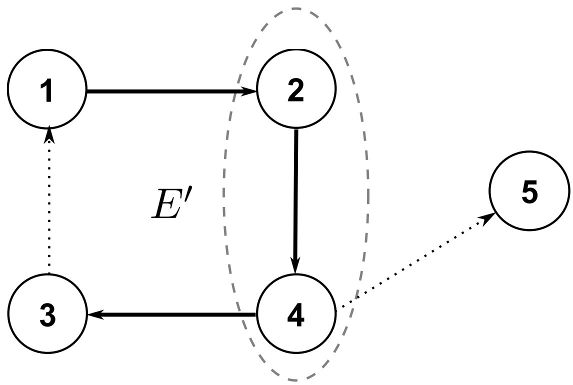

Figure 3.5: A directed graph for illustrating maximum input/output flow.

elements of a connected vertex set; hence the name of the definition. For the 5-zone example in Fig. 3.4, the connected vertex set is

$$\begin{aligned}\mathcal{W}(\mathcal{G}) = \{&\{1\},\{2\},\{3\},\{4\},\{5\},\\ &\{1,2\},\{1,3\},\{2,4\},\{3,4\},\{4,5\},\\ &\{1,2,3\},\{1,2,4\},\{1,3,4\},\{2,3,4\},\{2,4,5\},\{3,4,5\},\\ &\{1,2,3,4\},\{1,2,4,5\},\{1,3,4,5\},\{2,3,4,5\},\{1,2,3,4,5\}\},\end{aligned}$$

and the size of the connected vertex set $|\mathcal{W}(\mathcal{G})|$ *is 21.*

Definition 3.2 (Maximum Input/Output Flow). *For a directed graph* $\mathcal{G}(V,E)$ *where* $\forall e \in E, f(e)$ *denotes the flow in* e *and* $-T_e^- \leq f(e) \leq T_e^+$*, for all* $S \subseteq V, E' \subseteq E$*, the maximum input flow* $I(S|E')$ *and the maximum output flow* $O(S|E')$ *on* E' *are defined as follows:*

$$I(S|E') = \sum_{v\in S, w\in S^c:(v,w)\in E'} T_{(v,w)}^- + \sum_{v\in S, w\in S^c:(w,v)\in E'} T_{(w,v)}^+, \tag{3.22}$$

$$O(S|E') = \sum_{v\in S, w\in S^c:(v,w)\in E'} T_{(v,w)}^+ + \sum_{v\in S, w\in S^c:(w,v)\in E'} T_{(w,v)}^-. \tag{3.23}$$

Notice that maximum input/output flows are properties of the network since they are defined only on the basis of line capacities $T_e^{+/-}$.

Example 3.2 (Maximum Input/Output Flow). *In Fig. 3.5, every edge of the directed graph has flow capacities in both directions. For the edge* $(1,2)$*, the maximum capacity for the direction* $1 \to 2$ *is* $T_{(1,2)}^+$*, and the maximum capacity for the direction* $2 \to 1$ *is* $T_{(1,2)}^-$*. Maximum input/output flows are defined on an edge subset* $E' = \{(1,2),(2,4),(4,3)\} \subset E$ *for a vertex subset* S*. In Fig. 3.5,* $S = \{2,4\}$*, so the maximum input flow is* $I(S|E') = T_{(1,2)}^+ + T_{(4,3)}^-$*, and the maximum output flow is* $O(S|E') = T_{(1,2)}^- + T_{(4,3)}^+$*.*

Let us return to our problem. For the sake of brevity, let us denote $\mathcal{W}(\mathcal{G}(Z,E))$ for the network $\mathcal{G}(Z,E))$ for our original problem as $\mathcal{W}(\mathcal{G})$ from now on. Using this new notation, we define the three following sets F, F_p, F_r.

$$F = \{(r^+, r^-, p, f) \in \mathbb{R}_+^{|Z|} \times \mathbb{R}_+^{|Z|} \times \mathbb{R}^{|Z|} \times \mathbb{R}^{|E|} : (3.24) - (3.26)\}$$

$$p_z + \delta_z = \sum_{e=(z,\cdot)\in E} f_e - \sum_{e=(\cdot,z)\in E} f_e, \quad z \in Z \tag{3.24}$$

$$-r_z^- \leq p_z \leq r_z^+, \quad z \in Z \tag{3.25}$$

$$-T_e^- \leq f_e \leq T_e^+, \quad e \in E \tag{3.26}$$

$$F_p = \{(r^+, r^-, p) \in \mathbb{R}_+^{|Z|} \times \mathbb{R}_+^{|Z|} \times \mathbb{R}^{|Z|} : (3.27) - (3.28)\}$$

$$-I(S|E) \leq \sum_{z\in S}(p_z + \delta_z) \leq O(S|E), \quad S \in \mathcal{W}(\mathcal{G}) \tag{3.27}$$

$$-r_z^- \leq p_z \leq r_z^+, \quad z \in Z \tag{3.28}$$

$$F_r = \{(r^+, r^-) \in \mathbb{R}_+^{|Z|} \times \mathbb{R}_+^{|Z|} : (3.29) - (3.30)\}$$

$$\sum_{z\in S} r_z^- \geq \sum_{z\in S} \delta_z - O(S|E), \quad S \in \mathcal{W}(\mathcal{G}) \tag{3.29}$$

$$\sum_{z\in S} r_z^+ \geq -\sum_{z\in S} \delta_z - I(S|E), \quad S \in \mathcal{W}(\mathcal{G}) \tag{3.30}$$

Notice that F is a set defined with the same types of constraints as $F^{+/-}$ from (3.1) and (3.2), but considering upper and lower bounds for the balancing energy p_z at the same time in Eq. (3.25). Another difference is that F is in the space of $(r^{+/-}, p, f)$. The goal is to find a projection onto the space of $r^{+/-}$ *only*. The resulting projection is the set F_r and $F_r = F^+ \cap F^-$. F_p is a projection of F onto the space $(r^{+/-}, p)$, and it is used as an intermediate step to go from F to F_r in order to prove that F_r is indeed a projection of F. Since the proof for the set F is more general than the proofs for Eqs. (3.1) or (3.2), we present the proof for F in this section. From now on, for a set A defined in the space of variables (x, y), we denote $Proj_{(x)}(A)$ as the projection of the set A onto the space of x.

Theorem 3.4. *$Proj_{(r^+,r^-)}(F) = F_r$.*

Proof. The proof is based on two steps. First, in Claim 3.4.2, we show that the projection of F onto the space of (r^+, r^-, p) is F_p. Second, in Claim 3.4.1, we show that the projection of F_p onto the space of (r^+, r^-) is F_r. Claim 3.4.2 and Claim 3.4.1 together imply that $Proj_{(r^+,r^-)}(F) = F_r$. Q.E.D.

Claim 3.4.1. *$Proj_{(r^+,r^-)}(F_p) = F_r$.*

Claim 3.4.2. *$Proj_{(r^+,r^-,p)}(F) = F_p$.*

Proof of Claim 3.4.1

In this proof, we use V instead of Z as the vertex set in the set F_p and F_r.

Lemma 3.5. *For a graph* $\mathcal{G}(V,E)$,

$$I(S_1 \setminus S_2|E') + O(S_2 \setminus S_1|E') \leq I(S_1|E') + O(S_2|E'), \quad \forall S_1, S_2 \subseteq V, E' \subseteq E.$$

Proof. For the sake of compactness, without loss of generality, we leave out the conditions $(v,w) \in E'$ or $(w,v) \in E'$ under the summation sign. We can divide $I(S_1 \setminus S_2|E')$ into two terms:

$$\begin{aligned} I(S_1 \setminus S_2|E') = \sum_{v\in S_1\setminus S_2, w\in S_1^c} (T^-_{(v,w)} + T^+_{(w,v)}) \\ + \sum_{v\in S_1\setminus S_2, w\in S_1\cap S_2} (T^-_{(v,w)} + T^+_{(w,v)}). \end{aligned} \tag{3.31}$$

Observe that since $S_1 \setminus S_2 \subseteq S_2^c$, the second term

$$\sum_{v\in S_1\setminus S_2, w\in S_1\cap S_2} (T^-_{(v,w)} + T^+_{(w,v)}) \leq \sum_{v\in S_2^c, w\in S_1\cap S_2} (T^-_{(v,w)} + T^+_{(w,v)}). \tag{3.32}$$

By changing v and w, we can obtain

$$\sum_{v\in S_2^c, w\in S_1\cap S_2} (T^-_{(v,w)} + T^+_{(w,v)}) = \sum_{v\in S_1\cap S_2, w\in S_2^c} (T^+_{(v,w)} + T^-_{(w,v)}). \tag{3.33}$$

In a similar way, $O(S_2 \setminus S_1|E')$ can be divided into two terms:

$$\begin{aligned} O(S_2 \setminus S_1|E') = \sum_{v\in S_2\setminus S_1, w\in S_2^c} (T^+_{(v,w)} + T^-_{(w,v)}) \\ + \sum_{v\in S_2\setminus S_1, w\in S_1\cap S_2} (T^+_{(v,w)} + T^-_{(w,v)}). \end{aligned} \tag{3.34}$$

Since $S_2 \setminus S_1 \subseteq S_1^c$, the second term

$$\sum_{v\in S_2\setminus S_1, w\in S_1\cap S_2} (T^+_{(v,w)} + T^-_{(w,v)}) \leq \sum_{v\in S_1^c, w\in S_1\cap S_2} (T^+_{(v,w)} + T^-_{(w,v)}). \tag{3.35}$$

By changing v and w, we can obtain

$$\sum_{v\in S_1^c, w\in S_1\cap S_2} (T^+_{(v,w)} + T^-_{(w,v)}) = \sum_{v\in S_1\cap S_2, w\in S_1^c} (T^-_{(v,w)} + T^+_{(w,v)}). \tag{3.36}$$

Now observe that the sum of the first term of (3.31) and the right-hand-side of (3.36) is equal to $I(S_1|E')$. Likewise, the sum of the first term of (3.34) and the right-hand-side of (3.33) is equal to $O(S_2|E')$. Thus, $I(S_1 \setminus S_2|E') + O(S_2 \setminus S_1|E') \leq I(S_1|E') + O(S_2|E')$. Q.E.D.

Algorithm 3: Finding feasible $\hat{p}$ to F_p from $(\hat{r}^+, \hat{r}^-) \in F_r$

Input: $\mathcal{G} = (V, E)$, $(\hat{r}^+, \hat{r}^-) \in F_r$
Output: $\hat{p}$
Start with an empty set $R \leftarrow \emptyset$;
while $R \neq V$ **do**
  Choose $v \in V \setminus R$ such that $R \cup v \subseteq \mathcal{W}(\mathcal{G})$;
  Fix $\hat{p}_v$ satisfying (3.37) - (3.39);

$$-\hat{r}_v^- \leq \hat{p}_v \leq \hat{r}_v^+ \tag{3.37}$$

$$\hat{p}_v \geq -\sum_{w \in R \cap S} \hat{p}_w - \sum_{w \in S \setminus \{R \cup v\}} \hat{r}_w^+ - \sum_{w \in S} \delta_w - I(S|E), \qquad S \in \mathcal{W}(\mathcal{G}) : v \in S \tag{3.38}$$

$$\hat{p}_v \leq -\sum_{w \in R \cap S} \hat{p}_w + \sum_{w \in S \setminus \{R \cup v\}} \hat{r}_w^- - \sum_{w \in S} \delta_w + O(S|E), \qquad S \in \mathcal{W}(\mathcal{G}) : v \in S \tag{3.39}$$

  $R \leftarrow R \cup v$;
end while

Claim 3.4.1 $Proj_{(r^+, r^-)}(F_p) = F_r$.
Proof. First, we show that $Proj_{(r^+, r^-)}(F_p) \subseteq F_r$. From (3.28),

$$-\sum_{v \in S} r_v^- \leq \sum_{v \in S} p_v \leq \sum_{v \in S} r_v^+. \tag{3.40}$$

Now it is easy to see that Eqs. (3.40) and (3.27) imply Eqs. (3.29) and (3.30).

Second, we show that $F_r \subseteq Proj_{(r^+, r^-)}(F_p)$. It suffices to show that, for all $(\hat{r}^+, \hat{r}^-) \in F_r$, there exists $\hat{p}$ such that $(\hat{r}^+, \hat{r}^-, \hat{p}) \in F_p$. We show that we can find such $\hat{p}$ from Algorithm 3 and that it always exists. If it exists, it is easy to show that $\hat{p}$ satisfies Eq. (3.28) from Eq. (3.37). Also, observe that $\hat{p}$ satisfies Eq. (3.27) because for all $S \in \mathcal{W}(\mathcal{G})$, over the course of the while statement, there exists v, R such that $S \not\subseteq R, S \subseteq R \cup v$. Then Eqs. (3.38) and (3.39) for S with such v, R become Eq. (3.27).

Now, we show the existence of such $\hat{p}$ in Algorithm 3. We use mathematical induction. Denote R_i and v_i as the node sets and the nodes we get from Algorithm 3 as it iterates over the while statement. For the first step of the induction we consider the case where $R_1 = \emptyset$. The lower bound of (3.38) $\leq$ the upper bound of (3.37) is implied by Eq. (3.30) and the upper bound of (3.39) $\geq$ the lower bound of (3.37) is implied by Eq. (3.29). For showing why the lower bound of (3.38) $\leq$ the upper bound of (3.39), pick $S_1, S_2 \in \{S \in \mathcal{W}(\mathcal{G}) : v_1 \in S\}$. From

(3.29) for $S_2 \setminus S_1$ and (3.30) for $S_1 \setminus S_2$ [5] using Lemma 3.5,

$$\begin{aligned}\sum_{w\in S_1\setminus S_2} r_w^+ + \sum_{w\in S_2\setminus S_1} r_w^- \geq& \\ &-\sum_{w\in S_1\setminus S_2}\delta_w + \sum_{w\in S_2\setminus S_1}\delta_w - I(S_1\setminus S_2|E) - O(S_2\setminus S_1|E) \\ &\quad \geq -\sum_{w\in S_1\setminus S_2}\delta_w + \sum_{w\in S_2\setminus S_1}\delta_w - I(S_1|E) - O(S_2|E). \end{aligned} \quad (3.41)$$

Since $\sum_{w\in(S_1\cap S_2)\setminus v_1}(r_w^+ + r_w^-) \geq 0$, (3.41) implies

$$\sum_{w\in S_1\setminus v_1} r_w^+ + \sum_{w\in S_2\setminus v_1} r_w^- \geq -\sum_{w\in S_1}\delta_w + \sum_{w\in S_2}\delta_w - I(S_1|E) - O(S_2|E), \quad (3.42)$$

which is equivalent to the lower bound of (3.38) for $S_1 \leq$ the upper bound of (3.39) for S_2. Thus, $\hat{p}_{v_1}$ satisfying Eqs. (3.37) - (3.39) exists for the case where $R_1 = \emptyset$.

For the next step of mathematical induction, assume that, for $i \geq 1$, there exists $\hat{p}_{v_k}$ for $1 \leq k \leq i$ satisfying (3.37) - (3.39). For $R_{i+1} = R_i \cup v_i$ and $v_{i+1} \in V \setminus R_{i+1}$, our goal is to show that all the possible combinations of the upper bounds and the lower bounds from Eqs. (3.37) - (3.39) can be implied by other inequalities so that we can show that $\hat{p}_{v_{i+1}}$ exists. First, we show it for the combinations of upper bounds and lower bounds between (3.37) and (3.38) - (3.39). Here, we show one out of the two cases: the lower bound of (3.38) $\leq$ the upper bound of (3.37). The other case can be shown in a similar fashion. The set $\mathcal{W}(\mathcal{G})$ can be divided into two cases : i) $R_{i+1} \cap S = \emptyset$ and ii) $R_{i+1} \cap S \neq \emptyset$. For case i), $\sum_{w\in R_{i+1}\cap S}\hat{p}_w = 0$ and $\sum_{w\in S\setminus\{R_{i+1}\cup v_{i+1}\}}\hat{r}_w^+ = \sum_{w\in S\setminus v_{i+1}}\hat{r}_w^+$, so (3.30) implies that the lower bound of (3.38) $\leq$ the upper bound of (3.37). For case ii), from the set $\{v : v \in R_{i+1} \cap S\}$, pick the node with the largest index l. Observe that $\sum_{w\in R_{i+1}\cap S}\hat{p}_w = \sum_{w\in R_l\cap S}\hat{p}_w + \hat{p}_{v_l}$ and $\sum_{w\in S\setminus R_{i+1}}\hat{r}_w^+ = \sum_{w\in S\setminus\{R_l\cup v_l\}}\hat{r}_w^+$. This can be proven by contradiction. Assume that it is not true. Then $\exists v_m$ such that $m \neq l, v_m \in R_{i+1}, v_m \notin R_l$, and $v_m \in S$. This contradicts the fact that l is the largest index. Thus, (3.38) with R_l and v_l implies that the lower bound of (3.38) $\leq$ the upper bound of (3.37).

For showing why the lower bound of (3.38) $\leq$ the upper bound of (3.39), pick $S_1, S_2 \in \{S \in \mathcal{W}(\mathcal{G}) : v \in S\}$. We have four different cases to show : i) $R_{i+1} \cap S_1 = \emptyset$, $R_{i+1} \cap S_2 = \emptyset$, ii) $R_{i+1} \cap S_1 \neq \emptyset$, $R_{i+1} \cap S_2 = \emptyset$, iii) $R_{i+1} \cap S_1 = \emptyset$, $R_{i+1} \cap S_2 \neq \emptyset$, iv) $R_{i+1} \cap S_1 \neq \emptyset$, $R_{i+1} \cap S_2 \neq \emptyset$. Since it is similar to the other cases, here we only show the argument for case ii) where $R_{i+1} \cap S_1 \neq \emptyset$, $R_{i+1} \cap S_2 = \emptyset$. From the set $\{v : v \in R_{i+1} \cap (S_1 \setminus S_2)\}$,

[5] It is possible that $S_1 \setminus S_2 \notin \mathcal{W}(\mathcal{G})$ or $S_2 \setminus S_1 \notin \mathcal{W}(\mathcal{G})$, but in this case there exist disjoint $S_A, S_B \in \mathcal{W}(\mathcal{G})$ such that $S_A \cup S_B = S_1 \setminus S_2$ or $S_A \cup S_B = S_2 \setminus S_1$, and we can get the same results as Eq. (3.41) by summing up (3.29) or (3.30) for S_A and the same for S_B.

pick the node with the largest index l. Similar to what we have shown above, observe that $\sum\limits_{w\in R_{i+1}\cap(S_1\setminus S_2)} \hat{p}_w = \sum\limits_{w\in R_l\cap(S_1\setminus S_2)} \hat{p}_w + \hat{p}_{v_l}$ and $\sum\limits_{w\in(S_1\setminus S_2)\setminus R_{i+1}} \hat{r}^+_w = \sum\limits_{w\in(S_1\setminus S_2)\setminus\{R_l\cup v_l\}} \hat{r}^+_w$. From (3.38) for $S_1 \setminus S_2$ with R_l, v_l and (3.29) for $S_2 \setminus S_1$ using Lemma 3.5, following a similar process as in (3.41) and (3.42) we get the inequality,

$$\begin{aligned}\sum_{w\in R_{i+1}\cap S_1} \hat{p}_w + \sum_{w\in S_1\setminus R_{i+1}} r^+_w + \sum_{w\in S_2} r^-_w \geq \\ - \sum_{w\in S_1} \delta_w + \sum_{w\in S_2} \delta_w - I(S_1|E) - O(S_2|E),\end{aligned} \quad (3.43)$$

which is equivalent to the lower bound of (3.38) for $S_1 \leq$ the upper bound of (3.39) for S_2.

Thus, $\hat{p}_{v_{i+1}}$ satisfying (3.37) - (3.39) exists and it proves the existence of $\hat{p}$. Q.E.D.

Proof of Claim 3.4.2

Lemma 3.6. *For a graph $\mathcal{G}(V,E)$ for all $S_1, S_2 \subseteq V, E' \subseteq E$,*

$$O(S_1 \cup S_2|E') + O(S_1 \cap S_2|E') = O(S_1|E') + O(S_2|E') - \Phi(S_1, S_2|E')$$

$$I(S_1 \cup S_2|E') + I(S_1 \cap S_2|E') = I(S_1|E') + I(S_2|E') - \Phi(S_1, S_2|E')$$

where

$$\Phi(S_1, S_2|E') = \sum_{v,w\in(S_1\setminus S_2)\cup(S_2\setminus S_1):(v,w)\in E'} (T^+_{(v,w)} + T^-_{(v,w)}).$$

Proof. Since it follows an almost identical reasoning, we only show the case of Maximum Output Flow. For the sake of compactness, without loss of generality, we leave out the conditions $(v, w) \in E'$ or $(w, v) \in E'$ under the summation sign. Notice that $O(S|E')$ consists of the terms related to $T^+_{(v,w)}$ and those of $T^-_{(v,w)}$. In this proof, the patterns for $T^+_{(v,w)}$ and $T^-_{(v,w)}$ are exactly the same and what is important is the relationship of summations, so we omit $T^+_{(v,w)}$ and $T^-_{(v,w)}$ over the course of the equations. Notice that the right-hand-side can be written as follows:

$$\begin{aligned}O(S_1|E') + O(S_2|E') - \Phi(S_1, S_2|E') = \\ \sum_{v\in S_1, w\in S_1^c} + \sum_{v\in S_2, w\in S_2^c} - \sum_{v\in S_1\setminus S_2, w\in S_2\setminus S_1} - \sum_{v\in S_2\setminus S_1, w\in S_1\setminus S_2}\end{aligned} \quad (3.44)$$

Since

$$\sum_{v\in S_1, w\in S_1^c} = \sum_{v\in S_1\setminus S_2, w\in S_2\setminus S_1} + \sum_{v\in S_1\setminus S_2, w\in (S_1\cup S_2)^c}$$

$$+\sum_{v\in S_1\cap S_2,w\in S_2\setminus S_1}+\sum_{v\in S_1\cap S_2,w\in(S_1\cup S_2)^c} \tag{3.45}$$

$$\sum_{v\in S_2,w\in S_2^c}=\sum_{v\in S_2\setminus S_1,w\in S_1\setminus S_2}+\sum_{v\in S_2\setminus S_1,w\in(S_1\cup S_2)^c}$$
$$+\sum_{v\in S_1\cap S_2,w\in S_1\setminus S_2}+\sum_{v\in S_1\cap S_2,w\in(S_1\cup S_2)^c} \tag{3.46}$$

the first terms of (3.45) and (3.46) are crossed out with the third and the fourth term of (3.44). From the rest of the terms, observe that

$$\sum_{v\in S_1\setminus S_2,w\in(S_1\cup S_2)^c}+\sum_{v\in S_2\setminus S_1,w\in(S_1\cup S_2)^c}$$
$$+\sum_{v\in S_1\cap S_2,w\in(S_1\cup S_2)^c}=\sum_{v\in(S_1\cup S_2),w\in(S_1\cup S_2)^c} \tag{3.47}$$

$$\sum_{v\in S_1\cap S_2,w\in S_2\setminus S_1}+\sum_{v\in S_1\cap S_2,w\in S_1\setminus S_2}$$
$$+\sum_{v\in S_1\cap S_2,w\in(S_1\cup S_2)^c}=\sum_{v\in(S_1\cap S_2),w\in(S_1\cap S_2)^c}. \tag{3.48}$$

The right-hand-side of (3.47) is $O(S_1\cup S_2|E')$ and the right-hand-side of (3.48) is $O(S_1\cap S_2|E')$. Thus, $O(S_1\cup S_2|E')+O(S_1\cap S_2|E')=O(S_1|E')+O(S_2|E')-\Phi(S_1,S_2|E')$. Q.E.D.

Definition 3.3 (Net Output Flow)**.** *For a directed graph $\mathcal{G}(V,E)$ where $\forall e\in E, \hat{f}(e)$ denotes the flow in e, for all $S\subseteq V, E'\subseteq E$, the Net Output Flow on E', $\Gamma(S|E')$ is defined as follows:*

$$\Gamma(S|E')=\sum_{(v,w)\in E':v\in S}\hat{f}_{(v,w)}-\sum_{(v,w)\in E':w\in S}\hat{f}_{(v,w)}. \tag{3.49}$$

Lemma 3.7. *For a graph $\mathcal{G}(V,E)$,*

$$\Gamma(S_1|E')-\Gamma(S_2|E')=\Gamma(S_1\setminus S_2|E')-\Gamma(S_2\setminus S_1|E'),\forall S_1,S_2\subseteq V,E'\subseteq E.$$

Proof. For the sake of compactness, without loss of generality, we leave out the conditions $(v,w)\in E'$ under the summation sign.

$$\Gamma(S_1|E')=\sum_{v\in S_1\setminus S_2}\hat{f}_{(v,w)}+\sum_{v\in S_1\cap S_2}\hat{f}_{(v,w)}$$
$$-\sum_{w\in S_1\setminus S_2}\hat{f}_{(v,w)}-\sum_{w\in S_1\cap S_2}\hat{f}_{(v,w)} \tag{3.50}$$

$$\Gamma(S_2|E')=\sum_{v\in S_2\setminus S_1}\hat{f}_{(v,w)}+\sum_{v\in S_1\cap S_2}\hat{f}_{(v,w)}$$
$$-\sum_{w\in S_2\setminus S_1}\hat{f}_{(v,w)}-\sum_{w\in S_1\cap S_2}\hat{f}_{(v,w)} \tag{3.51}$$

Observe that

$$\begin{aligned}\Gamma(S_1|E') - \Gamma(S_2|E') &= (\sum_{v\in S_1\setminus S_2} \hat{f}_{(v,w)} - \sum_{w\in S_1\setminus S_2} \hat{f}_{(v,w)}) \\ &\quad - (\sum_{v\in S_2\setminus S_1} \hat{f}_{(v,w)} - \sum_{w\in S_2\setminus S_1} \hat{f}_{(v,w)}) \\ &= \Gamma(S_1 \setminus S_2|E') - \Gamma(S_2 \setminus S_1|E').\end{aligned} \tag{3.52}$$

Q.E.D.

Lemma 3.8. *For a graph $\mathcal{G}(V,E)$,*

$$\Gamma(S_1 \cup S_2|E') + \Gamma(S_1 \cap S_2|E') = \Gamma(S_1|E') + \Gamma(S_2|E'), \forall S_1, S_2 \subseteq V, E' \subseteq E.$$

Proof. It can be easily shown by the fact that $\sum_{v\in(S_1\cup S_2)} + \sum_{v\in(S_1\cap S_2)} = \sum_{v\in S_1} + \sum_{v\in S_2}$. Q.E.D.

Algorithm 4: Finding feasible $\hat{f}$ to F from $(\hat{r}^+, \hat{r}^-, \hat{p}) \in F_p$

Input: $\mathcal{G} = (V,E)$, $(\hat{r}^+, \hat{r}^-, \hat{p}) \in F_p$
Output: $\hat{f}$
Start with an empty set $Q \leftarrow \emptyset$;
while $Q \neq E$ **do**
1. Choose $(v,w) \in E \setminus Q$;
2. Fix $\hat{f}_{(v,w)}$ satisfying (3.53) - (3.57);

$$-T^-_{(v,w)} \leq \hat{f}_{(v,w)} \leq T^+_{(v,w)} \tag{3.53}$$

For all $S \in \mathcal{W}(\mathcal{G}) : v \in S, w \notin S$

$$\hat{f}_{(v,w)} \geq \sum_{u\in S}(\hat{p}_u + \delta_u) - \Gamma(S|Q) - O(S|E) + O(S|Q \cup (v,w)) \tag{3.54}$$

$$\hat{f}_{(v,w)} \leq \sum_{u\in S}(\hat{p}_u + \delta_u) - \Gamma(S|Q) + I(S|E) - I(S|Q \cup (v,w)) \tag{3.55}$$

For all $S \in \mathcal{W}(\mathcal{G}) : v \notin S, w \in S$

$$\hat{f}_{(v,w)} \geq -\sum_{u\in S}(\hat{p}_u + \delta_u) + \Gamma(S|Q) - I(S|E) + I(S|Q \cup (v,w)) \tag{3.56}$$

$$\hat{f}_{(v,w)} \leq -\sum_{u\in S}(\hat{p}_u + \delta_u) + \Gamma(S|Q) + O(S|E) - O(S|Q \cup (v,w)) \tag{3.57}$$

3. $Q \leftarrow Q \cup (v,w)$;

end while

Claim 3.4.2 $Proj_{(r^+,r^-,p)}(F) = F_p$.

Proof. First, we show that $Proj_{(r^+,r^-,p)}(F) \subseteq F_p$. Notice that (3.25) and (3.28) are identical. So, it suffices to show that Eqs. (3.24) and (3.26) imply

Eq. (3.27). From Eq. (3.24),

$$\sum_{v\in S}(p_v+\delta_v)=\sum_{v\in S,w\in S^c} f_{(v,w)}-\sum_{v\in S,w\in S^c} f_{(w,v)},\quad S\in\mathcal{W}(\mathcal{G}). \tag{3.58}$$

Now, it is easy to see that Eqs. (3.58) and (3.26) imply Eq. (3.27). Second, we show that $F_p \subseteq Proj_{(r^+,r^-,p)}(F)$. It suffices to show that for all $(\hat{r}^+,\hat{r}^-,\hat{p}) \in F_p$, there exists $\hat{f}$ such that $(\hat{r}^+,\hat{r}^-,\hat{p},\hat{f}) \in F$. We show that we can find such $\hat{f}$ from Algorithm 4 and that it always exists. If it exists, it is easy to show that $\hat{f}$ satisfies (3.26) from (3.53). Also, observe that $\hat{f}$ satisfies (3.24) from Eqs. (3.54) - (3.57). For all $v \in V$, let $E(v)=\{e\in E: e=(v,\cdot)\cup e=(\cdot,v)\}$. During the course of Algorithm 4, when we pick (v,w) such that $E(v)\subset Q\cup(v,w)$, with such Q and $S=\{v\}$, Eqs. (3.54) and (3.55) become Eq. (3.24). Likewise, when we pick (w,v) such that $E(v)\subset Q\cup(w,v)$, with such Q and $S=\{v\}$, Eqs. (3.56) and (3.57) become Eq. (3.24).

Now, we show the existence of such $\hat{f}$ in Algorithm 4. We use mathematical induction. For the first step we consider the case where $Q_1=\emptyset$. Then, $\Gamma(S|Q_1)=0$ for all $S\in\mathcal{W}(\mathcal{G})$. We want to show that (3.27) implies all the possible combinations of upper bounds and lower bounds among Eqs. (3.53) - (3.57). This can be done through Lemma 3.5 and 3.6. For the next step of mathematical induction, assume that for $i\geq 1$, there exists $\hat{f}_{(v_k,w_k)}$ for $1\leq k\leq i$ satisfying Eqs. (3.53) - (3.57). For $Q_{i+1}=Q_i\cup(v_i,w_i)$ and $(v_{i+1},w_{i+1})\in E\setminus Q_{i+1}$, our goal is to show that all the possible combinations of the upper bounds and the lower bounds from Eqs. (3.53) - (3.57) can be implied by other inequalities so that we can show that $\hat{f}_{(v_{i+1},w_{i+1})}$ exists. First we show this for the combinations of upper bounds and lower bounds between Eqs. (3.53) and (3.54) - (3.57). This can be done through Lemma 3.5 and 3.7. Lastly, we need to show that the lower bound of (3.54) $\leq$ the upper bound of (3.57) and the upper bound of (3.55) $\geq$ the lower bound of (3.56). This can be done through Lemma 3.6 and 3.7. Thus, $\hat{f}_{(v_{i+1},w_{i+1})}$ satisfying Eqs. (3.53) - (3.57) exists and it proves the existence of $\hat{f}$. Q.E.D.

Corollary 3.9. *$Proj_{(r^+,r^-)}(F^{+/-})=F_r^{+/-}$, where*

$$F_r^-=\{r^-\in\mathbb{R}_+^{|Z|}:(3.29)\},F_r^+=\{r^+\in\mathbb{R}_+^{|Z|}:(3.30)\}.$$

Theorem 3.4 and Corollary 3.9 show that the set F_r is indeed an explicit representation of the projection of F on the space of the first-stage variables $r^{+/-}$, resulting in $Proj_{(r^+,r^-)}(F^{+/-})=F_r^{+/-}$.

Theorem 3.10. *F_r is a minimal representation on the space of (r^+,r^-).*

Proof. Since the proof for the set of inequalities (3.29) is similar to the case for Eq. (3.30), we show here the case for Eq. (3.29). In order to show that Eq. (3.29) is a minimal representation on the space of r^-, to arrive at a contradiction, first let us assume that there exists a set $S' \in \mathcal{W}(\mathcal{G})$ such that there exist mutually different sets $S'_1, \ldots, S'_n \in \mathcal{W}(\mathcal{G})$ by which the inequality constructed in the form of Eq. (3.29) dominates the inequality for the set S'. Formally, this means that there exist coefficients $\alpha_1, \ldots, \alpha_n \geq 0$ that satisfy the following conditions (3.59) and (3.60):

$$\alpha_1 \sum_{v \in S'_1} r_v^- + \cdots + \alpha_n \sum_{v \in S'_n} r_v^- \leq \sum_{v \in S'} r_v^- \tag{3.59}$$

$$\alpha_1 \sum_{v \in S'_1} \delta_v + \cdots + \alpha_n \sum_{v \in S'_n} \delta_v - \alpha_1 O(S'_1|E) - \cdots \\ - \alpha_n O(S'_n|E) \geq \sum_{v \in S'} \delta_v - O(S'|E) \tag{3.60}$$

In order to satisfy the inequalities (3.59) and (3.60) for all possible values of r_v^- and δ_v, $\sum_{i:v \in S'_i} \alpha_i = 1$ for all $v \in S'$ and $\sum_{i:v \in S'_i} \alpha_i = 0$ for all $v \in V \setminus S'$. This implies that for all $i \in \{1, \ldots, n\}$, $S'_i \subseteq S'$ and $\bigcup_{i=1}^n S'_i = S'$. Notice that the left-hand side and the right-hand side of Eq. (3.59) are equal, and (3.60) becomes

$$O(S'|E) \geq \alpha_1 O(S'_1|E) + \cdots + \alpha_n O(S'_n|E). \tag{3.61}$$

Since the right-hand side of Eq. (3.61)

$$\alpha_1 O(S'_1|E) + \cdots + \alpha_n O(S'_n|E) = \\ \sum_{i:v \in S'_i} \alpha_i \cdot \Big(\sum_{v \in S', w \in (S')^c} T^+_{(v,w)} + \sum_{v \in S', w \in (S')^c} T^-_{(w,v)} \Big) + \tilde{O} \\ = O(S'|E) + \tilde{O},$$

where

$$\tilde{O} = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \Big(\sum_{v \in S'_i, w \in (S'_i)^c \cap S_j} T^+_{(v,w)} + \sum_{v \in S'_i, w \in (S'_i)^c \cap S_j} T^-_{(w,v)} \Big) > 0,$$

this contradicts the initial assumption. Q.E.D.

Corollary 3.11. *$F_r^{+/-}$ is a minimal representation on the space of $r^{+/-}$.*

Theorem 3.10 and Corollary 3.11 show that the sets of inequalities (3.29) and (3.30) are indeed minimal representations for the projection of F (or $F^{+/-}$). Notice that the inequalities (3.29) and (3.30) have an intuitive explanation. For certain combinations of zones, the sum of the reserve capacities should cover the total imbalances for the zones net of the maximum input [resp. output] flow for upward [resp. downward] reserves. For the case of infinite line capacities, where $T^{+/-}$ is infinity, our multi-area problem amounts to a single-zone problem. This can be checked from Eqs. (3.29) and (3.30), where the only non-redundant constraints are when $S \in \mathcal{W}(\mathcal{G})$ is equal to the set of all the zones Z, which is equivalent to the case where we aggregate all the zones in one region.

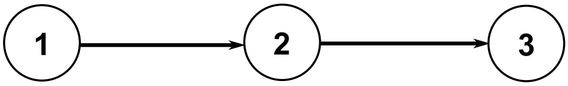

Figure 3.6: A directed graph for illustrating reformulations.

Example 3.3 (Minimal Projection)**.** *In order to illustrate the reformulation more explicitly, we provide a three-node example in Fig. 3.6. Firstly, the connected vertex set for the three-node graph is*

$$\mathcal{W}(\mathcal{G}) = \{\{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,2,3\}\}.$$

Accordingly, we write down all the constraints for F_r^+ as follows:

$$\begin{aligned}
&r_1^+ \geq -\delta_1 - T^-_{(1,2)} \\
&r_2^+ \geq -\delta_2 - T^+_{(1,2)} - T^-_{(2,3)} \\
&r_3^+ \geq -\delta_3 - T^+_{(2,3)} \\
&r_1^+ + r_2^+ \geq -\delta_1 - \delta_2 - T^-_{(2,3)} \\
&r_2^+ + r_3^+ \geq -\delta_2 - \delta_3 - T^+_{(1,2)} \\
&r_1^+ + r_2^+ + r_3^+ \geq -\delta_1 - \delta_2 - \delta_3.
\end{aligned}$$

F_r^- can be written down in a similar fashion.

As in Example 3.3, the representation $r^{+/-} \in F^{+/-}$ in constraint (3.3b) can be replaced by $A^{+/-} r^{+/-} \geq \xi^{+/-}$, where A^+ and A^- are linear maps, and there are uncertain parameters only in the right-hand-sides ξ^+ and ξ^-. The resulting new formulation is only involving the first-stage variables. Now that we have derived an explicit representation with inequalities, this new formulation enables us to apply the integer programming techniques for joint chance-constrained programs, introduced in chapter 1. Thanks to Corollary 3.11, we are guaranteed that our representation is minimal, so that the number of inequalities within the probabilistic constraints is minimized. This is crucial for the computational performance of our method, since, as the number of inequalities increases, the linear programming relaxation gap tends to be larger. We can now introduce a tightening step using integer programming techniques. In the following subsection, we introduce one such technique known as strong extended formulation.

3.5.2 Strengthened Formulation

Thanks to the projection formulation in the previous subsection, our problem (3.3) can be represented as follows.

$$
\begin{aligned}
\min \ & \sum_{z \in Z} (r_z^+ + r_z^-) \\
\text{s.t. } & \Pr\{A^{+/-} r^{+/-} \geq \xi^{+/-}\} \geq 1 - \epsilon^{+/-} \\
& r^{+/-} \geq 0
\end{aligned}
\tag{3.62}
$$

The main difference between this representation and Eq. (3.3b) is that the polyhedron on the space of the variables $r^{+/-}$ in the probabilistic constraint is now explicitly known in the form of a set of inequalities.

Again, we use sample approximation approach to reformulate the probabilistic constraint. For $i \in [N]$, δ_{zi} denotes the imbalance of scenario i at zone z and $T_{ei}^{+/-}$ the transmission network capacities of line e for scenario i. Notice that, in Eq. (3.62), $\xi^{+/-}$ is a linear combination of δ_z and $T_e^{+/-}$, as in Example 3.3. Here, $\xi_i^{+/-}$ refers to the right-hand-side vector under scenario i. By introducing new binary variables $u_i^{+/-}$ for each scenario i, our problem can be reformulated as follows.

$$
\begin{aligned}
\min \ & \sum_{z \in Z} (r_z^+ + r_z^-) \\
\text{s.t. } & A^{+/-} r^{+/-} + \xi_i^{+/-} u_i^{+/-} \geq \xi_i^{+/-}, \quad \forall i \in [N] \\
& \sum_{i \in N} u_i^{+/-} \leq \left\lfloor \epsilon^{+/-} N \right\rfloor \\
& r^{+/-} \geq 0, u^{+/-} \in \{0,1\}^N,
\end{aligned}
\tag{3.63}
$$

where $u_i^{+/-} = 1$ indicates that, under scenario i, the constraint of balancing the system is violated; thus, the number of violated scenarios should be less than or equal to $\epsilon^{+/-} N$, according to the direction of the reserves.

The reformulation Eq. (3.63) is a mixed-integer linear programming problem. Although it is a form that can be plugged into a commercial solver, this step alone cannot solve large-scale instances to optimality due to the big LP relaxation gap. In order to close this gap, we need to exploit the specific structure of the formulation. In Eq. (3.63), our main constraint has a form that is widely studied, the mixing set, that is introduced in chapter 1. Note that, when y is equal to the j-th row of $A^{+/-} r^{+/-}$, and $h_i = \xi_{ij}$, where ξ_{ij} denotes the j-th component of the vector ξ_i, our formulation in Eq. (3.63) has the form of a mixing set. In fact, there are multiple mixing sets, as many as the number of rows of $A^{+/-} r^{+/-}$. Each of these sets is tightened independently through the special types of inequalities named mixing inequalities. This allows us access to a set of techniques that help us close the LP relaxation gap. By utilizing the extended formulation Eq. (1.28), we obtain a new reformulation, the strengthened minimal projection formulation.

3.5.3 Strengthened Minimal Projection Formulation

Formulation

The strengthened minimal projection formulation using $F_r^{+/-}$ (Eq. (3.29), Eq. (3.30)) and EG in Eq. (1.27) is formally defined as follows:

$$\begin{aligned}
&\min \sum_{z\in Z}(r_z^+ + r_z^-) \\
&\text{s.t. } \sum_{z\in S} r_z^{+/-} + \sum_{i=1}^{q^{+/-}} (h_{S,i}^{+/-} - h_{S,i+1}^{+/-}) w_{S,i}^{+/-} \geq h_{S,1}^{+/-}, S \in \mathcal{W}(\mathcal{G}) \\
&\quad w_{S,i}^{+/-} - w_{S,i+1}^{+/-} \geq 0, \quad \forall i \in [q^{+/-} - 1], S \in \mathcal{W}(\mathcal{G}) \\
&\quad u_{\sigma_{S,i}^{+/-}}^{+/-} - w_{S,i}^{+/-} \geq 0, \quad \forall i \in [q^{+/-}], S \in \mathcal{W}(\mathcal{G}) \\
&\quad \sum_{i=1}^{N} u_i^{+/-} \leq q^{+/-} \\
&\quad r^{+/-} \geq 0, u^{+/-} \in \{0,1\}^N, w^{+/-} \in \{0,1\}^{q^{+/-}\cdot|\mathcal{W}(\mathcal{G})|},
\end{aligned} \tag{3.64}$$

where $q^{+/-} = \lfloor \epsilon^{+/-} N \rfloor$.

For $S \in \mathcal{W}(\mathcal{G})$,

$$h_{S,\sigma_{S,i}^+}^+ = -\sum_{v\in S} \delta_{v,i} - I_i(S|E) \tag{3.65a}$$

$$h_{S,\sigma_{S,i}^-}^- = \sum_{v\in S} \delta_{v,i} - O_i(S|E), \tag{3.65b}$$

where $\sigma_{S,i}^{+/-}$ are the permutations that rearrange the indices as $h_{S,1}^+ \geq h_{S,2}^+ \geq \cdots \geq h_{S,N}^+$ and $h_{S,1}^- \geq h_{S,2}^- \geq \cdots \geq h_{S,N}^-$, and for $i \in [N]$

$$I_i(S|E) = \sum_{v\in S, w\in S^c:(v,w)\in E} T_{(v,w),i}^- + \sum_{v\in S, w\in S^c:(w,v)\in E} T_{(w,v),i}^+, \tag{3.66}$$

$$O_i(S|E) = \sum_{v\in S, w\in S^c:(v,w)\in E} T_{(v,w),i}^+ + \sum_{v\in S, w\in S^c:(w,v)\in E} T_{(w,v),i}^-. \tag{3.67}$$

Implementation

The formulation of Eq. (3.64) is a Mixed-Integer Linear Programming problem which can be directly solved by commercial solvers such as CPLEX and GUROBI. However, in order to implement the algorithm, three elements are required: the connected vertex set $\mathcal{W}(\mathcal{G})$, the coefficients of mixing sets $h_{S,i}^{+/-}$ and the permutations $\sigma_{S,i}^{+/-}$ for all $S \in \mathcal{W}(\mathcal{G})$ and $i \in [N]$.

Generation of Connected Vertex Set

Algorithm 5: Generation of $\mathcal{W}(\mathcal{G})$

Input: $\mathcal{G} = (V, E)$
Output: $\mathcal{W}$
Select a start node $v_0 \in V$;
Initialize $\mathcal{W} = \{\{v_0\}\}, V_{sel} = \{v_0\}, E_{sel} = \emptyset$;
while $E_{sel} \neq E$ **do**
  Choose
  $e = (v, w) \in E(V_{sel}) = \{e' \in E : \exists v' \in V_{sel} \text{ s.t. } e' = (v', \cdot) \text{ or } e' = (\cdot, v')\}$;
  $E_{sel} \leftarrow E_{sel} \cup \{e\}$;
  if $v, w \in V_{sel}$ **then**
    $\mathcal{W}^v \leftarrow \{S \in \mathcal{W} : v \in S\}$;
    $\mathcal{W}^w \leftarrow \{S \in \mathcal{W} : w \in S\}$;
    for $S_1 \in \mathcal{W}^v, S_2 \in \mathcal{W}^w$ **do**
      $\mathcal{W} \leftarrow \mathcal{W} \cup \{S_1 \cup S_2\}$;
    end for
  else
    (WLOG assume $v \in V_{sel}$ and $w \notin V_{sel}$);
    $\mathcal{W} \leftarrow \mathcal{W} \cup \{\{w\}\}$;
    $V_{sel} \leftarrow V_{sel} \cup \{w\}$;
    $\mathcal{W}^v \leftarrow \{S \in \mathcal{W} : v \in S\}$;
    for $S \in \mathcal{W}^v$ **do**
      $\mathcal{W} \leftarrow \mathcal{W} \cup \{S \cup \{w\}\}$;
    end for
  end if
end while

The connected vertex set of a graph $\mathcal{G} = (V, E)$ can be generated using Algorithm 5. The size of the resulting connected vertex set $\mathcal{W}(\mathcal{G})$ varies according to the topology of the graph $\mathcal{G}$. When the graph is radial and all the nodes are connected (such as chains), the size of the connected vertex set is minimal and it is $|\mathcal{W}(\mathcal{G})| = 1/2 \cdot |V|(|V| + 1)$. The worst-case scenario is when the graph is a complete graph where all the nodes are connected to each other. In this case, the size of the connected vertex set is $|\mathcal{W}(\mathcal{G})| = 2^{|V|} - 1$. For the graph in Fig. 3.4, there are 5 zones, so the size ($|\mathcal{W}(\mathcal{G})| = 21$) is within the range of $4 \cdot 5/2 = 10 \leq 21 \leq 2^5 - 1 = 31$.

The $\mathcal{W}(\mathcal{G})$ Generation algorithm presented in this section has a worst-case complexity[6] of $\mathcal{O}(|E| \cdot 2^{|V|})$. This is significantly lower than the complexity of general-purpose projection methods. For instance, the Fourier-Motzkin-Elimination algorithm [Mot36], which is a well-known general projection method, is known for its double exponential complexity, that is $\mathcal{O}((|E| + |V|)^{2^{(|E|+|V|)}})$ for our case. A recently developed general algorithm [JMMT20] enjoys a sin-

[6] The algorithm scans every $e \in E$, and for each e in a worst-case scenario it can scan all the pairs of $v \in V$, which has a complexity of $\mathcal{O}(2^{|V|})$.

gle exponential complexity, but with much higher exponents; namely $\mathcal{O}((|E|+|V|)^{2.5(|E|+2|V|)} \cdot (|E|+2|V|)^3)$ for our problem.

Sorting

Now that we have $|\mathcal{W}(\mathcal{G})|$, we are ready to generate the remaining elements $h_{S,i}^{+/-}$ and $\sigma_{S,i}^{+/-}$. An easy way to obtain the parameters $h_{S,i}^{+/-}$, $\sigma_{S,i}^{+/-}$ is to calculate the right-hand-side for each scenario i in (3.65) and sort these right-hand side parameters in non-increasing order. Then, the resulting non-increasing sequence becomes $h_{S,i}^{+/-}$ and the corresponding permutation of the indices becomes $\sigma_{S,i}^{+/-}$. As the size of the connected vertex set can be exponential with respect to the number of zones, the time for the process of calculating $h_{S,i}^{+/-}$ and $\sigma_{S,i}^{+/-}$ is non-negligible. In the next subsection, we present the calculation time for pre-processing as well as the solver time for the optimization problem when we present the computational results for a case study.

3.5.4 Computational Results

In this section, we compare our strengthened minimal projection method with the alternatives of previous sections. First, we compare the projection method with another exact method using Benders' Decomposition in section 3.4. Additionally, we also compare with the heuristic method by [PBA+21] introduced in section 3.3.

Comparison with an Exact Method

In this subsection, we compare our method with another exact method that also guarantees to solve the problem to optimality [CP22], introduced in section 3.4. This method uses the theory in [Lue14], which is based on Bender's Decomposition [Ben62]. Roughly, it tries to solve the two-stage chance constraint problem directly instead of reformulating it. It does so by generating inequalities of the type of Eq. (1.26) through Bender's Decomposition. The computational results of this alternative method are much faster than solving the Big-M formulation of Eq. (3.7) to optimality shown in section 3.4.3. However, this direct approach to solving two-stage chance-constrained problems is not scalable to the size of realistic instances. The method in [CP22], for example, requires approximately 30 minutes to solve instances of four zones with 5,000 samples to optimality. On the other hand, our method using the formulation of Eq. (3.64) can achieve the optimal solution for the same size of instances within 1 second. One of the reasons for the subpar performance of this alternative method is the inequality generation step. For methods based on Bender's Decomposition in chance-constrained problems, in order to generate a single inequality that is similar to Eq. (1.26), a linear program should be solved N times, the number of scenarios. As the size of the problem increases, the number of inequalities and scenarios increase at the same time. On the contrary, through the minimal projection step and the additional strengthening step, our formulation is already in a compact form that does not require any inequality generating steps. Since this method has been found to not scale to large instances, we proceed to compare our method with a heuristic method using the Big-M formulation of Eq. (3.7).

3.5.5 Case Study: Comparison with a Heuristic Method

For the comparison, a case study of the Nordic system is considered. In this case study, as indicated in Fig. 3.7, three Nordic countries (Norway, Sweden and Finland) are involved, and they account for 10 bidding zones with 15 links. The reference data for imbalances for each zone and the network capacity are sourced from [Boe17]. For the imbalances, we generate samples from a normal distribution with zero mean and a standard deviation equal to the reference imbalances. For the network capacity, we add perturbations to the reference data for each sample. The perturbations are distributed according to a normal distribution with zero mean and a standard deviation equal to 5% of the value of the reference data. For all the figures Fig. 3.8 - 3.11, the bar charts refer to the mean of 100 simulations in which the middle lines indicate the standard deviation. Throughout the case study, we use GUROBI version 9.51 as optimization solver with JuMP embedded in the programming language Julia, and computing equipment with a SkyLake CPU (2.3GHz).

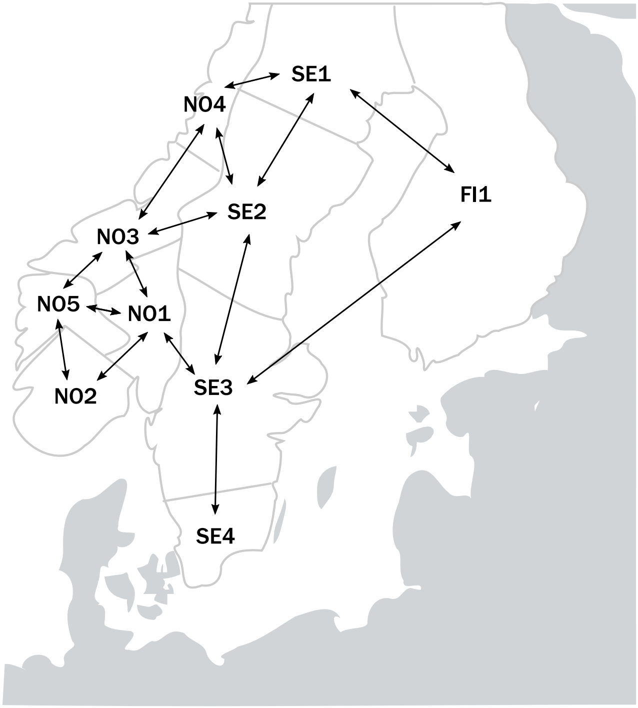

Figure 3.7: Bidding zones and transmission network lines for a case study of the Nordic countries.

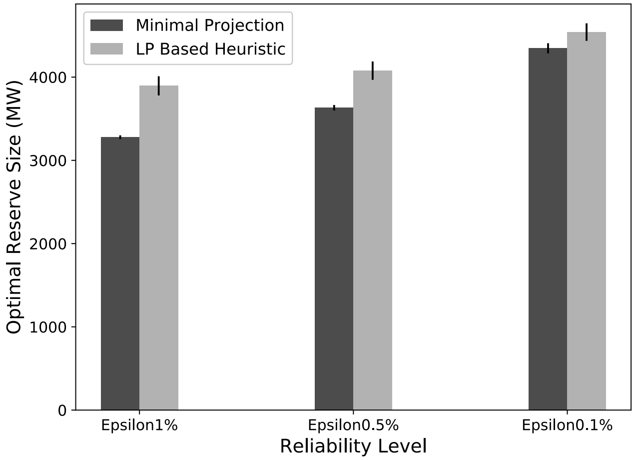

Figure 3.8: Comparison between Strengthened Minimal Projection Method and the LP Based Heuristic Method when the sample size is $N = 25,000$ in terms of optimal objective function value.

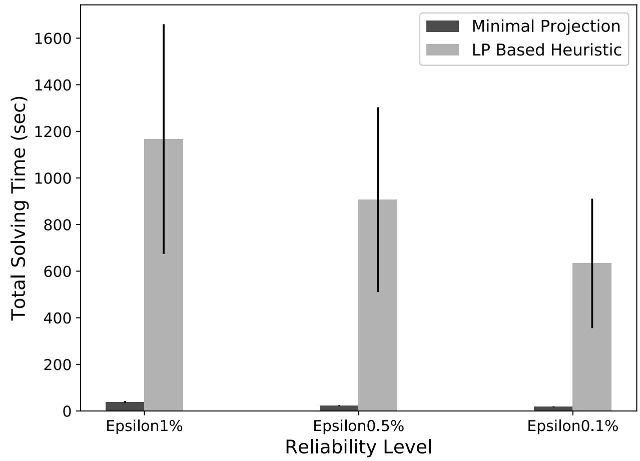

Figure 3.9: Comparison between Strengthened Minimal Projection Method and the LP Based Heuristic Method when the sample size is $N = 25,000$ in terms of total solving time.

We compare the minimal projection based method and the LP based heuristic method in [PBA+21], introduced in section 3.3. In Fig. 3.8 and Fig. 3.9, the results comparing the optimal reserve sizes and the total solving time[7] are presented for varying degrees of reliability levels (ϵ) when the sample size is $N = 25,000$. The Minimal Projection Method can be solved notably faster than the LP Based Heuristic, *and* finds the optimal solution. This seemingly counter-intuitive result can be explained by the fact that the strengthened minimal projection formulation (3.64) often has a smaller size than the formulation (3.7) in terms of the number of variables and constraints. Notice that $q << N$, thus each set of constraints in Eq. (3.64) is repeated q times whereas that in Eq. (3.7) is N times. Additionally, even though the size of the connected vertex set $\mathcal{W}(\mathcal{G})$ is exponential, it is often the case that $h^{+/-}$ in (3.65) are all negative, resulting in adding redundant constraints that can be ignored or automatically removed in pre-processing steps of commercial optimization solvers. This phenomenon happens more often when the capacities of lines $T^{+/-}$ are sufficiently large compared to the level of imbalances δ. In an extreme case where $T^{+/-}$ is infinity, the only constraints left are $\sum_{z\in Z} r^{+/-} \geq \mp \sum_{z\in Z} \delta_z$, which is equivalent to a single-zone problem.

The gap between the optimal solution and the sub-optimal solution from the LP method when $\epsilon = 1\%$ is around 18.9%. As ϵ becomes smaller, since q also becomes smaller, the optimization problem becomes less complex, resulting in a smaller gap between the optimal solution and the sub-optimal solution and

[7] The total solving time includes the data pre-processing time and the time for optimization solved by the commercial solver. The data pre-processing time refers to the time for generating a connected vertex set and sorting, which are necessary for actually running the solver.

a faster solving time. However, the high value of the standard deviation in the optimal solution for $\epsilon = 0.1\%$ implies that the sample size is not sufficient. When we increase the sample size, then the gap also increases.

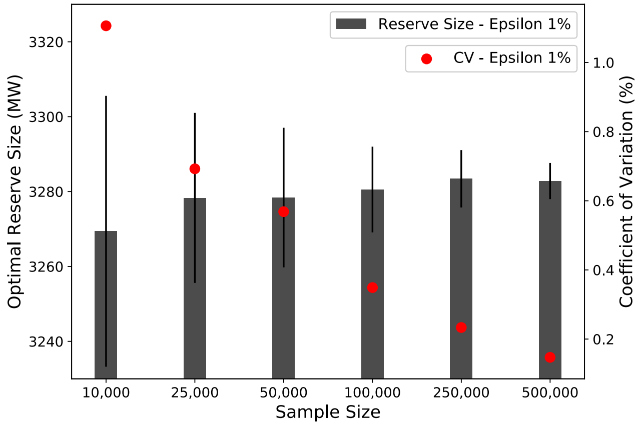

Figure 3.10: Sensitivity analysis for the sample approximation approach over sample size when epsilon is 1%.

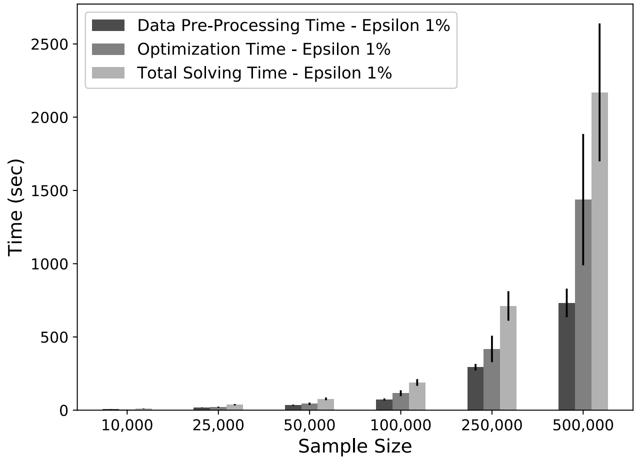

Figure 3.11: Solving time for the Strengthened Minimal Projection Method over sample size when epsilon is 1%.

Additionally, in Fig. 3.10, we present a sensitivity analysis of the sample approximation approach with respect to the sample size for our case study. There are two issues when the number of samples is low. First, the variance of the optimal objective function value is high. In Fig. 3.10, this is captured with the coefficient of variance (CV), which is the variance divided by its mean. One can observe that CV is the highest (1.1%) for the case of $N = 10,000$, and as the sample size increases, the CV decreases and the value of reserve stabilizes. Secondly, a typical phenomenon of the sample approximation method that is introduced in [LA08] is an underestimation of the true objective function value, in the sense that the resulting optimal objective function value tends to be lower than the true optimal objective function value of the problem. This can be observed in Fig. 3.10.

Lastly, we analyze the solving time for the Strengthened Minimal Projection Method over different sample sizes in Fig. 3.11. Notice that $\epsilon = 1\%$ is the most computationally complex problem among the three different levels of ϵ. Our method allows us to solve $N = 500,000$ samples in less than 30 minutes in terms of optimization time. In general, the data pre-processing time is non-negligible due to the exponential size of the connected vertex set $\mathcal{W}(\mathcal{G})$; however, the bottleneck complexity is $\mathcal{O}(N \log N)$ due to the sorting algorithm that is still scalable. In practice, if one needs to solve the problem dynamically, adding new samples to an already-sorted list (which can be expected to be the case in practice, based on information communicated to us by Nordic TSOs) is much easier than sorting the entire list, and in this case the data pre-processing time is negligible.

3.6 Conclusion

In this chapter, several methods are introduced for solving the chance-constrained multi-area reserve sizing problem. Even though the basic formulation is the same for all the methods, there can be various approaches especially through different types of reformulations. First, a heuristic algorithm is presented using "Big-M" method, that is one of the most generic way of reformulating such type of problem. Later, other two algorithms are based on integer programming techniques, mixing sets, mixing inequalities, and the strong extended formulation. The algorithm in section 3.4 is the Branch-and-Cut algorithm based on Benders' Decomposition. Finally, the last method using the minimal projection concept is exhibited. By identifying a minimal representation of the projected set of our feasible region, this approach can deal with instances of realistic size, and this is shown in a case study of the Nordic system.

All the three methods introduced in this section assume transportation-based networks. This means that all the approaches can also be used in different domains. Nevertheless, in future work, it is possible to extend the model with different approximations of power flow constraints in order to represent power systems more accurately. For example, a DC (Direct Current) approximation can be an option. Furthermore, from the perspective of better calculation, more recent IP techniques can be applied for solving the minimal projection formulation.

4 Conclusion

4.1 Summary of the contributions

The increasing need for considering uncertainty in power systems calls for advancements in optimization under uncertainty, providing decision-makers with the tools to make more informed choices. As practical requirements become more specific and complex, standard methodologies are no longer sufficient. In addition to algorithmic developments, modeling plays a crucial role as one of the essential foundations for further enhancements. Modeling not only bridges the gap between practical needs and current computational capabilities, but also contributes to improving the performance of specific problem types. This dissertation presents two primary examples to illustrate the importance and potential of modeling. The first example focuses on pricing under uncertainty, specifically in the context of real-time markets. The second example addresses the determination of the optimal reserve size while satisfying explicit probabilistic reliability constraints across multiple zones. The thesis is organized into two main parts, each corresponding to one of these distinct examples.

Chapter 2 studies pricing under uncertainty in multi-interval real-time markets. Considering the inter-temporal constraints of market participants, some system operators operate their markets in a multi-interval manner. One inevitable consequence of such multi-interval consideration is the rolling implementation of market clearing, where the market horizon is shifted at each time step, leading to the addition of new future information while losing past information. The analysis of the effects of this procedure is not straightforward, even in deterministic cases.

This chapter first discusses theories in deterministic cases, starting with the assumption of a fixed horizon, then addressing rolling implementation with a fixed horizon, and finally examining rolling implementation with a moving horizon. A fundamental finding in this analysis, which aims to mitigate the distortions resulting from rolling operation, is that the model for optimal dispatch decisions and the optimization model for pricing are not the same. This seemingly counterintuitive fact is discussed in terms of minimizing lost opportunity costs over a longer horizon, which aligns more closely with reality. Later, this is empirically demonstrated through a case study.

The analysis in the deterministic case allows us to extend our exploration to the uncertain case. In doing so, two alternative metrics are introduced, which are generalizations of the concept of lost opportunity cost. By unveiling the relationships between all these metrics, a method is proposed that minimizes one of the metrics, serving as an upper bound for all the others. This approach, which identifies the potential for reformulation, employs the stochastic gradient algorithm to solve the proposed method instead of resorting to multi-stage stochastic programming. As a result, it becomes possible to obtain reasonably close to optimal solutions for realistic-scale problems within a short time frame (5 - 15 minutes). This serves as an exemplary demonstration of the power of modeling because it establishes a connection between typical multi-stage stochastic programming and unconstrained stochastic programming by identifying their underlying relationship and addressing a practical need.

Chapter 3 addresses the multi-area reserve dimensioning problem. The goal of this problem is to find optimal allocations of reserves under reliability constraints explicitly specified as probabilistic constraints. Starting from a two-stage chance-constrained programming formulation, three different reformulations are explored, each leading to different methodologies. The last approach, in particular, exploits the minimal description of projection on first-stage variables and manages to solve real-world problems to optimality. Thanks to the simplicity of the method and its high performance, it has actually been adopted by one of the Nordic TSOs (Transmission System Operators) in their operation. This exemplifies the necessity of modeling, especially in problems involving integer (binary) variables.

Bibliography

[ACE20] ACER. ACER decision on the implementation framework for mfrr platform: Annex i. implementation framework for the European platform for the exchange of balancing energy from frequency restoration reserves with manual activation in accordance with article 20 of Commission regulation (EU) 2017/2195 of 23 November 2017 establishing a guideline on electricity balancing. Technical report, ACER, 2020.

[Age23] International Energy Agency. Renewable energy market update outlook for 2023 and 2024, 2023. URL: `https://iea.blob.core.windows.net/assets/63c14514-6833-4cd8-ac53-f9918c2e4cd9/RenewableEnergyMarketUpdate_June2023.pdf`.

[ANS00] A. Atamtürk, G. L. Nemhauser, and M. W. Savelsbergh. The mixed vertex packing problem. *Math. Program*, 89(1), 2000.

[ÁPJE23] D. Ávila, A. Papavasiliou, M. Junca, and L. Exizidis. Applying high-performance computing to the european resource adequacy assessment. *IEEE Transactions on Power Systems*, 2023.

[ÁPL21] D. Ávila, A. Papavasiliou, and N. Löhndorf. Parallel and distributed computing for stochastic dual dynamic programming. *Computational Management Science*, pages 1–28, 2021.

[Bal06] Ross Baldick. *Applied optimization: formulation and algorithms for engineering systems.* Cambridge University Press, 2006.

[Ben62] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1), 1962.

[BGC05] François Bouffard, Francisco D. Galiana, and Antonio J. Conejo. Market-clearing with stochastic security. *IEEE Transactions on Power Systems*, 20(4):1818–1826, 2005.

[BH22] D. Biggar and M. Hesamzadeh. Do we need to implement multi-interval real-time markets? *The Energy Journal*, 43(2):111–131, 2022.

[BL11] John R Birge and Francois Louveaux. *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

[Blo82] Jeremy A Bloom. Long-range generation planning using decomposition and probabilistic simulation. *IEEE Transactions on Power Apparatus and Systems*, (4), 1982.

[Boe17] J. Boe. Balancing energy activation with network constraints. Master's thesis, Norwegian University of Science and Technology, 2017.

[Bot98] L. Bottou. Online learning and stochastic approximations. *Online learning in neural networks*, 17(9), 1998.

[Bot12] Léon Bottou. Stochastic gradient descent tricks. *Lecture Notes in Computer Science Neural Networks: Tricks of the Trade*, page 421–436, 2012.

[Bru16] Kenneth Brunix. *Improved modeling of unit commitment decisions under uncertainty.* PhD thesis, KULeuven, 2016.

[BSdG+22] Stian Backe, Christian Skar, Pedro Crespo del Granado, Ozgu Turgut, and Asgeir Tomasgard. Empire: An open-source model based on multi-horizon programming for energy transition analyses. *SoftwareX*, 17:100877, 2022.

[BTGN09] A. Ben-Tal, Laurent El Ghaoui, and Arkadi Semenovich Nemirovski. *Robust optimization.* Princeton University Press, 2009.

[CAI13] CAISO. Business practice manual for market operations, version 31, 2013.

[CC59] A. Charnes and W. W. Cooper. Chance-constrained programming. *Management Science*, 6(1), 1959.

[CC63] A. Charnes and W. W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11(1), 1963.

[CGG13] Y. Chen, P. Gribik, and J. Gardner. Incorporating post zonal reserve deployment transmission constraints into energy and ancillary service co-optimization. *IEEE Transactions on Power Systems*, 29(2), 2013.

[Com] European Commission. Energy and the green deal. URL: https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal/energy-and-green-deal_en.

[Com17a] European Commission. Commission regulation (EU) 2017/1485 of 2 August 2017 establishing a guideline on electricity transmission system operation, 2017.

[Com17b] European Commission. Commission regulation (EU) 2017/2195 of 23 November 2017 establishing a guideline on electricity balancing, 2017.

[Com19] European Commission. Communication from the Commission to the European parliament, the European council, the council, the European economic and social committee and the committee of the regions, 2019. URL: https://eur-lex.europa.eu/resource.html?uri=cellar:b828d165-1c22-11ea-8c1f-01aa75ed71a1.0002.02/DOC_1&format=PDF.

[CP22] J. Cho and A. Papavasiliou. A branch-and-cut algorithm for chance-constrained multi-area reserve sizing. Technical report, UCLouvain CORE Discussion Paper, 2022. URL: https://dial.uclouvain.be/pr/boreal/object/boreal:260463.

[Dan55] George B. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3-4), 1955.

[Dan67] John M. Danskin. The theory of max-min and its application to weapons allocation problems. *Ökonometrie und Unternehmensforschung / Econometrics and Operations Research*, 1967.

[DMPFG10] V De Matos, Andrew B Philpott, Erlon C Finardi, and Ziming Guan. Solving long-term hydro-thermal scheduling problems. Technical report, Technical report, Electric Power Optimization Centre, University of Auckland, 2010.

[DPMD19] Oscar Dowson, Andy Philpott, Andrew Mason, and Anthony Downward. A multi-stage stochastic optimization model of a pastoral dairy farm. *European Journal of Operational Research*, 274(3):1077–1089, 2019.

[DVSD+19] K. De Vos, N. Stevens, O. Devolder, A. Papavasiliou, B. Hebb, and J. Matthys-Donnadieu. Dynamic dimensioning approach for operating reserves: Proof of concept in Belgium. *Energy policy*, 124, 2019.

[Ene22] Energinet. Explanatory document for the amended Nordic LFC block methodology for FRR dimensioning in accordance with Article 157(1) of the Commission Regulation (EU) 2017/1485 of 2 August 2017 establishing a guideline on electricity transmission system operation, 2022. URL: `https://consultations.entsoe.eu/system-operations/nordic-tsos-proposal-on-frr-dimensioning/supporting_documents/20210824%206_190513%20Explanatory%20document%20for%20FRR%20dimensioning%20rules%20final%20%20TSO%20approved.pdf`.

[FBP10] BC Flach, LA Barroso, and MVF Pereira. Long-term optimal allocation of hydro generation for a price-maker company in a competitive market: latest developments and a stochastic dual dynamic programming approach. *IET generation, transmission & distribution*, 4(2):299–314, 2010.

[FER13] FERC. Make-whole payments and related bidding strategies, 2013.

[GAC14] Esteban Gil, Ignacio Aravena, and Raúl Cárdenas. Generation capacity expansion planning under hydro uncertainty using stochastic mixed integer programming and scenario reduction. *IEEE Transactions on Power Systems*, 30(4):1838–1847, 2014.

[GCT20] Ye Guo, Cong Chen, and Lang Tong. Pricing multi-interval dispatch under uncertainty part ii: Generalization and performance. *IEEE Transactions on Power Systems*, 36(5), 2020.

[GCT21] Ye Guo, Cong Chen, and Lang Tong. Pricing multi-interval dispatch under uncertainty part i: Dispatch-following incentives. *IEEE Transactions on Power Systems*, 36(5), 2021.

[GG61] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6), 1961.

[GP01] O. Günlük and Y. Pochet. Mixing mixed-integer inequalities. *Math. Program*, 90(3), 2001.

[HGKK15] J. De Haan, M. Gibescu, D. Klaar, and W. Kling. Sizing and allocation of frequency restoration reserves for lfc block cooperation. *Electric Power Systems Research*, 128, 2015.

[Hog16] W.W. Hogan. Electricity market design: Optimization and market equilibrium. Presentation, 2016. URL:

https://scholar.harvard.edu/whogan/files/hogan_ihs_110916.pdf.

[Hog20] W.W. Hogan. Electricity market design: Multi-interval pricing models. Presentation, 2020. URL: https://scholar.harvard.edu/files/whogan/files/hogan_yesenergy_202720r.pdf.

[HSZ+19] Bowen Hua, Dane A. Schiro, Tongxin Zheng, Ross Baldick, and Eugene Litvinov. Pricing in multi-interval real-time markets. *IEEE Transactions on Power Systems*, 34(4):2696–2705, 2019.

[JMMT20] R. J. Jing, M. Moreno-Maza, and D. Talaashrafi. Complexity estimates for Fourier-Motzkin Elimination. *International Workshop on Computer Algebra in Scientific Computing*, 2020. Springer.

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4), 1984.

[KLT16] Dheepak Krishnamurthy, Wanning Li, and Leigh Tesfatsion. An 8-zone test system based on iso new england data: Development and application. *IEEE Transactions on Power Systems*, 31(1):234–246, 2016.

[KSHdM02] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2), 2002.

[KW52] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3), 1952.

[LA08] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2), 2008.

[LAN10] J. Luedtke, S. Ahmed, and G. L. Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Math. Program*, 122(2), 2010.

[Lap10] G Laporte. A concise guide to the traveling salesman problem. *Journal of the Operational Research Society*, 61(1), 2010.

[LHZ13] J. D. Lyon, K. W. Hedman, and M. Zhang. Reserve requirements to efficiently manage intra-zonal congestion. *IEEE Transactions on Power Systems*, 29(1), 2013.

[Lue14] J. Luedtke. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Math. Program*, 146(1), 2014.

[LW20] Nils Löhndorf and David Wozabal. Gas storage valuation in incomplete markets. *European Journal of Operational Research*, 2020.

[LWM13] Nils Löhndorf, David Wozabal, and Stefan Minner. Optimizing trading decisions for hydro storage systems using approximate dual dynamic programming. *Operations Research*, 61(4):810–823, 2013.

[Mic15] Joel Mickey. Multi-interval real-time market overview. Board of Directors Meeting, ERCOT Public, 2015.

[Mot36] T. S. Motzkin. *Beitrage zur theorie der linearen Ungleichungen.* PhD thesis, University of Base, 1936.

[Mur83] Katta G. Murty. *Linear Programming.* John Wiley, 1983.

[MZPP14] Juan M. Morales, Marco Zugno, Salvador Pineda, and Pierre Pinson. Electricity market clearing with improved scheduling of stochastic production. *European Journal of Operational Research*, 235(3):765–774, 2014.

[Nat] United Nations. For a livable climate: Net-zero commitments must be backed by credible action. URL: `https://www.un.org/en/climatechange/net-zero-coalition`.

[NJLS09] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[OEC14] OECD/IEA. The power of transformation - wind, sun and the economics of flexible power systems, 2014. URL: `https://www.oecd.org/publications/the-power-of-transformation-9789264208032-en.htm`.

[OMN04] U. Aytun Ozturk, Mainak Mazumdar, and Bryan A. Norman. A solution to the stochastic unit commitment problem using chance constrained programming. *IEEE Transactions on Power Systems*, 19(4), 2004.

[oStUSEOotP21] The United States Department of State and the United States Executive Office of the President. The long-term strategy of the United States: Pathways to net-zero greenhouse gas emissions by 2050, 2021. URL:

`https://www.whitehouse.gov/wp-content/uploads/2021/10/US-Long-Term-Strategy.pdf`.

[PBA+21] A. Papavasiliou, A. Bouso, S. Apelfröjd, E. Wik, T. Gueuning, and Y. Langer. Multi-area reserve dimensioning using chance-constrained optimization. *IEEE Transactions on Power Systems*, 2021.

[PBDS20] A. Papavasiliou, M. Bjorndal, G. Doorman, and N. Stevens. Hierarchical balancing in zonal markets. *2020 17th International Conference on the European Energy Market (EEM)*, 2020.

[PBM13] Roberto J Pinto, CarmenL T Borges, and Maria EP Maceira. An efficient parallel algorithm for large scale hydrothermal system operation planning. *IEEE Transactions on Power Systems*, 28(4):4888–4896, 2013.

[PF21] Andy Philpott and Michael Ferris. Dynamic risked equilibrium. *Operations Research*, 2021.

[PFW16] Andy Philpott, Michael Ferris, and Roger Wets. Equilibrium, uncertainty and risk in hydro-thermal electricity systems. *Mathematical Programming*, 157(2):483–513, 2016.

[PG08] Andrew B Philpott and Ziming Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008.

[PJ92] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

[PMCS17] Anthony Papavasiliou, Yuting Mou, Léopold Cambier, and Damien Scieur. Application of stochastic dual dynamic programming to the real-time dispatch of storage under renewable supply uncertainty. *IEEE Transactions on Sustainable Energy*, 9(2):547–558, 2017.

[PO13] Anthony Papavasiliou and Shmuel S. Oren. Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network. *Operations Research*, 61(3):578–592, 2013.

[POR14] Anthony Papavasiliou, Shmuel S Oren, and Barry Rountree. Applying high performance computing to transmission-constrained stochastic unit commitment for renewable energy integration. *IEEE Transactions on Power Systems*, 30(3):1109–1120, 2014.

[PP91] Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1-3), 1991.

[Pre74] Andras Prekopa. Programming under probabilistic constraints with a random technology matrix. *Optimization*, 5(2), 1974.

[Pre03] Andras Prekopa. Probabilistic programming. *Handbooks in operations research and management science*, 10, 2003.

[PS17] Anthony Papavasiliou and Yves Smeers. Remuneration of flexibility using operating reserve demand curves: A case study of Belgium. *The Energy Journal*, 38(01), 2017.

[PZBT20] B. Park, Z. Zhou, A. Botterud, and P. Thimmapuram. Probabilistic zonal reserve requirements for improved energy management and deliverability with wind power uncertainty. *IEEE Transactions on Power Systems*, 35(6), 2020.

[PZP10] Geoffrey Pritchard, Golbon Zakeri, and Andrew Philpott. A single-settlement, energy-only electric power market for unpredictable and intermittent participants. *Operations research*, 58(4):1210–1219, 2010.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. `doi:10.1214/aoms/1177729586`.

[RMKA16] L. Roald, S. Misra, T. Krause, and G. Andersson. Corrective control to handle forecast uncertainty: A chance constrained optimal power flow. *IEEE Transactions on Power Systems*, 32(2), 2016.

[Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 1958.

[RS15] Daniel Ralph and Yves Smeers. Risk trading and endogenous probabilities in investment equilibria. *SIAM Journal on Optimization*, 25(4):2589–2611, 2015.

[RW91] R. Tyrrell Rockafellar and Roger J-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147, 1991.

[Sch17] Dane A. Schiro. Flexibility procurement and reimbursement: A multi-period pricing approach. FERC Technical Conference, 2017. URL: `https://cms.ferc.gov/sites/default/`

files/2020-05/20170623123635-Schiro_FERC2017_Final.pdf.

[Sha12] Alexander Shapiro. Time consistency of dynamic risk measures. *Operations Research Letters*, 40(6):436–439, 2012.

[SHdM00] Alexander Shapiro and Tito Homem-de Mello. On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization*, 11(1), 2000.

[SJM18] C. Singh, P. Jirutitijaroen, and J. Mitra. *Electric power grid reliability evaluation: models and methods.* John & Sons Wiley, 2018.

[SZZL16] Dane A. Schiro, Tongxin Zheng, Feng Zhao, and Eugene Litvinov. Convex hull pricing in electricity markets: Formulation, analysis, and implementation challenges. *IEEE Transactions on Power Systems*, 31(5):4068–4075, 2016.

[TBL96] Samer Takriti, John R Birge, and Erik Long. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, 11(3), 1996.

[TSO19] Nordic TSOs. Nordic system operation agreement (SOA) - annex load-frequency control & reserves (LFCR), 2019. URL: https://eepublicdownloads.entsoe.eu/clean-documents/SOC%20documents/Nordic/Nordic%20SOA_Annex%20LFCR.pdf.

[Van00] François Vanderbeck. Exact algorithm for minimising the number of setups in the one-dimensional cutting stock problem. *Operations Research*, 48(6), 2000.

[VMLA13] M. Vrakopoulou, K. Margellos, J. Lygeros, and G. Andersson. A probabilistic framework for reserve scheduling and N-1 security assessment of systems with high wind power penetration. *IEEE Transactions on Power Systems*, 28(4), 2013.

[VSW69] R. M. Van Slyke and Roger Wets. l-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4), 1969.

[WC21] F. Wang and Y. Chen. Market implications of short-term reserve deliverability enhancement. *IEEE Transactions on Power Systems*, 36(2), 2021.

[WGW11] Qianfan Wang, Yongpei Guan, and Jianhui Wang. A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output. *IEEE Transactions on Power Systems*, 27(1), 2011.

[WH15] Beibei Wang and Benjamin F. Hobbs. Real-time markets for flexiramp: A stochastic unit commitment-based analysis. *IEEE Transactions on Power Systems*, 31(2):846–860, 2015.

[Wil02] R. Wilson. Architecture of power markets. *Econometrica*, 70(4), 2002.

[WP18] K. Wellenius and S. L. Pope. Challenges of estimating opportunity costs of energy-limited resources and implications for efficient local market power mitigation, 2018. URL: `https://www.caiso.com/Documents/FTIConsultingWhitePaper-Challenges-EstimatingOpportunityCosts-Energy-LimitedResources-etc.pdf`.

[YCWZ09] H. Yu, C. Y. Chung, K. P. Wong, and J. H. Zhang. A chance constrained transmission network expansion planning method with consideration of load and wind farm uncertainties. *IEEE Transactions on Power Systems*, 24(3), 2009.

[ZGYT21] Haifei Zhang, Hongwei Ge, Jinlong Yang, and Yubing Tong. Review of vehicle routing problems: Models, classification and solving algorithms. *Archives of Computational Methods in Engineering*, 29(1), 2021.

[ZKAB17] Victor M. Zavala, Kibaek Kim, Mihai Anitescu, and John Birge. A stochastic electricity market clearing formulation with consistent pricing properties. *Operations research*, 65(3):557–576, 2017.

[ZL08] T. Zheng and E. Litvinov. Contingency-based zonal reserve modeling and pricing in a co-optimized energy and reserve market. *IEEE transactions on Power Systems*, 23(2), 2008.

[ZL11] Hui Zhang and Pu Li. Chance constrained programming for optimal power flow under uncertainty. *IEEE Transactions on Power Systems*, 26(4), 2011.

[ZZL19] Jinye Zhao, Tongxin Zheng, and Eugene Litvinov. A multi-period market design for markets with intertemporal constraints. *IEEE Transactions on Power Systems*, 2019.