THEORY AND COMPUTATION OF THE MULTISTAGE STOCHASTIC EQUILIBRIA WITH RISK-AVERSE PLAYERS

by

Jiajie Shen

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Industrial and Systems Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2023

Date of final oral examination: 06/07/2023

The dissertation is approved by the following members of the Final Oral Committee:

Michael C. Ferris, Professor, Computer Science
Jeffery T. Linderoth, Professor, Industrial and Systems Engineering
James Luedtke, Professor, Industrial and Systems Engineering
Carla Michini, Assistant Professor, Industrial and Systems Engineering
Thomas F. Rutherford, Professor, Agricultural and Applied Economics

© Copyright by Jiajie Shen 2023
All Rights Reserved

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis primarily focuses on addressing the multistage stochastic equilibrium problem involving risk-averse players. The impetus behind this research stems from the recent surge in interest surrounding such problem domains, given their wide-ranging applications in electricity power system management, natural gas markets, and other scientific, engineering, and economic fields. Despite the growing interest and the imperative to tackle these challenges, there exists a notable dearth of both modeling tools and algorithms that can aid researchers in effectively handling this type of problem. Consequently, to address this intricate issue, our study encompasses several contributions: (i) the introduction of a standardized set of test instances that serve as benchmarks for evaluating algorithms tailored to solve similar problems; (ii) the proposal of three decomposition frameworks specifically designed to tackle this class of problem; and (iii) the development of a novel graph framework capable of representing stochastic problems in a comprehensive manner.

In Chapter 1, a thorough overview of stochastic MOPECs with risk-averse players is presented, aiming to provide a comprehensive understanding of the fundamental mathematical underpinnings within this realm. The chapter delves into an in-depth exploration of key mathematical concepts relevant to this area, including the Nash equilibrium problem, variational inequality problem, stochastic problem incorporating a scenario tree, and the incorporation of a coherent risk measure. By elucidating these concepts, readers gain a nuanced understanding of the mathematical essence underlying stochastic MOPECs with risk-averse players. Furthermore, the chapter elucidates a range of formulations utilized in addressing these complex problems, thereby offering readers a comprehensive view of the diverse approaches employed in tackling this particular class of problems

Chapter 2 encompasses a detailed exposition of three prominent problem instances, namely the economic dispatch example, capacity expansion example, and hydroelectricity example, which have emerged as widely employed scenarios within the field. Each example is accompanied by a meticulous algebraic formulation, rendering a comprehensive mathematical representation of the respective problem instance. Furthermore, this chapter thoroughly investigates three distinct categories of market constraints, which hold paramount significance within the modern economic landscape and find substantial applications across diverse domains. The inclusion of these diverse problem types and the consideration of various market constraints collectively contribute to an expansive coverage of the prevailing stochastic MOPECs with risk-averse players encountered in real-world scenarios, thereby offering comprehensive insights into the realm of these problems.

Chapter 3 is dedicated to exploring the player-based inner structure of the problem,

with a particular emphasis on the stochastic PNEP (price-incentivized Nash Equilibrium Problem) involving risk-averse players. A *conjugate-based reformulation* is employed to ensure the independence of each player's subproblem when the market price $\boldsymbol{p}$ is held constant. This key property has motivated the development of numerous player-based algorithms for solving such problems; however, their applicability to stochastic MOPECs with risk-averse players is limited. In this chapter, two existing methods are initially examined, followed by the introduction of an ADMM (Alternating Direction Method of Multipliers)-based algorithm specifically designed to address the stochastic PNEP with risk-averse players. The advantages of the proposed ADMM-based algorithm over the previously discussed methods are demonstrated. Moreover, numerical results are presented in Chapter 3 to illustrate various properties of the ADMM-based algorithm and to investigate the influence of algorithmic parameters on its performance.

In Chapter 4, a new primal-MOPEC-dual-risk algorithm was introduced, presenting an *equilibrium reformulation* as a novel approach to tackle the stochastic MOPEC (Multiobjective Programming with Equilibrium Constraints) with risk-averse players. This approach leveraged a decomposition strategy motivated by the inherent nonlinearity introduced by the inclusion of the probability vector $\boldsymbol{\mu}$ in the multistage problem formulation. Consequently, when addressed by the PATH solver, the problem manifests as a highly nonlinear complementarity problem. However, by fixing the probability vector $\boldsymbol{\mu}$, the problem undergoes simplification and, in the case of quadratic player objective functions, can even be formulated as a linear complementarity problem. The resulting subproblem can then be efficiently solved through a series of linear programming problems. In contrast to the ADMM-based algorithm discussed in Chapter 3, the proposed primal-MOPEC-dual-risk algorithm offers enhanced versatility in addressing general stochastic MOPECs with risk-averse players, particularly in scenarios where players exhibit diverse Nash behavior during their interactions. The efficacy of the algorithm is validated through its application to all three examples, encompassing all three market constraints, demonstrating its superiority over the classical PATH solver in terms of efficiency and performance.

Chapter 5 provides stage-based decomposition approach to deal with primal-MOPEC subproblem defined in Chapter 4, we introduced a primal-MOPEC-dual-risk decomposition algorithm. When dealing with scenario trees of substantial size, a notable challenge arises due to the resultant increase in dimensions within the primal-MOPEC subproblem. To address this challenge, a stage-based decomposition method is devised, allowing for the partitioning of the comprehensive problem into smaller subproblems that are indexed according to time stages. This partitioning facilitates enhanced computational efficiency and enables the successful resolution of large-scale stochastic MOPECs with risk-averse

players.

In Chapter 6, the formulation of the multistage stochastic MOPEC with risk-averse players is introduced, providing a comprehensive mathematical framework for analysis. Additionally, a novel two-level graph representation is presented as a means to represent the general stochastic problem. This representation offers a structured visualization that aids in understanding the complex interactions and dependencies within the problem. By employing this two-level graph representation, the intricacies of the general stochastic problem can be effectively communicated and analyzed.

Chapter 7 presents the primary conclusions drawn from this thesis.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor Michael C. Ferris for serving as an exceptional advisor throughout my journey towards obtaining my PhD. The path to a doctoral degree can be arduous, often leading individuals astray. However, my advisor has consistently provided invaluable assistance in advancing my knowledge, skills, and research vision, thereby helping me overcome numerous challenges. While fostering my independence, they have offered invaluable ideas, insights, and guidance that have proven instrumental in my progress. I deeply appreciate their training, patience, and encouragement, as they have made my graduate studies both productive and enjoyable. Without their unwavering support, my dissertation would not have come to fruition.

I am also immensely thankful to Professor Jeffery T. Linderoth, James Luedtke, Carla Michini and Thomas Rutherford for generously dedicating their time and effort to serve on my doctoral committee. Their invaluable contributions, suggestions, and encouragement in various aspects have been instrumental in shaping my research over the years.

Furthermore, I would like to extend my heartfelt appreciation to my collaborator, Dr. Olivier Huber. Our collaboration has not only fostered insightful discussions but has also facilitated the development of my skills in numerous areas, including posing thought-provoking questions, delivering presentations, and implementing computational methodologies.

To my dear friends at the Wisconsin Institutes for Discovery and Optimization groups, I am grateful for their unwavering support and companionship throughout the years. Although there are too many names to mention individually, their presence has greatly enriched my academic journey.

Lastly, I would like to express my deepest gratitude to my parents for their continuous encouragement and support. Their passion and enthusiasm have been a constant source of inspiration and energy throughout my life. I would also like to acknowledge my girlfriend, Dr. Yaqing Wu, for her immeasurable love and support. Her presence has brought immense joy to my graduate school experience, and I am incredibly grateful for her endless care and support.

1 INTRODUCTION

Nash equilibrium, and game theory, in general, has gained significant importance in a wide range of fields, including economics [62], social science [57], algorithmic computer science [20, 18], engineering sciences [37], and others.

The origin of the concept of equilibrium can be traced back to Cournot's research [19] on an oligopolistic economy, followed by von Neumann's pioneering work on zero-sum two-person games [69, 86]. Let $\mathcal{A}$ denote the set of the players. For each player $a \in \mathcal{A}$, let $x_a \in \mathbb{R}^{d_a}$ represent the strategy of player a, where d_a is a positive integer. The vector consisting of all these strategies is denoted by $\boldsymbol{x} \coloneqq (x_a)_{a\in\mathcal{A}} \in \mathbb{R}^d$, where $d \coloneqq \sum_{a\in\mathcal{A}} d_a$. All other players' strategies except the strategy of player a is denoted by $x_{-a} \coloneqq (x_{a'})_{a'\in\mathcal{A},a'\neq a} \in \mathbb{R}^{d_{-a}}$, where $d_{-a} \coloneqq d - d_a$. Each player a seeks to optimize a function f_a dependent on x_a and x_{-a}, and constrained by a set X_a. The classical Nash equilibrium problem

$$\begin{aligned} \min_{x_a} \quad & f_a(x_a; x_{-a}) \\ \text{s.t.} \quad & x_a \in X_a \end{aligned} \tag{1.1}$$

was formally introduced by Nash [68, 67]. In this problem, the aim of player a is to select a strategy $x_a \in X_a$ that minimizes the objective function $f_a(x_a; x_{-a})$, where $f : \mathbb{R}^{d_a+d_{-a}} \to \mathbb{R}$, given that the other players' strategies x_{-a} remain fixed. The generalized Nash equilibrium problem (where $X_a = X_a(x_{-a})$, meaning the constraint set X_a is dependent upon the actions of the other players) was formally introduced by Debreu in [22]. In the period between the Arrow and Debreu seminal paper [4] and the 1990s, the primary focus was on studying the existence and uniqueness of equilibria in the economic field. Since the 1990s, there have been many contributions to the calculation of equilibria.

Over the past two decades, the desire to solve complex and large-scale equilibrium problems has been at the heart of research interest for numerous researchers and practitioners. One category of such problems, which is particularly challenging, involves uncertainty and is referred to as stochastic equilibrium problems. In fact, although substantial work has been done to investigate deterministic equilibrium problems, stochastic equilibrium problems are still challenging to researchers, and there lacks a practical, efficient and robust computational framework to deal with them. The main goal of this thesis is to develop practical methods to solve this kind of problem.

The stochastic equilibrium problem is a vital tool in strategic decision-making, market power analysis, addressing inefficiencies, contract design, and environmental concerns. It also aids in understanding interactions between sectors like agriculture, industry, and

transportation. Stochastic equilibrium models provide insights into complex systems, enabling informed decisions and effective policy design.

In strategic decision-making, these models assess outcomes under uncertainty. By incorporating stochastic elements, decision-makers can identify optimal strategies that balance risk and reward. They evaluate scenarios, enabling the assessment of strategy robustness. Stochastic equilibrium models are instrumental in studying market power [53, 21, 51, 8]. They capture uncertain behavior and identify conditions for its emergence. Researchers analyze market structures, dynamics, and strategic interactions to design regulatory interventions. Addressing inefficiencies benefits from stochastic equilibrium models. These models quantify costs and benefits, facilitating the design of efficient interventions. By considering uncertainty, researchers assess intervention effectiveness and identify market failures. Contract design benefits from stochastic equilibrium models [72, 44]. They model stochastic agent behavior, evaluate contract structures, and analyze risk-sharing and efficiency. Such models allow for robust contracts that provide the right incentives. Environmental concerns [82, 65], like carbon pricing and climate change, benefit from stochastic equilibrium models. They integrate uncertainties in modeling carbon pricing mechanisms, simulating market behavior, and evaluating policy consequences. Stochastic equilibrium models enable the analysis of sectoral interactions. They help understand ripple effects and trade-offs between sectors. By considering interconnectedness, policymakers can design integrated policies for sustainable outcomes.

In conclusion, stochastic equilibrium models are essential in strategic decision-making, market power analysis, addressing inefficiencies, contract design, and environmental concerns. They provide valuable insights and facilitate evidence-based decision-making and policy formulation.

1.1 Previous work

1.1.1 Deterministic equilibrium problems

Deterministic Nash equilibrium problems and complementarity problems have been applied extensively over the last two decades. An overivew of the application areas of the Nash equilibrium problem and complementarity problem is given in [39]. Much research has been devoted to utilizing deterministic Nash equilibrium problems and complementarity problems to model market problems in a natural gas system [45, 43, 47, 28, 30, 31, 1], hydrothermal electricity markets [73, 84, 75, 64, 13, 6, 7, 14, 59], and transportation problems [3, 79, 87, 5, 88, 61]. A detailed and comprehensive complementarity model for computing

market equilibrium values in the European natural gas system is presented in [30], which includes 52 countries that produce, consume, or ship gas to Europe. A new multiseasonal, multiyear, natural gas market equilibrium model is presented in [45] based on the concept of a competitive equilibrium involving the market participants: producers, storage reservoir operators, peak gas operators, pipeline operators,marketers, and consumers. These papers in particular are motivational for our example problems.

1.1.2 Algorithms and tools to solve deterministic equilibrium problems

Because of its wide application, there have been many contributions to the calculation of equilibria in the deterministic setting. These works include natural Jacobi- or Gauss-Seidel-type decomposition methods [55, 36], Variational Inequality (VI)-type methods [25, 35, 66], Nikaido-Isoda (NI)-function-type [50, 54, 85, 27], penalty methods [70, 34], Oridinary Differential Equation (ODE)-based methods [15, 42] and local Newton methods [32].

Among these methods, one of the most important is utilizing the property that these kinds of problems can be transformed into an equivalent deterministic variational inequality or the mixed complementarity problem format. They can be solved by algorithms that are designed to solve complementarity problems or variational inequality problems, such as the PATH solver [25], MILES [78], NE/SQP [71], SMOOTH [17], QPCOMP [10], SEMISMOOTH [9], interior point methods [16], semismooth least squares method [58], KKT based method [33], derivative-free method [48] and Lemke's method [80]. This kind of method is often used by researcher because it is effective for a lot of applications mentioned above even though the underlying problems do not necessarily satisfy theoretical requisites (such as the monotonicity property of the variational inequality or complementarity problem).

1.1.3 Stochastic equilibrium problems

The study of equilibrium problems with uncertainty has received significant attention in recent years [53, 21, 51, 8]. Research has been conducted to analyze its applications in the natural gas market, with works such as [53] that used a stochastic Nash-Cournot modeling technique for the European gas market, [21] that presented a stochastic Stackelberg game for the European gas market, [51] that used a simulation and sample-path approach to study the European gas market, and [8] that used a gradient-based quasi-Monte Carlo simulation technique to analyze the stochastic Mixed Complementarity Problem or Variational Inequality. The study of stochastic equilibrium problems has also been extended to the

electricity power system, as demonstrated in works such as [72, 44, 82, 65]. For instance, [89] examined the modeling and computation of a two-settlement oligopolistic equilibrium in a congested electricity network.

1.1.4 Algorithms to solve stochastic equilibrium problems

Although uncertainty plays a critical role in modeling of a real life system, the curse of dimension is an unavoidable issue in this area, especially in the multistage setting. Much effort has been extended to design efficient algorithms to solve this type of problem. In [51] a sample path approximation method was developed to solve the stochastic equilibrium problem. Scenario reduction is used in [46], which is a common method in multistage stochastic programming programming, to reduce the computational time of large-scale stochastic equilibrium problems. A heuristic method called rolling horizon was developed in [24] to make this kind of problem tractable. Besides, a natural way to deal with such kind of problem is to relax the coupling constraints of the extensive form of the problem so that it can be decomposed into smaller subproblems. The Bender's approach was developed in [44, 29, 83] to decompose this problem into a first stage master problem and several second stage recourse problems. In [49], the linking constraints were relaxed between different stages and the ADMM method was utilized as the decomposition method. In [76], the progressive hedging was utilized to decompose the stochastic varitional inequality by scenarios and successfully proved the convergence under the assumption of the monotonicity. Furthermore, they even proved the convergence rate is linear if the operator is affine and the feasible region is polyhedral.

Decomposition by players is a commonly used method to solve equilibrium problems. Methods using decomposition by agents mainly lie in two classes: (i) best response schemes (ii) gradient-based schemes. Best-response schemes are usually faster than gradient-based schemes but not as stable as the latter. However, even for the gradient-based schemes, most works just focused on the equilibrium problems with the assumption that their corresponding complementarity problems must be strongly monotone. In recent years, much work also applied this decomposition method to solve stochastic equilibrium problems. A survey about the analysis and basic algorithms to solve the stochastic variational inequality before 2013 is given in [81]. The convergence of the best-responses schemes remains an open problem under stochastic regimes. For the gradient-based schemes, they also require the expectation-valued complementarity mappings to be monotone when solving the problems with risk-neutral agents.

1.2 Background Material

1.2.1 Quasi Variational Inequality, Variational Inequality and Mixed Complementarity Problem

For a given continuous function $F : \mathbb{R}^d \to \mathbb{R}^d$ and a point-to-set mapping $K : \mathbb{R}^d \to \mathbb{R}^d$, where $K(x)$ is a closed convex set for each $x \in R^d, x^* \in K(x^*)$ is a solution to the Quasi Variational Inequality $QVI(K, F)$ if

$$\langle F(x^*), x - x^* \rangle \geq 0, \quad \forall x \in K(x^*) \tag{QVI}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product.

Alternatively, $QVI(K, F)$ has another representation based on the concept of a normal cone. Let $S \subset \mathbb{R}^d$ be a closed, convex set. The normal cone of S is the set-valued mapping $\mathcal{N}_S : \mathbb{R}^d \to 2^{\mathbb{R}^d}$, given by

$$\mathcal{N}_S(x) = \begin{cases} \{g \in \mathbb{R}^n | g^T(z - x) \leq 0, \forall z \in S\} & \text{if } x \in S \\ \emptyset & \text{if } x \notin S \end{cases} \tag{1.2}$$

It is clear to see that $x^* \in K(x^*)$ is a solution to the $QVI(K, F)$ if and only if $-F(x^*)$ is an element of $\mathcal{N}_{K(x^*)}(x^*)$, which is equivalent to

$$0 \in F(x^*) + \mathcal{N}_{K(x^*)}(x^*) \tag{1.3}$$

Based on the definition of the QVI, if $K(x)$ is restricted to be a fixed closed convex set K for every x, then $x^* \in K$ is a solution to the Variational Inequality $VI(K, F)$ if

$$\langle F(x^*), x - x^* \rangle \geq 0, \quad \forall x \in K \tag{VI}$$

As above, it follows that, $x^* \in K$ is a solution of $VI(K, F)$ if and only if

$$0 \in F(x^*) + \mathcal{N}_K(x^*) \tag{1.4}$$

If the feasible convex set K in $VI(K, F)$ is restricted to be a box $B = \{x \in \mathbb{R}^d | l_i \leq x_i \leq u_i, \quad \text{for} \quad i = 1, \ldots, d\}$ with $l_i \leq u_i$ and $l_i \in \mathbb{R} \cup \{-\infty\}$ and $u_i \in \mathbb{R} \cup \{+\infty\}$, then we call this special variational inequality problem $VI(B, F)$ a mixed complementary problem $MCP(B, F)$. It follows from elementary calculations that $x^* \in B$ is a solution to this MCP

if and only if the following conditions hold for each $i = 1, \ldots, d$:

$$\begin{aligned} x_i^* &= l_i, \quad \text{if } F_i(x^*) \geq 0 \\ l_i \leq x_i^* &\leq u_i, \quad \text{if } F_i(x^*) = 0 \\ x_i^* &= u_i, \quad \text{if } F_i(x^*) \leq 0 \end{aligned} \tag{MCP}$$

In shorthand notation, the above MCP problem can also be written as $F(x^*) \perp x^* \in [l, u]$. Note that for each index i, a solution x^* identifies which of the constraints $x_i = l_i, x_i = u_i$ or $F_i(x) = 0$ is active.

Merit function of Mixed compementarity problem

Throughout this thesis, we choose the the classical Fischer Burmeister (FB) merit function $\Psi(x)$ defined in [41] as the merit function, which is a quantitative way to measure the quality of the point we have. In fact, a point x is a solution of the Mixed Complementarity Problem if and only if $x \in \arg\min_{z \in [l,u]} \Psi(z)$ and $\Psi(x) = 0$. Here the merit function

$$\Psi(x) = \frac{1}{2}\Phi(x)^T\Phi(x) = \frac{1}{2}\|\Phi(x)\|^2 \tag{1.5}$$

is continuously differentiable. The operator $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ is defined componentwise as follows:

$$\Phi_i(x) := \begin{cases} \phi(x_i - l_i, F_i(x)) & \text{if } \ i \in I_l, \\ -\phi(u_i - x_i, -F_i(x)) & \text{if } \ i \in I_u, \\ \phi(x_i - l_i, \phi(u_i - x_i, -F_i(x))) & \text{if } \ i \in I_{lu}, \\ -F_i(x) & \text{if } \ i \in I_f. \end{cases} \tag{1.6}$$

where we define $\phi : \mathbb{R}^2 \to \mathbb{R}$ by

$$\phi(a, b) := \sqrt{a^2 + b^2} - a - b \tag{1.7}$$

and I_l, I_u, I_{lu}, I_f is a partition of the set $\{1, \ldots, n\}$, where

$$I_l := \{1 \leq i \leq n | -\infty < l_i < u_i = +\infty\}, \tag{1.8}$$

$$I_u := \{1 \leq i \leq n | -\infty = l_i < u_i < +\infty\}, \tag{1.9}$$

$$I_{lu} := \{1 \leq i \leq n | -\infty < l_i < u_i < +\infty\}, \tag{1.10}$$

$$I_f := \{1 \leq i \leq n | -\infty = l_i < u_i = +\infty\}. \tag{1.11}$$

In the implementation of the algorithm, the stopping criteria is $\Psi(x) < \epsilon$, where ϵ is an arbitrary parameter chosen at the start.

1.2.2 Equilibrium problems: MOPECs, GNEPs, NEPs and PNEPs

The objective of multiple optimization problems with equilibrium constraints (MOPECs) with mutiple players and equilibrium constraints is to find a profile of strategies and market prices that satisfy the following conditions: (1) each player's strategy is an optimal response to the other players' strategies and market prices, and (2) the market prices are the reasonable responses from the markets in response to the players' strategies. It is important to note that the strategy set of each player may depend on the strategies of other players and the market prices. Recall the basic definitions of $\mathcal{A}, x_a, x_{-a}, \boldsymbol{x}$ from Section 1.1. The market price is denoted by $p \in \mathbb{R}^\alpha$, where α is a positive integer.

In the MOPEC, each player $a \in \mathcal{A}$, taking the other players' strategies x_{-a} and market price p at their given values, solves the following minimization problem:

$$\begin{aligned} P_a(x_{-a}, p): \quad & \min_{x_a} \quad f_a(x_a; x_{-a}, p) \\ & \text{s.t.} \quad x_a \in X_a(x_{-a}, p) \end{aligned} \tag{1.12}$$

where $f_a : \mathbb{R}^{d+\alpha} \to \mathbb{R}$ is the given cost function of player a. $X_a : \mathbb{R}^{d-a+\alpha} \to \mathbb{R}^{d_a}$ is a given point-to-set mapping. For each fixed x_{-a} and p, $X_a(x_{-a}, p)$ is a subset of $\mathbb{R}^{d_a}$, which is the feasible strategy set of player a with fixed other players' strategies given by x_{-a} and market price given by p. The market price p, taking all players' strategies $\boldsymbol{x}$ at given values, satisfies the equilibrium constraint:

$$0 \in F(p; \boldsymbol{x}) + \mathcal{N}_{K(\boldsymbol{x})}(p) \tag{1.13}$$

where $F : \mathbb{R}^{\alpha+d} \to \mathbb{R}^\alpha$ is a function that embodies the equilibrium condition and can be influenced by the price p and the strategies of all players $\boldsymbol{x}$, typically representing the supply-demand relationship in economics. $K : \mathbb{R}^d \to \mathbb{R}^\alpha$ is a point-to-set mapping and $K(\boldsymbol{x})$ is a set in $\mathbb{R}^\alpha$ given the fixed strategy vector $\boldsymbol{x}$. $\mathcal{N}_{K(\boldsymbol{x})} : \mathbb{R}^\alpha \to \mathbb{R}^\alpha$ is the point-to-set mapping to represent the normal cone of set $K(x)$. Let $Q^{VI}(\boldsymbol{x}) \coloneqq \{p \in K(\boldsymbol{x}) \mid 0 \in F(p; \boldsymbol{x}) + \mathcal{N}_{K(\boldsymbol{x})}(p)\}$ denote the set of market prices p that safisfy the equilibrium contraint (1.13).

A vector $(\boldsymbol{x}, p) = \big((x_a)_{a\in\mathcal{A}}, p\big)$ is said to be feasible to MOPEC if $x_a \in X_a(x_{-a}, p)$ for each $a \in \mathcal{A}$ and $p \in K(\boldsymbol{x})$. The goal of MOPEC is to find a vector $(\boldsymbol{x}^*, p^*) = \big((x_a^*)_{a\in\mathcal{A}}, p^*\big) \in \mathbb{R}^{d+\alpha}$ such that

$$\begin{aligned} & x_a^* \quad \text{is an optimal solution of} \quad P_a(x_{-a}^*, p^*) \qquad \text{for each } a \in \mathcal{A} \\ & p^* \quad \text{belongs to set} \quad Q^{VI}(\boldsymbol{x}^*) \end{aligned} \tag{1.14}$$

A vector $(\boldsymbol{x}^*, p^*) = \big((x_a^*)_{a\in\mathcal{A}}, p^*\big)$ satisfying (1.14) is called an equilibrium point of MOPEC.

If there is no market price p and corresponding market constraint involved in the problem, each player $a \in \mathcal{A}$ will solve the minimization problem with the fixed exogenous

parameter x_{-a}:

$$P_a^{\mathcal{G}}(x_{-a}): \quad \min_{x_a} \quad f_a(x_a; x_{-a}) \qquad \text{s.t.} \quad x_a \in X_a(x_{-a}) \tag{1.15}$$

In this case, we call the above problem a generalized Nash equilibrium problem (GNEP). A vector $\boldsymbol{x}^* = (x_a^*)_{a\in\mathcal{A}}$ satisfying the conditions

$$x_a^* \quad \text{is an optimal solution of} \quad P_a^{\mathcal{G}}(x_{-a}^*) \quad \text{for all } a \in \mathcal{A}. \tag{1.16}$$

is called an equilibrium point of GNEP.

Furthermore, if in a GNEP, the strategy set of each player is not impacted by the strategies of other players, meaning $X_a(x_{-a}) = X_a$ is a fixed set, it is called a Pure Nash equilibrium problem (NEP). The solution to this special type of GNEP is known as an equilibrium point of NEP.

If there are no other players' strategies x_{-a} involved in each player's optimization problem in a MOPEC, we will have each player $a \in \mathcal{A}$ solves a minimization problem parameterized by the fixed market price p:

$$P_a^{\mathcal{P}}(p): \quad \min_{x_a} \quad f_a(x_a; p) \qquad \text{s.t.} \quad x_a \in X_a(p) \tag{1.17}$$

The problem with the above minimization problems $P_a^{\mathcal{P}}(p)$ for each player a and the market constraint (1.13) is called the price-incentivized Nash equilibrium problem (PNEP). A vector $(\boldsymbol{x}^*, p^*) = \left((x_a^*)_{a\in\mathcal{A}}, p^*\right)$ satisfying the conditions

$$\begin{aligned} &x_a^* \quad \text{is an optimal solution of} \quad P_a^{\mathcal{P}}(p^*) \qquad \text{for all } a \in \mathcal{A}. \\ &p^* \quad \text{belongs to set} \quad Q^{VI}(\boldsymbol{x}^*) \end{aligned} \tag{1.18}$$

is called an equilibrium point of PNEP.

Throughout this thesis, we make the following assumptions regarding the smoothness and convexity of functions involved in the MOPEC.

Assumption 1.1. *Suppose the vector (x, p) is a feasible vector to MOPEC. Assume the set $K(\boldsymbol{x})$ is a closed, convex set for each fixed $\boldsymbol{x}$. For $a \in \mathcal{A}$, assume the feasible strategy set $X_a(x_{-a}, p)$ is a closed, convex set in $\mathbb{R}^{d_a}$ for each fixed x_{-a} and p, and the function f is a continuously differentiable function with $f_a(\cdot; x_{-a}, p)$ is differentiable and convex for each fixed x_{-a} and p. F is assumed to be a continuous function.*

We also assume the Mangasarian-Fromovitz Constraint Qualification (MFCQ) holds for all nonlinear constraints. This assumption enables the computation of a solution to the equilibrium problems from their associated MCP and vice versa, given that the constraints are explicitly given as equalities and inequalities. The use of MFCQ also allows for the representation of all normal cones using the algebraic information derived from the constraints.

1.2.3 Equivalence between equilibrium problems and their variational forms

The results shown in this section are simple extensions of existing results in [52]. We first show the equivalence between the equilibrium problems and their associated QVIs.

Proposition 1.2. *For each player $a \in \mathcal{A}$, if $f_a(\cdot; x_{-a}, p)$ is a convex function given (x_{-a}, p), and $X_a(x_{-a}, p)$ is a closed convex set for each (x_{-a}, p), f is a continuously differentiable function, then $(\boldsymbol{x}^*, p^*)$ is a solution to the MOPEC if and only if it is a solution to the $QVI(\mathcal{K}, G)$ where*

$$\begin{aligned} \mathcal{K}(\boldsymbol{x}, p) &= \prod_{a \in \mathcal{A}} X_a(x_{-a}, p) \times K(\boldsymbol{x}) \\ G(\boldsymbol{x}, p) &= (\nabla_{x_1} f_1(x_1; x_{-1}, p)^T, \ldots, \nabla_{x_{|\mathcal{A}|}} f_{|\mathcal{A}|}(x_{|\mathcal{A}|}; x_{-|\mathcal{A}|}, p)^T, F(p; \boldsymbol{x})^T)^T \end{aligned} \tag{1.19}$$

Proof. ($\Rightarrow$) Let $(\boldsymbol{x}^*, p^*)$ be a solution to the MOPEC. For each player $a \in \mathcal{A}$, since each player's objective function $f_a(\cdot; x_{-a}, p)$ is a convex function for any (x_{-a}, p), then the first order conditions are necessary and sufficient. For that reason we will have

$$\langle \nabla_{x_a} f_a(x_a^*; x_{-a}^*, p^*), x_a - x_a^* \rangle \geq 0, \quad \forall x_a \in X_a(x_{-a}^*, p^*) \text{ for } a \in \mathcal{A}$$

Also, since $K(\boldsymbol{x}^*)$ is closed convex set, then from

$$0 \in F(p^*; \boldsymbol{x}^*) + \mathcal{N}_{K(\boldsymbol{x}^*)}(p^*) \tag{1.20}$$

we will have the variational inequality

$$\langle F(p^*; \boldsymbol{x}^*), p - p^* \rangle \geq 0, \quad \forall p \in K(\boldsymbol{x}^*) \tag{1.21}$$

Thus $(\boldsymbol{x}^*, p^*)$ is a solution of the $QVI(\mathcal{K}, G)$.

($\Leftarrow$) If $(\boldsymbol{x}^*, p^*)$ is a solution to the $QVI(\mathcal{K}, G)$, then we will have

$$\begin{aligned}\langle \nabla_{x_a} f_a(x_a^*; x_{-a}^*, p^*), x_a - x_a^* \rangle \geq 0, \quad \forall x_a \in X_a(x_{-a}^*, p^*) \text{ for each } a \in \mathcal{A} \\ \langle F(p^*; \boldsymbol{x}^*), p - p^* \rangle \geq 0, \quad \forall p \in K(\boldsymbol{x}^*)\end{aligned}$$

Since $f_a(\cdot; x_{-a}^*, p^*)$ is convex for (x_{-a}^*, p^*), we have

$$x_a^* \in \arg \min_{x_a \in X_a(x_{-a}^*, p^*)} f_a(x_a; x_{-a}^*, p^*), \text{ for each } a \in \mathcal{A} \tag{1.22}$$

Thus $(\boldsymbol{x}^*, p^*)$ is a solution of the corresponding MOPEC. □

Proposition 1.3. *Suppose* $X_a(x_{-a}, p) = \{x_a \in [l_a, u_a] | h_a(x_a; x_{-a}, p) = 0,\ g_a(x_a; x_{-a}, p) \leq 0\}$ *where* $h_a : \mathbb{R}^{d+\alpha} \to \mathbb{R}^{\chi_a}$ *is an affine function,* $g_a : \mathbb{R}^{d+\alpha} \to \mathbb{R}^{\psi_a}$ *is continuously differentiable and a convex function of* x_a *given* (x_{-a}, p) *and* $l_a \leq u_a, l_a \in \mathbb{R}^{d_a} \cup \{-\infty\}^{d_a}$*, and* $u_a \in \mathbb{R}^{d_a}$*, and* $u_a \in \mathbb{R}^{d_a} \cup \{\infty\}^{d_a}$*. Suppose* $K(\boldsymbol{x}) = \mathbb{R}_+^{\alpha}$*,* $f_a(\cdot; \cdot, \cdot)$ *is continuously differentiable,* $f_a(\cdot; x_{-a}, p)$ *is a convex function, and* $X_a(x_{-a}, p)$ *is a closed convex set for each given* x_{-a} *and price* p*. Then* $(\boldsymbol{x}^*, p^*)$ *is a solution to the MOPEC if and only if* $(\boldsymbol{x}^*, p^*, \boldsymbol{\gamma}^*, \boldsymbol{\eta}^*)$ *is a solution to the* $MCP(B, G)$*, provided the constraint qualification holds at* $(\boldsymbol{x}^*, p^*)$ *with*

$$\begin{aligned}B = \prod_{a \in \mathcal{A}} [l_a, u_a] \times \mathbb{R}_+^{d_p} \times \mathbb{R}^{\chi} \times \mathbb{R}_-^{\psi}, \quad \chi = \sum_{a \in \mathcal{A}} \chi_a, \quad \psi = \sum_{a \in \mathcal{A}} \psi_a, \\ G(\boldsymbol{x}, p, \boldsymbol{\gamma}, \boldsymbol{\eta}) = \Big((\nabla_{x_1} f_1(x_1; x_{-1}, p) - \nabla_{x_1} h_1(x_1; x_{-1}, p)\gamma_1 - \nabla_{x_1} g_1(x_1; x_{-1}, p)\eta_1)^T, \ldots, \\ (\nabla_{x_{|\mathcal{A}|}} f_{|\mathcal{A}|}(x_{|\mathcal{A}|}; x_{-|\mathcal{A}|}, p) - \nabla_{x_{|\mathcal{A}|}} h_{|\mathcal{A}|}(x_{|\mathcal{A}|}; x_{-|\mathcal{A}|}, p)\gamma_{|\mathcal{A}|} - \nabla_{x_{|\mathcal{A}|}} g_{|\mathcal{A}|}(x_{|\mathcal{A}|}; x_{-|\mathcal{A}|}, p)\eta_{|\mathcal{A}|})^T, \\ F(p; \boldsymbol{x})^T, h_1(x_1; x_{-1}, p)^T, \ldots, h_{|\mathcal{A}|}(x_{|\mathcal{A}|}; x_{-|\mathcal{A}|}, p)^T, g_1(x_1; x_{-1}, p)^T, \ldots, g_{|\mathcal{A}|}(x_{|\mathcal{A}|}; x_{-|\mathcal{A}|}, p)^T \Big)^T\end{aligned} \tag{1.23}$$

where $\boldsymbol{\gamma} = (\gamma_a)_{a \in \mathcal{A}}, \boldsymbol{\eta} = (\eta_a)_{a \in \mathcal{A}}$.

Proof. ($\Rightarrow$) Let $(\boldsymbol{x}^*, p^*)$ be a solution to the MOPEC. Using the KKT conditions of each optimization player, the VI, and the constraint qualification at $(\boldsymbol{x}^*, p^*)$, there exists $(\boldsymbol{\gamma}^*, \boldsymbol{\eta}^*)$ such that

$$\begin{aligned}\nabla_{x_a} f_a(x_a^*; x_{-a}^*, p^*) - \nabla_{x_a} h_a(x_a^*; x_{-a}^*, p^*)\gamma_a^* - \nabla_{x_a} g_a(x_a^*; x_{-a}^*, p^*)\eta_a^* \perp l_a \leq x_a^* \leq u_a, \quad \forall a \in \mathcal{A} \\ F(p^*; \boldsymbol{x}^*) \perp 0 \leq p^* \\ h_a(x_a^*; x_{-a}^*, p^*) \perp \gamma_a^* \text{ free} \quad \forall a \in \mathcal{A} \\ g_a(x_a^*; x_{-a}^*, p^*) \perp \eta_a^* \leq 0 \quad \forall a \in \mathcal{A}\end{aligned} \tag{1.24}$$

So $(\boldsymbol{x}^*, p^*, \boldsymbol{\gamma}^*, \boldsymbol{\eta}^*)$ is a solution to the MCP(B, G).

($\Leftarrow$) Let $(\boldsymbol{x}^*, p^*, \boldsymbol{\gamma}^*, \boldsymbol{\eta}^*)$ be a solution to the MCP(F, G), then $(\boldsymbol{x}^*, p^*, \boldsymbol{\gamma}^*, \boldsymbol{\eta}^*)$ satisfies (1.23). When $(\boldsymbol{x}^*, p^*)$ satisfies the constraint qualification, we will have

$$\begin{aligned}\mathcal{N}_{X_a(x^*_{-a},p^*)}(x^*_a) =&\{-\nabla_{x_a} h_a(x^*_a; x^*_{-a}, p^*)\gamma_a - \nabla_{x_a} g_a(x^*_a; x^*_{-a}, p^*)\eta_a| \\ & h_a(x^*_a; x^*_{-a}, p^*) \perp \gamma_a, g_a(x^*_a; x^*_{-a}, p^*) \perp \eta_a \leq 0\} \\ & + \mathcal{N}_{[l_a,u_a]}(x^*_a), \quad \forall a \in \mathcal{A}\end{aligned} \tag{1.25}$$

Then with the convexity condition, we know that $(\boldsymbol{x}^*, p^*)$ is a solution to the MOPEC. □

1.3 Multistage stochastic equilibrium problems based on a scenario tree

1.3.1 The notion of a scenario tree

Before introducing a stochastic Nash equilibrium, we first introduce a structure that is commonly used to represent a discrete-time stochastic process, with a finite set of events in each stage. Such a process can be modeled using a *scenario tree* specified by the pair $(\mathcal{N}, \mathcal{E})$, where $n \in \mathcal{N}$ is the set of scenario nodes and $\varepsilon \in \mathcal{E}$ is the set of edges. By convention we number the root node $n = 1$. A path between two nodes is defined as a sequence of adjacent nodes, where each adjacent pair of nodes can be represented by an edge in the set $\mathcal{E}$. The set of paths from node n to node m is denoted by $\mathcal{P}(n, m)$. If $n = 1$ is the root node, then the notation $\mathcal{P}(1, m)$ can be simplified to $\mathcal{P}(m)$. In this case, $\mathcal{P}(m)$ specifies the set of nodes that form a path from the root node 1 to node m. Let $\bar{\mathcal{P}}(n, m)$ denote the path from node n to node m but excluding the node n. The previous node in the path can be denoted as n_-, the node before that as n_{--}, and so on. The set of children of node $n \in \mathcal{N}$ is denoted by n_+, and its cardinality is denoted by $|n_+|$. The set of leaves of the scenario tree, that is, nodes with no children, is denoted by $\mathcal{L}$. The set of successors of node n is $\mathcal{S}(n) = \{\{n\} \cup n_+ \cup n_{++} \cup \ldots.\}$ where n_{++} is defined in the obvious way. The set of successors of node n excluding node n is $\bar{\mathcal{S}}(n) = \{n_+ \cup n_{++} \cup \ldots.\}$. The set of successors of node n excluding node n and its children nodes n_+ is $\bar{\bar{\mathcal{S}}}(n) = \{n_{++} \cup \ldots.\}$. The depth $\delta(n)$ of node n is the number of nodes on the path to node 1, so $\delta(1) = 1$ and we assume without loss of generality that every leaf node has the same depth, say $\delta_{\mathcal{L}}$. The depth of a node can be interpreted as a time index $t = 1, 2, \ldots, T = \delta_{\mathcal{L}}$. The corresponding time stage set of this tree could be denoted by $\mathcal{T} = \{1, 2, \ldots, T\}$. A pictorial representation of a scenario tree with three-time stages is given in Figure 1.1. The conditional probability of the event represented by node n is denoted ϕ_n and so ϕ_n represents the probability of

moving from n_- to n. Parameter $\pi_n = \prod_{m \in \mathcal{P}(n)} \phi_m$ is the probability of the scenario at node n. For the root node $n = 1$, $\phi_1 = \pi_1 = 1$. The sequence that travels from a root node to a leaf node is called a scenario, i.e., $(1, 2, 4)$. Denote $\mathcal{N}(t) \subseteq \mathcal{N}$ as the set of nodes representing the realizations of the random variable at time stage t. It can be seen that for the example of Figure 1.1 $\mathcal{N}(1) = \{1\}, \mathcal{N}(2) = \{2, 3\}$ and $\mathcal{N}(3) = \{4, 5, 6, 7, 8\}$.

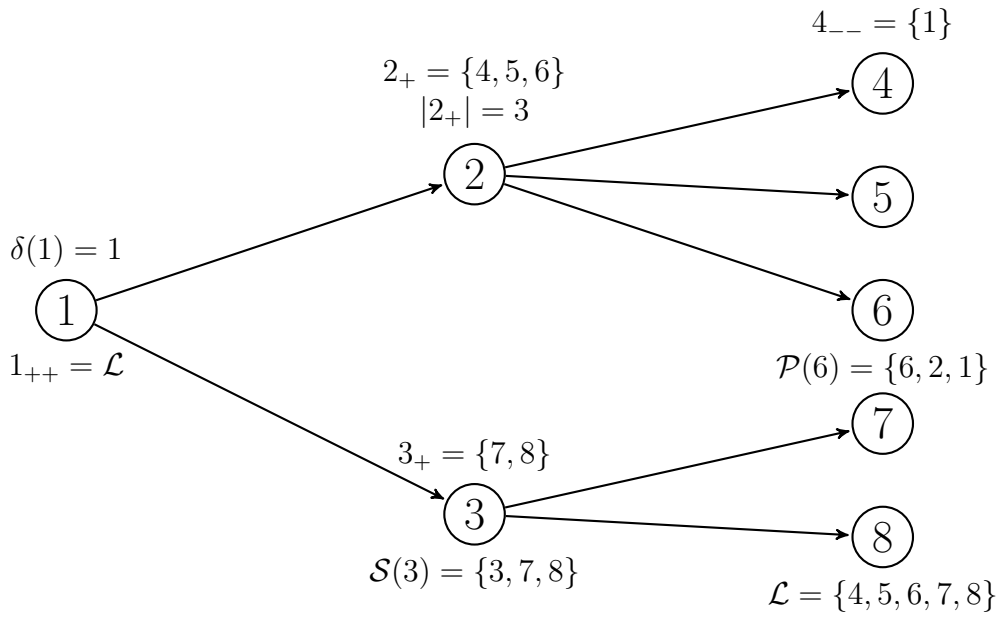


Figure 1.1: A scenario tree with nodes $\mathcal{N} = \{1, 2, \ldots, 8\}$, and $T = 3$

1.3.2 Stochastic MOPEC defined on a scenario tree

Following the introduction of the concept of a scenario tree, a stochastic MOPEC can be constructed based on it. For the stochastic problem, the strategy of each player x_a could be decomposed into the strategy of player on each scenario node $x_{an} \in \mathbb{R}^{d_{an}}$. The entire strategy of player a can then represented as $\boldsymbol{x}_{a\cdot} \coloneqq (x_{an})_{n \in \mathcal{N}} \in \mathbb{R}^{d_a}$, where $d_a \coloneqq \sum_{n \in \mathcal{N}} d_{an}$. The vector consisting of all strategies of players other than player a at scenario node n is denoted by $x_{-an} \coloneqq (x_{a'n})_{a' \in \mathcal{A}, a' \neq a} \in \mathbb{R}^{d_{-an}}$, where $d_{-an} \coloneqq \sum_{a' \in \mathcal{A}, a' \neq a} d_{a'n} = d_n - d_{an}$. The vector including entire strategies of all players excluding player a at all stages is denoted by $\boldsymbol{x}_{-a\cdot} \coloneqq (x_{-an})_{n \in \mathcal{N}}$.

Additionally, there will be an individual, independent market for each scenario node $n \in \mathcal{N}$, associated with a market price $p_n \in \mathbb{R}^{\alpha_n}$ for each market, where α_n is a positive integer. This will only be influenced by its corresponding market and the strategies involved in that market. The price vector involving all time stages is denoted by $\boldsymbol{p} \coloneqq (p_n)_{n \in \mathcal{N}} \in \mathbb{R}^{\alpha}$, where $\alpha = \sum_{n \in \mathcal{N}} \alpha_n$. Due to the independence of the prices at each stage, the feasible

region of the price vector $\boldsymbol{p}$ is $K \coloneqq \prod_{n\in\mathcal{N}} K_n \subseteq \mathbb{R}^{\alpha}$, where $K_n \subseteq \mathbb{R}^{\alpha_n}$ is a closed, convex set and represents the constraints on the price p_n.

For each $n \in \mathcal{N}$, there will be a cost $f_{an}(x_{an}; x_{-an}, p_n)$ of player a, where $f : \mathbb{R}^{d_n+\alpha_n} \to \mathbb{R}$ is a continuously differentiable function and $f(\cdot; x_{-an}, p_n)$ is a convex function given fixed other players' strategies x_{-an} and market price p_n. Based on this, players could use different risk measures to build their costs on the scenario tree. We start with the most commonly used risk measure, which is expectation. Based on the structure of the scenario tree $(\mathcal{N}, \mathcal{E})$, the expected disbenefit of future outcomes at node n of player a is then defined recursively to be

$$\begin{aligned}
&\mathbb{E}_n\Big(\big(f_{am}(x_{am}; x_{-am}, p_m)\big)_{m\in\bar{\mathcal{S}}(n)}\Big) \\
&= \begin{cases} 0, & n \in \mathcal{L} \\ \sum_{m\in n_+} \phi_m \cdot \Big(f_{am}(x_{am}; x_{-am}, p_m) + \mathbb{E}_m\Big(\big(f_{al}(x_{al}; x_{-al}, p_l)\big)_{l\in\bar{\mathcal{S}}(m)}\Big)\Big) & n \in \mathcal{N}\backslash\mathcal{L} \end{cases}
\end{aligned} \tag{1.26}$$

Based on (1.26), we can get

$$\mathbb{E}_n\Big(\big(f_{am}(x_{am}; x_{-am}, p_m)\big)_{m\in\bar{\mathcal{S}}(n)}\Big) = \sum_{m\in\bar{\mathcal{S}}(n)} \Big(\prod_{l\in\bar{\mathcal{P}}(n,m)} \phi_l\Big) \cdot f_{am}(x_{am}; x_{-am}, p_m) \tag{1.27}$$

The total cost the player wish to minimize is:

$$\begin{aligned}
& f_{a1}(x_{a1}; x_{-a1}, p_1) + \mathbb{E}_1\Big(\big(f_{an}(x_{an}; x_{-an}, p_n)\big)_{n\in\bar{\mathcal{S}}(1)}\Big) \\
=\ & f_{a1}(x_{a1}; x_{-a1}, p_1) + \sum_{n\in\bar{\mathcal{S}}(1)} \Big(\prod_{l\in\bar{\mathcal{P}}(1,n)} \phi_l\Big) \cdot f_{an}(x_{an}; x_{-an}, p_n) \\
=\ & f_{a1}(x_{a1}; x_{-a1}, p_1) + \sum_{n\in\bar{\mathcal{S}}(1)} \pi_n \cdot f_{an}(x_{an}; x_{-an}, p_n)
\end{aligned} \tag{1.28}$$

We also assume the stochastic MOPEC in this thesis has a special feasible region for each player:

$$X_a(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}) = \{(x_{an})_{n\in\mathcal{N}} | x_{an} \in \mathcal{X}_{an}, \quad \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, \quad \forall n \in \mathcal{N}\} \tag{1.29}$$

where $\mathcal{X}_{an} \subseteq \mathbb{R}^{d_{an}}$ is a closed, convex set, $\mathcal{K}_{an} \subseteq \mathbb{R}^{\psi_{an}}$ is a closed, convex cone, and $\mathcal{G}_{an} : \mathbb{R}^{d_{an_-}+d_n+\alpha_n} \to \mathbb{R}^{\psi_{an}}$ is a continuously differentiable function and $\mathcal{G}_{an}(\cdot, \cdot; x_{-an}, p_n)$ is a convex function for each fixed x_{-an} and p_n.

In this formulation, it is assumed that the algebraic function $\mathcal{G}_{an}$ only depend on the player a's decision at the current node and its parent node, the strategies of other players

at the current node, and the market price at the current node. This assumption means that the problem assumes an independent market at each node, and transitions between scenario nodes are limited to within each individual player. Furthermore, it is argued that, for each player, considering only their decisions at the current node and its parent node is sufficient to capture general situations, with the aid of simple transformations.

Thus, after using a scenario tree to represent the multistage stochastic process, each player a's multistage stochastic optimization parameterized by other players' strategies $\boldsymbol{x}_{-a\cdot}$ and market price $\boldsymbol{p}$ can be represented as the following *extensive format*:

$$\begin{aligned} P_a^{\mathcal{RN}}(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}): \quad \min_{\boldsymbol{x}_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \sum_{n \in \bar{\mathcal{S}}(1)} \pi_n \cdot f_{an}(x_{an}; x_{-an}, p_n) \\ \text{s.t.} \quad & \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, \qquad \forall n \in \mathcal{N} \\ & x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N} \end{aligned} \tag{1.30}$$

At scenario node n, the market price at that node p_n, taking all players' strategies at that node $\boldsymbol{x}_{\cdot n} = (x_{an})_{a \in \mathcal{A}}$ as given, satisfies the equilibrium constraint:

$$0 \in F_n(p_n; \boldsymbol{x}_{\cdot n}) + \mathcal{N}_{K_n}(p_n) \tag{1.31}$$

where $F_n : \mathbb{R}^{\alpha_n + \sum_{a \in \mathcal{A}} d_{an}} \to \mathbb{R}^{\alpha_n}$ is a given continuous function that embodies the equilibrium condition and is influenced by the price p_n and the strategies of all players $\boldsymbol{x}_{\cdot n}$ at scenario node n, typically representing the supply-demand relationship in economics.

Let $Q_n^{VI}(\boldsymbol{x}) \coloneqq \{p_n | p_n \in K_n, p_n \text{ solves } (1.31)\}$ denote the set of market price p_n that safisfies the equilibrium contraint at scenario node n. Then the combination of each players' optimization problem (1.30) and the market constraint (1.31) leads to the following definition for a stochastic MOPEC with risk-neutral players:

Definition 1.4. $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \left((x_{an}^*)_{a \in \mathcal{A}, n \in \mathcal{N}}, (p_n^*)_{n \in \mathcal{N}}\right)$ is an equilibrium point of the multistage stochastic MOPEC with risk-neutral players if and only if

$$\begin{aligned} \boldsymbol{x}_{a\cdot}^* \quad & \text{is an optimal solution of} \quad P_a^{\mathcal{RN}}(\boldsymbol{x}_{-a\cdot}^*, \boldsymbol{p}^*) \qquad \text{for all } a \in \mathcal{A}. \\ p_n^* \quad & \text{belongs to set} \quad Q_n^{VI}(\boldsymbol{x}^*) \qquad\qquad\qquad\quad \text{for all } n \in \mathcal{N}. \end{aligned} \tag{1.32}$$

Remark 1.5. A special case of the stochastic MOPEC is the dynamic MOPEC problem, which can also be represented by a "special" tree strucutre. An example of a dynamic problem with a total of 4 stages is shown in Figure 1.2.

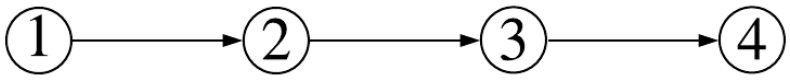


Figure 1.2: Dynamic equilibrium with 4-time stages

In the above dynamic equilibrium problem, the strategy of each player x_a could be decomposed into the strategies of player at each time stage t, denoted by $x_{at} \in \mathbb{R}^{d_{at}}$, where t is the time stage index and d_{at} is the dimension of the strategy. The entire strategy of player a can then be represented as $\boldsymbol{x}_{a\cdot} := (x_{at})_{t\in\mathcal{T}} \in \mathbb{R}^{d_a}$, where $d_a := \sum_{t\in\mathcal{T}} d_{at}$. The vector consisting of all strategies of players other than player a at time stage t is denoted by $x_{-at} := (x_{a't})_{a'\in\mathcal{A},a'\neq a} \in \mathbb{R}^{d_{-at}}$, where $d_{-at} := \sum_{a'\in\mathcal{A},a'\neq a} d_{a'n} = d_t - d_{at}$. The vector including the entire strategies of all players excluding player a at all stages is denoted by $\boldsymbol{x}_{-a\cdot} := (x_{-at})_{t\in\mathcal{T}}$. $p_t \in \mathbb{R}^{\alpha_t}$ is the market price at stage t. Each player a is trying to solve the optimization problem parameterized by other players' strategies $\boldsymbol{x}_{-a\cdot}$ and market price $\boldsymbol{p}$:

$$\begin{aligned} P_a^{\mathcal{D}}(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}): \quad \min_{\boldsymbol{x}_{a\cdot}} \quad & \sum_{t\in\mathcal{T}} f_{at}(x_{at}; x_{-at}, p_t) \\ \text{s.t.} \quad & \mathcal{G}_{at}(x_{at-1}, x_{at}; x_{-at}, p_t) \in \mathcal{K}_{at}, \quad \forall t \in \mathcal{T} \\ & x_{at} \in \mathcal{X}_{at}, \quad \forall t \in \mathcal{T} \end{aligned} \tag{1.33}$$

where $f : \mathbb{R}^{d_t+\alpha_t} \to \mathbb{R}$ is a continuously differentiable function and $f(\cdot; x_{-at}, p_t)$ is a convex function given fixed other players' strategies x_{-at} and market price p_t, $\mathcal{X}_{at} \subseteq \mathbb{R}^{d_{at}}$ is a closed, convex set, $\mathcal{K}_{at} \subseteq \mathbb{R}^{\psi_{at}}$ is a closed, convex cone, and $\mathcal{G}_{at} : \mathbb{R}^{d_{at-1}+d_t+\alpha_t} \to \mathbb{R}^{\psi_{at}}$ is a continuously differentiable function and $\mathcal{G}_{at}(\cdot, \cdot; x_{-at}, p_t)$ is a convex function for each fixed x_{-at} and p_t. There is an equilibrium market constraint for each $t \in \mathcal{T}$:

$$0 \in F_t(p_t; \boldsymbol{x}_{\cdot t}) + \mathcal{N}_{K_t}(p_t) \tag{1.34}$$

Define $Q_t^{VI}(\boldsymbol{x}) := \{p_t | p_t \in K_t, p_t \text{ solves } (1.34)\}$. A point $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \left((x_{at}^*)_{a\in\mathcal{A},t\in\mathcal{T}}, (p_t^*)_{t\in\mathcal{T}}\right)$ is an equilibrium point of an dynamic equilibrium problem if and only if

$$\begin{aligned} \boldsymbol{x}_{a\cdot}^* \quad & \text{is an optimal solution of} \quad P_a^{\mathcal{D}}(\boldsymbol{x}_{-a\cdot}^*, \boldsymbol{p}^*) \qquad \text{for all } a \in \mathcal{A}. \\ p_t^* \quad & \text{belongs to set} \quad Q_t^{VI}(\boldsymbol{x}^*) \qquad\qquad\qquad\quad \text{for all } t \in \mathcal{T}. \end{aligned} \tag{1.35}$$

The above dynamic equilibrium problem can also be represented by a stochastic MOPEC with the format (1.32) based on the following scenario tree structure. The set of scenario nodes of the tree is equivalent to the set of time stage $\mathcal{T}$. The set of edges $\mathcal{E}$ could be

constructed by letting $t_+ = t + 1$ when $t < T$. Because each node t only has one child node, we define the conditional probability $\phi_t = 1$ for every $t \in \mathcal{T}$. This formulation of the dynamic MOPEC fits perfectly in the format (1.32), indicating that the dynamic MOPEC is a special case of the stochastic MOPEC.

In fact, the structure of multistage stochastic MOPEC with risk-averse players could be represented by the Figure 1.3:

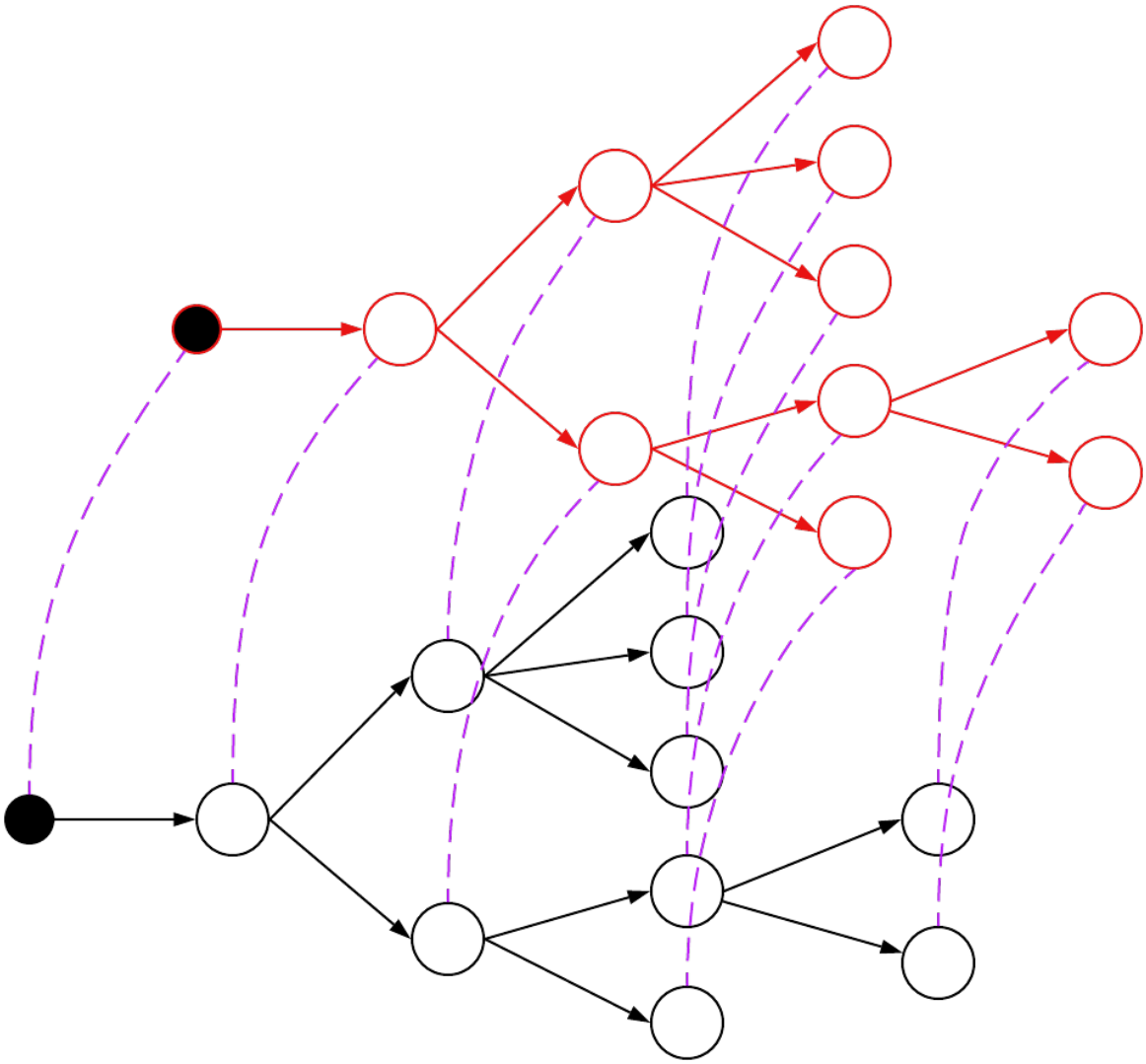


Figure 1.3: Visualization of the structure of multistage stochastic MOPEC with risk-averse players

Figure 1.3 illustrates the problem structure, wherein the black circle denotes the initial state, and the white circles represent the nodes of the scenario tree. The purple dashed

line symbolizes the interactions occurring between players and markets within the system.

1.4 Coherent risk measures

In this section, we delve into the concept of coherent risk measures, which play a crucial role in this thesis. Unlike most stochastic MOPECs that utilize the expectation operator $\mathbb{E}(\cdot)$ as a tool to measure the risk of uncertainty in their objectives, this thesis incorporates the use of coherent risk measures to cater to different requirements in the treatment of uncertainty.

The linearity property of the expectation operator makes it a widely used risk mapping tool. However, different players may have varying risk preferences, such as considering the worst-case scenario to be more important or the expectation of costs above a certain percentile φ.

To address these varying risk preferences, this thesis focuses on incorporating a popular risk mapping tool, the coherent risk measure, which is defined on a discrete sample space $\mathcal{I} = \{1, 2, 3, \ldots, |\mathcal{I}|\}$. In this framework, the decision maker is faced with a random cost $Z_i, i \in \mathcal{I}$, and the risk-adjusted disbenefit of Z is represented by $\rho(Z)$. The definition of the coherent risk measure is provided below.

Definition 1.6. A coherent risk measure ρ on probability space $(\Omega, \mathcal{I}, \pi)$ is a function from $\mathcal{L}^2(\Omega, \mathcal{I}, \pi)$ to $(-\infty, +\infty]$ and satisfies the following five axioms:

1. $\rho(C) = C$ for all constant C.
2. $\rho((1-\lambda)Z + \lambda Z') \leq (1-\lambda)\rho(Z) + \lambda\rho(Z')$ for $\lambda \in [0, 1]$
3. $\rho(Z) \leq \rho(Z')$ if $Z \leq Z'$ almost surely.
4. $\rho(Z) \leq 0$ when $\|Z^k - Z\|_2 \to 0$ with $\rho(Z^k) \leq 0$.
5. $\rho(\lambda Z) = \lambda\rho(Z)$ for $\lambda > 0$.

In fact, the coherent risk measure $\rho(Z)$ has a very important property that it has a dual representation expressed as

$$\rho(Z) = \sigma_{\mathcal{D}}(Z) = \sup_{\mu \in \mathcal{D}} \langle \mu, Z \rangle$$

where $\mathcal{D} \subseteq \{\mu \in \mathbb{R}_+^{|\mathcal{I}|} | \sum_{i \in \mathcal{I}} \mu_i = 1\}$ is a convex set and is the risk set of the coherent risk measure. $\sigma_{\mathcal{D}}(\cdot)$ is also called a *support function evaluation* on the risk set $\mathcal{D}$. Note that $\mu \in \mathcal{D}$

is a multi-dimensional vector with the property that $\sum_{i\in\mathcal{I}}\mu_i = 1$ and $\mu_i \geq 0$, where μ_i is the ith component of the vector μ. Here are four examples of coherent risk measures and their dual representations:

Example 1 (Expected value via support function evaluation) It is easy to see that the expectation operator $\mathbb{E}(\cdot)$ is a coherent risk measure with the dual risk set $\mathcal{D}$ being a singleton and $\mathcal{D} = \{\phi\}$. In this case, we know

$$\mathbb{E}(Z) = \sup_{\mu\in\mathcal{D}} \mu^T Z = \sigma_{\{\phi\}}(Z) = \sup_{\mu\in\{\phi\}} \mu^T Z = \phi^T Z,$$

where $\phi = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_{|\mathcal{I}|} \end{bmatrix}$.

Example 2 (CVaR via support function evaluation) $CVaR_\varphi(Z) \triangleq \min_{\kappa\in\mathbb{R}} \left\{\kappa + \frac{1}{1-\varphi}\mathbb{E}\big[Z-\kappa\big]_+\right\}$ is the value of the support function of $\mathcal{D}_{CVaR}(\varphi)$ at Z, where

$$\mathcal{D}_{CVaR}(\varphi) := \left\{\mu \in \frac{1}{1-\varphi}\prod_{i\in\mathcal{I}}[0,\phi_i] | \sum_{i\in\mathcal{I}}\mu_i = 1\right\}$$

Example 3 (Convex combination of Expectation and CVaR)

$$\overline{CVaR}(\lambda,\varphi)(Z) = (1-\lambda)\cdot\mathbb{E}(Z) + \lambda\cdot CVaR_\varphi(Z) = \sigma_{(1-\lambda)\{\phi\}+\lambda\mathcal{D}(\varphi)}(Z)$$

Example 4 (K-Deviance via support function evaluation) $Dev_K(Z)$ is the value of the support function of $\mathcal{D}_{Dev}(K)$ at Z, where

$$\mathcal{D}_{Dev}(K) := \left\{\mu_i = \pi_i\cdot(1+\nu_i - \sum_{j\in\mathcal{I}}\phi_j\cdot\nu_j) \mid 0\leq\nu_i\leq K, \forall i\in\mathcal{I}\right\}$$

1.4.1 Two different mathematical formulation representations for two-stage stochastic MOPEC with risk-averse players

We start with the two stage stochastic MOPEC as we transition from risk-neutral players to risk-averse players, so $\mathcal{N} = \{1\}\cup 1_+$. It is assumed that all players will adopt coherent risk measures to reflect their risk preference. For a two-stage problem, if each player replaces the expectation operator $\mathbb{E}(\cdot)$ with their preferable risk measure $\rho_{a1}(\cdot)$ in equation (1.28), then their objectives will become:

$$\min_{\boldsymbol{x}_{a\cdot}} \quad f_{a1}(x_{a1}; x_{-a1}, p_1) + \rho_{a1}\Big(\big[f_{an}(x_{an}; x_{-an}, p_n)\big]_{n\in 1_+}\Big) \tag{1.36}$$

In fact, if the stochastic process is represented by a discrete scenario tree, we could have two different algebraic formulations to represent the stochastic MOPECs: *Equilibrium reformulation* and *Conjugate-based reformulation*.

1.4.1.1 Equilibrium reformulation

Under the assumption that each coherent risk measure has the corresponding dual representation, then we could represent each player's coherent risk measure $\rho_{a1}(\cdot)$ by their dual format:

$$\rho_{a1}(\cdot) = \sigma_{\mathcal{D}_{a1}}(\cdot) = \sup_{\mu_{a1_+} \in \mathcal{D}_{a1}} \langle \mu_{a1_+}, \cdot \rangle$$

where $\mathcal{D}_{a1}$ is the risk set of ρ_{a1} and $\mu_{a1_+} = (\mu_{an})_{n \in 1_+}$denotes the vector that collects the risk probability variable of each player at each node. Based on these, the optimization problem for each player a could be written in minimax problem format:

$$\begin{aligned} P_a^{\mathcal{RA}-\mathcal{E}}(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}): \quad \min_{\boldsymbol{x}_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \max_{\mu_{a1_+} \in \mathcal{D}_{a1}} \sum_{n \in 1+} \mu_{an} \cdot f_{an}(x_{an}; x_{-an}, p_n) \\ \text{s.t.} \quad & \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, \qquad \forall n \in \mathcal{N} \\ & x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N} \end{aligned} \tag{1.37}$$

Combining each player's optimization problems (1.37) and market constraints (1.31), we could define the two-stage stochastic MOPEC with risk-averse players utilizing dual representation of coherent risk measure:

Definition 1.7. $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \left((x^*_{an})_{a \in \mathcal{A}, n \in \mathcal{N}}, (p^*_n)_{n \in \mathcal{N}}\right)$ is an equilibrium point of the above two-stage stochastic MOPEC with risk-averse players utilizing dual representation of coherent risk measure if and only if

$$\begin{aligned} \boldsymbol{x}^*_{a\cdot} \quad & \text{is an optimal solution of} \quad P_a^{\mathcal{RA}-\mathcal{E}}(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*) \qquad \text{for all } a \in \mathcal{A} \\ p^*_n \quad & \text{belongs to set} \quad Q_n^{VI}(\boldsymbol{x}^*) \qquad\qquad\qquad\qquad\quad \text{for all } n \in \mathcal{N} \end{aligned} \tag{1.38}$$

An *equilibrium reformulation* of the above problem (1.38) was proposed in [56]. It is shown that the player's optimization problem $P_a^{\mathcal{RA}-\mathcal{E}}(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*)$ can be represented by an equilibrium problem. For each player $a \in \mathcal{A}$, there exists one addtional player who is controlling the vector μ_{a1_+} and trying to solve the following optimization problem:

$$\max_{\mu_{a1_+} \in \mathcal{D}_{a1}} \sum_{n \in 1+} \mu_{an} \cdot f_{an}(x_{an}; x_{-an}, p_n) \tag{1.39}$$

with given $\boldsymbol{x}$ and $\boldsymbol{p}$. We denote this problem by

$$R_a(\boldsymbol{x}, \boldsymbol{p}) : \max_{\mu_{a1_+} \in \mathcal{D}_{a1}} \sum_{n \in 1+} \mu_{an} \cdot f_{an}(x_{an}; x_{-an}, p_n) \tag{1.40}$$

After moving the risk-averse part out, each player $a \in \mathcal{A}$ is trying to solve the following problem with given $(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}, \mu_{a1_+})$:

$$\begin{aligned} P_a^{\mathcal{F}}(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}, \mu_{a1_+}) : \quad \min_{\boldsymbol{x}_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \sum_{n \in 1_+} \mu_{an} \cdot f_{an}(x_{an}; x_{-an}, p_n) & \\ \text{s.t.} \quad & \mathcal{G}_{an}(x_{a,n_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, & \forall n \in \mathcal{N} \\ & x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N} & \end{aligned} \tag{1.41}$$

Combing the above formulations (1.40)-(1.41) and market constraint (1.31), [56] gives the definition for the equilibrium point of two-stage stochastic MOPEC with risk-averse players in *equilibrium reformulation* format:

Definition 1.8. $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*) = \left((x^*_{an})_{a\in\mathcal{A}, n\in\mathcal{N}}, (p^*_n)_{n\in\mathcal{N}}, (\mu^*_{an})_{a\in\mathcal{A}, n\in 1_+}\right)$ is an equilibrium point of the above two-stage stochastic MOPEC with risk-averse players in *equilibrium reformulation* format if and only if

$$\begin{aligned} \boldsymbol{x}^*_{a\cdot} \quad & \text{is an optimal solution of} \quad P_a^{\mathcal{F}}(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*, \mu^*_{a1_+}) && \text{for all } a \in \mathcal{A} \\ p^*_n \quad & \text{belongs to set} \quad Q_n^{VI}(\boldsymbol{x}^*) && \text{for all } n \in \mathcal{N} \\ \mu^*_{a1_+} \quad & \text{is an optimal solution of} \quad R_a(\boldsymbol{x}^*, \boldsymbol{p}^*) && \text{for all } a \in \mathcal{A} \end{aligned} \tag{1.42}$$

We have the following equivalence between equilibrium points of the above two formulations:

Proposition 1.9. *If* $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*) = \left((x^*_{an})_{a\in\mathcal{A}, n\in 1_+}, (p^*_n)_{n\in\mathcal{N}}, (\mu^*_{an})_{a\in\mathcal{A}, n\in 1_+}\right)$ *is an equilibrium point of two-stage stochastic MOPEC with risk-averse players in equilibrium reformulation format (1.42) then its paritial solution* $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \left((x^*_{an})_{a\in\mathcal{A}, n\in\mathcal{N}}, (p^*_n)_{n\in\mathcal{N}}\right)$ *is an equilibrium point of two-stage stochastic MOPEC with risk-averse players in format (1.38).*

Proof. If $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*) = \left((x^*_{an})_{a\in\mathcal{A}, n\in 1_+}, (p^*_n)_{n\in\mathcal{N}}, (\mu^*_{an})_{a\in\mathcal{A}, n\in 1_+}\right)$ is an equilibrium point of (1.42). For any $a \in \mathcal{A}$, let $\boldsymbol{x}_{a\cdot} \in X_a(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*) = \{(x_{an})_{n\in\mathcal{N}} | \mathcal{G}_{an}(x_{a,n_-}, x_{an}; x^*_{-an}, p^*_n) \in$

$\mathcal{K}_{an}, x_{an} \in \mathcal{X}_{an}, \forall n \in \mathcal{N}\}$ be arbitrary. We will have

$$\begin{aligned} & f_{a1}(x_{a1}; x^*_{-a1}, p^*_1) + \max_{\mu_{a1+} \in \mathcal{D}_{a1}} \sum_{m \in 1+} \mu_{am} \cdot f_{am}(x_{am}; x^*_{-am}, p^*_m) \\ \geq \quad & f_{a1}(x_{a1}; x^*_{-a1}, p^*_1) + \sum_{m \in 1+} \mu^*_{am} \cdot f_{am}(x_{am}; x^*_{-am}, p^*_m) \\ \geq \quad & f_{a1}(x^*_{a1}; x^*_{-a1}, p^*_1) + \sum_{m \in 1+} \mu^*_{am} \cdot f_{am}(x^*_{am}; x^*_{-am}, p^*_m) \\ = \quad & f_{a1}(x^*_{a1}; x^*_{-a1}, p^*_1) + \max_{\mu_{a1+} \in \mathcal{D}_{a1}} \sum_{m \in 1+} \mu_{am} \cdot f_{am}(x^*_{am}; x^*_{-am}, p^*_m) \end{aligned} \tag{1.43}$$

This shows that

$$\boldsymbol{x}^*_{a\cdot} \in \quad \arg \min_{\boldsymbol{x}_{a\cdot} \in X_a(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*)} f_{a1}(x_{a1}; x^*_{-a1}, p^*_1) + \max_{\mu_{a1+} \in \mathcal{D}_{a1}} \sum_{m \in 1+} \mu_{am} \cdot f_{am}(x_{am}; x^*_{-am}, p^*_m) \tag{1.44}$$

With the market constraints:

$$0 \in F_n(p^*_n; \boldsymbol{x}^*_{\cdot n}) + \mathcal{N}_{K_n}(p^*_n), \quad \forall n \in \mathcal{N} \tag{1.45}$$

we could have $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ is an equilibrium point of (1.38). This completes the proof of the proposition. □

1.4.1.2 Conjugate-based reformulation

In this reformulation, we consider the scenario where the risk set $\mathcal{D}_{a1}$ is a polyhedral for each player $a \in \mathcal{A}$. The reason is that the situation where the risk set $\mathcal{D}_{a1}$ is a polyhedral set is really common and there is an equivalent reformulation of the support function $\sigma_{\mathcal{D}_{a1}}$ whenever $\mathcal{D}_{a1}$ is a polyhedral set. Assuming $\mathcal{D}_{a1}$ to be the closed convex hull of the set of vertices $\{\mu^k_{a1_+} | k \in \Lambda_{a1}\}$, where Λ_{a1} is a finite index set for the extreme points of risk set $\mathcal{D}_{a1}$, we can express it as follows:

$$\rho_{a1}(Z) = \sigma_{\mathcal{D}_{a1}}(Z) = \sup_{\mu_{a1_+} \in \mathcal{D}_{a1}} \langle \mu_{a1_+}, Z \rangle = \max_{k \in \Lambda_{a1}} \langle \mu^k_{a1_+}, Z \rangle = \max_{k \in \Lambda_{a1}} \sum_{n \in 1_+} \mu^k_{an} \cdot Z_n \tag{1.46}$$

The third equality in equation (1.46) is a result of the linearity property of the support function. Let $\bar{\theta} \in \mathbb{R}$ denote the risk-adjusted disbenefit associated with all random future outcomes $(Z_n)_{n \in 1_+}$. It follows that:

$$\bar{\theta} = \max_{k \in \Lambda_{a1}} (\mu^k_{a1_+})^T Z = \begin{cases} \min & \theta \\ \text{s.t.} & \theta \geq \sum_{n \in 1_+} \mu^k_{an} \cdot Z_n, \ \ \forall k \in \Lambda \end{cases}$$

Here $\theta \in \mathbb{R}$ is a free variable in the minimization problem on the righthand side.

Let's look back at our two-stage stochastic MOPEC (1.36). Let $\bar{\theta}_{a1} \in \mathbb{R}$ denote the risk-adjusted disbenefit of all random outcomes of player a. We would have

$$\begin{aligned}
\bar{\theta}_{a1} &= \rho_{a1}\Big(\big[f_{an}(x_{an}; x_{-an}, p_n)\big]_{n\in 1_+}\Big) = \max_{\mu_{a1_+}\in\mathcal{D}_{a1}} \sum_{n\in 1+} \mu_{an}\cdot f_{an}(x_{an}; x_{-an}, p_n) \\
&= \max_{k\in\Lambda_{a1}} \sum_{n\in 1+} \mu^k_{an}\cdot f_{an}(x_{an}; x_{-an}, p_n) \\
&= \begin{cases} \min & \theta_{a1} \\ \text{s.t.} & \theta_{a1} \geq \sum_{n\in 1_+} \mu^k_{an}\cdot f_{an}(x_{an}; x_{-an}, p_n), \quad \forall k \in \Lambda_{a1} \end{cases}
\end{aligned} \tag{1.47}$$

Here $\theta_{a1} \in \mathbb{R}$ is a free variable, $\{\mu^k_{a1+} | k \in \Lambda_{a1}\}$ are the set of extreme points of polyhedral risk set $\mathcal{D}_{a1}$. Then we will have

$$\begin{aligned}
& \min_{\boldsymbol{x}_{a\cdot}} && f_{a1}(x_{a1}; x_{-a1}, p_1) + \rho_{a1}\Big(\big[f_{an}(x_{an}; x_{-an}, p_n)\big]_{n\in 1_+}\Big) \\
(=)\quad & \min_{\boldsymbol{x}_{a\cdot}} && f_{a1}(x_{a1}; x_{-a1}, p_1) + \max_{\mu_{a1_+}\in\mathcal{D}_{a1}} \sum_{n\in 1+} \mu_{an}\cdot f_{an}(x_{an}; x_{-an}, p_n) \\
(=)\quad & \min_{\boldsymbol{x}_{a\cdot}} && f_{a1}(x_{a1}; x_{-a1}, p_1) + \max_{k\in\Lambda_{a1}} \sum_{n\in 1+} \mu^k_{an}\cdot f_{an}(x_{an}; x_{-an}, p_n) \\
(=)\quad & \min_{\boldsymbol{x}_{a\cdot}, \theta_{ai}} && f_{a1}(x_{a1}; x_{-a1}, p_1) + \theta_{a1} \\
& \text{s.t.} && \theta_{a1} \geq \sum_{n\in 1+} \mu^k_{an}\cdot f_{an}(x_{an}; x_{-an}, p_n), && \forall k \in \Lambda_{a1}
\end{aligned} \tag{1.48}$$

Based on this, we can achieve the *conjugate-based reformulation* of a two-stage stochastic equilibrium problem with risk-averse players as follows:

Each player a, with given $\boldsymbol{x}_{-a\cdot}$ and $\boldsymbol{p}$, will solve the optimization problem:

$$\begin{aligned}
P^{\mathcal{RA}-\mathcal{C}}_a(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}): \quad & \min_{\boldsymbol{x}_{a\cdot}, \theta_{a1}} && f_{a1}(x_{a1}; x_{-a1}, p_1) + \theta_{a1} \\
& \text{s.t.} && \theta_{a1} \geq \sum_{n\in 1+} \mu^k_{an}\cdot f_{an}(x_{an}; x_{-an}, p_n), \quad \forall k \in \Lambda_{a1} \\
& && \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, \qquad \forall n \in \mathcal{N} \\
& && x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N}
\end{aligned} \tag{1.49}$$

Based on the players' optimization problems (1.49) and the market constraint (1.31), we have the definition for an equilibrium point of a two-stage stochastic MOPEC with risk-averse players in the *conjugate-based reformulation* format:

Definition 1.10. $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\theta}^*) = \left((x^*_{an})_{a\in\mathcal{A},n\in\mathcal{N}}, (p^*_n)_{n\in\mathcal{N}}, (\theta^*_{a1})_{a\in\mathcal{A}}\right)$ is an equilibrium point of a two-stage stochastic MOPEC with risk-averse players in the *conjugate-based reformulation* format if and only if

$$\begin{array}{rlll} (\boldsymbol{x}^*_{a\cdot}, \theta^*_{a1}) & \text{is an optimal solution of} & P_a^{\mathcal{RA}-\mathcal{C}}(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*) & \text{for all } a \in \mathcal{A} \\ p^*_n & \text{belongs to set} & Q_n^{VI}(\boldsymbol{x}^*) & \text{for all } n \in \mathcal{N} \end{array} \tag{1.50}$$

1.5 Dynamic multistage risk measures

Now let's transition our focus to the multistage setting. We incorporate the risk measures discussed above into a multistage setting in which players make decisions over several time stages to minimize risk-adjusted expected disbenefit.

For a multistage decision problem, we require a dynamic version of risk. The concept of coherent risk measures was introduced in [74] and is described for general Markov decision problems in [77]. Formally one defines a probability space $(\Omega, \mathcal{F}, \pi)$ and a filtration $\{\emptyset, \Omega\} = \mathcal{F}_1 \subset \mathcal{F}_2 \cdots \subset \mathcal{F}_T \subset \mathcal{F}$ of $\sigma-$fields where all data in node 0 is assumed to be deterministic and decisions at time t are $\mathcal{F}_t-$measurable random variables (see [77]). Working with finite probability spaces defined by a scenario tree simplifies this description.

Given a tree defined by $\mathcal{N}$, suppose for each player $a \in \mathcal{A}$ the random sequence of actions $(x_{an})_{n\in\mathcal{N}}$ results in a random sequence of disbenefits $\{f_{an}(x_{an}; x_{-an}, p_n), n \in \mathcal{N}\}$ parameterized by $(x_{-an})_{n\in\mathcal{N}}$ and $(p_n)_{n\in\mathcal{N}}$. We seek to measure the risk of this disbenefit sequence when viewed by a decision maker at node 1. At node n the decision maker is endowed with coherent risk measure ρ_{an} with a corresponding a one-step risk set $\mathcal{D}_{an}$ that measures the risk of random risk-adjusted costs accounted for in $m \in n_+$ for each player a.

In this situation, we can get an extended version of risk disbenefit at node n of player a:

$$\begin{aligned} & \rho_{an}\left(\left(f_{am}(x_{am}; x_{-am}, p_m)\right)_{m\in\bar{\mathcal{S}}(n)}\right) \\ = \; & \rho_{an}\left(\left(f_{am}(x_{am}; x_{-am}, p_m) + \rho_{am}\left(\left[f_{al}(x_{al}; x_{-al}, p_l) + \rho_{al}\left(\cdots\right)\right]_{l\in m_+}\right)\right)_{m\in n_+}\right) \end{aligned} \tag{1.51}$$

Similar to equation (1.26), the risk disbenefit of future outcomes at node n of player a

can also be defined recursively to be:

$$\begin{aligned}
&\rho_{an}\Big(\big(f_{am}(x_{am};x_{-am},p_m)\big)_{m\in\bar{\mathcal{S}}(n)}\Big)\\
&=\begin{cases}0, & n\in\mathcal{L}\\ \rho_{an}\Bigg(\Big(f_{am}(x_{am};x_{-am},p_m)+\rho_{am}\Big(\big(f_{al}(x_{al};x_{-al},p_l)\big)_{l\in\bar{\mathcal{S}}(m)}\Big)\Big)_{m\in n_+}\Bigg), & n\in\mathcal{N}\backslash\mathcal{L}\end{cases}
\end{aligned}\tag{1.52}$$

Based on above equation (1.52), the total cost of player a is:

$$\begin{aligned}
&f_{a1}(x_{a1};x_{-a1},p_1)+\rho_{an}\Big(\big(f_{am}(x_{am};x_{-am},p_m)\big)_{m\in\bar{\mathcal{S}}(n)}\Big)\\
=\;& f_{a1}(x_{a1};x_{-a1},p_1)+\rho_{a1}\Bigg(\Big[f_{am}(x_{am};x_{-am},p_m)\\
&+\rho_{am}\Big([f_{al}(x_{al};x_{-al},p_l)+\rho_{al}(\cdots)]_{l\in m_+}\Big)\Big]_{m\in 1+}\Bigg)
\end{aligned}\tag{1.53}$$

Thus, for each player $a\in\mathcal{A}$, he is trying to solve the following problem when involving coherent risk measure $\{\rho_{an}(\cdot)|n\in\mathcal{N}\backslash\mathcal{L}\}$:

$$\begin{aligned}
\min_{\boldsymbol{x}_{a\cdot}}\quad & f_{a1}(x_{a1};x_{-a1},p_1)+\rho_{a1}\Bigg(\Big[f_{am}(x_{am};x_{-am},p_m)\\
&+\rho_{am}\Big([f_{al}(x_{al};x_{-al},p_l)+\rho_{al}(\cdots)]_{l\in m_+}\Big)\Big]_{m\in 1+}\Bigg)\\
\text{s.t.}\quad & \mathcal{G}_{an}(x_{an_-},x_{an};x_{-an},p_n)\in\mathcal{K}_{an}, \qquad \forall n\in\mathcal{N}\\
& x_{an}\in\mathcal{X}_{an},\quad \forall n\in\mathcal{N}
\end{aligned}\tag{1.54}$$

1.5.1 Equilibrium reformulation

According to the equivalent dual representation

$$\rho_{an}(\cdot)=\sigma_{\mathcal{D}_{an}}(\cdot)=\sup_{\mu_{an_+}\in\mathcal{D}_{an}}\langle\mu_{an_+},\cdot\rangle$$

here $\mathcal{D}_{an}$ is the corresponding risk set for coherent risk measure ρ_{an} and $\mu_{an_+}=(\mu_{am})_{m\in n_+}$ is a vector consisting all risk probability of player a at nodes that are children nodes of node n. Then each player's multistage optimization problem (1.54) can be written in the

following format:

$$
\begin{aligned}
P_a^{\mathcal{RA}-\mathcal{EM}}(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}): \quad \min_{\boldsymbol{x}_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \max_{\mu_{a1_+} \in \mathcal{D}_{a1}} \sum_{n \in 1+} \mu_{an} \cdot \Big[f_{an}(x_{an}; x_{-an}, p_n) \\
& + \max_{\mu_{an_+} \in \mathcal{D}_{an}} \sum_{m \in n+} \mu_{am} \cdot \big[f_{am}(x_{am}; x_{-am}, p_m) + \max_{\mu_{am_+} \in \mathcal{D}_{am}} \cdots \big] \Big] \\
\text{s.t.} \quad & \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, \qquad \forall n \in \mathcal{N} \\
& x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N}
\end{aligned}
\tag{1.55}
$$

Combining each player's multistage optimization problems (1.55) and market constraints (1.31), we could define the multistage stochastic MOPEC with risk-averse players utilizing dual representation of coherent risk measure:

Definition 1.11. $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \big((x^*_{an})_{a \in \mathcal{A}, n \in \mathcal{N}}, (p^*_n)_{n \in \mathcal{N}}\big)$ is an equilibrium point of the above multistage stochastic MOPEC with risk-averse players utilizing dual representation of coherent risk measure if and only if

$$
\begin{array}{lll}
\boldsymbol{x}^*_{a\cdot} & \text{is an optimal solution of} \quad P_a^{\mathcal{RA}-\mathcal{EM}}(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*) & \text{for all } a \in \mathcal{A} \\
p^*_n & \text{belongs to set} \quad Q_n^{VI}(\boldsymbol{x}^*) & \text{for all } n \in \mathcal{N}
\end{array}
\tag{1.56}
$$

[56] also proposed an *equilibrium reformulation* of the above multistage problem (1.56). For each player $a \in \mathcal{A}$ and scenario node n, there exists one addtional player who is controlling the vector μ_{an_+} and tyring to solve the following optimization problem:

$$
R_{an}(\boldsymbol{x}, \boldsymbol{p}, \mu_{a\bar{\bar{\mathcal{S}}}(n)}): \quad \max_{\mu_{an_+} \in \mathcal{D}_{an}} \sum_{m \in n+} \mu_{am} \cdot \Big[f_{am}(x_{am}; x_{-am}, p_m) + \sum_{l \in m_+} \mu_{al} \cdot \big[f_{al}(x_{al}; x_{-al}, p_l) + \cdots \big] \Big]
\tag{1.57}
$$

with given $\boldsymbol{x}$, $\boldsymbol{p}$ and $\mu_{a\bar{\bar{\mathcal{S}}}(n)} := (\mu_{al})_{l \in \bar{\bar{\mathcal{S}}}(n)}$.

After moving the risk-averse part out, each player $a \in \mathcal{A}$ is trying to solve the following problem with given $(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}, \boldsymbol{\mu}_{a\cdot})$, where $\boldsymbol{\mu}_{a\cdot} = (\mu_{an})_{n \in \bar{\mathcal{S}}(1)}$:

$$
\begin{aligned}
P_a^{\mathcal{F}}(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}, \boldsymbol{\mu}_{a\cdot}): \quad \min_{\boldsymbol{x}_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \sum_{n \in 1+} \mu_{an} \cdot \Big[f_{an}(x_{an}; x_{-an}, p_n) \\
& + \sum_{m \in n+} \mu_{am} \cdot \big[f_{am}(x_{am}; x_{-am}, p_m) + \cdots \big] \Big] \\
\text{s.t.} \quad & \mathcal{G}_{an}(x_{a,n_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, \qquad \forall n \in \mathcal{N} \\
& x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N}
\end{aligned}
\tag{1.58}
$$

Combing the above formulations (1.57)-(1.58) and market constraint (1.31), [56] gives the definition for the equilibrium point of multistage stochastic MOPEC with risk-averse players in the *equilibrium reformulation* format:

Definition 1.12. $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*) = \left((x^*_{an})_{a\in\mathcal{A},n\in\mathcal{N}}, (p^*_n)_{n\in\mathcal{N}}, (\mu^*_{an})_{a\in\mathcal{A},n\in\bar{\mathcal{S}}(1)}\right)$ is an equilibrium point of the above multistage stochastic MOPEC with risk-averse players in *equilibrium reformulation* format if and only if

$$\begin{array}{rlll} \boldsymbol{x}^*_{a\cdot} & \text{is an optimal solution of} \quad P^{\mathcal{F}}_a(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*, \boldsymbol{\mu}^*_{a\cdot}) & \text{for all } a \in \mathcal{A} & \\ p^*_n & \text{belongs to set} \quad Q^{VI}_n(\boldsymbol{x}^*) & \text{for all } n \in \mathcal{N} & (1.59) \\ \mu^*_{an_+} & \text{is an optimal solution of} \quad R_{an}(\boldsymbol{x}^*, \boldsymbol{p}^*, \mu^*_{a\bar{\mathcal{S}}(n)}) & \text{for all } a \in \mathcal{A}, n \in \mathcal{N}\backslash\mathcal{L} & \end{array}$$

We have the following equivalence between equilibrium points between above two formulations:

Proposition 1.13. *If* $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*) = \left((x^*_{an})_{a\in\mathcal{A},n\in\mathcal{N}}, (p^*_n)_{n\in\mathcal{N}}, (\mu^*_{an})_{a\in\mathcal{A},n\in\bar{\mathcal{S}}(1)}\right)$ *is an equilibrium point of multistage stochastic MOPEC with risk-averse players in equilibrium reformulation format (1.59) then its paritial solution* $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \left((x^*_{an})_{a\in\mathcal{A},n\in\mathcal{N}}, (p^*_n)_{n\in\mathcal{N}}\right)$ *is an equilibrium point of two-stage stochastic MOPEC with risk-averse players in format (1.56).*

Proof. If $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*) = \left((x^*_{an})_{a\in\mathcal{A},n\in\mathcal{N}}, (p^*_n)_{n\in\mathcal{N}}, (\mu^*_{an})_{a\in\mathcal{A},n\in\bar{\mathcal{S}}(1)}\right)$ is an equilibrium point of (1.59). For any $a \in \mathcal{A}$, let $\boldsymbol{x}_{a\cdot} \in X_a(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*) = \{(x_{an})_{n\in\mathcal{N}} | \mathcal{G}_{an}(x_{a,n_-}, x_{an} | x^*_{-an}, p^*_n) \in \mathcal{K}_{an}, x_{an} \in$

$\mathcal{X}_{an}, \forall n \in \mathcal{N}\}$ be arbitrary. We will have

$$\begin{aligned}
& f_{a1}(x_{a1}; x^*_{-a1}, p^*_1) + \max_{\mu_{a1+} \in \mathcal{D}_{a1}} \Big\{ \sum_{m \in 1+} \mu_{am} \cdot \Big[f_{am}(x_{am}; x^*_{-am}, p^*_m) \\
& \qquad + \max_{\mu_{am_+} \in \mathcal{D}_{am}} \Big\{ \sum_{l \in m+} \mu_{al} \cdot \big[f_{al}(x_{al}; x^*_{-al}, p^*_l) + \cdots \big] \Big\} \Big] \Big\} \\
\geq \quad & f_{a1}(x_{a1}; x^*_{-a1}, p^*_1) + \sum_{m \in 1+} \mu^*_{am} \cdot \Big[f_{am}(x_{am}; x^*_{-am}, p^*_m) \\
& \qquad + \sum_{l \in m+} \mu^*_{al} \cdot \big[f_{al}(x_{al}; x^*_{-al}, p^*_l) + \cdots \big] \Big] \\
\geq \quad & f_{a1}(x^*_{a1}; x^*_{-a1}, p^*_1) + \sum_{m \in 1+} \mu^*_{am} \cdot \Big[f_{am}(x^*_{am}; x^*_{-am}, p^*_m) \\
& \qquad + \sum_{l \in m+} \mu^*_{al} \cdot \big[f_{al}(x^*_{al}; x^*_{-al}, p^*_l) + \cdots \big] \Big] \\
= \quad & f_{a1}(x^*_{a1}; x^*_{-a1}, p^*_1) + \max_{\mu_{a1+} \in \mathcal{D}_{a1}} \Big\{ \sum_{m \in 1+} \mu_{am} \cdot \Big[f_{am}(x^*_{am}; x^*_{-am}, p^*_m) \\
& \qquad + \max_{\mu_{am_+} \in \mathcal{D}_{am}} \Big\{ \sum_{l \in m+} \mu_{al} \cdot \big[f_{al}(x^*_{al}; x^*_{-al}, p^*_l) + \cdots \big] \Big\} \Big] \Big\}
\end{aligned} \tag{1.60}$$

This shows that

$$\begin{aligned}
\boldsymbol{x}^*_{a\cdot} \in \quad & \arg \min_{\boldsymbol{x}_{a\cdot} \in X_a(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*)} f_{a1}(x_{a1}; x^*_{-a1}, p^*_1) + \max_{\mu_{a1+} \in \mathcal{D}_{a1}} \Big\{ \sum_{m \in 1+} \mu_{am} \cdot \Big[f_{am}(x_{am}; x^*_{-am}, p^*_m) \\
& + \max_{\mu_{am_+} \in \mathcal{D}_{am}} \Big\{ \sum_{l \in m+} \mu_{al} \cdot \big[f_{al}(x_{al}; x^*_{-al}, p^*_l) + \cdots \big] \Big\} \Big] \Big\}
\end{aligned} \tag{1.61}$$

With the market constraints:

$$0 \in F_n(p^*_n; \boldsymbol{x}^*_{\cdot n}) + \mathcal{N}_{K_n}(p^*_n), \quad \forall n \in \mathcal{N} \tag{1.62}$$

we could have $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ is an equilibrium point of (1.56). This completes the proof of the proposition. □

1.5.2 Conjuate-based reformulation

Similar to the two-stage case, let's assume the risk sets $\{\mathcal{D}_{an}|a \in \mathcal{A}, n \in \mathcal{N}\}$ are polyhedrals and $\{\mu^k_{an_+}|k \in \Lambda_{an}\}$ are extreme points for risk set $\mathcal{D}_{an}$. Thus, we will have

$$\rho_{an}(Z) = \sigma_{\mathcal{D}_{an}}(Z) = \sup_{\mu_{an_+} \in \mathcal{D}_{an}} \langle \mu_{an_+}, Z \rangle = \max_{k \in \Lambda_{an}} (\mu^k_{an_+})^T Z = \max_{k \in \Lambda_{an}} \sum_{m \in n_+} \mu^k_{am} Z_m$$

Here Z_m is the random outcome at scenario node m.

Let $\theta_{an} \in \mathbb{R}$ denote the risk-adjusted disbenefit of all random future outcomes. We could have the definition of $\{\theta_{an}|a \in \mathcal{A}, n \in \mathcal{N}\}$ in a recursive way:

$$\theta_{an} = \begin{cases} 0, & n \in \mathcal{L} \\ \max_{k \in \Lambda_{an}} \sum_{m \in n+} \mu^k_{am} \cdot \big(f_{am}(x_{am}; x_{-am}, p_m) + \theta_{am}\big), & n \in \mathcal{N} \backslash \mathcal{L} \end{cases} \tag{1.63}$$

Replace $\rho_{an}(\cdot)$ in (1.53) by variable θ_{an}, each player's optimization problem will be in the following format with given $\boldsymbol{x}_{-a\cdot}$ and $\boldsymbol{p}$:

$$\begin{aligned} P_a^{\mathcal{RA}-\mathcal{CM}}(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}): \quad \min_{\boldsymbol{x}_{a\cdot}, \boldsymbol{\theta}_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \theta_{a1} \\ \text{s.t.} \quad & \theta_{an} \geq \sum_{m \in n+} \mu^k_{am} \cdot \big(f_{am}(x_{am}; x_{-am}, p_m) + \theta_{am}\big), \quad \forall k \in \Lambda_{an}, n \in \mathcal{N} \backslash \mathcal{L} \\ & \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, \qquad \forall n \in \mathcal{N} \\ & x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N} \\ & \theta_{an} = 0, \quad \forall n \in \mathcal{L} \end{aligned} \tag{1.64}$$

Based on players' optimization problems (1.64) and market constraint (1.31), we could have the definition for the equilibrium point of multistage stochastic MOPEC with risk-averse players in the *conjugate-based reformulation* format:

Definition 1.14. $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\theta}^*) = \Big((x^*_{an})_{a \in \mathcal{A}, n \in \mathcal{N}}, (p^*_n)_{n \in \mathcal{N}}, (\theta^*_{an})_{a \in \mathcal{A}, n \in \mathcal{N}}\Big)$ is an equilibrium point of a multistage stochastic MOPEC with risk-averse players in the *conjugate-based reformulation* format if and only if

$$\begin{aligned} (\boldsymbol{x}^*_{a\cdot}, \boldsymbol{\theta}^*_{a\cdot}) \quad & \text{is an optimal solution of} \quad P_a^{\mathcal{RA}-\mathcal{CM}}(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*) \qquad \text{for all } a \in \mathcal{A} \\ p^*_n \quad & \text{belongs to set} \quad Q_n^{VI}(\boldsymbol{x}^*) \qquad \text{for all } n \in \mathcal{N} \end{aligned} \tag{1.65}$$

1.6 Main contributions and goals

The multistage SMOPEC with risk-averse players is a topic of increasing interest in the academic community due to its broad applications in various fields, but there is still limited research on its theoretical properties and practical methods for solving it. The main contributions of this thesis are as follows:

- An investigation into some theoretical properties of the multistage stochastic MOPEC with risk-averse players based on discrete scenario trees for the problem classes introduced in Chapter 2, providing a deeper understanding of this challenging problem.
- The development of three different decomposition algorithms for this problem, based on discrete scenario trees under specific conditions:
  - A player-based decomposition algorithm for the multistage stochastic PNEP with risk-averse players in the *conjugate-based reformulation* on discrete scenario trees.
  - A risk-MOPEC-based decomposition algorithm for the multistage stochastic MOPEC with risk-averse players in the *equilibrium reformulation* on discrete scenario trees.
  - A stage-based decomposition algorithm for the multistage stochastic MOPEC with fixed risk probabilities players in the *equilibrium reformulation* based on discrete scenario trees.
- An extension of the multistage stochastic MOPEC with risk-averse players to a general uncertainty setting and an investigation of sample average approximation methods for constructing an accurate approximation scenario tree for any multistage stochastic MOPEC with risk-averse players under general uncertain stochastic process.

1.7 Outline of the thesis

The present thesis has been structured to achieve the aforementioned objectives. At the beginning, three main types of test instances, namely economic dispatch, capacity expansion, and hydroelectric examples, are introduced in Chapter 2. These instances cover a wide range of applications of stochastic MOPECs and will be utilized in subsequent chapters to evaluate the performance of various algorithms. Following the introduction of the test instances, Chapters 3 to 5 present different decomposition approaches from three distinct perspectives for solving stochastic MOPECs based on discrete scenario trees. Player-based

decomposition approaches are initially discussed in Chapter 3 since they are the most frequently used decomposition approaches for stochastic PNEPs with risk-averse players using a conjugate-based reformulation. We develop a new ADMM-like player-based algorithm for solving stochastic PNEPs. However, player-based algorithms are incapable of solving general stochastic MOPECs, especially when the problem is formulated in *equilibrium reformulation* or when players have a general Nash interaction with each other. Thus, we develop a new primal-MOPEC-dual-risk decomposition approach in Chapter 4, which is a powerful decomposition framework capable of solving more general stochastic MOPECs with risk-averse players. Nonetheless, when the scenario tree is of large scale, the primal-MOPEC subproblem in the primal-MOPEC-dual-risk decomposition approach becomes computationally intractable. Hence, in Chapter 5, we develop a stage-based decomposition approach to further decompose the primal-MOPEC subproblem in Chapter 4 into smaller subproblems indexed by stages. Finally, we extend our approach to stochastic MOPECs with a general distribution setting and its approximation using discrete scenario trees in Chapter 6. This provides us with a general framework for solving stochastic MOPECs with general distributions. The thesis concludes with a summary of the key findings in Chapter **??**. The detailed organization of the thesis is outlined and visualized in Figure 1.4.

1. **Chapter 2:** This chapter introduces the main test problem instances that will be used in the computational experiments of later chapters.

2. **Chapter 3:** This chapter focuses on the player-based decomposition algorithm to solve the stochastic PNEP with risk-averse players, which is based on a conjugate-based reformulation.

3. **Chapter 4:** This chapter presents a risk-MOPEC-based decomposition algorithm to solve the general stochastic MOPEC with risk-averse players, which is based on an equilibrium reformulation.

4. **Chapter 5:** This chapter introduces a stage-based decomposition algorithm to solve the subproblem involved in each iteration of the risk-MOPEC-based decomposition algorithm. This algorithm is designed to handle large-scale general stochastic MOPECs with risk-averse players.

5. **Chapter 6:** This chapter extends the analysis to stochastic MOPEC problems under general probability distributions, introduces a new two-level graph representation for the general stochastic problem, and describes the sample-average approximation (SAA) method to fit a two stochastic MOPEC into a discrete scenario tree-based stochastic MOPEC.

6. **Chapter 7** concludes the thesis project by summarizing its contributions and suggesting future research directions.

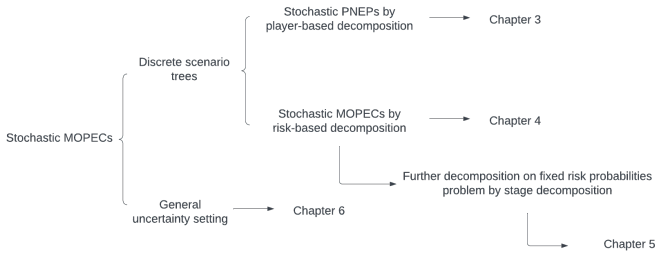


Figure 1.4: Visualization of the organization of the thesis

The aim of this chapter is to present a series of problem instances that will be utilized in the subsequent computational experiments. The structure of these instances is contingent on the specific market constraints and reformulation techniques utilized, particularly when the players exhibit risk-averse behavior. Each problem instance will be thoroughly examined to assess its classification within the problem types introduced in chapter 1, alongside an analysis of its structural components. Furthermore, a comparative analysis of the instances will be conducted to identify their unique characteristics and challenges.

The chapter is structured as follows: Section 2.1 provides the definition of three types of market constraint; Sections 2.2 through 2.4 present comprehensive details of three instances, including the formulations of problems under different risk-averse reformulations and market constraints, and discuss the problem types to which each instance belongs; Section 2.5 provides information about the scenario trees to be used in the computational experiments.

2.1 Different types of equilibrium problems depending on players' interactions and the price demand function

In this chapter, we demonstrate how certain problems arising from the literature fit into our framework. The general format of the equilibrium problem, as expressed in equations (1.12) - (1.15), has numerous applications in various fields.

In the general economic equilibrium problems, the variables that determine the equilibrium can be categorized into the following categories:

p the corresponding price vector for goods

x_a the player's action on the goods

An general economic equilibrium problem could be separated into following different categories according to its market equilibrium constraint $0 \in F(p; \boldsymbol{x}) + \mathcal{N}_K(p)$:

- *Supply demand variational inequality with price unrelated demand*. The first type is defined by the market constraint function $F(p; \boldsymbol{x}) = S(\boldsymbol{x}) - D$, where $S(\boldsymbol{x})$ represents the supply of goods from all players and D represents a given fixed demand in the market. This constraint embodies the classical supply-demand relationship, where the market clearing price is determined by the intersection of the upward sloping

supply curve and the downward sloping demand curve. The market constraint $0 \in S(\boldsymbol{x}) - D + \mathcal{N}_K(p)$ approximates this intersection. In particular, when the feasible region $K = \mathbb{R}^{\alpha}_{+}$, it reduces to the complementarity constraint $0 \leq S(\boldsymbol{x}) - D \perp p \geq 0$, implying that a valid price exists only when supply equals demand, in agreement with classical economic theory.

- *Supply demand variational inequality with price related demand*. The second type is when the market constraint is defined by a function $F(p; \boldsymbol{x}) = S(\boldsymbol{x}) - D(p)$. Compared to the first type, the difference is that the market demand D is not a fixed parameter but a function of price p. This means that the demand in the second type is responsive to the price, whereas in the first type the demand remains unchanged with the price. Both types of market constraints have numerous applications in economics and are dependent on the price sensitivity of the demand. However, the first type is often used when the demand is relatively insensitive to price changes, whereas the second type is more applicable when demand is price sensitive.

- *Price decided by inverse demand function instead of market equilibrium constraint*. The third type will not have the price p variable directly and thus the corresponding market equilibrium constraint $0 \in F(p; \boldsymbol{x}) + \mathcal{N}_K(p)$ doesn't exist. This situation is applied when the market is an oligopoly. In this case, each player is not a price-taker anymore but is powerful enough to change the price according to his action x_a such that price p becomes a function of the combination of all players' actions $p(\boldsymbol{x})$. A common example of this is the Nash-Cournot equilibrium problem, where the objective of player a is defined as $c_a(x_a) - \langle p(\sum_{a \in \mathcal{A}} x_a), x_a \rangle$, where $x_a \in \mathbb{R}^{d_a}$ for $a \in \mathcal{A}$.

2.2 Multistage economic dispatch system

We first consider a general multistage economic dispatch equilibrium problem with multiple players. The general multistage economic dispatch equilibrium is a very general framework that many short-term to medium-term planning problems can be fit in, such as security constrained economic dispatch problem and planning problem in day-ahead market. According to the notation in chapter 1, let $\mathcal{A}$ denote the set of all players and $(\mathcal{N}, \mathcal{E})$ denote the scenario tree that represents the stochastic process of this problem. Additionally, let $\mathcal{I}$ denote the set of different locations. The product (such as electricity or gas) is sold at different locations $i \in \mathcal{I}$.

For each player a at scenario node n, $v_{ain} \in \mathbb{R}$ represents the ending storage of the product of player a at location i. There are lower and upper bounds, $\bar{v}^l_{ai}$ and $\bar{v}^u_{ai}$, respectively,

on the storage level for each node n. The product is sold by player a at location i, and the amount of sales is represented by $s_{ain} \in \mathbb{R}_+$. The production level of the product by player a at location i is represented by $q_{ain} \in \mathbb{R}_+$, and is subject to a fixed capacity constraint W_{ai}. Thus, the variable of each player a at scenario node n, x_{an}, can be decomposed into the following components:

$$x_{an} \quad := \big((v_{ain})_{i\in\mathcal{I}}, (s_{ain})_{i\in\mathcal{I}}, (q_{ain})_{i\in\mathcal{I}}\big) \tag{2.1}$$

with the corresponding feasible region:

$$\mathcal{X}_{an} = \prod_{i\in\mathcal{I}} [\bar{v}^l_{ai}, \bar{v}^u_{ai}] \times \mathbb{R}_+^{|\mathcal{I}|} \times \prod_{i\in\mathcal{I}} [0, W_{ai}]$$

The transition constraint $\mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}$ corresponds to

$$v_{ain} - v_{ain_-} + s_{ain} - q_{ain} = 0, \quad \forall i \in \mathcal{I} \tag{2.2}$$

where

$$\begin{aligned} \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \quad &= \big(v_{ain} - v_{ain_-} + s_{ain} - q_{ain}\big)_{i\in\mathcal{I}} \\ \mathcal{K}_{an} \quad &= \{0\}^{|\mathcal{I}|} \end{aligned} \tag{2.3}$$

Equation (2.2) specifies the storage transition between parent node n_- and node n at location i, where each storage level is decreased by the sales amount s_{ain} to the market and increased by the production amount q_{ain} from player a.

In this type of problem, it is assumed that there will be a price $p_{in} \in \mathbb{R}_+$ for each location $i \in \mathcal{I}$ at scenario node $n \in \mathcal{N}$. The disbenefit $f_{an}(x_{an}; x_{-an}, p_n)$ of each player a at scenario node n corresponds to

$$\sum_{i\in\mathcal{I}} \big(c_{ain}(q_{ain}, s_{ain}, v_{ain}) - p_{in} \cdot s_{ain}\big) \tag{2.4}$$

where $c_{ain} : \mathbb{R}^3 \to \mathbb{R}$ is a convex function that measures the cost of activity of player a because of his action $(v_{ain}, s_{ain}, q_{ain})$ at location i. $\sum_{i\in\mathcal{I}} c_{ain}(q_{ain}, s_{ain}, v_{ain})$ is the summation of these costs at all locations. The benefit from the sales at location i is given by $p_{in} \cdot s_{ain}$. $\sum_{i\in\mathcal{I}} p_{in} \cdot s_{ain}$ is the summation of these benefits at all locations of player a at scenario node n. Combining the above two, the total disbenefits of player a at scenario node n is equation (2.4).

2.2.1 Formulation of player's problem

According to (1.54), the risk-averse optimization problem for each player a given price $(p_{in})_{i\in\mathcal{I},n\in\mathcal{N}}$ is:

For each $a \in \mathcal{A}$ solves

$$
\begin{aligned}
\min_{\boldsymbol{v}_{a\cdot},\boldsymbol{s}_{a\cdot},\boldsymbol{q}_{a\cdot}} \quad & \sum_{i\in\mathcal{I}}\big(c_{ai1}(q_{ai1},s_{ai1},v_{ai1}) - p_{i1}\cdot s_{ai1}\big) \\
& + \rho_{a1}\Bigg(\Big[\sum_{i\in\mathcal{I}}\big(c_{aim}(q_{aim},s_{aim},v_{aim}) - p_{im}\cdot s_{aim}\big) \\
& + \rho_{am}\Big(\Big[\sum_{i\in\mathcal{I}}\big(c_{ail}(q_{ail},s_{ail},v_{ail}) - p_{il}\cdot s_{ail}\big) + \rho_{al}(\cdots)\Big]_{l\in m_+}\Big)\Big]_{m\in 1+}\Bigg) \\
\text{s.t.} \quad & v_{ain} = v_{ain\_} - s_{ain} + q_{ain}, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& q_{ain} \le W_{ai}, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& \bar{v}^l_{ai} \le v_{ain} \le \bar{v}^u_{ai}, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& q_{ain}, s_{ain} \ge 0, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N}
\end{aligned}
$$

where $\boldsymbol{v}_{a\cdot} = (v_{ain})_{i\in\mathcal{I},n\in\mathcal{N}}$ is the vector consisting of all ending storages controlled by player a, $\boldsymbol{s}_{a\cdot} = (s_{ain})_{i\in\mathcal{I},n\in\mathcal{N}}$ is the vector consisting of all sales of player a, and $\boldsymbol{q}_{a\cdot} = (q_{ain})_{i\in\mathcal{I},n\in\mathcal{N}}$ is the vector consisting of all production of player a.

2.2.1.1 Players' problem in the *equilibrium reformulation*

As stated in section 1.5.1, the optimal solution to the optimization problem of each player a can be determined as a component of an equilibrium point of the following equilibrium

problem:

$$
\begin{aligned}
\min_{\boldsymbol{v}_{a\cdot},\boldsymbol{s}_{a\cdot},\boldsymbol{q}_{a\cdot}} \quad & \sum_{i\in\mathcal{I}}\big(c_{ai1}(q_{ai1},s_{ai1},v_{ai1})-p_{i1}\cdot s_{ai1}\big) \\
& +\sum_{n\in 1+}\mu_{an}\cdot\Big[\sum_{i\in\mathcal{I}}\big(c_{ain}(q_{ain},s_{ain},v_{ain})-p_{in}\cdot s_{ain}\big) \\
& +\sum_{m\in n+}\mu_{am}\cdot\Big[\sum_{i\in\mathcal{I}}\big(c_{aim}(q_{aim},s_{aim},v_{aim})-p_{im}\cdot s_{aim}\big)+\cdots\Big]\Big] \\
\text{s.t.} \quad & v_{ain}=v_{ain_-}-s_{ain}+q_{ain}, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& q_{ain}\le W_{ai}, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& \bar{v}^l_{ai}\le v_{ain}\le \bar{v}^u_{ai}, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& q_{ain},s_{ain}\ge 0, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
\max_{\mu_{an_+}\in\mathcal{D}_{an}} \quad & \sum_{m\in n+}\mu_{am}\cdot\Big[\sum_{i\in\mathcal{I}}\big(c_{aim}(q_{aim},s_{aim},v_{aim})-p_{im}\cdot s_{aim}\big) \\
& +\sum_{l\in m_+}\mu_{al}\cdot\Big[\sum_{i\in\mathcal{I}}\big(c_{ail}(q_{ail},s_{ail},v_{ail})-p_{il}\cdot s_{ail}\big)+\cdots\Big]\Big] && \forall n\in\mathcal{N}\backslash\mathcal{L}
\end{aligned}
\tag{2.5}
$$

2.2.1.2 Players' optimization problem in the *conjugate-based reformulation*

According to section 1.5.2, the optimal solution to the optimization problem of each player a is equivalent to the optimal solution of the following optimization problem:

$$
\begin{aligned}
\min_{\boldsymbol{v}_{a\cdot},\boldsymbol{s}_{a\cdot},\boldsymbol{q}_{a\cdot},\boldsymbol{\theta}_{a\cdot}} \quad & \sum_{i\in\mathcal{I}}\big(c_{ai1}(q_{ai1},s_{ai1},v_{ai1})-p_{i1}\cdot s_{ai1}\big)+\theta_{a1} \\
\text{s.t.} \quad & \theta_{an}\ge\sum_{m\in n+}\mu^k_{am}\cdot\big(\sum_{i\in\mathcal{I}}\big(c_{aim}(q_{aim},s_{aim},v_{aim})-p_{im}\cdot s_{aim}\big)+\theta_{am}\big), && \forall k\in\Lambda_{an}, n\in\mathcal{N}\backslash\mathcal{L} \\
& v_{ain}=v_{ain_-}-s_{ain}+q_{ain}, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& q_{ain}\le W_{ai}, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& \bar{v}^l_{ai}\le v_{ain}\le \bar{v}^u_{ai}, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& q_{ain},s_{ain}\ge 0, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
& \theta_{an}=0, \quad \forall n\in\mathcal{L}
\end{aligned}
\tag{2.6}
$$

2.2.2 Three different types of market constraints

In this example, it is assumed that there is an individual market for each location $i\in\mathcal{I}$ at each scenario node $n\in\mathcal{N}$ and the corresponding price is $p_{in}\in\mathbb{R}$. It is also assumed that

there are three types of market constraints according to section 2.1.

2.2.2.1 *Supply demand variational inequality with price unrelated demand (Type I)*

If the market constraint belongs to *Type I* as defined in section 2.1, it is assumed that the demand is price-insensitive and is represented by parameter D_{in}, as there is an individual market for each location i. The demand for each market is satisfied by the sum of the supply from all players at that location $\sum_{a\in\mathcal{A}} s_{ain}$. As a result, the market constraint $0 \in F_n(p_n; \boldsymbol{x}_{\cdot n}) + \mathcal{N}_{K_n}(p_n)$ at scenario node n corresponds to

$$0 \leq \sum_{a\in\mathcal{A}} s_{ain} - D_{in} \perp p_{in} \geq 0, \quad \forall i \in \mathcal{I} \tag{2.7}$$

where the feasible region of price $p_n = (p_{in})_{i\in\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ is

$$K_n = \mathbb{R}_+^{|\mathcal{I}|}$$

and the function

$$F_n(p_n; \boldsymbol{x}_{\cdot n}) = \begin{bmatrix} \sum_{a\in\mathcal{A}} s_{a1n} - D_{1n} \\ \sum_{a\in\mathcal{A}} s_{a2n} - D_{2n} \\ \vdots \\ \sum_{a\in\mathcal{A}} s_{a|\mathcal{I}|n} - D_{|\mathcal{I}|n} \end{bmatrix}$$

The complementarity constraint, as defined in equation (2.7), embodies a fundamental principle in economics, where $\sum_{a\in\mathcal{A}} s_{ain} - D_{in}$ represents the classical supply and demand relationship in economics. This constraint states that a positive price at location i and scenario node n is valid only if the total supply from the players, $\sum_{a\in\mathcal{A}} s_{ain}$, is equal to the demand, D_{in}, from the market. If the total supply exceeds the demand, no price is valid. Moreover, the constraint also assumes that, in a feasible solution, supply must always be greater than demand.

2.2.2.2 *Supply demand variational inequality with price related demand (Type II)*

If the market constraint belongs to *Type II* as defined in section 2.1, it is assumed that the demand is price-sensitive and is a function parameterized by the price p_{in} at each location i, which can be represented by $d_{in}(p_{in})$. In this case, we have the feasible region of price $p_n = (p_{in})_{i\in\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ as

$$K_n = \mathbb{R}_+^{|\mathcal{I}|}$$

and the function

$$F_n(p_n; \boldsymbol{x}_{\cdot n}) = \begin{bmatrix} \sum_{a\in\mathcal{A}} s_{a1n} - d_{1n}(p_{1n}) \\ \sum_{a\in\mathcal{A}} s_{a2n} - d_{2n}(p_{2n}) \\ \vdots \\ \sum_{a\in\mathcal{A}} s_{a|\mathcal{I}|n} - d_{|\mathcal{I}|n}(p_{|\mathcal{I}|n})) \end{bmatrix}$$

As a result, the market constraint $0 \in F_n(p_n; \boldsymbol{x}_{\cdot n}) + \mathcal{N}_{K_n}(p_n)$ at scenario node n corresponds to

$$0 \leq \sum_{a\in\mathcal{A}} s_{ain} - d_{in}(p_{in}) \perp p_{in} \geq 0, \quad \forall i \in \mathcal{I} \tag{2.8}$$

The demand is modeled as a decreasing linear function, $d_{in}(p_{in}) = D_{in} \cdot (1 - \gamma_{in} \cdot (p_{in} - \bar{p}_{in}))$, with slope $-D_{in} \cdot \gamma_{in}$ and intersecting the point $(\bar{p}_{in}, D_{in})$. The fixed parameters $(\gamma_{in})_{i\in\mathcal{I},n\in\mathcal{N}}$ are determined by market properties. $(D_{in})_{i\in\mathcal{I},n\in\mathcal{N}}$ are the same parameters as the price-independent demands in the previous market type, while $(\bar{p}_{in})_{i\in\mathcal{I},n\in\mathcal{N}}$ are the market prices obtained from the previous type's solution under demand $(D_{in})_{i\in\mathcal{I},n\in\mathcal{N}}$.

2.2.2.3 *Price decided by inverse demand function instead of market equilibrium constraint (Type III)*

For this type of problem, it is assumed that there are no price variables, and thus, there are no market constraints. Instead, the price p_{in} is replaced by a function $P_{in}(\boldsymbol{s}_{\cdot in})$ of all players' sales, where $\boldsymbol{s}_{\cdot in} = (s_{ain})_{a\in\mathcal{A}}$. This function is derived from the equality of supply and demand in (2.8):

$$\sum_{a\in\mathcal{A}} s_{ain} - D_{in} \cdot (1 - \gamma_{in} \cdot (p_{in} - \bar{p}_{in})) = 0, \quad \forall i \in \mathcal{I}, \tag{2.9}$$

From equation (2.9) we can derive an inverse function of p_{in}:

$$p_{in} = P_{in}(\boldsymbol{s}_{\cdot in}) = \bar{p}_{in} + \frac{1}{\gamma_{in} \cdot D_{in}}(D_{in} - \sum_{a\in\mathcal{A}} s_{ain}), \quad \forall i \in \mathcal{I}, \ \ \forall n \in \mathcal{N}$$

The equation of supply equaling demand (2.9) is equivalent to the market constraint (2.8) when $p_{in} > 0$. This type of model differs from the previous two in that the market is an oligopoly, where each player is no longer a price-taker but instead has the ability to influence the price through their action s_{ain}. Furthermore, it is noteworthy that in this type of market constraint, the disbenefit $f_{an}(x_{an}; x_{-an}, p_n)$ for each player a at scenario node n

corresponds to

$$\sum_{i\in\mathcal{I}}\Big(c_{ain}(q_{ain},s_{ain},v_{ain})-\big[\bar{p}_{in}+\frac{1}{\gamma_{in}\cdot D_{in}}(D_{in}-\sum_{a\in\mathcal{A}}s_{ain})\big]\cdot s_{ain}\Big) \tag{2.10}$$

2.2.3 Classification of the problem in different formulations and market constraints

As discussed, we have two types of formulation of player's risk-averse problem:

- *Equilibrium reformulation*
- *Conjugate-based reformulation*

with three different equilibrium market constraints:

- *Supply demand variational inequality with price unrelated demand (Type I)*
- *Supply demand variational inequality with price related demand (Type II)*
- *Price decided by inverse demand function instead of market equilibrium constraint (Type III)*

We begin with the simple situation where all players are risk-neutral. In cases where the market constraint is *Type I* or *Type II*, the transition constraint $(v_{ain}-v_{ain\_}+s_{ain}-q_{ain})_{i\in\mathcal{I}}$ doesn't involve market prices or other players' variables. As a result, market prices only appear in the players' objective function, leading to a price-incentive Nash equilibrium problem (PNEP). In this case, each player's problem will exhibit Nash behavior. However, when the market constraint is a *Type III* market constraint, the objective function of each player will include other players' variables, resulting in the entire equilibrium problem being a pure Nash equilibrium problem (NEP).

When considering the risk-averse players, the situation becomes more complicated. Initially, we consider the risk-averse problem formulated in the *equilibrium reformulation* (2.5). If the market constraint is a *type I* or *type II* market constraint, the feasible regions of players will not be impacted by market prices, other players' variables, or risk probability vector y, as the risk probabilities will only exist in the objective function of each player's problem. Consequently, the whole problem is a typical MOPEC with each player exhibiting Nash behavior under these market constraints. If the market constraint is of *Type III*, there will be no market price variable p, and each player will only be affected by risk probability vector y in the objective function. In this case ,the problem is also a NEP.

When the risk-averse formulation is based on the *conjugate-based reformulation* (2.6), the situation is very different from that of the *equilibrium reformulation*. In case the market

constraint is of *Type I* or *Type II*, the market price p_{in} will apper in the constraint $\theta_{an} \geq \sum_{m \in n+} \mu_{am}^k \cdot \left(\sum_{i \in \mathcal{I}} \left(c_{aim}(q_{aim}, s_{aim}, v_{aim}) - p_{im} \cdot s_{aim}\right) + \theta_{am}\right)$ of each player's optimization problem. This means that the optimization probelm of each player will exhibit general Nash equilibrium behavior. However, since the feasible region and the objective function will only be affected by the market price, these problems are PNEPs. If the market constraint is of *Type III*, there will be no market price variable p_{in}, and each player will have a cost-to-go constraint as follows: $\theta_{an} \geq \sum_{m \in n+} \mu_{am}^k \cdot \left(\sum_{i \in \mathcal{I}} \left(c_{aim}(q_{aim}, s_{aim}, v_{aim}) - \left[\bar{p}_{in} + \frac{1}{\gamma_{in} \cdot D_{in}} (D_{in} - \sum_{a \in \mathcal{A}} s_{ain})\right] \cdot s_{aim} \right) + \theta_{am} \right)$. In this case, the problem is a GNEP.

The following Table 2.1 provides a conclusion of the type of problem to which each part belongs:

Market constraint Type	Risk-averse formulation	Equilibrium problem type
Type I	*Conjugate-based*	PNEP with each player GNE
Type I	*Equilibrium*	MOPEC with each player NE
Type II	*Conjugate-based*	PNEP with each player GNE
Type II	*Equilibrium*	MOPEC with each player NE
Type III	*Conjugate-based*	GNEP
Type III	*Equilibrium*	NEP

Table 2.1: Problem types of different risk-averse formulations and market constraint types of general dispatch equilibrium

2.3 Multistage capacity expansion equilibrium problems

In the second example we consider a multistage capacity expansion equilibrium problem. Compared to the general dispatch problem, such a model is typically employed in the context of long-term power system markets, including the planning of markets over several years. In this framework, the players are denoted by $a \in \mathcal{A}$ and the locations are denoted by $i \in \mathcal{I}$. Additionally, the stochastic process is characterized by scenario tree $(\mathcal{N}, \mathcal{E})$.

In this example, there exists an individual market for each location, and each player generates all their supply at a single location and sells it to different markets i. The amount of sales made by player a to market i at scenario node n is denoted by $s_{ain} \in \mathbb{R}_+$. The total production capacity of player a at scenario node n is denoted by the variable $C_{an} \in \mathbb{R}_+$. The total sales $\sum_{i \in \mathcal{I}} s_{ain}$ of each player a at scenario node n is subject to the constraint: $C_{an} - \sum_{i \in \mathcal{I}} \psi_n \cdot s_{ain} \geq 0$, where ψ_n is the degradation coefficient from production to sales at scenario node n. A special property of this problem is that the capacity C_{an} can be expanded by player from stage to stage. The capacity investment made by player a at scenario node n to expand their generation capacity from C_{an} to C_{am}, where m can be any child node of

node n, is denoted as $u_{an} \in \mathbb{R}_+$. The constraint that describes the capacity expansion of each player a at scenario node $n \in \mathcal{N}\backslash\mathcal{L}$ corresponds to: $C_{am} = C_{an} + u_{an}, \forall m \in n_+$. Based on these, the variable of each player a at scenario node $n \in \mathcal{N}\backslash\mathcal{L}$, x_{an}, can be decomposed into the following components:

$$x_{an} \quad := \big((s_{ain})_{i\in\mathcal{I}}, C_{an}, u_{an}\big) \tag{2.11}$$

with the corresponding feasible region:

$$\mathcal{X}_{an} = \mathbb{R}_+^{|\mathcal{I}|+2}$$

The variable x_{an} of each player a at scenario node $n \in \mathcal{L}$ can be decomposed into the following components:

$$x_{an} \quad := \big((s_{ain})_{i\in\mathcal{I}}, C_{an}\big) \tag{2.12}$$

with the corresponding feasible region:

$$\mathcal{X}_{an} = \mathbb{R}_+^{|\mathcal{I}|+1}$$

The transition constraint $\mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}$ corresponds to

$$\begin{aligned} C_{an} - C_{an_-} - u_{an_-} &= 0 \\ C_{an} - \sum_{i\in\mathcal{I}} \psi_n \cdot s_{ain} &\geq 0 \end{aligned} \tag{2.13}$$

where

$$\begin{aligned} \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \quad &= \begin{bmatrix} C_{an} - C_{an_-} - u_{an_-} \\ C_{an} - \sum_{i\in\mathcal{I}} \psi_n \cdot s_{ain} \end{bmatrix} \\ \mathcal{K}_{an} \quad &= \{0\} \times \mathbb{R}_+ \end{aligned} \tag{2.14}$$

In this type of problem, it is assumed that there is a price $p_{in} \in \mathbb{R}_+$ for each location $i \in \mathcal{I}$ at scenario node $n \in \mathcal{N}$. $G_{an} : \mathbb{R}^{|\mathcal{I}|+1} \to \mathbb{R}$ is a smooth convex function that measures the cost of player a's activity and $G_{an}\big((s_{ain})_{i\in\mathcal{I}}, C_{an}\big)$ represents the cost of player a at scenario node n. $I_{an} : \mathbb{R} \to \mathbb{R}$ is a convex function that measures the cost of capacity expansion made by player a at scenario node n. The benefit from sales at location i is given by $p_{in} \cdot s_{ain}$ and $\sum_{i\in\mathcal{I}} p_{in} \cdot s_{ain}$ represents the summation of these benefits from all locations of player a at scenario node n. Define $\boldsymbol{s}_{a\cdot n} = (s_{ain})_{i\in\mathcal{I}}$. Combining all the disbenefits described above, the

total disbenefit $f_{an}(x_{an}; x_{-an}, p_n)$ of each player a at scenario node $n \in \mathcal{N}\backslash\mathcal{L}$ corresponds to

$$G_{an}(\boldsymbol{s}_{a\cdot n}, C_{an}) + I_{an}(u_{an}) - \sum_{i\in\mathcal{I}} p_{in} \cdot s_{ain} \tag{2.15}$$

and the disbenefit $f_{an}(x_{an}; x_{-an}, p_n)$ of each player a at scenario node $n \in \mathcal{L}$ corresponds to

$$G_{an}(\boldsymbol{s}_{a\cdot n}, C_{an}) - \sum_{i\in\mathcal{I}} p_{in} \cdot s_{ain} \tag{2.16}$$

2.3.1 Formulation of player's problem

According to (1.54), the risk-averse optimization problem for each player a given price $(p_{in})_{i\in\mathcal{I},n\in\mathcal{N}}$ is:

$$\begin{aligned}
\min_{\boldsymbol{u}_{a\cdot},\boldsymbol{s}_{a\cdot},\boldsymbol{C}_{a\cdot}} \quad & G_{a1}(\boldsymbol{s}_{a\cdot 1}, C_{a1}) + I_{a1}(u_{a1}) - \sum_{i\in\mathcal{I}} p_{i1} \cdot s_{ai1} \\
& + \rho_{a1}\Bigg(\Big[G_{an}(\boldsymbol{s}_{a\cdot n}, C_{an}) + I_{an}(u_{an}) - \sum_{i\in\mathcal{I}} p_{in} \cdot s_{ain} \\
& + \rho_{an}\Big(\Big[G_{am}(\boldsymbol{s}_{a\cdot m}, C_{am}) + I_{am}(u_{am}) - \sum_{i\in\mathcal{I}} p_{im} \cdot s_{aim} \\
& + \rho_{am}(\cdots + \rho_{al}([G_{ah}(\boldsymbol{s}_{a\cdot h}, C_{ah}) - \sum_{i\in\mathcal{I}} p_{ih} \cdot s_{aih}]_{h\in l_+}) \dots) \Big]_{m\in n_+} \Big) \Big]_{n\in 1+} \Bigg) \\
\text{s.t.} \quad & C_{an} = C_{an_-} + u_{an_-}, && \forall n \in \mathcal{N}\backslash\mathcal{L} \\
& C_{an} - \sum_{i\in\mathcal{I}} \psi_n \cdot s_{ain} \geq 0, && \forall n \in \mathcal{N} \\
& u_{an}, C_{an} \geq 0, && \forall n \in \mathcal{N} \\
& s_{ain} \geq 0, && \forall i \in \mathcal{I},\ \ n \in \mathcal{N}
\end{aligned}$$

where $\boldsymbol{u}_{a\cdot} = (u_{an})_{n\in\mathcal{N}\backslash\mathcal{L}}$ is the vector consisting of all capacity investment of player a, $\boldsymbol{s}_{a\cdot} = (s_{ain})_{i\in\mathcal{I},n\in\mathcal{N}}$ is the vector consisting of all production of player a, and $\boldsymbol{C}_{a\cdot} = (C_{an})_{n\in\mathcal{N}}$ is the vector consisting of all production capacity of player a. Similar to the multistage dispatch system equilibrium problem, the objective of each player is to minimize the total disbenefits.

2.3.1.1 Players' problem in the *equilibrium reformulation*

As stated in section 1.5.1, the optimal solution to the optimization problem of each player a is equivalent to the partial component of equilibrium point of the following equilibrium

problem:

$$
\begin{aligned}
\min_{\boldsymbol{u}_{a\cdot},\boldsymbol{s}_{a\cdot},\boldsymbol{C}_{a\cdot}} \quad & G_{a1}(\boldsymbol{s}_{a\cdot 1}, C_{a1}) + I_{a1}(u_{a1}) - \sum_{i\in\mathcal{I}} p_{i1}\cdot s_{ai1} \\
& + \sum_{n\in 1_+} \mu_{an}\cdot\Bigg(G_{an}(\boldsymbol{s}_{a\cdot n}, C_{an}) + I_{an}(u_{an}) - \sum_{i\in\mathcal{I}} p_{in}\cdot s_{ain} \\
& + \sum_{m\in n_+} \mu_{am}\cdot\Bigg(G_{am}(\boldsymbol{s}_{a\cdot m}, C_{am}) + I_{am}(u_{am}) - \sum_{i\in\mathcal{I}} p_{im}\cdot s_{aim} \\
& + \cdots + \sum_{h\in l_+} \mu_{ah}\cdot\Big(G_{ah}(\boldsymbol{s}_{a\cdot h}, C_{ah}) - \sum_{i\in\mathcal{I}} p_{ih}\cdot s_{aih}\Big)\Bigg)\Bigg) \\
\text{s.t.} \quad & C_{an} = C_{an_-} + u_{an_-}, && \forall n\in\mathcal{N}\backslash\mathcal{L} \\
& C_{an} - \sum_{i\in\mathcal{I}} \psi_n\cdot s_{ain} \geq 0, && \forall n\in\mathcal{N} \\
& u_{an}, C_{an} \geq 0, && \forall n\in\mathcal{N} \\
& s_{ain} \geq 0, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N} \\
\max_{\mu_{an_+}\in\mathcal{D}_{an}} \quad & \sum_{m\in n+} \mu_{am}\cdot\Big[G_{am}(\boldsymbol{s}_{a\cdot m}, C_{am}) + I_{am}(u_{am}) - \sum_{i\in\mathcal{I}} p_{im}\cdot s_{aim} \\
& + \sum_{l\in m_+} \mu_{al}\cdot\Big[G_{al}(\boldsymbol{s}_{a\cdot l}, C_{al}) + I_{al}(u_{al}) - \sum_{i\in\mathcal{I}} p_{il}\cdot s_{ail} + \cdots\Big]\Big], && \forall n\in\mathcal{N}\backslash\mathcal{L}
\end{aligned}
\tag{2.17}
$$

2.3.1.2 Players' optimization problem in the *conjugate-based reformulation*

According to section 1.5.2, we will have the optimal solution of optimization problem of each player a is equivalent to the optimal solution of the following optimization problem:

$$
\begin{aligned}
\min_{\boldsymbol{u}_{a\cdot},\boldsymbol{s}_{a\cdot},\boldsymbol{C}_{a\cdot},\boldsymbol{\theta}_{a\cdot}} \quad & G_{a1}(\boldsymbol{s}_{a\cdot 1}, C_{a1}) + I_{a1}(u_{a1}) - \sum_{i\in\mathcal{I}} p_{i1}\cdot s_{ai1} + \theta_{a1} \\
\text{s.t.} \quad & \theta_{an} \geq \sum_{m\in n+} \mu_{am}^{k}\cdot\big(G_{am}(\boldsymbol{s}_{a\cdot m}, C_{am}) + I_{am}(u_{am}) \\
& \qquad - \sum_{i\in\mathcal{I}} p_{im}\cdot s_{aim} + \theta_{am}\big), && \forall k\in\Lambda_{an}, n\in\mathcal{N}\backslash\big(\mathcal{N}(T-1)\cup\mathcal{N}(T)\big) \\
& \theta_{an} \geq \sum_{m\in n+} \mu_{am}^{k}\cdot\big(G_{am}(\boldsymbol{s}_{a\cdot m}, C_{am}) - \sum_{i\in\mathcal{I}} p_{im}\cdot s_{aim}\big), && \forall k\in\Lambda_{an}, n\in\mathcal{N}(T-1) \\
& C_{an} = C_{an_-} + u_{an_-}, && \forall n\in\mathcal{N}\backslash\mathcal{L} \\
& C_{an} - \sum_{i\in\mathcal{I}} \psi_n\cdot s_{ain} \geq 0, && \forall n\in\mathcal{N} \\
& u_{an}, C_{an} \geq 0, && \forall n\in\mathcal{N} \\
& s_{ain} \geq 0, && \forall i\in\mathcal{I},\ \ n\in\mathcal{N}
\end{aligned}
\tag{2.18}
$$

2.3.2 Three different types of market constraints

2.3.2.1 *Supply demand variational inequality with price unrelated demand (Type I)*

Similar to the dispatch example, if the demand at location i and scenario node n is price-insensitive, it could be represented as parameter D_{in}. The market constraint $0\in F_n(p_n;\boldsymbol{x}_{\cdot n}) + \mathcal{N}_{K_n}(p_n)$ at scenario node n corresponds to

$$
0 \leq \sum_{a\in\mathcal{A}} s_{ain} - D_{in} \perp p_{in} \geq 0, \quad \forall i\in\mathcal{I} \tag{2.19}
$$

The complementarity constraint (2.19) states that a positive price at location i and scenario node n is valid only if the total supply from players, denoted by $\sum_{a\in\mathcal{A}} s_{ain}$, is equal to the demand D_{in} from the market. If the total supply exceeds the demand, no price is valid. The constraint also assumes that, if the solution is feasible, supply must always exceed demand.

2.3.2.2 *Supply demand variational inequality with price related demand (Type II)*

In *Type II* problem, the demand is price-sensitive and market constraint $0 \in F_n(p_n; \boldsymbol{x}_{\cdot n}) + \mathcal{N}_{K_n}(p_n)$ at scenario node n corresponds to

$$0 \leq \sum_{a \in \mathcal{A}} s_{ain} - d_{in}(p_{in}) \perp p_{in} \geq 0, \quad \forall i \in \mathcal{I} \tag{2.20}$$

The demand function $d_{in}(p_{in})$ is a decreasing linear function $d_{in}(p_{in}) = D_{in} \cdot (1 - \gamma_{in} \cdot (p_{in} - \bar{p}_{in}))$ with slope $-D_{in} \cdot \gamma_{in}$ and intersecting the point $(\bar{p}_{in}, D_{in})$. The parameters $(\gamma_{in})_{i \in \mathcal{I}, n \in \mathcal{N}}$ are determined by market properties, while $(D_{in})_{i \in \mathcal{I}, n \in \mathcal{N}}$ are the same parameters as the price-independent demands in the previous type. The values of $(\bar{p}_{in})_{i \in \mathcal{I}, n \in \mathcal{N}}$ are the solution market prices of the previous type under demand $(D_{in})_{i \in \mathcal{I}, n \in \mathcal{N}}$.

2.3.2.3 *Price decided by inverse demand function instead of market equilibrium constraint (Type III)*

In *Type III*, the price p_{in} is replaced by a function $P_{in}(\boldsymbol{s}_{\cdot in})$ of all players' sales. This function is also derived from the equality of supply and demand constraint (2.9). Thus, the inverse demand function of price is:

$$p_{in} = P_{in}(\boldsymbol{s}_{\cdot in}) = \bar{p}_{in} + \frac{1}{\gamma_{in} \cdot D_{in}}(D_{in} - \sum_{a \in \mathcal{A}} s_{ain}), \quad \forall i \in \mathcal{I}, \ \ \forall n \in \mathcal{N}$$

The disbenefit $f_{an}(x_{an}; x_{-an}, p_n)$ of each player a at scenario node $n \in \mathcal{N} \backslash \mathcal{L}$ corresponds to

$$G_{an}(\boldsymbol{s}_{a \cdot n}, C_{an}) + I_{an}(u_{an}) - \sum_{i \in \mathcal{I}} \big[\bar{p}_{in} + \frac{1}{\gamma_{in} \cdot D_{in}}(D_{in} - \sum_{a \in \mathcal{A}} s_{ain})\big] \cdot s_{ain} \tag{2.21}$$

and disbenefit $f_{an}(x_{an}; x_{-an}, p_n)$ of each player a at scenario node $n \in \mathcal{L}$ corresponds to

$$G_{an}(\boldsymbol{s}_{a \cdot n}, C_{an}) - \sum_{i \in \mathcal{I}} \big[\bar{p}_{in} + \frac{1}{\gamma_{in} \cdot D_{in}}(D_{in} - \sum_{a \in \mathcal{A}} s_{ain})\big] \cdot s_{ain} \tag{2.22}$$

2.3.3 Classification of the problem in different formulations and market constraints

As discussed, we have two types of formulation of player's risk-averse problem:

- *Equilibrium reformulation*
- *Conjugate-based reformulation*

with three different equilibrium market constraints:

- *Supply demand variational inequality with price unrelated demand (Type I)*
- *Supply demand variational inequality with price related demand (Type II)*
- *Price decided by inverse demand function instead of market equilibrium constraint (Type III)*

When all players are risk-neutral and the market constraint is of *Type I* or *Type II*, the constraints $C_{an} - C_{an\_} - u_{an\_} = 0$ and $C_{an} - \sum_{i\in\mathcal{I}} \psi_n \cdot s_{ain} \geq 0$ do not take into account market prices or other players' variables. Therefore, the problem can be classified a PNEP. On the other hand, when the market constraint is of *Type III*, the objective of each player includes other players' variables, resulting in the problem being classified as NEP.

When players are risk-averse, we first consider the risk-averse problem formulated in the *equilibrium reformulation* (2.5). If the market constraint is a *type I* or *type II* market constraint, similar to the previous example, the entire problem is a typical MOPEC, with each player exhibiting Nash behavior under these market constraints. If the market constraint is of *type III*, the problem becomes a NEP, as there is no market price variable p and each player is only impacted by risk probability vector y in the objective function.

When the risk-averse formulation is based on the *conjugate-based reformulation* (2.6), the situation differs significantly from the *equilibrium reformulation*. In case the market constraint is a *Type I* or *Type II* market constraint, each player's optimization problem will exhibit general Nash equilibrium behavior with market price p_{in} appearing in the cost-to-go constraint. Nevertheless, since the feasible regions and the objective functions will only be affected by the market price, these problems are PNEPs. On the other hand, if the market constraint is of *type III*, the problem becomes a GNEP, similar to the previous example, as there is no market price.

The following Table 2.2 summarizes the type of problem for each part.

Market constraint Type	Risk-averse formulation	Equilibrium problem type
Type I	*Conjugate-based*	PNEP with each player GNE
Type I	*Equilibrium*	MOPEC with each player NE
Type II	*Conjugate-based*	PNEP with each player GNE
Type II	*Equilibrium*	MOPEC with each player NE
Type III	*Conjugate-based*	GNEP
Type III	*Equilibrium*	NEP

Table 2.2: Problem types of different risk-averse formulations and market constraint types of general capacity expansion equilibrium

Remark 2.1. Upon comparing Table 2.1 and Table 2.2, it can be observed that there are no discernible differences regarding the distinct types of equilibrium problems for the dispatch

and capacity expansion examples. However, the feasible region of each player in the dispatch equilibrium problem is a compact convex set, whereas the feasible region of each player in the capacity expansion equilibrium is an unbounded convex set. This dissimilarity in feasible region properties can give rise to distinct properties of the equilibrium points in general equilibrium problems. Notably, the latter is considerably more challenging to solve.

2.4 Multistage hydro electricity system with water network

The last example that we include in this thesis is Example 1 in [38]. This example considers a water network comprised of three hydroelectric generators and one leaving node, as depicted in Figure 2.1. Let $\mathcal{I} = \{0, 1, 2, 3\}$ denote the set of locations of these hydroelectric generators and leaving node 0. The electricity generation and storage at locations 1 and 3 are controlled by firm A, whereas the electricity generation and storage at locations 2 are controlled by firm B. Let $\mathcal{A} = \{A, B\}$ denote the set of players and let $\mathcal{I}_a$ denote the set of locations that are controlled by player $a \in \mathcal{A}$. Thus, in this example, we have $\mathcal{I}_A = \{1, 3\}$ and $\mathcal{I}_B = \{2\}$. The arrows in Figure 2.1 represent the directions of water flows between locations. Let ξ_i denote the set of locations with incoming water flow from location i and let $\tilde{\xi}_i$ denote the set of locations with outgoing water flow to location i.

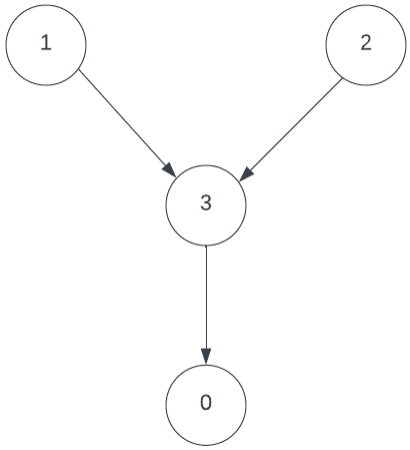


Figure 2.1: Water network with hydroelectric production at location 1, 2, and 3 and an leaving node 0

At each scenario node n, the water flow travels from location i to location j is denoted by

the variable $u_{ijn} \in \mathbb{R}_+$, and the water level at location i is denoted by the variable $v_{in} \in \mathbb{R}_+$. There are lower and upper bounds, $\bar{v}^l_{ai}$ and $\bar{v}^u_{ai}$, respectively, on the water level for each node n. Let $\boldsymbol{u}_{i\cdot n} := (u_{ijn})_{j\in\xi_i}$ denote the vector consisting of all water flows that travel out of location i at scenario node n. Player a controls the water flows $\boldsymbol{u}_{i\cdot n}$ and water level v_{in} if $i \in \mathcal{I}_a$. Based on these variables, the variable of each player a at scenario node $n \in \mathcal{N}$, x_{an}, can be decomposed into the following components:

$$x_{an} \quad := \big((u_{ijn})_{i\in\mathcal{I}_a, j\in\xi_i}, (v_{in})_{i\in\mathcal{I}_a}\big) \tag{2.23}$$

with the corresponding feasible region:

$$\mathcal{X}_{an} = \mathbb{R}_+^{\sum_{i\in\mathcal{I}_a}|\xi_i|} \times \prod_{i\in\mathcal{I}_a} [\bar{v}^l_{in}, \bar{v}^u_{in}]$$

At location i and scenario node n, the parameter $T_{ij} \in \mathbb{R}$ allows for dynamic consumption of the water flow between different locations, and ϖ_{in} denotes the random water inflows from nature. The water level is increased by incoming water flows from other locations $\sum_{j\in\tilde{\xi}_i} T_{ji}u_{jin}$, as well as random inflows supplies ϖ_{in}, but is decreased by outgoing water flows $\sum_{j\in\xi_i} T_{ij}u_{ijn}$. This transition is represented by the constraint:

$$v_{in_-} + \sum_{j\in\tilde{\xi}_i} T_{ji}u_{jin} - \sum_{j\in\xi_i} T_{ij}u_{ijn} + \varpi_{in} - v_{in} \geq 0$$

The inequality in this constraint allows for the free disposal of water levels to maintain them in a feasible range. Thus, the transition constraint $\mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}$ corresponds to:

$$v_{in_-} + \sum_{j\in\tilde{\xi}_i} T_{ji}u_{jin} - \sum_{j\in\xi_i} T_{ij}u_{ijn} + \varpi_{in} - v_{in} \geq 0, \quad \forall i \in \mathcal{I}_a$$

where

$$\begin{aligned} \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) &= \big(v_{in_-} + \sum_{j\in\tilde{\xi}_i} T_{ji}u_{jin} - \sum_{j\in\xi_i} T_{ij}u_{ijn} + \varpi_{in} - v_{in}\big)_{i\in\mathcal{I}_a} \\ \mathcal{K}_{an} &= \mathbb{R}_+^{|\mathcal{I}_a|} \end{aligned} \tag{2.24}$$

The cost of electricity generation and water level at location i and scenario node n is measureby by a continuously differetiable convex function $C_{in} : \mathbb{R}^{|\xi_i|+1} \to \mathbb{R}$, so that the cost at locations i and scenario node n equals to $C_{in}(\boldsymbol{u}_{i\cdot n}, v_{in})$. The selling benefit at location i and scenario node n is $p_n \cdot g_{in}(\boldsymbol{u}_{i\cdot n})$, where $p_n \in \mathbb{R}$ is the price determined by the market

at scenario node n and $g_{in}(\boldsymbol{u}_{i\cdot n})$ is the amount of electricity generated from water flow between locations. The function $g_{in} : \mathbb{R}^{|\xi_i|} \to \mathbb{R}$ is a concave function. Thus, the disbenefit $f_{an}(x_{an}; x_{-an}, p_n)$ of each player a at scenario node $n \in \mathcal{N}$ corresponds to

$$\sum_{i\in\mathcal{I}} \big(C_{in}(\boldsymbol{u}_{i\cdot n}, v_{in}) - p_n \cdot g_{in}(\boldsymbol{u}_{i\cdot n})\big) \tag{2.25}$$

2.4.1 Formulation of player's problem

According to (1.54), the risk-averse optimization problem for each player a given price $(p_n)_{n\in\mathcal{N}}$ is:

$$\begin{aligned}
\min_{(\boldsymbol{u}_{i\cdot},\boldsymbol{v}_{i\cdot})_{i\in\mathcal{I}_a}} \quad & \sum_{i\in\mathcal{I}_a} \big(C_{i1}(\boldsymbol{u}_{i\cdot 1}, v_{i1}) - p_1 \cdot g_{i1}(\boldsymbol{u}_{i\cdot 1})\big) \\
& + \rho_{a1}\Bigg(\Big[\sum_{i\in\mathcal{I}_a} \big(C_{in}(\boldsymbol{u}_{i\cdot n}, v_{in}) - p_n \cdot g_{in}(\boldsymbol{u}_{i\cdot n})\big) \\
& + \rho_{an}\Big(\Big[\sum_{i\in\mathcal{I}_a} \big(C_{im}(\boldsymbol{u}_{i\cdot m}, v_{im}) - p_m \cdot g_{im}(\boldsymbol{u}_{i\cdot m})\big) + \rho_{am}(\cdots)\Big]_{m\in n_+}\Big)\Big]_{n\in 1+}\Bigg) \\
\text{s.t.} \quad & v_{in_-} + \sum_{j\in\tilde{\xi}_i} T_{ji} u_{jin} - \sum_{j\in\xi_i} T_{ij} u_{ijn} + \varpi_{in} - v_{in} \geq 0, && \forall i \in \mathcal{I}_a,\ \ n \in \mathcal{N} \\
& \bar{v}^l_{in} \leq v_{in} \leq \bar{v}^u_{in}, && \forall i \in \mathcal{I}_a,\ \ n \in \mathcal{N} \\
& u_{ijn} \geq 0 && \forall i \in \mathcal{I}_a,\ \ j \in \xi_i,\ \ n \in \mathcal{N}
\end{aligned}$$

where $\boldsymbol{u}_{i\cdot} = (u_{ijn})_{j\in\xi_i, n\in\mathcal{N}}$ is the vector consisting of all water flow originating from location $i \in \mathcal{I}$, $\boldsymbol{v}_{i\cdot} = (v_{in})_{n\in\mathcal{N}}$ is the vector consisting of all water level states at location i.

2.4.1.1 Players' problem in the *equilibrium reformulation*

As stated in section 1.5.1, the optimal solution to the optimization problem of each player a is equivalent to the partial component of equilibrium point of the following equilibrium

problem:

$$
\begin{aligned}
\min_{(\boldsymbol{u}_{i\cdot\cdot},\boldsymbol{v}_{i\cdot})_{i\in\mathcal{I}_a}} \quad & \sum_{i\in\mathcal{I}_a}\big(C_{i1}(\boldsymbol{u}_{i\cdot 1},v_{i1})-p_1\cdot g_{i1}(\boldsymbol{u}_{i\cdot 1})\big) \\
& +\sum_{n\in 1+}\mu_{an}\cdot\Big[\sum_{i\in\mathcal{I}_a}\big(C_{in}(\boldsymbol{u}_{i\cdot n},v_{in})-p_n\cdot g_{in}(\boldsymbol{u}_{i\cdot n})\big) \\
& +\sum_{m\in n+}\mu_{am}\cdot\Big[\sum_{i\in\mathcal{I}_a}\big(C_{im}(\boldsymbol{u}_{i\cdot m},v_{im})-p_m\cdot g_{im}(\boldsymbol{u}_{i\cdot m})\big)+\cdots\Big]\Big] \\
\text{s.t.} \quad & v_{in_-}+\sum_{j\in\tilde{\xi}_i}T_{ji}u_{jin}-\sum_{j\in\xi_i}T_{ij}u_{ijn}+\varpi_{in}-v_{in}\geq 0, && \forall i\in\mathcal{I}_a,\ \ n\in\mathcal{N} \\
& \bar{v}_{ai}^{\,l}\leq v_{ain}\leq\bar{v}_{ai}^{\,u}, && \forall i\in\mathcal{I}_a,\ \ n\in\mathcal{N} \\
& u_{ijn}\geq 0 && \forall i\in\mathcal{I}_a,\ \ j\in\xi_i,\ \ n\in\mathcal{N} \\
\max_{\mu_{an_+}\in\mathcal{D}_{an}} \quad & \sum_{m\in n+}\mu_{am}\cdot\Big[\sum_{i\in\mathcal{I}_a}\big(C_{im}(\boldsymbol{u}_{i\cdot m},v_{im})-p_m\cdot g_{im}(\boldsymbol{u}_{i\cdot m})\big) \\
& +\sum_{l\in m_+}\mu_{al}\cdot\Big[\sum_{i\in\mathcal{I}_a}\big(C_{il}(\boldsymbol{u}_{i\cdot l},v_{il})-p_l\cdot g_{il}(\boldsymbol{u}_{i\cdot l})\big)+\cdots\Big]\Big] && \forall n\in\mathcal{N}\backslash\mathcal{L}
\end{aligned}
\tag{2.26}
$$

2.4.1.2 Players' optimization problem in the *conjugate-based reformulation*

According to section 1.5.2, we will have the optimal solution of optimization problem of each player a is equivalent to the optimal solution of the following optimization problem:

$$
\begin{aligned}
\min_{(\boldsymbol{u}_{i\cdot\cdot},\boldsymbol{v}_{i\cdot})_{i\in\mathcal{I}_a},\boldsymbol{\theta}_{a\cdot}} \quad & \sum_{i\in\mathcal{I}_a}\big(C_{i1}(\boldsymbol{u}_{i\cdot 1},v_{i1})-p_1\cdot g_{i1}(\boldsymbol{u}_{i\cdot 1})\big)+\theta_{a1} \\
\text{s.t.} \quad & \theta_{an}\geq\sum_{m\in n+}\mu_{am}^{k}\cdot\big(\sum_{i\in\mathcal{I}_a}\big(C_{im}(\boldsymbol{u}_{i\cdot m},v_{im})-p_m\cdot g_{im}(\boldsymbol{u}_{i\cdot m})\big)+\theta_{am}\big), && \forall k\in\Lambda_{an},n\in\mathcal{N}\backslash\mathcal{L} \\
& v_{in_-}+\sum_{j\in\tilde{\xi}_i}T_{ji}u_{jin}-\sum_{j\in\xi_i}T_{ij}u_{ijn}+\varpi_{in}-v_{in}\geq 0, && \forall i\in\mathcal{I}_a,\ \ n\in\mathcal{N} \\
& \bar{v}_{ai}^{\,l}\leq v_{ain}\leq\bar{v}_{ai}^{\,u}, && \forall i\in\mathcal{I}_a,\ \ n\in\mathcal{N} \\
& u_{ijn}\geq 0 && \forall i\in\mathcal{I}_a,\ \ j\in\xi_i,\ \ n\in\mathcal{N} \\
& \theta_{an}=0,\quad \forall n\in\mathcal{L}
\end{aligned}
\tag{2.27}
$$

2.4.2 Three different types of market constraints

2.4.2.1 *Supply demand variational inequality with price unrelated demand (Type I)*

Similar to the previous two examples, for a *Type I* problem, denote $D_n\in\mathbb{R}_+$ be the fixed parameter representing the demand of electricity from public market at scenario node n,

and the feasible region of price $p_n \in \mathbb{R}$ is

$$K_n = \mathbb{R}_+$$

and the function

$$F_n(p_n; \boldsymbol{x}_{\cdot n}) = \sum_{a \in \mathcal{A}} \big(\sum_{i \in \mathcal{I}_a} g_{in}(\boldsymbol{u}_{i \cdot n}) \big) - D_n$$

In this case, the general market constraint is equivalent to the following market complementarity constraint for each location $i \in \mathcal{I}$ and scenario node $n \in \mathcal{N}$:

$$0 \leq \sum_{a \in \mathcal{A}} \big(\sum_{i \in \mathcal{I}_a} g_{in}(\boldsymbol{u}_{i \cdot n}) \big) - D_n \perp p_n \geq 0, \quad \forall n \in \mathcal{N} \tag{2.28}$$

$D_n \in \mathbb{R}_+$ is a fixed parameter meaning that the demand is not related to the market price p_n. The term $\sum_{a \in \mathcal{A}} \big(\sum_{i \in \mathcal{I}_a} g_{in}(\boldsymbol{u}_{i \cdot n}) \big)$ regards the total supply sold from all players $a \in \mathcal{A}$ into the market. Thus, $\sum_{a \in \mathcal{A}} \big(\sum_{i \in \mathcal{I}_a} g_{in}(\boldsymbol{u}_{i \cdot n}) \big) - D_n$ represents the classical supply minus demand in economics.

2.4.2.2 *Supply demand variational inequality with price related demand (Type II)*

In *Type II* problem, we have the price-sensitive demand

$$d_n(p_n) = D_n \cdot \big(1 - \gamma_n \cdot (p_n - \bar{p}_n) \big)$$

for each scenario node $n \in \mathcal{N}$. The function d_n is a decreasing linear function with slope $-D_n \cdot \gamma_n$ that will cross through the point $(\bar{p}_n, D_n)$. Here $\{\gamma_n\}_{n \in \mathcal{N}}$ are parameters that are decided by the market property, $\{D_n\}_{n \in \mathcal{N}}$ are the same parameters as the price unrelated demands in previous type and $\{\bar{p}_n\}_{n \in \mathcal{N}}$ are the solution market prices of the previous type under demand $\{D_n\}_{n \in \mathcal{N}}$. In this case, the market complementarity constraint for each location $i \in \mathcal{I}$ at scenario node $n \in \mathcal{N}$ is as follows:

$$0 \leq \sum_{a \in \mathcal{A}} \big(\sum_{i \in \mathcal{I}_a} g_{in}(\boldsymbol{u}_{i \cdot n}) \big) - D_n \cdot (1 - \gamma_n \cdot (p_n - \bar{p}_n)) \perp p_n \geq 0, \quad \forall n \in \mathcal{N} \tag{2.29}$$

2.4.2.3 *Price decided by inverse demand function instead of market equilibrium constraint (Type III)*

The price p_n is replaced by a function $P_n(\boldsymbol{u}_{\cdot n})$ of all players' actions, where $\boldsymbol{u}_{\cdot n} = (\boldsymbol{u}_{ijn})_{i,j\in\mathcal{I}}$. This function is from the supply equals demand equality:

$$\sum_{a\in\mathcal{A}}\Big(\sum_{i\in\mathcal{I}_a} g_{in}(\boldsymbol{u}_{i\cdot n})\Big) - D_n\cdot(1-\gamma_n\cdot(p_n-\bar{p}_n)) = 0, \quad \forall n\in\mathcal{N}, \tag{2.30}$$

from which we can have an inverse function of p_{in}:

$$p_n = \bar{p}_n + \frac{1}{\gamma_n\cdot D_n}\Big(D_n - \sum_{a\in\mathcal{A}}\Big(\sum_{i\in\mathcal{I}_a} g_{in}(\boldsymbol{u}_{i\cdot n})\Big)\Big), \quad \forall n\in\mathcal{N}$$

The supply equals to demand equality (2.30) is equal to the market constraint (2.29) when $p_n > 0$. Like before, a big difference between this type of model and the previous twos is that the market here is an oligopolistic market.

2.4.3 Classification of the problem in different formulations and market constraints

As discussed, we have two types of formulation of player's risk-averse problem:

- *Equilibrium reformulation*
- *Conjugate-based reformulation*

with three different equilibrium market constraints:

- *Supply demand variational inequality with price unrelated demand (Type I)*
- *Supply demand variational inequality with price related demand (Type II)*
- *Price decided by inverse demand function instead of market equilibrium constraint (Type III)*

Because each player a has the constraints $v_{in_-} + \sum_{j\in\tilde{\xi}_i} T_{ji}u_{jin} - \sum_{j\in\xi_i} T_{ij}u_{ijn} + \varpi_{in} - v_{in} \geq 0$ that will involve other players' variables $\{u_{jin} | j\in\tilde{\xi}_i\}$. Thus for any type of reformulations, the problem with Type I or Type II market constraint is a MOPEC with general Nash behavior, and the problem in Type III is a GNEP.

The following Table 2.3 provides a conclusion of the type of problem to which each part belongs:

Market constraint Type	Risk-averse formulation	Equilibrium problem type
Type I	*Conjugate-based*	PNEP with each player GNE
Type I	*Equilibrium*	MOPEC with each player GNE
Type II	*Conjugate-based*	PNEP with each player GNE
Type II	*Equilibrium*	MOPEC with each player GNE
Type III	*Conjugate-based*	GNEP
Type III	*Equilibrium*	GNEP

Table 2.3: Problem types of different risk-averse formulations and market constraint types of hydro electricity system equilibrium with water network

2.5 Data for testing scenario trees

The scenario trees we used for the test instances are presented in Table 2.4. We mainly used two scenario trees for our computational experiments in this study. Both scenario trees will have 4 stages. The difference is that in scenario tree 1 each parent node that is not a leaf node will have 3 children nodes, instead in scenario tree 2 each parent node that is not a leaf node will have 5 children nodes. The conditional probabilities of children nodes are the same if their parent nodes are located in the same time stage.

scenario tree	$\lvert\mathcal{T}\rvert$	size of children node	$\lvert\mathcal{N}\rvert$
tree1	3	3	12
tree2	4	3	40
tree3	4	5	156

Table 2.4: Test scenario trees

2.6 Repository

This section elucidates the procedure for obtaining access to each test instance mentioned in this thesis. The entirety of the associated files can be found at the web address: https://www.cs.wisc.edu/ ferris/stocheq. It is noteworthy that the test instances encompassed within this chapter pertain exclusively to problems featuring a singular market demand type, and these instances will be utilized in subsequent chapters to evaluate the efficacy of the various proposed algorithms. The generation of said test instances can be accomplished through the utilization of the provided source codes. The inclusive content of the files encompasses the following items: [add specific details of the files].

- Main file: genprob.gms

- Three files correspond to three different instances for generating problem data and problems.: soldispatch.gms, solcapex.gms, solhydro.gms
- Scenario tree data files: tree1.gdx, tree2.gdx, tree3.gdx

Below is a simple example of implementation:

Listing 2.1: Multiple optimizations with equilibrium constraints(MOPEC)

```
gams genprob.gms --data=tree1 --problem=dispatch --dataType=1 --
   randomSeed=1 --quad=1 --demandType=1 --lambda=0.1 --varphi=0.4
```

Option	Description	Default
data	the scenario tree for the stochastic problem. Candidates: tree1, tree2, tree3	tree1
problem	problem type: three types of problem that could be chosen: dispatch, capex, hydro	dispatch
dataType	two different ways for generation of random data. Candidates: 1, 2	1
randomSeed	random seeds set to generate data	1
quad ϵ	quadratic term of player's objective	1
demandType	Type of market demand function. Candidates: 1, 2, 3	1
lambda	λ value of risk measure $\overline{CVaR}(\lambda, \varphi)$	0.1
varphi	φ value of risk measure $\overline{CVaR}(\lambda, \varphi)$	0.5

Table 2.5: General options

Upon execution of the aforementioned code, a singular problem instance will be generated and subsequently resolved using the default PATH solver, employing its default configuration. As a result of this process, a gdx file will be generated, containing the output described below.

Output	Description
modelStatus	model status for the problem
solverStatus	solver status for the problem
problem instances data	all problem data that is generated by randomseed

Table 2.6: GDX Output

The parameters in gdx file can be gained by the folloing terminal command:

Listing 2.2: gdx output

```
gdxdump %problem%_tree%data%_dT%dataType%_rs%
   randomSeed%_qd%quad%_mkt%demandType%_la%lambda%
   _vphi%varphi%.gdx
```

The primary purpose of the GDX output file is to convey the status of the model and solver utilized by the default PATH solver. Additionally, it serves as a convenient means to provide problem data from the model, facilitating the regeneration of the data for the user's convenience.

3 ADMM-BASED DECOMPOSITION METHOD FOR MULTISTAGE STOCHASTIC EQUILIBRIUM WITH RISK-AVERSE PLAYERS

The main contribution of this chapter is the development of an ADMM-based algorithm to solve the stochastic PNEP with risk-averse players using a *conjugate-based reformulation* as described in chapter 1. The algorithm decomposes the original problem into collections of subproblems indexed by players and solves them iteratively. This decomposition approach is motivated by the fact that in PNEP, the system can be solved by solving a series of smaller player's optimization problems when the market price $\boldsymbol{p}$ is given. This property allows us to tackle the computational intractability of the entire PNEP using traditional methods like the PATH solver. This important property still remains in the stochastic setting with risk-averse players using a *conjugate-based reformulation*. However, the proposed approach cannot handle the stochastic PNEP with risk-averse players using an *equilibrium reformulation* or the more general stochastic MOPEC with risk-averse players, especially when each player has general Nash behavior interaction with each other, because players' problems cannot be solved independently with given $\boldsymbol{p}$. To address these scenarios, the risk-based decomposition algorithm presented in chapter 4 can solve more general stochastic MOPECs with risk-averse players, and the decomposition algorithm discussed in chapter 5 can be applied to solve the large-scale subproblem encountered in the algorithm from chapter 4.

The chapter is organized as follows: Section 3.1 briefly recalled problem addressed in chapter 1 and discusses its challenges when solving the entire system. Due to the hardness of these problems when solving as a whole system, section 3.2 focuses on player-based decomposition methods and presents three different algorithms based on this approach to solve the stochastic PNEP with risk-averse players using a *conjugate-based reformulation*. The first two algorithms are from the existing literature and are discussed in subsection 3.2.1, while the third algorithm is the new ADMM-based algorithm proposed in this chapter. The differences between these three algorithms are explained, highlighting the advantages of our algorithm in addressing a broader range of problems. Section 3.3 presents the numerical experiment results obtained using the proposed ADMM-based algorithm on the economic dispatch and capacity expansion examples introduced in chapter 2. Finally, section 3.4 provides a conclusion for this chapter, summarizing the key findings and contributions.

3.1 Problem statement

Solving stochastic equilibrium problems with risk-averse players in a general format poses significant challenges. In the deterministic case, player-based decomposition approaches, such as the Jacobi-based algorithm or Gauss-Seidel decomposition algorithm, are commonly employed to solve Nash equilibrium problems due to the advantageous inner structure of such problems. However, achieving convergence in the general case can be difficult, particularly when players exhibit general Nash behavior interactions. Therefore, this chapter focuses on studying the stochastic PNEP, where interactions between players are absent when the price $\boldsymbol{p}$ is given by the market constraint. Despite the lack of general interactions, this problem formulation finds wide applications in practice and aligns naturally with the concept of player-based decomposition. Notably, examples like the dispatch example and hydroelectric example discussed in chapter 2, featuring Type I and Type II market constraints, respectively, fall within this problem category.

Let us now recall the definition of stochastic MOPECs with risk-averse players using a *conjugate-based reformulation*. The reason for focusing solely on the *conjugate-based reformulation* rather than the *equilibrium reformulation* is that the latter involves the introduction of the variable $\boldsymbol{\mu}$ into each problem, rendering it no longer a PNEP even if the original problem is a PNEP. In the case of the stochastic PNEP defined on a scenario tree $(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ represents the set of scenario nodes and $\mathcal{E}$ represents the set of edges, the cost function f_{an} and the constraint function $\mathcal{G}_{an}$ for player a at scenario node n are assumed to be independent of other players' actions x_{-an}. Under this assumption, if player a is risk-averse and the player's problem is formulated using the *conjugate-based reformulation* introduced in chapter 1, the optimization problem for player a given price $\boldsymbol{p}$ can be expressed as follows:

$$
\begin{aligned}
\min_{\boldsymbol{x}_{a\cdot}, \boldsymbol{\theta}_{a\cdot}} \quad & f_{a1}(x_{a1}; p_1) + \theta_{a1} \\
\text{s.t.} \quad & \theta_{an} \geq \sum_{m \in n+} \mu_{am}^k \cdot \big(f_{am}(x_{am}; p_m) + \theta_{am}\big), \quad \forall k \in \Lambda_{an}, n \in \mathcal{N} \backslash \mathcal{L} \\
& \mathcal{G}_{an}(x_{an_-}, x_{an}; p_n) \in \mathcal{K}_{an}, \qquad \forall n \in \mathcal{N} \\
& x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N} \\
& \theta_{an} = 0, \quad \forall n \in \mathcal{L}
\end{aligned}
\tag{3.1}
$$

Without loss of generality, we also assume the following assumptions hold for the rest of this chapter.

Assumption 3.1. *For each $n \in \mathcal{N}$, constraint $\mathcal{G}_{an}(x_{an_-}, x_{an}; p_n) \in \mathcal{K}_{an}$ corresponds to the con-*

straints

$$\begin{aligned} \mathcal{G}^1_{an}(x_{an_-}, x_{an}; p_n) &= 0 \\ \mathcal{G}^2_{an}(x_{an_-}, x_{an}; p_n) &\leq 0 \end{aligned} \tag{3.2}$$

where

$$\mathcal{G}_{an}(x_{an_-}, x_{an}; p_n) = \begin{bmatrix} \mathcal{G}^1_{an}(x_{an_-}, x_{an}; p_n) \\ \mathcal{G}^2_{an}(x_{an_-}, x_{an}; p_n) \end{bmatrix}$$

and

$$\mathcal{K}_{an} = \{0\}^{\psi^1_{an}} \times \mathbb{R}^{\psi^2_{an}}_+$$

Here ψ^1_{an} and ψ^2_{an} are positive integers and represent the dimension of constraints $\mathcal{G}^1_{an}$ and $\mathcal{G}^2_{an}$ and $\psi^1_{an} + \psi^2_{an} = \psi_{an}$.

Assumption 3.2. *For each $n \in \mathcal{N}$, the market equilibrium constraint $0 \in F_n(p_n; \boldsymbol{x}_{\cdot n}) + \mathcal{N}_{K_n}(p_n)$ corresponds to the complementarity constraint*

$$0 \leq F_n(p_n; \boldsymbol{x}_{\cdot n}) \perp p_n \geq 0 \tag{3.3}$$

where

$$K_n = \mathbb{R}^{\alpha_n}_+$$

The assumptions stated in (3.2) and (3.3) simply require that the cone $\mathcal{K}_{an}$ and K_n respectively are positive orthants in their corresponding spaces. These are prevalent assumptions in various practical applications and cover a significant portion of the equilibrium problems encountered in different fields.

3.2 Player-based decomposition algorithms

This section introduces a player-based algorithm for solving the stochastic PNEP with risk-averse players using a *conjugate-based reformulation*. Equation (3.1) demonstrates that a stochastic PNEP with risk-averse players, when formulated using a *conjugate-based reformulation*, belongs to a type of PNEP where players have no direct interaction with each other once the market price $\boldsymbol{p}$ is given, as illustrated in Figure 3.1.

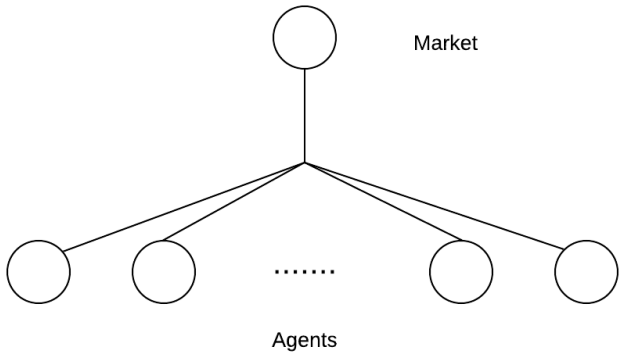


Figure 3.1: Equilibrium network with a centralized market

In the player-based decomposition method, each player's optimization problem (3.1) is not further subdivided. As a result, we can simplify the player's optimization problem and present the algorithms in a more concise manner. The general form of the player's optimization problem in the PNEP can be expressed as follows:

$$\begin{aligned} P_a^{\mathcal{SP}}(\boldsymbol{p}): \quad \min_{\boldsymbol{x}_a, \boldsymbol{\theta}_a} \quad & f_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) \\ \text{s.t.} \quad & h_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) = 0 \\ & g_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) \leq 0 \\ & \boldsymbol{x}_a \in \mathcal{X}_a, \boldsymbol{\theta}_a \in \Theta_a \end{aligned} \tag{3.4}$$

Here $f_a : \mathbb{R}^{d_a+|\mathcal{N}|+\alpha} \to \mathbb{R}$ is a twice continuously differentiable function and $f_a(\cdot; \boldsymbol{p})$ is convex for each given $\boldsymbol{p}$. g_a, h_a are differentiable and $g_a(\cdot; \boldsymbol{p}), h_a(\cdot; \boldsymbol{p})$ are convex with given $\boldsymbol{p}$.

Consider the stochastic problem (3.1), the cost function $f_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p})$ in (3.4) corresponds to

$$f_{a1}(x_{a1}; p_1) + \theta_{a1} \tag{3.5}$$

the equality constraint $h_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) = 0$ corresponds to

$$\mathcal{G}_{an}^1(x_{an_-}, x_{an}; p_n) = 0, \quad \forall n \in \mathcal{N} \tag{3.6}$$

and the inequality $g_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) \leq 0$ corresponds to

$$\begin{aligned} \mathcal{G}_{an}^2(x_{an_-}, x_{an}; p_n) \leq 0, \quad & \forall n \in \mathcal{N} \\ \sum_{m \in n+} \mu_{am}^k \cdot \big(f_{am}(x_{am}; p_m) + \theta_{am}\big) - \theta_{an} \leq 0, \quad & \forall k \in \Lambda_{an}, n \in \mathcal{N} \backslash \mathcal{L} \end{aligned} \tag{3.7}$$

The market constraint of equilibrium problem is

$$0 \leq F(\boldsymbol{p}; \boldsymbol{x}) \perp \boldsymbol{p} \geq 0 \tag{3.8}$$

which corresponds to

$$0 \leq F_n(p_n; \boldsymbol{x}_{\cdot n}) \perp p_n \geq 0, \quad \forall n \in \mathcal{N} \tag{3.9}$$

The structure of the PNEP problem (3.4), (3.8) is illustrated in Figure 3.1. One key characteristic of the PNEP is that the price directly influences each player, and when the price $\boldsymbol{p}$ is fixed, each player's problem becomes completely independent and can be solved in parallel. This implies that once the market price solution is known, the actions of other players can be obtained by solving a collection of small problems, and these subproblems can be solved in parallel. Hence, an intuitive approach for an iterative and decomposable algorithm is to initialize each iteration with a known market price and obtain the corresponding players' actions by solving each subproblem. Subsequently, the market price is updated using the new information obtained from all players. The underlying concept of this algorithmic approach is depicted in Figure 3.2:

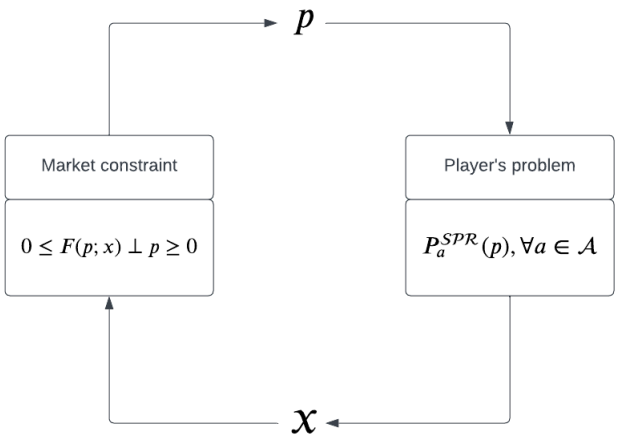


Figure 3.2: Player-based Algorithm Framework

However, many decomposition approaches rely on special properties of the problem to guarantee convergence. One of these important properties is the monotonicity and Lipschitz continuity of the reformulated mixed complementarity problem $0 \leq G(x) \perp x \geq 0$. Unfortunately, stochastic PNEPs with risk-averse players usually lack a monotone pseudo-gradient mapping, even if the original problem with risk-neutral players is monotone. Thus, most convergent algorithms utilizing the framework in Figure 3.1 are not applicable when players are risk-averse. Consequently, research on utilizing the framework in Figure

3.1 without requiring the monotonicity assumption is very limited. In this regard, we will illustrate two previous works that could be utilized to solve this type of problem. Following the introduction of these methods, we will discuss the restricted assumptions required by each approach. This discussion motivates us to develop a new ADMM-based decomposition approach that could be applied in a more general setting, which we will discuss in the next section.

3.2.1 Two decomposition methods to solve PNEP (3.4) with VI (3.8)

Method I: Newton-based decomposition method [11]

In this section, we present the first algorithm introduced in [11]. The authors of [11] assume that each player's optimization problem has a unique minimizer for any given $\boldsymbol{p}$. Let us define:

$$\big(\boldsymbol{x}_a(\boldsymbol{p}), \boldsymbol{\theta}_a(\boldsymbol{p})\big) = \arg\min_{\boldsymbol{x}_a, \boldsymbol{\theta}_a} \Big\{ f_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) : h_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) = 0, g_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) \leq 0, \boldsymbol{x}_a \in \mathcal{X}_a, \boldsymbol{\theta}_a \in \Theta_a \Big\}$$

Assume if we can get the algebraic representation of solution mapping $\big(\boldsymbol{x}_a(\boldsymbol{p}), \boldsymbol{\theta}_a(\boldsymbol{p})\big)$, we could replace $\boldsymbol{x}$ by $\boldsymbol{x}(\boldsymbol{p})$ in VI constraint (3.8) and define $R(\boldsymbol{p}) = F(\boldsymbol{p}; \boldsymbol{x}(\boldsymbol{p}))$ and have

$$0 \leq R(\boldsymbol{p}) \perp \boldsymbol{p} \geq 0 \tag{3.10}$$

From (3.8) we could get the equilibrium price $\boldsymbol{p}^*$ directly. However, it is usually impossible to compute the solution mapping $\boldsymbol{x}(\boldsymbol{p})$ and $R(\boldsymbol{p})$ in practical problem. To overcome this difficulty, an alternative is that we could try to get the a linear approximation of $R(\boldsymbol{p})$ around point $\bar{\boldsymbol{p}}$, which can be denoted by $\hat{R}(\boldsymbol{p}; \bar{\boldsymbol{p}})$. Such kind of approximation can be constructed as follows:

Assume that the solution mapping $\big(\boldsymbol{x}_a(\boldsymbol{p}), \boldsymbol{\theta}_a(\boldsymbol{p})\big)$ is continuously differentiable, we can denote the gradient of $\boldsymbol{x}_a(\boldsymbol{p})$ and $\boldsymbol{\theta}_a(\boldsymbol{p})$ as $\nabla_{\boldsymbol{p}} \boldsymbol{x}_a(\boldsymbol{p})$ and $\nabla_{\boldsymbol{p}} \boldsymbol{\theta}_a(\boldsymbol{p})$, respectively. Furthermore, let $\nabla_{\boldsymbol{p}} \boldsymbol{x}_a(\bar{\boldsymbol{p}}) = \nabla_{\boldsymbol{p}} \boldsymbol{x}_a(\boldsymbol{p})|_{\boldsymbol{p}=\bar{\boldsymbol{p}}}$ and $\nabla_{\boldsymbol{p}} \boldsymbol{\theta}_a(\bar{\boldsymbol{p}}) = \nabla_{\boldsymbol{p}} \boldsymbol{\theta}_a(\boldsymbol{p})|_{\boldsymbol{p}=\bar{\boldsymbol{p}}}$ represent the value of $\nabla_{\boldsymbol{p}} \boldsymbol{x}_a(\boldsymbol{p})$ and $\nabla_{\boldsymbol{p}} \boldsymbol{\theta}_a(\boldsymbol{p})$, respectively, at the point $\bar{\boldsymbol{p}}$. This allows us to obtain the following expression:

$$\nabla_{\boldsymbol{p}} R(\bar{\boldsymbol{p}}) = \nabla_{\boldsymbol{p}} \boldsymbol{x}(\bar{\boldsymbol{p}}) \cdot \nabla_{\boldsymbol{x}} F(\bar{\boldsymbol{p}}; \bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{p}} F(\bar{\boldsymbol{p}}; \bar{\boldsymbol{x}}) \tag{3.11}$$

where $\bar{\boldsymbol{x}} = \boldsymbol{x}(\boldsymbol{p})|_{\boldsymbol{p}=\bar{\boldsymbol{p}}}$.

We can construct a linear approximation of $R(\boldsymbol{p})$ based on (3.11) and have

$$\begin{aligned}\hat{R}(\boldsymbol{p};\bar{\boldsymbol{p}}) &= R(\bar{\boldsymbol{p}}) + \langle \nabla_{\boldsymbol{p}} R(\bar{\boldsymbol{p}}), \boldsymbol{p}-\bar{\boldsymbol{p}}\rangle \\ &= F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \langle \nabla_{\boldsymbol{p}}\boldsymbol{x}(\bar{\boldsymbol{p}})\cdot\nabla_{\boldsymbol{x}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{p}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}), \boldsymbol{p}-\bar{\boldsymbol{p}}\rangle\end{aligned} \tag{3.12}$$

Assume the matrix $\nabla_{\boldsymbol{p}}\boldsymbol{x}(\bar{\boldsymbol{p}})\cdot\nabla_{\boldsymbol{x}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{p}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}})$ is invertible, the solution of $0 \le \hat{R}(\boldsymbol{p};\bar{\boldsymbol{p}}) \perp \boldsymbol{p} \ge 0$ is

$$\boldsymbol{p} = \max\left\{0, \bar{\boldsymbol{p}} - \left(\nabla_{\boldsymbol{p}}\boldsymbol{x}(\bar{\boldsymbol{p}})\cdot\nabla_{\boldsymbol{x}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{p}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}})\right)^{-1}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}})\right\}$$

which results in a new market price $\boldsymbol{p}$, and the steps can be repeated iteratively to obtain the solution. The algorithmic steps for this process are presented in Algorithm 1.

Algorithm 1 Newton-based decomposition method

1: **Input and Initialization** Choose $\boldsymbol{p}^1 \in \mathbb{R}^{\alpha}$. Set $r = 1$.

2: **Step 1: Subproblems.** For each $a \in \mathcal{A}$ solve

$$\min_{\boldsymbol{x}_a,\boldsymbol{\theta}_a}\left\{f_a(\boldsymbol{x}_a,\boldsymbol{\theta}_a;\boldsymbol{p}^r) : h_a(\boldsymbol{x}_a,\boldsymbol{\theta}_a;\boldsymbol{p}^r) = 0, g_a(\boldsymbol{x}_a,\boldsymbol{\theta}_a;\boldsymbol{p}^r) \le 0, \boldsymbol{x}_a \in \mathcal{X}_a, \boldsymbol{\theta}_a \in \Theta_a\right\}$$

Let $(\boldsymbol{x}_a^r, \boldsymbol{\theta}_a^r) = (\boldsymbol{x}_a(\boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^r}, \boldsymbol{\theta}_a(\boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^r})$ denote the unique minimum in this problem given $\boldsymbol{p}^r$. Compute the gradients $\nabla_{\boldsymbol{p}}\boldsymbol{x}_a(\boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^r}$ and let $\nabla_{\boldsymbol{p}}\boldsymbol{x}_a^r = \nabla_{\boldsymbol{p}}\boldsymbol{x}_a(\boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^r}$. Define $\boldsymbol{x}^r = \begin{bmatrix}\boldsymbol{x}_1^r \\ \vdots \\ \boldsymbol{x}_{|\mathcal{A}|}^r\end{bmatrix}$ and $\nabla_{\boldsymbol{p}}\boldsymbol{x}^r = \begin{bmatrix}\nabla_{\boldsymbol{p}}\boldsymbol{x}_1^r \\ \vdots \\ \nabla_{\boldsymbol{p}}\boldsymbol{x}_{|\mathcal{A}|}^r\end{bmatrix}$.

3: **Step 2: Stopping Test.** If stopping criteria is met, Stop.

4: **Step 3: Update price.** Define $\nabla_{\boldsymbol{x}}F(\boldsymbol{p}^r;\boldsymbol{x}^r) = \nabla_{\boldsymbol{x}}F(\boldsymbol{p};\boldsymbol{x})|_{\boldsymbol{p}=\boldsymbol{p}^r,\boldsymbol{x}=\boldsymbol{x}^r}$ and $\nabla_{\boldsymbol{p}}F(\boldsymbol{p}^r;\boldsymbol{x}^r) = \nabla_{\boldsymbol{p}}F(\boldsymbol{p};\boldsymbol{x})|_{\boldsymbol{p}=\boldsymbol{p}^r,\boldsymbol{x}=\boldsymbol{x}^r}$.

$$\boldsymbol{p}^{r+1} = \max\left\{0, \boldsymbol{p}^r - \left(\nabla_{\boldsymbol{p}}\boldsymbol{x}^r\cdot\nabla_{\boldsymbol{x}}F(\boldsymbol{p}^r;\boldsymbol{x}^r) + \nabla_{\boldsymbol{p}}F(\boldsymbol{p}^r;\boldsymbol{x}^r)\right)^{-1}F(\boldsymbol{p}^r;\boldsymbol{x}^r)\right\}$$

5: **Step 4: Loop.** Set $r := r+1$ and go back to Step 1.

The practical performance of Algorithm 1 in equilibrium problems is not affected by the absence of a convergence result. However, to ensure the algorithm's effectiveness, we need to satisfy three primary assumptions as outlined in the above steps:

1. The solution mapping $\left(\boldsymbol{x}_a(\boldsymbol{p}), \boldsymbol{\theta}_a(\boldsymbol{p})\right)$ of the subproblem is a continuously differentiable function with respect to $\boldsymbol{p}$.

2. For any $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{x}} = \boldsymbol{x}(\boldsymbol{p})|_{\boldsymbol{p}=\bar{\boldsymbol{p}}}$, $\nabla_{\boldsymbol{p}}\boldsymbol{x}(\bar{\boldsymbol{p}}) \cdot \nabla_{\boldsymbol{x}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{p}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}})$ must be invertible and have nonnegative entries, otherwise we cannot guarantee a new price after this iteration.

3. $F(\boldsymbol{p};\boldsymbol{x})$ is continuously differentiable.

It is important to note that these three assumptions do not hold universally, and they may not be satisfied in a wide range of equilibrium problems. For instance, if the subproblem is a linear programming problem, the first assumption mentioned above will not be satisfied. Therefore, the applicability of Algorithm 1 may be limited in such cases.

Method II: Smoothing decomposition of equilibrium problem [12]

The approach proposed in [12] offers a solution for cases in which the solution mapping $(\boldsymbol{x}_a(\boldsymbol{p}), \boldsymbol{\theta}_a(\boldsymbol{p}))$ is not a continuously differentiable function with respect to $\boldsymbol{p}$. Instead of solving the original optimization problem (3.4) with respect to the market price $\boldsymbol{p}$, each player's optimization problem is modified by removing the inequality constraint $g_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p})$ and replacing it with a barrier term with scaling coefficient ε. Additionally, a regularization term $\frac{\varepsilon\cdot\omega}{2}(\|\boldsymbol{x}_a\|_2^2 + \|\boldsymbol{\theta}_a\|_2^2)$ is added to the objective function of player a. Thus, the modified optimization problem for player a is as follows:

$$
\begin{aligned}
P_a^{\varepsilon,\omega}(\boldsymbol{p}): \quad \min_{\boldsymbol{x}_a,\boldsymbol{\theta}_a} \quad & f_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) - \varepsilon \sum_i \log\{-g_{ai}(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p})\} + \mathcal{I}_{\mathcal{X}_a}(\boldsymbol{x}_a) \\
& + \mathcal{I}_{\Theta_a}(\boldsymbol{\theta}_a) + \frac{\varepsilon\cdot\omega}{2}(\|\boldsymbol{x}_a\|_2^2 + \|\boldsymbol{\theta}_a\|_2^2) \\
\text{s.t.} \quad & h_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) = 0
\end{aligned}
\tag{3.13}
$$

here $\mathcal{I}_{\mathcal{X}_a}$ and $\mathcal{I}_{\Theta_a}$ are the indicator functions on sets $\mathcal{X}_a$ and Θ_a, respectively. The optimization problem $P_a^{\varepsilon,\omega}(\boldsymbol{p})$ is obtained by removing the inequality constraint $g_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p})$ from the original optimization problem (3.4) and adding a barrier term with scaling coefficient ε and a regularization term $\frac{\varepsilon\cdot\omega}{2}(\|\boldsymbol{x}_a\|_2^2 + \|\boldsymbol{\theta}_a\|_2^2)$ to the player a's optimization problem. The solution mapping $(\boldsymbol{x}_a^{\varepsilon,\omega}(\boldsymbol{p}), \boldsymbol{\theta}_a^{\varepsilon,\omega}(\boldsymbol{p}))$ is a continuously differentiable function with the help of the barrier term and regularization term. Let $(\boldsymbol{x}_a^{\varepsilon,\omega}(\boldsymbol{p}), \boldsymbol{\theta}_a^{\varepsilon,\omega}(\boldsymbol{p}))$ denote the solution of problem $P_a^{\varepsilon,\omega}(\boldsymbol{p})$ with given $\boldsymbol{p}$. Let $\nabla_{\boldsymbol{p}}\boldsymbol{x}_a^{\varepsilon,\omega}(\boldsymbol{p}), \nabla_{\boldsymbol{p}}\boldsymbol{\theta}_a^{\varepsilon,\omega}(\boldsymbol{p})$ denote the gradient of $\boldsymbol{x}_a^{\varepsilon,\omega}(\boldsymbol{p}), \boldsymbol{\theta}_a^{\varepsilon,\omega}(\boldsymbol{p})$, respectively, and $\nabla_{\boldsymbol{p}}\boldsymbol{x}_a^{\varepsilon,\omega}(\boldsymbol{p})|_{\boldsymbol{p}=\bar{\boldsymbol{p}}}, \nabla_{\boldsymbol{p}}\boldsymbol{\theta}_a^{\varepsilon,\omega}(\boldsymbol{p})|_{\boldsymbol{p}=\bar{\boldsymbol{p}}}$ denote the value of $\nabla_{\boldsymbol{p}}\boldsymbol{x}_a^{\varepsilon,\omega}(\boldsymbol{p}), \nabla_{\boldsymbol{p}}\boldsymbol{\theta}_a^{\varepsilon,\omega}(\boldsymbol{p})$ at point $\bar{\boldsymbol{p}}$. The function $R^{\varepsilon,\omega}(\boldsymbol{p}) = F(\boldsymbol{p}; \boldsymbol{x}^{\varepsilon,\omega}(\boldsymbol{p}))$ can be used to obtain the gradient of $R^{\varepsilon,\omega}$ at point $\bar{\boldsymbol{p}}$, which is given by:

$$
\nabla_{\boldsymbol{p}}R^{\varepsilon,\omega}(\bar{\boldsymbol{p}}) = \nabla_{\boldsymbol{p}}\boldsymbol{x}^{\varepsilon,\omega}(\bar{\boldsymbol{p}}) \cdot \nabla_{\boldsymbol{x}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{p}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) \tag{3.14}
$$

where $\bar{\boldsymbol{x}} = \boldsymbol{x}^{\varepsilon,\omega}(\boldsymbol{p})|_{\boldsymbol{p}=\bar{\boldsymbol{p}}}$.

We can construct a linear approximation of $R^{\varepsilon,\omega}(\boldsymbol{p})$ at the point $\bar{\boldsymbol{p}}$. Denote this linear approximation by $\hat{R}^{\varepsilon,\omega}(\boldsymbol{p};\bar{\boldsymbol{p}})$, then we will have

$$\begin{aligned}\hat{R}^{\varepsilon,\omega}(\boldsymbol{p};\bar{\boldsymbol{p}}) &= R^{\varepsilon,\omega}(\bar{\boldsymbol{p}}) + \langle \nabla_{\boldsymbol{p}} R^{\varepsilon,\omega}(\bar{\boldsymbol{p}}), \boldsymbol{p}-\bar{\boldsymbol{p}}\rangle \\ &= F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \langle \nabla_{\boldsymbol{p}}\boldsymbol{x}^{\varepsilon,\omega}(\bar{\boldsymbol{p}})\cdot\nabla_{\boldsymbol{x}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{p}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}), \boldsymbol{p}-\bar{\boldsymbol{p}}\rangle\end{aligned} \tag{3.15}$$

Assume the matrix $\nabla_{\boldsymbol{p}}\boldsymbol{x}^{\varepsilon,\omega}(\bar{\boldsymbol{p}})\cdot\nabla_{\boldsymbol{x}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{p}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}})$ is invertible, the solution of $0 \le \hat{R}^{\varepsilon,\omega}(\boldsymbol{p};\bar{\boldsymbol{p}}) \perp \boldsymbol{p} \ge 0$ is

$$\boldsymbol{p} = \max\left\{0, \bar{\boldsymbol{p}} - \left(\nabla_{\boldsymbol{p}}\boldsymbol{x}(\bar{\boldsymbol{p}})\cdot\nabla_{\boldsymbol{x}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}}) + \nabla_{\boldsymbol{p}}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}})\right)^{-1}F(\bar{\boldsymbol{p}};\bar{\boldsymbol{x}})\right\}$$

which help us get a new market price $\boldsymbol{p}$ and we keep repeating the above steps. The above process can be illustrated by following Algorithm 2.

Algorithm 2 Smoothing Newton-based decomposition method

1: **Input and Initialization** Choose $\boldsymbol{p}^1 \in \mathbb{R}^\alpha, \varepsilon^1 > 0$ and $\omega^1 \ge 0$. Set $r = 1$.

2: **Step 1: Subproblems.** For each $a \in \mathcal{A}$ solve

$$\min_{\boldsymbol{x}_a,\boldsymbol{\theta}_a}\Big\{ f_a(\boldsymbol{x}_a,\boldsymbol{\theta}_a;\boldsymbol{p}^r) - \varepsilon^r\sum_i \log\{-g_{ai}(\boldsymbol{x}_a,\boldsymbol{\theta}_a;\boldsymbol{p}^r)\} + \mathcal{I}_{\mathcal{X}_a}(\boldsymbol{x}_a) + \mathcal{I}_{\Theta_a}(\boldsymbol{\theta}_a) + \frac{\varepsilon^r\cdot\omega^r}{2}(\|\boldsymbol{x}_a\|_2^2 + \|\boldsymbol{\theta}_a\|_2^2)$$
$$: h_a(\boldsymbol{x}_a,\boldsymbol{\theta}_a;\boldsymbol{p}^r) = 0\Big\}$$

Let $(\boldsymbol{x}_a^r,\boldsymbol{\theta}_a^r) = (\boldsymbol{x}_a^{\varepsilon^r,\omega^r}(\boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^r}, \boldsymbol{\theta}_a^{\varepsilon^r,\omega^r}(\boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^r})$ denote the unique minimum in this problem given $\boldsymbol{p}^r$. Compute the gradients $\nabla_{\boldsymbol{p}}\boldsymbol{x}_a^{\varepsilon^r,\omega^r}(\boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^r}$ and define $\nabla_{\boldsymbol{p}}\boldsymbol{x}_a^r = \nabla_{\boldsymbol{p}}\boldsymbol{x}_a^{\varepsilon^r,\omega^r}(\boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^r}$. Define $\boldsymbol{x}^r = \begin{bmatrix}\boldsymbol{x}_1^r\\ \vdots \\ \boldsymbol{x}_{|\mathcal{A}|}^r\end{bmatrix}$ and $\nabla_{\boldsymbol{p}}\boldsymbol{x}^r = \begin{bmatrix}\nabla_{\boldsymbol{p}}\boldsymbol{x}_1^r\\ \vdots \\ \nabla_{\boldsymbol{p}}\boldsymbol{x}_{|\mathcal{A}|}^r\end{bmatrix}$.

3: **Step 2: Stopping Test.** If stopping criteria is met, Stop.

4: **Step 3: Update price.** Define $\nabla_{\boldsymbol{x}}F(\boldsymbol{p}^r;\boldsymbol{x}^r) = \nabla_{\boldsymbol{x}}F(\boldsymbol{p};\boldsymbol{x})|_{\boldsymbol{p}=\boldsymbol{p}^r,\boldsymbol{x}=\boldsymbol{x}^r}$ and $\nabla_{\boldsymbol{p}}F(\boldsymbol{p}^r;\boldsymbol{x}^r) = \nabla_{\boldsymbol{p}}F(\boldsymbol{p};\boldsymbol{x})|_{\boldsymbol{p}=\boldsymbol{p}^r,\boldsymbol{x}=\boldsymbol{x}^r}$.

$$\boldsymbol{p}^{r+1} = \max\left\{0, \boldsymbol{p}^r - \left(\nabla_{\boldsymbol{p}}\boldsymbol{x}^r\cdot\nabla_{\boldsymbol{x}}F(\boldsymbol{p}^r;\boldsymbol{x}^r) + \nabla_{\boldsymbol{p}}F(\boldsymbol{p}^r;\boldsymbol{x}^r)\right)^{-1}F(\boldsymbol{p}^r;\boldsymbol{x}^r)\right\}$$

5: **Step 4: Update Smoothing Parameters.** Determine ε^{r+1} and ω^{r+1} such that

$$0 < \varepsilon^{r+1} < \varepsilon^r, \quad 0 < \omega^{r+1} < \omega^r$$

6: **Step 5: Loop.** Set $r := r+1$ and go back to Step 1.

Remark

The aforementioned methods have been proposed as decomposition approaches to solve PNEP. However, there are still cases of PNEP that may not be solved by the above two algorithms in practice. For instance, the algorithm presented in [11] requires the subproblem's solution mapping $\boldsymbol{x}(p)$ to be a continuously differentiable function, which is a really strong requirement that may not be satisfied by a linear programming subproblem or when the subproblem is not strongly quadratic. The convergence of the outer iteration of Algorithm 2 is also not clearly addressed in [12]. Additionally, both methods necessitate the computation of the gradients $\nabla_{\boldsymbol{p}}\boldsymbol{x}_a(p)$ for each player a and the computation of the inverse $\left(\nabla_{\boldsymbol{p}}\boldsymbol{x}^r \cdot \nabla_{\boldsymbol{x}}F(\boldsymbol{p}^r;\boldsymbol{x}^r) + \nabla_{\boldsymbol{p}}F(\boldsymbol{p}^r;\boldsymbol{x}^r)\right)^{-1}$, which is computational expansive and may even be intractable if the problem is of large scale.

3.2.2 ADMM-based algorithm

In this section, we propose an ADMM-based decomposition algorithm designed to solve stochastic PNEPs. We begin by presenting a mathematical reformulation of each player's optimization problem and demonstrating the equivalence between solving the reformulated problem and the original problem. This reformulated problem allows us to develop an ADMM-based algorithm that decomposes the large original problem into smaller subproblems and solves it iteratively. We also proved that if the ADMM-based algorithm generates a series of converging points, the resulting point is the solution of the original equilibrium problem, thereby verifying the correctness of the algorithm.

The ADMM-based algorithm decomposes the problem into smaller subproblems with fixed prices, solves the subproblems in a Gauss-Seidel iterative manner, and uses the players' solutions to improve market prices. Although the ADMM method has been extensively studied in centralized network problems, its extension to equilibrium problems is unclear. Therefore, in this section, we present an ADMM-based algorithm to solve multistage stochastic equilibria with risk-averse players. We first provide an overview of the algorithm's conceptual framework before delving into the specifics of each step and the parameter updates.

Before discussing the algorithm, we introduce an equivalent reformulated problem for the PNEP, where each player solves a convex optimization problem. In the reformulated problem, an artificial variable $\boldsymbol{y}_a \in \mathbb{R}^{\alpha}$ controlled by player a is added, and an additional term $\boldsymbol{p}^T\boldsymbol{y}_a$ is added in the objective of player a. The reformulated optimization problem of

player a is as follows:

$$
\begin{aligned}
P_a^{\mathcal{SPR}}(\boldsymbol{p}): \quad \min_{\boldsymbol{x}_a, \boldsymbol{\theta}_a, y_a} \quad & f_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) + \boldsymbol{p}^T \boldsymbol{y}_a \\
\text{s.t.} \quad & h_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) = 0 \\
& g_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) \leq 0 \\
& \boldsymbol{x}_a \in \mathcal{X}_a, \boldsymbol{\theta}_a \in \Theta_a, \boldsymbol{y}_a \geq 0
\end{aligned}
\tag{3.16}
$$

The market constraint $0 \leq F(\boldsymbol{p}; \boldsymbol{x}) \perp \boldsymbol{p} \geq 0$ is also changed by $\boldsymbol{y} = (\boldsymbol{y}_a)_{a \in \mathcal{A}}$ and the new market constraint is as follows:

$$
F(\boldsymbol{p}; \boldsymbol{x}) - \sum_{a \in \mathcal{A}} \boldsymbol{y}_a = 0 \perp \boldsymbol{p} \text{ free} \tag{3.17}
$$

Theorem 3.3. *$(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*)$ is a solution of PNEP (3.4) and (3.8) if and only if $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{y}^*, \boldsymbol{p}^*)$ is the solution of MOEPC (3.16) and (3.17).*

Proof. Since for PNEP (3.4) and (3.8), each player is solving a convex optimization problem, f_a is twice continuously differentiable and g_a is continuously differentiable and satisfies the constraint qualification, $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*)$ is a solution of PNEP if and only if it satisfies the KKT conditions:

$$
\begin{aligned}
0 \in \nabla_{\boldsymbol{x}_a} f_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) + \nabla_{\boldsymbol{x}_a} h_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \nu_a^* + \nabla_{\boldsymbol{x}_a} g_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \beta_a^* + \mathcal{N}_{\mathcal{X}_a}(\boldsymbol{x}_a^*), \quad \forall a \in \mathcal{A} \\
0 \in \nabla_{\boldsymbol{\theta}_a} f_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) + \nabla_{\boldsymbol{\theta}_a} h_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \nu_a^* + \nabla_{\boldsymbol{\theta}_a} g_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \beta_a^* + \mathcal{N}_{\Theta_a}(\boldsymbol{\theta}_a^*), \quad \forall a \in \mathcal{A} \\
0 = h_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \perp \nu_a^* \text{ free}, \quad \forall a \in \mathcal{A} \\
0 \geq g_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \perp \beta_a^* \geq 0, \quad \forall a \in \mathcal{A} \\
0 \leq F(\boldsymbol{p}^*; \boldsymbol{x}^*) \perp \boldsymbol{p}^* \geq 0
\end{aligned}
\tag{3.18}
$$

For the PNEP (3.16) and (3.17), $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*, \boldsymbol{y}^*)$ is a solution if and only if it satisfies the

following KKT conditions

$$\begin{aligned}
0 \in \nabla_{\boldsymbol{x}_a} f_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) + \nabla_{\boldsymbol{x}_a} h_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*)\nu_a^* + \nabla_{\boldsymbol{x}_a} g_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*)\beta_a^* + \mathcal{N}_{\mathcal{X}_a}(\boldsymbol{x}_a^*), \quad \forall a \in \mathcal{A} \\
0 \in \nabla_{\boldsymbol{\theta}_a} f_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) + \nabla_{\boldsymbol{\theta}_a} h_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*)\nu_a^* + \nabla_{\boldsymbol{\theta}_a} g_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*)\beta_a^* + \mathcal{N}_{\Theta_a}(\boldsymbol{\theta}_a^*), \quad \forall a \in \mathcal{A} \\
0 = h_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \perp \nu_a^* \text{ free}, \quad \forall a \in \mathcal{A} \\
0 \geq g_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \perp \beta_a^* \geq 0, \quad \forall a \in \mathcal{A} \\
0 = F(\boldsymbol{p}^*; \boldsymbol{x}^*) - \sum_{a \in \mathcal{A}} \boldsymbol{y}_a^* \perp \boldsymbol{p}^* \text{ free} \\
0 \leq \boldsymbol{p}^* \perp \boldsymbol{y}_a^* \geq 0, \quad \forall a \in \mathcal{A}
\end{aligned} \tag{3.19}$$

Having the two KKT conditions, we start the main part of our proof.

First, we show that: $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*)$ solves (3.18) ($\Rightarrow$) there exist $\boldsymbol{y}^* \geq 0$ such that $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*, \boldsymbol{y}^*)$ satisfies (3.19).

Suppose

$$\boldsymbol{y}_a^* = \frac{1}{|\mathcal{A}|} F(\boldsymbol{p}^*; \boldsymbol{x}^*).$$

Then with the condition $0 \leq F(\boldsymbol{p}^*; \boldsymbol{x}^*) \perp \boldsymbol{p}^* \geq 0$ we will have

$$\begin{aligned}
0 \leq \boldsymbol{p}^* \perp \boldsymbol{y}_a^* \geq 0, \quad \forall a \in \mathcal{A} \\
0 = F(\boldsymbol{p}^*; \boldsymbol{x}^*) - \sum_{a \in \mathcal{A}} \boldsymbol{y}_a^*,
\end{aligned} \tag{3.20}$$

which proves $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*, \boldsymbol{y}^*)$ satisfies (3.19).

Next, we show that $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*, \boldsymbol{y}^*)$ satisfies (3.19) ($\Rightarrow$) $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*)$ satisfies (3.18). This is not difficult to show since from $0 \leq \boldsymbol{p}^* \perp \boldsymbol{y}_a^* \geq 0, \ \ \forall a \in \mathcal{A}$ and $0 = F(\boldsymbol{p}^*; \boldsymbol{x}^*) - \sum_{a \in \mathcal{A}} \boldsymbol{y}_a^* \perp \boldsymbol{p}^*$ we will have

$$0 \leq F(\boldsymbol{p}^*; \boldsymbol{x}^*) \perp \boldsymbol{p}^* \geq 0$$

From the above, $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*)$ is a solution of competitive MOPEC if and only if $(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*, \boldsymbol{y}^*)$ is the solution of MOEPC (3.16) and (3.17) □

After building the equivalence between the original problem (3.4), (3.8) and the reformulated problem (3.16)-(3.17), it becomes evident that solving the reformulated problem can yield the equilibrium point of the original problem. To achieve this, an artificial variable $\boldsymbol{y}_a \geq 0$ is introduced, which is controlled by each player a. The inequality market constraint

$$F(\boldsymbol{p}; \boldsymbol{x}) \geq 0$$

is transformed into an equality constraint

$$F(\boldsymbol{p};\boldsymbol{x}) - \sum_{a\in\mathcal{A}} \boldsymbol{y}_a = 0$$

During the iteration of the player-based decomposition algorithm, when solving player a's optimization problem, the original problem (3.16) is replaced by a modified objective function with an additional penalty term: $\frac{\omega}{2}\|F(\boldsymbol{p};\boldsymbol{x}) - \sum_{a\in\mathcal{A}} \boldsymbol{y}_a\|^2$. Therefore, the objective function becomes:

$$f_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}) + \boldsymbol{p}^T \boldsymbol{y}_a + \frac{\omega}{2}\|F(\boldsymbol{p};\boldsymbol{x}) - \sum_{a\in\mathcal{A}} \boldsymbol{y}_a\|^2 \tag{3.21}$$

where ω represents the penalty parameter that controls the strength of the penalty term.

When only one player is present, and the equality constraint $F(\boldsymbol{p};\boldsymbol{x}) - \sum_{a\in\mathcal{A}} \boldsymbol{y}_a = 0$ is part of the optimization problem, and the objective function has the additional Lagrange term associated with this equality constraint, we could claim that the objective function is the augmented Lagrangian function that associates with the equality constraint. The augmented Lagrangian method could be applied naturally here by iteratively updating the primal variables $(\boldsymbol{x}_a, \boldsymbol{\theta}_a, \boldsymbol{y}_a)$, and the Lagrange multipliers for equality constraint until convergence. However, when multiple players are present, the ADMM method could be applied by updating the primal variables $(\boldsymbol{x}_a, \boldsymbol{\theta}_a, \boldsymbol{y}_a)_{a\in\mathcal{A}}$ in a Gauss-Seidel manner, and subsequently the Lagrange multipliers until convergence.

In the case of our problem, although we don't have this augmented Lagrangian function, we treat the market price $\boldsymbol{p}$ as the Lagrangian multiplier associated with the market constraint (3.17) and the objective function (3.21) as the augmented Lagrangian function. When all players are risk-neutral, a large family of stochastic PNEP is equivalent to a system optimization problem, such as the economic dispatch problem and capacity expansion example with *Type I* or *Type II* market constraint in chapter 2. This equivalence has been shown in [72]. In this situation, the market price $\boldsymbol{p}$ is the Lagrangian multiplier associated with the market constraint in the system optimization problem with an appropriate coefficient, and the ADMM method is perfectly suited to be applied with strong theoretical guarantees. This motivates us to transfer the idea of the ADMM algorithm into the risk-averse case.

At iteration r with initial price $\boldsymbol{p}^r$, we solve each player a's optimization problem in a Gauss-Seidel manner to obtain the udpated player's actions $(\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}^{r+1}a, \boldsymbol{y}^{r+1}a)$.Then, we update the price variable by the projected first-order update with step size $\varepsilon \cdot \omega$ as follows:

$$\boldsymbol{p}^{r+1} = \max\left\{0, \boldsymbol{p}^r - \varepsilon\cdot\omega\cdot\left(F(\boldsymbol{p}^r;\boldsymbol{x}^{r+1}) - \sum_{j\in\mathcal{A}} \boldsymbol{y}_j^{r+1}\right)\right\}$$

where $0 < \varepsilon \leq 1$ is the parameter to control the step size of the projected first-order update on market price. The complete Algorithm for solving the reformulated problem is as follows:

Algorithm 3 ADMM-based algorithm for problem (3.16) and (3.17)

1: set $r = 0$, choose a starting point $(\boldsymbol{x}^0, \boldsymbol{\theta}^0, \boldsymbol{p}^0)$, parameter $\omega > 0, 0 < \varepsilon \leq 1$.
2: **while** stopping criterion not met **do**
3: **for** each player $a \in \mathcal{A}$ **do**
4:

$$\begin{aligned}\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}_a^{r+1}, \boldsymbol{y}_a^{r+1} = \arg\min_{\boldsymbol{x}_a, \boldsymbol{\theta}_a, \boldsymbol{y}_a} \Bigg\{ & f_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}^r) + (\boldsymbol{p}^r)^T \boldsymbol{y}_a \\ & + \frac{\omega}{2} \| F(\boldsymbol{p}^r; \boldsymbol{x}_{a<j}^{r+1}, \boldsymbol{x}_a, \boldsymbol{x}_{a>j}^r) - \boldsymbol{y}_a - \sum_{j<a} \boldsymbol{y}_j^{r+1} - \sum_{j>a} \boldsymbol{y}_j^r \|^2 \\ & \text{s.t. } h_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}^r) = 0, \;\; g_a(\boldsymbol{x}_a, \boldsymbol{\theta}_a; \boldsymbol{p}^r) \leq 0, \;\; \boldsymbol{x}_a \in \mathcal{X}_a, \;\; \boldsymbol{\theta}_a \in \Theta_a, \;\; \boldsymbol{y}_a \geq 0 \Bigg\}\end{aligned}$$

5: **end for**
6:

$$\boldsymbol{p}^{r+1} = \max \left\{ 0, \boldsymbol{p}^r - \varepsilon \cdot \omega \cdot \left(F(\boldsymbol{p}^r; \boldsymbol{x}^{r+1}) - \sum_{j \in \mathcal{A}} \boldsymbol{y}_j^{r+1} \right) \right\} \tag{3.22}$$

7: $r = r + 1$
8: **end while**

By iteratively updating the player's actions and the price variable, the algorithm aims to converge to the equilibrium point of the original problem (3.4) and (3.8). In practice, in the subsequent presentation of numerical results, we aim to demonstrate the convergence of the algorithm by examining several illustrative examples. In these examples, we choose relatively small values for ω and ε to ensure convergence. However, it is important to note that excessively reducing the values of ω and ε can lead to a considerable slowdown in the algorithm's convergence rate, thus resulting in prolonged computational time required to reach a solution. Therefore, striking a suitable balance for these parameters becomes essential to achieve a desirable trade-off between convergence speed and solution accuracy.

3.2.3 Optimality result for ADMM-based algorithm

Theorem 3.4. *If the the points* $\left\{ \left(\boldsymbol{x}^r, \; \boldsymbol{\theta}^r, \; \nu^r, \; \beta^r, \; \boldsymbol{p}^r, \; \boldsymbol{y}^r \right) \right\}$ *generated by the Algorithm 3 converge to the point* $\left\{ \left(\boldsymbol{x}^*, \; \boldsymbol{\theta}^*, \; \nu^*, \; \beta^*, \; \boldsymbol{p}^*, \; \boldsymbol{y}^* \right) \right\}$*, then* $\left(\boldsymbol{x}^*, \boldsymbol{\theta}^*, \boldsymbol{p}^*, \boldsymbol{y}^* \right)$ *will be the equilibrium point of the PNEP (3.16)-(3.17).*

Proof. Since $\{\boldsymbol{p}^r\}$ converge to $\boldsymbol{p}^*$ when $r \to \infty$, thus we must have

$$F(\boldsymbol{p}^*; \boldsymbol{x}^*) - \sum_{j \in \mathcal{A}} \boldsymbol{y}_j^* = 0 \tag{3.23}$$

because of $\boldsymbol{p}^{r+1} = \boldsymbol{p}^r - \varepsilon \cdot \omega \cdot \left(F(\boldsymbol{p}^r; \boldsymbol{x}^{r+1}) - \sum_{j \in \mathcal{A}} \boldsymbol{y}_j^{r+1} \right)$. From (3.22) we can have the KKT conditions for the player a's subproblem at iteration $r+1$

$$\begin{aligned}
0 \in \nabla_{\boldsymbol{x}_a} f_a(\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}_a^{r+1}; \boldsymbol{p}^r) + \nabla_{\boldsymbol{x}_a} h_a(\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}_a^{r+1}; \boldsymbol{p}^r) \nu_a^{r+1} + \nabla_{\boldsymbol{x}_a} g_a(\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}_a^{r+1}; \boldsymbol{p}^r) \beta_a^{r+1} \\
+ \omega \cdot \nabla_{\boldsymbol{x}_a} F(\boldsymbol{p}^r; \boldsymbol{x}_{j \le a}^{r+1}, \boldsymbol{x}_{j>a}^r) \left(F(\boldsymbol{p}^r; \boldsymbol{x}_{j \le a}^{r+1}, \boldsymbol{x}_{j>a}^r) - \sum_{j \le a} \boldsymbol{y}_j^{r+1} - \sum_{j>a} \boldsymbol{y}_j^r \right) + \mathcal{N}_{\mathcal{X}_a}(\boldsymbol{x}_a^{r+1}) \\
0 \in \nabla_{\boldsymbol{\theta}_a} f_a(\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}_a^{r+1}; \boldsymbol{p}^r) + \nabla_{\boldsymbol{\theta}_a} h_a(\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}_a^{r+1}; \boldsymbol{p}^r) \nu_a^{r+1} \\
+ \nabla_{\boldsymbol{\theta}_a} g_a(\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}_a^{r+1}; \boldsymbol{p}^r) \beta_a^{r+1} + \mathcal{N}_{\Theta_a}(\boldsymbol{\theta}_a^{r+1}), \\
0 = h_a(\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}_a^{r+1}; \boldsymbol{p}^r) \perp \nu_a^{r+1} \text{ free} \\
0 \ge g_a(\boldsymbol{x}_a^{r+1}, \boldsymbol{\theta}_a^{r+1}; \boldsymbol{p}^r) \perp \beta_a^{r+1} \ge 0 \\
0 \le \boldsymbol{p}^r - \omega \cdot \left(F(\boldsymbol{p}^r; \boldsymbol{x}_{j \le a}^{r+1}, \boldsymbol{x}_{j>a}^r) - \sum_{j \le a} \boldsymbol{y}_j^{r+1} - \sum_{j>a} \boldsymbol{y}_j^r \right) \perp \boldsymbol{y}_a^{r+1} \ge 0
\end{aligned} \tag{3.24}$$

And since $\left\{ \begin{pmatrix} \boldsymbol{x}^r & \boldsymbol{\theta}^r & \nu^r & \beta^r & \boldsymbol{p}^r & \boldsymbol{y}^r \end{pmatrix} \right\}$ converge to $\left\{ \begin{pmatrix} \boldsymbol{x}^* & \boldsymbol{\theta}^* & \nu^* & \beta^* & \boldsymbol{p}^* & \boldsymbol{y}^* \end{pmatrix} \right\}$, and from (3.24) and (3.23) we will have

$$\begin{aligned}
0 \in \nabla_{\boldsymbol{x}_a} f_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) + \nabla_{\boldsymbol{x}_a} h_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \nu_a^* + \nabla_{\boldsymbol{x}_a} g_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \beta_a^* + \mathcal{N}_{\mathcal{X}_a}(\boldsymbol{x}_a^*) \\
0 \in \nabla_{\boldsymbol{\theta}_a} f_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) + \nabla_{\boldsymbol{\theta}_a} h_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \nu_a^* + \nabla_{\boldsymbol{\theta}_a} g_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \beta_a^* + \mathcal{N}_{\Theta_a}(\boldsymbol{\theta}_a^*), \\
0 = h_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \perp \nu_a^* \text{ free} \\
0 \ge g_a(\boldsymbol{x}_a^*, \boldsymbol{\theta}_a^*; \boldsymbol{p}^*) \perp \beta_a^* \ge 0 \\
0 \le \boldsymbol{p}^* \perp \boldsymbol{y}_a^* \ge 0
\end{aligned} \tag{3.25}$$

□

3.2.4 Structural comparisons between algorithms

This section aims to provide a comparison between our proposed algorithm and two existing decomposition algorithms, namely the Newton-based decomposition algorithm (Algorithm 1) in [12] and the smoothing decomposition algorithm (Algorithm 2) [11]. All three algorithms aim to solve the PNEP (3.4) and (3.8) in a distributed manner. However, there exist differences among these algorithms, particularly in the major iteration of price updating.

- Algorithm 1 and 2 update the price by solving a linear approximation of $0 \leq \hat{R}(\boldsymbol{p}) \perp \boldsymbol{p} \geq 0$ directly.
- Algorithms 3 uses only first-order information to update the price $\boldsymbol{p}$ by a gradient-based method: $\boldsymbol{p} = \bar{\boldsymbol{p}} - \varepsilon \cdot \omega \cdot F(\bar{\boldsymbol{p}}; \boldsymbol{x}(\bar{\boldsymbol{p}}))$.

When solving the subproblem:

- With fixed $\boldsymbol{p}$, Algorithm 1 solves the original subproblem.
- With fixed $\boldsymbol{p}$, Algorithm 2 solves an approximated subproblem using smooth approximation and regularization.
- With fixed $\boldsymbol{p}$, in Algorithm 3, we minimize the original problem with penalization of the market constraint via an Augmented Lagrangian in a Gauss-Seidel way.

Besides the above major differences, there are also some other differences that can be summarized in the following Table 3.1.

	penalty term	penalty size	require $\nabla_p x(p)$	p update	specific requirement
Algorithm 1 [11]	No	No	Yes	Newton update	with fixed $\boldsymbol{p}$, each subproblem has a unique solution
Algorithm 2 [12]	$\frac{\omega_r}{2}(\|\boldsymbol{x}_a\|^2 + \|\boldsymbol{\theta}_a\|^2)$	$\omega_r \searrow 0$	Yes	Using Ipopt and many points every iteration	No, but to show epiconvergence, origianl subproblem must have a unique solution with $\boldsymbol{p}$
This work	$\frac{\omega_r}{2}\|F(\boldsymbol{p}^r; \boldsymbol{x}_{a<j}^{r+1}, \boldsymbol{x}_{a>j}^r) - \boldsymbol{y}_a - \sum_{j<a} \boldsymbol{y}_j^{r+1} - \sum_{j>a} \boldsymbol{y}_j^r\|^2$	ω_r constant	No	first order update with step size $\gamma\omega_r$	No

Table 3.1: Comparison of the difference of three main algorithms

3.3 Numerical experiments

This section presents an empirical evaluation and demonstration of the effectiveness of the ADMM-based player decomposition algorithm in solving stochastic PNEP. The algorithm is tested on the dispatch example and hydroelectric example in chapter 2, and computational experiments are conducted using the coherent risk measure $\overline{CVaR}(\lambda, \varphi)(Z)$ defined in chapter 1. This coherent risk measure can be formulated in the dual representation of coherent risk measure by a *conjugate-reformulation*, and it also has a specific primal formulation. An analysis of the practical efficiency of the two formulations is presented, and the ADMM-based algorithm is tested on both formulations to find out which is is

performing better in practice. The results indicate that the *conjugate-reformulation* is more efficient in practice. The numerical results also demonstrate how the parameters λ, φ, and the quadratic coefficient term in players' objective function affect the performance of ADMM-based algorithm. Specifically, larger values of λ and the quadratic coefficient term make the problem more difficult. The section also examines the impact of the algorithm's parameter ω and the optimal selection of ω on the algorithm's performance. The numerical results show that a too large ω causes the algorithm to converge slowly, while a too small ω causes divergence of the algorithm. The optimal ω is a parameter that needs to be selected in between. The experiments aim to provide insights into the practical usefulness of the algorithm and identify possible limitations and areas for future research.

3.3.1 Computational setting

The proposed algorithm was implemented in GAMS 26.1 and all examples were run on a single server with an Intel Xeon processor (36 cores 3.1Ghz) and 768G RAM. The CONOPT3 solver was used to solve each optimization subproblem, and each algorithm was limited to a single thread. In the analysis of the algorithm's performance, we will first examine the impact of the parameters γ and ω. Next, we will investigate how the risk measure of the original problem affects the algorithm's performance with fixed parameters γ and ω. Finally, we will propose a special initial point technique for dealing with the same problems but in an increasing risk-averse setting.

3.3.2 Parameters setting of test problems

We are testing our ADMM-based algorithm on the stochastic economic dispatch and capacity expansion examples with *Type I* and *Type II* market constraints and risk-averse players in chapter 2. Hence, we first present more details of both examples.

In the economic dispatch example, we assume in the player's objective function that

$$c_{ain}(q_{ain}, s_{ain}, v_{ain}) = \frac{1}{2}\begin{bmatrix} q_{ain} & s_{ain} & v_{ain} \end{bmatrix}\begin{bmatrix} \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}\begin{bmatrix} q_{ain} \\ s_{ain} \\ v_{ain} \end{bmatrix} + \begin{bmatrix} c^q_{ain} & c^s_{ain} & c^v_{ain} \end{bmatrix}\begin{bmatrix} q_{ain} \\ s_{ain} \\ v_{ain} \end{bmatrix} \tag{3.26}$$

The parameter ϵ is used to control the degree of convexity of each player's objective function. Its value will be varied in different experiments later. The parameters c^q_{ain}, c^s_{ain} and c^v_{ain} are uniformly distributed over the interval $[1, 10]$ for all $a \in \mathcal{A}$, $i \in \mathcal{I}$ and $n \in \mathcal{N}$. In the supply-demand market constraint, the demand D_{in} is uniformly distributed over the

interval $[20 \cdot |\mathcal{A}|, 50 \cdot |\mathcal{A}|]$, where $|\mathcal{A}|$ is the cardinality of set $\mathcal{A}$. The production capacity W_{ai} is uniformly distributed over the interval $[40, 75]$. The lower bound of storage $\bar{v}_{ai}^{l}$ is fixed at 20, and the upper bound of storage $\bar{v}_{ai}^{u}$ is fixed at 100. In the *Type II* market constraint the parameter γ_{in} is fixed at 10^{-2}.

In the capacity expansion example, we assume in the player's objective function that

$$\begin{aligned} G_{an}(\boldsymbol{s}_{a\cdot n}, C_{an}) &= \sum_{i\in\mathcal{I}}[\frac{1}{2}\cdot\epsilon\cdot s_{ain}^2 + c_{ain}^s \cdot s_{ain}] + \frac{1}{2}\cdot\epsilon\cdot C_{an}^2 \\ I_{an}(u_{an}) &= \frac{1}{2}\cdot\frac{\epsilon}{10}\cdot u_{an}^2 + c_{an}^u \cdot u_{an} \end{aligned} \tag{3.27}$$

The parameter ϵ controls the degree of convexity in each player's objective function, which will be varied in different experiments later. The coefficient c_{ain}^s follows a uniform distribution between $[1, 10]$, and c_{an}^u follows a uniform distribution between $[5, 20]$. In the supply-demand market constraint, the demand D_{in} follows a uniform distribution between $[7.5 \cdot |\mathcal{A}|, 12.5 \cdot |\mathcal{A}|]$. Parameter ψ_n is uniformly distributed over the interval $[0.5, 1]$ for each $n \in \mathcal{N}$. In the *Type II* market constraint the parameter γ_{in} is fixed at 10^{-2}.

3.3.3 Comparisons of different formulations for coherent risk measure $\overline{CVaR}(\lambda, \varphi)$

In this subsection, we intend to assess the effectiveness of the ADMM-based algorithm on the economic dispatch example with risk-averse players, where the risk measure $\overline{CVaR}(\lambda, \varphi)(\cdot)$ is employed. As discussed in the introduction chapter, $\overline{CVaR}(\lambda, \varphi)(\cdot)$ is a coherent risk measure that enables us to formulate the problem using a *conjugate-based reformulation*. Additionally, $\overline{CVaR}(\lambda, \varphi)(\cdot)$ has an unique characteristic in that it has an alternative algebraic representation referred to as the *primal formulation*:

$$\begin{aligned} \overline{CVaR}(\lambda,\varphi)\Big((f_{am}(x_{am}, x_{-am}; p_m)_{m\in n_+})\Big) &= (1-\lambda)\cdot\mathbb{E}\Big((f_{am}(x_{am}, x_{-am}; p_m)_{m\in n_+})\Big) \\ &\quad + \lambda\cdot CVaR_{\varphi}\Big((f_{am}(x_{am}, x_{-am}; p_m)_{m\in n_+})\Big) \\ &= (1-\lambda)\cdot\sum_{m\in n_+}\phi_m\cdot f_{am}(x_{am}, x_{-am}; p_m) \\ &\quad + \lambda\cdot\min_{\kappa_{an}\in\mathbb{R}}\Big\{\kappa_{an} + \frac{1}{1-\varphi}\mathbb{E}\big([f_{am}(x_{am}, x_{-am}; p_m) \\ &\quad - \kappa_{an}]_+\big)_{m\in n_+}\Big\} \end{aligned} \tag{3.28}$$

Let us analyze the computational efficiency of the *conjugate-based reformulation* and

primal formulation of the coherent risk measure $\overline{CVaR}(\lambda,\varphi)(\cdot)$. The primal formulation introduces additional variables $\{\kappa_{an}\}_{n\in\mathcal{N}\setminus\mathcal{L}}$, $\{\delta_{an}\}_{n\in\mathcal{N}\setminus\{1\}}$, where $\delta_{an}\in\mathbb{R}^{+}$, and additional constraints ($\delta_{an}\geq c^{q}_{ain}(q_{ain})-p_{in}\cdot\tau_a\cdot s_{ain}+\theta_{an}-\kappa_{an-}$) in each player's optimization problem. These are total $2\cdot|\mathcal{N}|-|\mathcal{L}|-1$ additional variables and $|\mathcal{N}|-1$ additional constraints. Therefore, the whole system involves an additional $|\mathcal{A}|\cdot\big(2\cdot|\mathcal{N}|-|\mathcal{L}|-1\big)$ variables and $|\mathcal{A}|\cdot\big(|\mathcal{N}|-1\big)$ constraints when using the primal formulation.

The computational complexity of the primal formulation will depend on the problem size and the solution method used. If the problem size is small, the additional variables and constraints introduced by the primal formulation may not have a significant impact on the computational time. However, for large-scale problems, the additional computational burden introduced by the primal formulation could be substantial. The solution method used will also affect the computational efficiency of the primal formulation. If an efficient solution method is used to solve the primal formulation, the computational time may not increase significantly compared to the *conjugate-based reformulation*. However, if an inefficient solution method is used, the additional variables and constraints introduced by the primal formulation may lead to a substantial increase in computational time.

For the dual formulation, although no additional variables will be involved, there will be additional $\sum_{n\in\mathcal{N}\setminus\mathcal{L}}\big(|\mathcal{K}_{an}|-1\big)$ constraints, where $|\mathcal{K}_{an}|$ is the cardinality of the set $\mathcal{K}_{an}$, which is the set of extreme points of the risk set.

$$\mathcal{D}^{\lambda,\varphi}_{an}=\Big\{\boldsymbol{\mu}_a\in\mathbb{R}^{|n+|}:\mu_{am}=(1-\lambda)\cdot\phi_m+\lambda\cdot\eta_{am},\ 0\leq\eta_m\leq\frac{\phi_m}{1-\varphi},\ \ m\in n_+ \\ \sum_{m\in n_+}\eta_{am}=1\Big\} \tag{3.29}$$

In reality, the value of $|\mathcal{K}_{an}|$ is determined by the value of $\frac{\phi_m}{1-\varphi}$ for each $m\in n_+$. Specifically, if $\frac{\phi_m}{1-\varphi}>1$ for all $m\in n_+$, then $|\mathcal{K}_{an}|=|n_+|$. However, if $\frac{\phi_m}{1-\varphi}\leq 1$ for all $m\in n_+$, then in the worst case scenario, $|\mathcal{K}_{an}|$ could be greater than $\binom{|n_+|}{2}\cdot 2$. Therefore, in the worst-case scenario, each player's optimization problem will involve more than an additional $\sum_{n\in\mathcal{N}\setminus\mathcal{L}}\binom{|n_+|}{2}\cdot 2$ constraints. However, in most cases, the value of φ is typically large enough to avoid the worst-case scenario. From a theoretical perspective, it is challenging to compare the computational efficiency of the primal and dual formulations. Thus, the analysis will be conducted from a numerical perspective instead.

The effect of size ω on algorithm's performance: Based on primal representation and dual *conjugate-based reformulation* representation of the risk measure $\overline{CVaR}(\lambda,\varphi)$

In this subsection, we aim to evaluate the effectiveness of the ADMM-based algorithm on the economic dispatch example with risk-averse players, where $\overline{CVaR}(\lambda,\varphi)(\cdot)$ is used

as the risk measure. In this series of experiments, we investigate the effect of varying values of ω on the performance of the ADMM-based algorithm, using both primal formulation and dual *conjugate-based reformulation* of the risk measure $\overline{CVaR}(\lambda, \varphi)$ for the economic dispatch example. The experiments were conducted on scenario **tree 1** in chapter 2, with $|\mathcal{A}| = 5$ and $|\mathcal{I}| = 2$, where each player uses the $\overline{CVaR}(\lambda, \varphi)$ risk measure with $\lambda = 0.5$ and $\varphi = 0.95$. The quadratic term $\epsilon_{ain} = 1$ for all $a \in \mathcal{A}, i \in \mathcal{I}, n \in \mathcal{N}$. For each test, we set all prices p_{in} to 60, while all other players' variables are set to 0. The parameter ε is fixed at 0.5 for all experiments. The algorithms are run until the Fisher-Burmeister (FB) merit value is smaller than 10^{-4}. The numerical results are reported as the average/standard deviation on 4 randomly generated data sets and are presented in Tables 3.2 and 3.3. Table 3.2 shows the results obtained using the primal formulation, while Table 3.3 presents the results based on the dual *conjugate-based reformulation*.

ω	Initial Merit(Mean/std)	Iter # (Mean/std)	Final Merit(Mean/std)	t(sec) (Mean/std)
0.0005	252.690/4.7	4998.2/1031.0	$9.873 \times 10^{-5}/1.902 \times 10^{-6}$	823.7/175.4
0.001	278.058/7.3	2774.0/576.4	$9.497 \times 10^{-5}/6.260 \times 10^{-6}$	432.4/86.2
0.005	739.841/42.8	2000.8/473.2	$9.921 \times 10^{-5}/1.214 \times 10^{-7}$	276.0/71.9
0.01	1500.7/100.3	1814.8/438.0	$9.942 \times 10^{-5}/4.924 \times 10^{-7}$	249.2/65.9
0.05	7112.1/602.0	2463.8/1930.8	$9.958 \times 10^{-5}/1.404 \times 10^{-7}$	363.0/300.7
0.1	9320.6/262.8	-	-	1800/0
0.5	9890.7/1077.3	-	-	1800/0
1	9594.0/1400.7	-	-	1800/0

Table 3.2: Comparison of algorithm performance with respect to parameter ω based on primal representation with $\varepsilon = 0.5$ on scenario tree 1. '-' is caused by the divergence of the algorithm.

ω	Initial Merit(Mean/std)	Iter # (Mean/std)	Final Merit(Mean/std)	t(sec) (Mean/std)
0.0005	251.1/4.6	5009.8/1038.1	$9.806 \times 10^{-5}/2.332 \times 10^{-6}$	682.1/183.2
0.001	272.2/7.1	2774.8/581.1	$9.721 \times 10^{-5}/2.243 \times 10^{-6}$	346.1/98.1
0.005	700.8/49.6	2000.5/472.1	$9.873 \times 10^{-5}/7.146 \times 10^{-7}$	269.7/65.6
0.01	1416.1/105.8	1812.5/436.8	$9.923 \times 10^{-5}/5.470 \times 10^{-7}$	241.9/60.0
0.05	6933.7/599.5	2462.8/1931.8	$9.950 \times 10^{-5}/1.613 \times 10^{-7}$	346.9/270.6
0.1	9025.1/192.5	-	-	1800/0
0.5	9230.9/969.4	-	-	1800/0
1	8588.4/1033.2	-	-	1800/0

Table 3.3: Comparison of algorithm performance with respect to parameter ω based on dual representation with $\varepsilon = 0.5$ on scenario tree 1. '-' is caused by the divergence of the algorithm.

According to the results presented in Table 3.2 and Table 3.3, the choice of risk measure formulation, either primal or dual, does not have a significant impact on the number of

iterations required to achieve the desired accuracy. The experiments performed on both formulations require a similar number of steps with the same parameter ω. However, the total time required to achieve the desired accuracy differs between the two formulations, with the dual formulation requiring less time. This is likely due to the fact that for this particular test case, where $\varphi = 0.95$ and the upper tail of the risk is very small, not many additional constraints are required in the dual formulation.

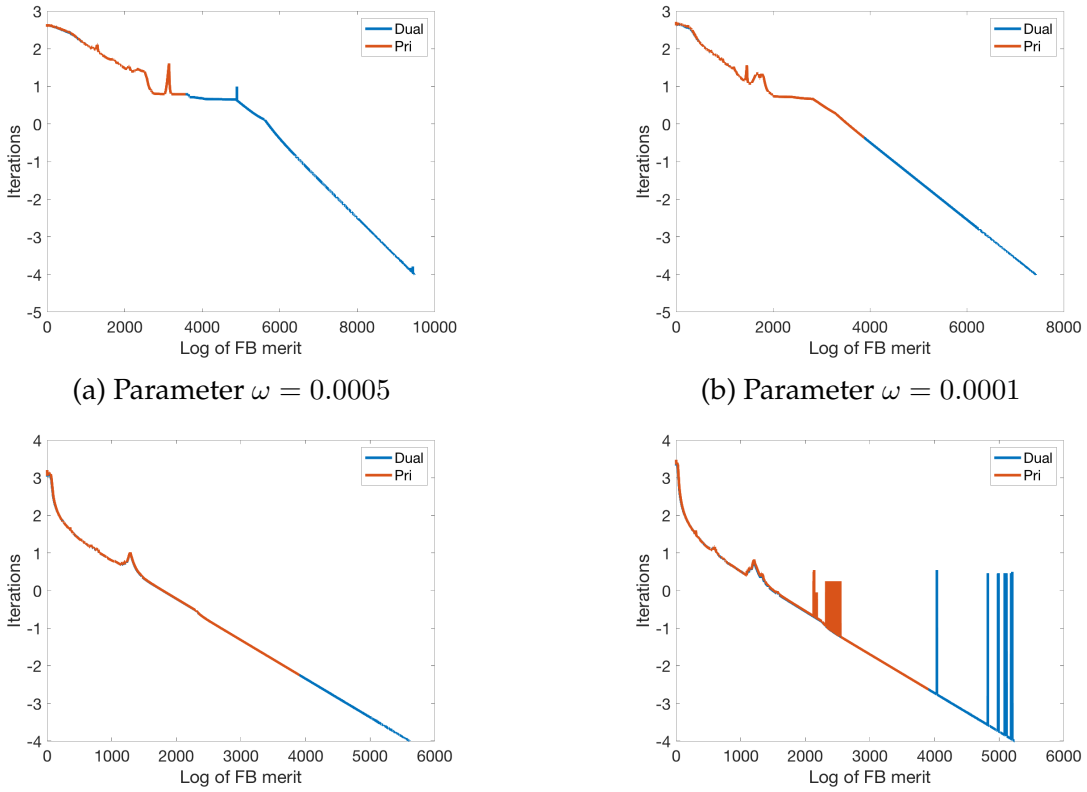
ω	Initial Merit	Iter #	Final Merit	t(sec)
0.0005	454.0	3576	6.122	3600
0.001	519.7	3877	4.381×10^{-1}	3600
0.005	1582.0	3906	5.700×10^{-3}	3600
0.01	3001.2	3907	2.400×10^{-3}	3600
0.05	7988.4	3899	$5.224 \cdot 10^{-4}$	3600
0.1	1176.3	3769	1.200×10^{-3}	3600
0.5	14120.8	3476	9.299×10^{-1}	3600
1	13017.3	3478	2.607	3600

Table 3.4: Comparison of different ω on algorithm performance based on primal representation with $\varepsilon = 0.7$ on scenario tree 2

ω	Initial Merit	Iter #	Final Merit	t(sec)
0.0005	450.0	9479	9.943×10^{-5}	2083.7
0.001	505.9	7422	9.708×10^{-5}	1534.6
0.005	1476.7	5611	9.980×10^{-5}	1101.2
0.01	2806.7	5241	9.998×10^{-5}	1025.5
0.05	7521.2	4599	$9.978 \cdot 10^{-5}$	883.7
0.1	11092.4	4471	9.987×10^{-5}	851.5
0.5	12695.6	13983	1.706×10^{-3}	3600
1	11531.3	13730	7.421×10^{-2}	3600

Table 3.5: Comparison of different ω on algorithm performance based on dual representation with $\varepsilon = 0.7$ on scenario tree 2

We also conducted experiments on a large scenario tree, tree 2, with the same parameter settings to validate our conclusions. Tables 3.4 and 3.5 show that the dual representation-based formulation achieves a much better result during the same restricted time compared to the primal representation-based formulation. This is because the subproblem with the dual representation takes less time to solve. Figure 3.3 displays the log of the Fisher-Burmeister residual along with the iterations. We can observe that in the earlier iterations (before the primal formulations stop), the plots of the primal and dual are almost identical. This indicates that both formulation approaches exhibit some acceleration effects on the major iterations. Therefore, the advantage of the dual formulation lies in the faster solution of the subproblem in that situation.



(a) Parameter $\omega = 0.0005$

(b) Parameter $\omega = 0.0001$

(c) Parameter $\omega = 0.005$

(d) Parameter $\omega = 0.001$

Figure 3.3: Comparison between number of iterations and FB merit in log scale on both primal and dual formulation with changing parameter ω.

Also from Tables 3.2, 3.3, 3.4 and 3.5, it is evident that the parameter size ω has a significant impact on the practical performance of the algorithm. The experiments demonstrate that the algorithm can converge if ω is small enough. However, when ω is too small, the Fisher-Burmeister residual decreases at a slow rate compared to a suitable ω. This occurs because the update step for the price from $\boldsymbol{p}^r$ to $\boldsymbol{p}^{r+1}$ becomes too small, which leads to a slow convergence rate to the optimal price and results in the dual merit residual remaining large.

The effect of size λ on algorithm's performance: Based on dual representation of the risk measure $\overline{CVaR}(\lambda, \varphi)$.

Economic dispatch example

From our numerical analysis, we have compared the primal and dual representations of a coherent risk measure and found that the dual formulation is superior when computing with a large value of φ. Furthermore, the dual formulation can represent the general case of a coherent risk measure. As a result, in all subsequent experiments, we will use the dual formulation. In this section, we will investigate how the properties of the risk measure

and the classification of a player's subproblem impact the practical performance of the algorithm.

ω	λ	α	Iter #	Final Merit	t(seconds)
0.05	0.1	0.95	5787	4.740×10^{-7}	4643.967
0.05	0.2	0.95	7626	3.530×10^{-5}	7200
0.05	0.3	0.95	7601	2.043×10^{-3}	7200
0.05	0.4	0.95	7593	2.441×10^{-2}	7200
0.05	0.5	0.95	7585	1.170×10^{-1}	7200
0.05	0.6	0.95	6487	3.707×10^{-1}	7200
0.05	0.7	0.95	7846	3.621×10^{-1}	7200
0.05	0.8	0.95	7879	6.819×10^{-1}	7200
0.05	0.9	0.95	7833	4.345×10^{-1}	7200

Table 3.6: Comparison of different risk measure on algorithm performance based on economic dispatch example with $\epsilon = 0$ and $\varepsilon = 1$ on scenario tree 3.

ω	λ	φ	Iter #	Final Merit	t(seconds)
0.05	0.1	0.95	6971	9.251×10^{-6}	7200
0.05	0.2	0.95	7624	3.471×10^{-4}	7200
0.05	0.3	0.95	7376	1.356×10^{-2}	7200
0.05	0.4	0.95	7593	-	7200
0.05	0.5	0.95	6858	5.170×10^{-1}	7200
0.05	0.6	0.95	7116	9.568×10^{-1}	7200
0.05	0.7	0.95	6036	1.3618	7200
0.05	0.8	0.95	6021	1.719	7200
0.05	0.9	0.95	6079	9.045×10^{-1}	7200

Table 3.7: Comparison of different risk measure on algorithm performance based on economic dispatch example with $\epsilon = 1$ and $\varepsilon = 1$ on scenario tree 3. ''-' is caused by the failure of the solver to solve the subproblem.

Tables 3.6, 3.7 present the numerical results of the ADMM-based Algorithm 3 applied to the economic dispatch example with the dual representation formulation, with varying values of the parameter ϵ_{ain}, as previously discussed.

The tables demonstrate that as the value of λ increases in $\overline{CVaR}(\lambda, \varphi)$, the algorithm requires more major iterations to attain the desired FB-merit residual value. Notably, when $\lambda \geq 0.5$, it becomes challenging for the algorithm to force the FB-merit residual value below 10^{-1}.

Capacity expansion example

In the second example, we still test how risk measure itself will affect the performance of the algorithm. In Table 3.8, we fix $\varphi = 0.95$ and change $\lambda = 0.1, 0.2, \ldots, 0.9$. In Tables 3.9 and 3.10, we fix $\alpha_{an} = 0.6$ and change $\lambda = 0.1, 0.5, 0.9$. Here Tables 3.8 and 3.9 are for linear subproblem and Table 3.10 is for quadratic subproblem.

ω	λ	φ	Iter #	Final Merit	t(seconds)
0.05	0.1	0.95	746	9.888×10^{-5}	185.9
0.05	0.2	0.95	398	9.992×10^{-5}	115.3
0.05	0.3	0.95	280	9.904×10^{-5}	81.4
0.05	0.4	0.95	429	9.863×10^{-5}	111.8
0.05	0.5	0.95	799	9.865×10^{-5}	196.1
0.05	0.6	0.95	1741	9.946×10^{-5}	421.5
0.05	0.7	0.95	4790	9.989×10^{-5}	1207.9
0.05	0.8	0.95	12692	1.460×10^{-3}	3600
0.05	0.9	0.95	12395	2.228×10^{-2}	3600

Table 3.8: Comparison of different risk measure on algorithm performance based on second example with $\epsilon = 0$ and $\varepsilon = 1$ on scenario tree 2

ω	λ	φ	Iter #	Final Merit	t(seconds)
0.05	0.1	0.6	2183	4.776×10^{-3}	3600
0.05	0.5	0.6	-	-	3600
0.05	0.9	0.6	2222	5.531×10^{-2}	3600

Table 3.9: Comparison of different risk measure on algorithm performance based on second example with $\epsilon = 0$ and $\varepsilon = 1$ on scenario tree 2. '-' is caused by the failure of the solver to solve the subproblem.

ω	λ	φ	Iter #	Final Merit	t(seconds)
0.05	0.1	0.6	1248	6.483×10^{-7}	1952
0.05	0.5	0.6	1708	7.493×10^{-7}	535.3
0.05	0.9	0.6	2072	9541×10^{-6}	3600

Table 3.10: Comparison of different risk measures on algorithm performance based on the second example with $\epsilon = 1$ and $\varepsilon = 1$ on scenario tree 2.

Remark 3.5. We can see although from these three tables that when λ is increasing in $\overline{CVaR}(\lambda, \varphi)$, the major iterations it takes is not in a strictly increasing trend, we can still see that in all three cases when λ is near 1 the problem is the hardest. It is difficult for the algorithm to force the FB-merit residual value small when $\lambda = 0.5$.

Initial point selection approach

To address the slow convergence issue that arises when λ is close to 1, we propose an initial point selection approach. The rationale behind this is that the slow convergence rate of the algorithm stems from the slow updates of price variables towards the optimal solution position. Specifically, the price update only utilizes first-order information, which is suitable for large-scale problems but suffers from a slow convergence rate. To tackle this issue, we propose an approach to select a good initial point for the price variables that

reduces the algorithm's running time in practice. Specifically, we suggest setting the initial point as the solution of the case when λ_0 is close to 0. Our experiments show that using this approach, the algorithm's running time is significantly reduced.

We perform experiments on the problems presented in Tables 3.6 and 3.7. For each problem, we solve the case when $\lambda = k \cdot 0.1$ for $k = 1, \ldots, 0.9$, using the solution of the case $\lambda_0 = (k-1) \cdot 0.1$ as the initial point. The numerical results are reported in the following tables.

ω	λ	φ	Iter #	Final Merit	t(seconds)
0.05	0.1	0.95	1679	8.891×10^{-7}	992.740
0.05	0.2	0.95	1685	9.934×10^{-7}	870.236
0.05	0.3	0.95	6905	9.998×10^{-7}	6471.4
0.05	0.4	0.95	6536	9.984×10^{-7}	5554.3
0.05	0.5	0.95	7284	1.740×10^{-6}	7200
0.05	0.6	0.95	2677	9.893×10^{-7}	2324.8
0.05	0.7	0.95	4756	5.545×10^{-6}	7200
0.05	0.8	0.95	6942	4.190×10^{-2}	7200
0.05	0.9	0.95	7452	5.201×10^{-3}	7200

Table 3.11: Comparison of different risk measure on sequentail initial point algorithm performance with $\epsilon = 0$ and $\varepsilon = 1$ on scenario tree 3.

ω	λ	φ	Iter #	Final Merit	t(seconds)
0.05	0.1	0.95	4855	9.521×10^{-7}	2458.4
0.05	0.2	0.95	5477	7.738×10^{-7}	2779.3
0.05	0.3	0.95	5589	4.520×10^{-6}	7200
0.05	0.4	0.95	7411	3.773×10^{-6}	7200
0.05	0.5	0.95	2543	9.805×10^{-6}	1696.7
0.05	0.6	0.95	1908	9.420×10^{-7}	999.2
0.05	0.7	0.95	2681	9.359×10^{-6}	3697.7
0.05	0.8	0.95	4750	9.850×10^{-7}	4114.9
0.05	0.9	0.95	7301	3.143×10^{-4}	7200

Table 3.12: Comparison of different risk measure on sequentail initial point algorithm performance with $\epsilon = 1$ and $\varepsilon = 1$ on scenario tree 3.

It should be noted that there is a clear difference between the results reported in Tables 3.6 and 3.11, as well as the difference between Tables 3.7 and 3.12. When the algorithm was initialized with an arbitrary starting point, it exhibited a slow convergence rate to the solution. However, in the case of solving the problem with a risk measure parameterized by λ and φ, if we start from a solution of the stochastic equilibrium with risk measure parameter $\bar{\lambda} < \lambda$, we can observe a faster convergence rate and achieve a good approximation of the solution point.

3.4 Conclusion

In this chapter, we proposed a new ADMM-based algorithm to solve the PNEP (3.4) with VI (3.8). The convergence of the algorithm was observed in all numerical tests on two multistage stochastic equilibrium problems with risk-averse players. It is observed that with all other conditions fixed, it will take longer for the algorithm to solve the problem if the players become more risk-averse. Although the convergence can be achieved from an arbitrary initial point, the numerical results show that the computational performance can be improved by choosing an initial point as the solution of the case with less risk-averse players. This suggests generating a homotopy approach for more reliable solution finding.

Overall, these results are somewhat disappointing. While they allow decomposition, they do not converge quickly and the choice of algorithmic parameters is problematirc. We believe subsequent methods are superior for solution of even these PNEP problems.

4 RISK DECOMPOSITION METHOD FOR MULTISTAGE STOCHASTIC EQUILIBRIUM WITH RISK-AVERSE PLAYERS

The main contribution of this chapter is the development of a primal-MOPEC-dual-risk algorithm to solve the stochastic MOPEC with risk-averse players using an *equilibrium reformulation* as discussed in chapter 1. The algorithm decomposes the original problem into two subproblems and solves them iteratively. This decomposition approach is motivated by the fact that the additional probability vector $\boldsymbol{\mu}$ introduces nonlinearity to the multistage problem, resulting in a highly nonlinear complementarity problem if solved by the PATH solver. By fixing the probability vector $\boldsymbol{\mu}$, the problem becomes simpler, and in the case of quadratic player objective functions, it can even be formulated as a linear complementarity problem. The other subproblem can be efficiently solved by solving a series of linear programming problems. Compared to the ADMM-based algorithm discussed in chapter 3, the proposed primal-MOPEC-dual-risk algorithm enables the solution of more general stochastic MOPECs with risk-averse players, especially in scenarios where players exhibit general Nash behavior interaction with each other. However, one challenge arises when the scenario tree has a large size, leading to a significant increase in the size of primal-MOPEC subproblem. To address this issue, a stage-based decomposition method is developed. This method decomposes the large-scale subproblem into smaller subproblems indexed by time stages. By dividing the problem in this way, computational efficiency is improved, making it more manageable to solve large-scale stochastic MOPECs with risk-averse players. The details of this stage-based decomposition approach will be further discussed in subsequent chapter 5.

The aim of this chapter is to address the challenging problem of solving multistage stochastic MOPEC with risk-averse players by proposing a new scheme based on risk decomposition and its variants (incorporating proximal terms). The first contribution of this chapter is to demonstrate the utilization of the risk decomposition algorithm in achieving the solution to the problem under certain specific conditions, as well as its variant incorporating proximal terms in solving the general case. Additionally, this chapter investigates and presents new theoretical properties of the problem. The third contribution is the demonstration of using the risk decomposition as an effective initial point setting strategy for the PATH solver when solving multistage stochastic MOPEC with risk-averse players. The proposed methods are tested on the problem instances presented in chapter 2 and the results indicate their effectiveness and superiority over the performance of the PATH solver.

The chapter is organized as follows: Section 4.1 provides a brief literature review of the progress made in solving stochastic MOPECs with risk-averse players in past decade. Section 4.2 recalls the problem using *equilibrium reformulation* addressed in chapter 1 and discusses its advantages over *conjugate-based reformulation*. However, it also highlights the drawbacks of solving the problem as a whole system. Due to the inadequacy of traditional methods attempting to solve the problem as a whole, a new primal-MOPEC-dual-risk algorithm is developed in section 4.3, which exploits the inherent structure in the problem using the *equilibrium reformulation* and decomposes it into two smaller and more manageable subproblems that can be solved separately and iteratively. Building upon the main framework of the primal-MOPEC-dual-risk algorithm, section 4.4 explores various enhancement techniques aimed at improving the practical performance of the algorithm. In section 4.3.2 and 4.4.3, a comparative analysis is conducted, evaluating the success rates of different algorithms from section 4.3 and 4.4 on problem instances from chapter 2. The statistical analysis investigates the properties of the solution under changing risk measure and the quadratic terms in objective functions. Finally, section 4.5 provides a comprehensive conclusion for this chapter, highlighting the advantages of our algorithm in solving such problems compared to traditional methods.

4.1 Literature review

In this chapter, a risk decomposition algorithm and its variants are presented to solve the multistage stochastic MOPEC with risk-averse players using an *equilibrium reformulation*. The algorithm focuses on multistage stochastic MOPEC problems with risk-averse players and complements the player-based decomposition algorithm and its variant presented in the previous chapter, which solves a specific type of problem: the price-incentive Nash equilibrium problem (PNEP) with risk-averse players. However, that approach is not capable of handling the more general stochastic MOPEC with risk-averse players, which has numerous applications in fields such as power system [40], electricity market [63], economics [23] and transportation [2].

There has been significant effort to find an efficient algorithm to solve the multistage stochastic MOPEC with risk-averse players. The PATH solver [25], which is a powerful tool for solving deterministic MOPEC and stochastic MOPEC with risk-neutral players, has been found unreliable in solving stochastic MOPEC with risk-averse players. A sequential approximation approach [60] has been proposed, but it requires solving a problem of the same size as the original problem in each iteration. Decomposition methods have gained increasing interest due to the large scale of the scenario tree in multistage problems.

In general, there are four types of decomposition algorithms: player-based, time-stage-based, scenario-based, and risk-based. Player-based decomposition has been extensively studied in many literatures and shown to be efficient for *potential games* in the deterministic setting, but it cannot lead to an equilibrium point for *non-potential games*. Scenario-based decomposition [76] has been shown to be an efficient method for large-scale multistage stochastic MOPEC, but it can only handle problems with a global monotonicity property and risk-neutral players. To the best of our knowledge, there are no time-stage-based or risk-based decomposition methods for this type of problem.

4.2 Model and structure

This chapter presents a risk-based decomposition algorithm called primal-MOPEC-dual-risk decomposition algorithm. The algorithm is specifically designed to address the multi-stage stochastic MOPEC with risk-averse players by employing an *equilibrium reformulation*. To establish a contextual understanding, it is important to recall the fundamental definitions outlined in section 1.5.1 of chapter 1. By revisiting these definitions, we can establish a comprehensive perspective on the problem at hand and its associated challenges.

Each player $a \in \mathcal{A}$ is trying to solve the following optimization problem with given $(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}, \boldsymbol{\mu}_{a\cdot})$, where $\boldsymbol{\mu}_{a\cdot} = (\mu_{an})_{n\in\bar{\mathcal{S}}(1)}$:

$$
\begin{aligned}
P_a^{\mathcal{F}}(\boldsymbol{x}_{-a\cdot}, \boldsymbol{p}, \boldsymbol{\mu}_{a\cdot}): \quad \min_{\boldsymbol{x}_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \sum_{n\in 1+} \mu_{an} \cdot \Big[f_{an}(x_{an}; x_{-an}, p_n) \\
& \qquad + \sum_{m\in n+} \mu_{am} \cdot \big[f_{am}(x_{am}; x_{-am}, p_m) + \cdots \big] \Big] \\
\text{s.t.} \quad & \mathcal{G}_{an}(x_{a,n_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, \qquad \forall n \in \mathcal{N} \\
& x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N}
\end{aligned}
\tag{4.1}
$$

For each player $a \in \mathcal{A}$ and scenario node n, there exists one addtional player who is controlling the vector $\mu_{an_+} = (\mu_{am})_{m\in n_+}$ and trying to solve the following optimization problem:

$$
R_{an}(\boldsymbol{x}, \boldsymbol{p}, \mu_{a\bar{\bar{\mathcal{S}}}(n)}): \quad \max_{\mu_{an_+}\in\mathcal{D}_{an}} \sum_{m\in n+} \mu_{am} \cdot \Big[f_{am}(x_{am}; x_{-am}, p_m) + \sum_{l\in m_+} \mu_{al} \cdot \big[f_{al}(x_{al}; x_{-al}, p_l) + \cdots \big] \Big]
\tag{4.2}
$$

with given $\boldsymbol{x}$, $\boldsymbol{p}$ and $\mu_{a\bar{\bar{\mathcal{S}}}(n)} := (\mu_{al})_{l\in\bar{\bar{\mathcal{S}}}(n)}$.

The market price at node n is p_n, taking all players' strategies at node $\boldsymbol{x}_{\cdot n} = (x_{an})_{a\in\mathcal{A}}$ as

given, satisfies the equilibrium constraint:

$$0 \in F_n(p_n; \boldsymbol{x}_{\cdot n}) + \mathcal{N}_{K_n}(p_n) \tag{4.3}$$

$Q_n^{VI}(\boldsymbol{x}) \coloneqq \{p_n | p_n \in K_n, p_n \text{ solves } (4.3)\}$ denotes the set of market price p_n that safisfies the equilibrium contraint at scenario node n.

From definition 1.12, $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*) = \left((x^*_{an})_{a\in\mathcal{A}, n\in\mathcal{N}}, (p^*_n)_{n\in\mathcal{N}}, (\mu^*_{an})_{a\in\mathcal{A}, n\in\bar{\mathcal{S}}(1)}\right)$ is an equilibrium point of the above multistage stochastic MOPEC with risk-averse players using an *equilibrium reformulation* if and only if

$$\begin{array}{rlll} \boldsymbol{x}^*_{a\cdot} & \text{is an optimal solution of} \quad P_a^{\mathcal{F}}(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*, \boldsymbol{\mu}^*_{a\cdot}) & \text{for all } a \in \mathcal{A} & \\ p^*_n & \text{belongs to set} \quad Q_n^{VI}(\boldsymbol{x}^*) & \text{for all } n \in \mathcal{N} & (4.4) \\ \mu^*_{an_+} & \text{is an optimal solution of} \quad R_{an}(\boldsymbol{x}^*, \boldsymbol{p}^*, \mu^*_{a,\bar{\bar{\mathcal{S}}}(n)}) & \text{for all } a \in \mathcal{A}, n \in \mathcal{N}\backslash\mathcal{L} & \end{array}$$

4.3 Primal-MOPEC-dual-risk decomposition algorithm

The integration of the multistage stochastic MOPEC with risk-averse players, through the utilization of an *equilibrium reformulation*, introduces numerous challenges. To effectively address these challenges, we propose a decomposition method. The primary objective of this method is to compute the risk measure probability vector, denoted as $\boldsymbol{\mu}$, independently from the stochastic MOPEC. The solution process entails iteratively solving the stochastic MOPEC with a fixed risk measure probability vector $\boldsymbol{\mu}$, followed by the resolution of the worst-case model, which relies on the fixed strategies of the players and the prevailing market prices. Notably, when $\boldsymbol{\mu}$ is held fixed, the stochastic MOPEC can often be approached as a comparatively simpler problem, amenable to solution techniques such as the PATH solver. In essence, the primal-MOPEC-dual-risk decomposition algorithm facilitates the partitioning of the original problem (4.4) into two distinct subproblems:

- Primal-MOPEC $\mathcal{F}(\boldsymbol{\mu}^*)$:

$$\begin{array}{rlll} \text{With given } \boldsymbol{\mu}^*, \text{ find} & (\boldsymbol{x}^*, \boldsymbol{p}^*) \text{ such that} & & \\ \boldsymbol{x}^*_{a\cdot} & \text{is an optimal solution of} \quad P_a^{\mathcal{F}}(\boldsymbol{x}^*_{-a\cdot}, \boldsymbol{p}^*, \boldsymbol{\mu}^*) & \text{for all } a \in \mathcal{A}. & \\ p^*_n & \text{belongs to set} \quad Q_n^{VI}(\boldsymbol{x}^*) & \text{for all } n \in \mathcal{N}. & (4.5) \end{array}$$

- Dual-risk problem $\mathcal{B}(\boldsymbol{x}^*, \boldsymbol{p}^*)$:

$$\begin{aligned} \text{With given } (\boldsymbol{x}^*, \boldsymbol{p}^*), \text{ find} \quad & \boldsymbol{\mu}^* \text{ such that} \\ \mu^*_{an_+} \quad & \text{is an optimal solution of} \quad R_{an}(\boldsymbol{x}^*, \boldsymbol{p}^*, \mu^*_{a,\bar{\bar{\mathcal{S}}}(n)}) \\ & \text{for all } a \in \mathcal{A}, n \in \mathcal{N} \backslash \mathcal{L} \end{aligned} \tag{4.6}$$

Starting at iteration $r \geq 1$ with an initial point $(\boldsymbol{x}^{r-1}, \boldsymbol{p}^{r-1}, \boldsymbol{\mu}^{r-1})$, the primal-MOPEC is first solved using the given risk measure probability vector $\boldsymbol{\mu}^{r-1}$. This yields the updated vectors of players' actions $\boldsymbol{x}^r$ and price $\boldsymbol{p}^r$, resulting in a simpler MOPEC problem with a reduced problem size compared to the original comprehensive problem. Typically, this type of problem can be effectively addressed by employing the PATH solver. Subsequently, the updated risk measure probability vector $\boldsymbol{\mu}^r$ is determined by solving the dual-risk, considering the obtained $(\boldsymbol{x}^r, \boldsymbol{p}^r)$. This iterative process is repeated until a point $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^r)$ is attained, satisfying the predetermined stopping criterion.

The utilization of such a decomposition approach offers two principal advantages over directly solving the entire system. Firstly, in the primal-MOPEC phase, the fixation of $\boldsymbol{\mu}$ results in a reduction of the problem's complexity. Specifically, it transforms the highly nonlinear complementarity problem into a more manageable nonlinear complementarity problem, and in certain cases, it may even simplify the problem to a linear complementarity problem, particularly when the objective function of each player is quadratic and the functions $\mathcal{G}$ and F exhibit linearity. This reduction in complexity enables more efficient and tractable solution procedures. The second advantage lies in the separability of the dual-risk calculation. The risk-averse problem $R_{an}(\boldsymbol{x}^*, \boldsymbol{p}^*, \mu^*_{a,\bar{\bar{\mathcal{S}}}(n)})$ can be solved by focusing solely on the value of $\boldsymbol{\mu}$ for player a and the nodes within the set $\bar{\bar{\mathcal{S}}}(n)$. This feature provides the flexibility to calculate the value of $\boldsymbol{\mu}$ in a reverse manner, following the reverse order of the time stage $\mathcal{T}$. Consequently, an algorithmic approach can be devised to obtain the values of the entire risk measure probability vector $\boldsymbol{\mu}$, leveraging the known values of $\boldsymbol{x}$ and $\boldsymbol{p}$.

By capitalizing on these advantages, the proposed algorithm facilitates an efficient and effective solution process for the multistage stochastic MOPEC with risk-averse players. The iterative nature of the algorithm, along with the partitioning of the problem and the utilization of suitable solvers, enhances computational efficiency and enables the attainment of viable solutions within a reasonable timeframe. We could have the following algorithm to get value of the entire $\boldsymbol{\mu}$ with given $\boldsymbol{x}$ and $\boldsymbol{p}$.

Algorithm 4 Algorithm for the dual-risk $\mathcal{B}(\boldsymbol{x}^*, \boldsymbol{p}^*)$

Input and Initialization Players' strategies $\boldsymbol{x}^*$ and market prices $\boldsymbol{p}^*$.
Solving dual-risk
for $t = |\mathcal{T}| - 1, \ldots, 1$ **do**
  for $a \in \mathcal{A}$ **do**
    for $n \in \mathcal{N}(t)$ **do**
      Compute $\mu^*_{an_+}$ that is the solution of risk-averse problem $R_{an}(\boldsymbol{x}^*, \boldsymbol{p}^*, \mu^*_{a,\bar{\bar{\mathcal{S}}}(n)})$.
    end for
  end for
end for

Based on the use of Algorithm 4 to calculate $\boldsymbol{\mu}$, the complete primal-MOPEC-dual-risk decomposition iterative algorithm is as follows:

Algorithm 5 Primal-MOPEC-dual-risk decomposition algorithm

Input and Initialization Choose $\boldsymbol{\mu}^1 = \mathbb{P}$, where $\mathbb{P}$ is vector for risk-neutral probabilities for each scenario node $n \in \mathcal{N}$ and $\mathbb{P}_n = \prod_{m \in \mathcal{P}(n)} \phi_m$. Set $r = 1$.
Step 1: Primal-step Solve the primal-MOPEC $\mathcal{F}(\boldsymbol{\mu}^r)$, get $(\boldsymbol{x}^r, \boldsymbol{p}^r)$ such that

$$\begin{array}{lll} \boldsymbol{x}^r_a & \text{is an optimal solution of} \quad P^{\mathcal{F}}_a(\boldsymbol{x}^r_{-a}, \boldsymbol{p}^r, \boldsymbol{\mu}^r) & \text{for all } a \in \mathcal{A} \\ \boldsymbol{p}^r & \text{belongs to set} \quad Q^{\mathcal{M}}(\boldsymbol{x}^r) & \end{array} \tag{4.7}$$

Step 2: Dual-step.
for $t = |\mathcal{T}| - 1, \ldots, 1$ **do**
  for $a \in \mathcal{A}$ **do**
    for $n \in \mathcal{N}(t)$ **do**
      Compute $\mu^{r+1}_{an_+}$ that is the solution of risk-averse problem $R_{an}(\boldsymbol{x}^r, \boldsymbol{p}^r, \mu^{r+1}_{a,\bar{\bar{\mathcal{S}}}(n)})$.
    end for
  end for
end for
Step 3: Stopping Test. If stopping criteria is met, stop and go to step 5.
Step 4: Loop. Set $r := r + 1$ and go back to Step 1.
Step 5: PATH solve (optional). Use the PATH solver to solve the whole problem with initial point $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$. Get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*)$ if the solution is found.

4.3.1 Stopping criteria

The selection of an appropriate stopping criterion in step 3 of the primal-MOPEC-dual-risk Algorithm 5 holds significant importance in practical applications, as it can significantly impact the algorithm's overall performance. The choice of a stopping criterion is highly dependent on the specific requirements and constraints present in real-world scenarios. Given

the flexibility in choosing the stopping criterion, multiple approaches can be adopted. One commonly employed stopping criterion is based on the Fischer Burmeister (FB) function Ψ, introduced in subsection 1.2.1, serving as a merit function for evaluating the convergence of the point $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$. This merit function enables the assessment of the solution's quality by quantifying the distance between the obtained solution and the ideal equilibrium solution. In practice, the algorithm is halted when the merit value $\Psi(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$ falls below a predefined threshold value. This threshold value serves as an indicator of the desired level of convergence, allowing the algorithm to terminate once an acceptable solution is achieved.

Although the primal-MOPEC-dual-risk decomposition Algorithm 5 provides a systematic approach for decomposing the problem into more manageable subproblems, its convergence to the equilibrium point of the problem is not guaranteed. As detailed in Chapter 2, it is possible to construct examples where the sequence $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^r)_{r=1}^{\infty}$ fails to converge. Despite this limitation, empirical results presented later illustrate that the Risk decomposition Algorithm 5 is effective in solving a majority of instances when compared to the PATH solver.

In the numerical experiments, even in cases where the Algorithm 5 does not yield a convergent series, it is still possible to obtain $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$ for a certain iteration r, which can serve as a promising initial point for the PATH solver. In such situations, a modified termination criterion can be employed. The algorithm can be stopped at iteration r if $\Psi(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1}) \geq \Psi(\boldsymbol{x}^{r-1}, \boldsymbol{p}^{r-1}, \boldsymbol{\mu}^r)$, and $(\boldsymbol{x}^{r-1}, \boldsymbol{p}^{r-1}, \boldsymbol{\mu}^r)$ can be chosen as the initial point for the PATH solver to solve the entire system. By leveraging this approach, empirical results in later sections will demonstrate how this technique enhances the success rate of the PATH solver in solving problem (4.4).

While convergence to the equilibrium point cannot be guaranteed in all cases, the empirical findings indicate the effectiveness of the Risk decomposition Algorithm 5 in addressing a wide range of instances. Moreover, by incorporating the PATH solver and leveraging the obtained intermediate points, the algorithm provides a practical strategy for improving the convergence and success rate of the overall solution process. These empirical results serve as valuable insights, shedding light on the algorithm's performance characteristics and its ability to tackle the complexities of the multistage stochastic MOPEC with risk-averse players.

4.3.2 Numerical Experiments

In order to assess the effectiveness of the primal-MOPEC-dual-risk decomposition approach and its variants, as proposed in section 4.3 and 4.4, a comprehensive series of computational experiments was conducted. The objective of these experiments was to rigorously evaluate the performance and efficacy of the proposed algorithms in addressing the complexities of the stochastic MOPEC with risk-averse players.

The proposed algorithms were implemented using the GAMS modeling software. The experiments were carried out on a single Intel Xeon server, which was equipped with 36 cores and boasted a processing power of 3.1 GHz, accompanied by a substantial 768 GB of RAM.

Throughout the experiments, a standardized FB residual tolerance of 10^{-6} was employed as a benchmark for determining the successful solution of the problem. This tolerance criterion served as a reliable metric for evaluating the convergence and solution quality, enabling consistent comparisons and objective assessments across different problem instances and algorithmic approaches.

By adopting this rigorous experimental framework, the aim was to provide a comprehensive and unbiased evaluation of the proposed primal-MOPEC-dual-risk decomposition approach and its variants. The experimental results, which will be presented in subsequent sections, reflect the performance characteristics and effectiveness of the algorithms in addressing the challenges posed by the multistage stochastic MOPEC with risk-averse players.

4.3.2.1 Problem parameters setting

To ensure a thorough evaluation, the impact of these approaches was tested across all three problem instance classes, encompassing a diverse set of scenarios characterized by the three types of market constraints defined in chapter 2. However, to focus on the important parts, we will mainly show the numerical results on the economic dispatch examples, which have the most represtative results and are already enough for us to get the main conclusions. For the other two examples, we will show the summary results and put the detail results in the appendix. To introduce variability and account for different problem settings, varying problem parameters were randomly generated. This systematic testing methodology allowed for a comprehensive exploration of the algorithms' performance across a wide range of problem configurations. Hence, we first present more details of these examples. In the economic dispatch example, we assume that the objective functions of players have the same algebraic representation as described in section 3.3.2. Specifically, the objective

function of player a in the economic dispatch example follows the format (3.26) and the objective function of player a in the capacity expansion example follows the format (3.27).

For the economic dispatch example, the parameters c^g_{ain} are uniformly distributed over the interval $[10, 20]$. Additionally, the parameters c^s_{ain} and c^v_{ain} are set to zero for all $a \in \mathcal{A}$, $i \in \mathcal{I}$, and $n \in \mathcal{N}$. The demand D_{in} and production capacity W_{ai} follow the same distribution as specified in section 3.3.2. The lower bound of storage $\bar{v}^l_{ai}$ and upper bound of storage $\bar{v}^u_{ai}$ remain constant, as described in section 3.3.2. For the *Type II* and *Type III* market constraints, the parameter γ_{in} is fixed at 10^{-2} for each $i \in \mathcal{I}$ and $n \in \mathcal{N}$.

In the context of the capacity expansion example, it is worth noting that the coefficient c^s_{ain} is subject to a uniform distribution ranging between the values of 10 and 20. Similarly, the coefficient c^u_{an} is uniformly distributed within the range of 35 to 45. Pertaining to the supply-demand market constraint, the demand variable D_{in} is uniformly distributed between the intervals of 40 multiplied by the cardinality of set $\mathcal{A}$, and 80 multiplied by the cardinality of set $\mathcal{A}$. It is important to mention that the parameter ψ_n is consistently set to the value of 1 for each node $n \in \mathcal{N}$. As for the *Type II* and *Type III* market constraints, the parameter γ_{in} remains fixed at a value of 10^{-2}.

In both the economic dispatch example and the capacity expansion example, it is assumed that the number of players, denoted by $|\mathcal{A}|$, is fixed at 2. Furthermore, the scenario involves a single location, represented by $|\mathcal{I}|$, which is also equal to 1.

In the hydroelectricity example, it is important to highlight that the objective function associated with player a follows a specific form. The details pertaining to this specific form will be elaborated upon in subsequent sections.

$$\begin{aligned} C_{in}(\boldsymbol{u}_{i\cdot n}, v_{in}) &= \sum_{j\in\xi_i}(\frac{1}{2}\cdot\epsilon\cdot u^2_{ijn} + c^u_{ijn}\cdot u_{ijn}) \\ g_{in}(\boldsymbol{u}_{i\cdot n}) &= \sum_{j\in\xi_i} c^g_{ijn}\cdot u_{ijn} \end{aligned} \tag{4.8}$$

For the hydroelectricity example, the coefficient c^u_{ijn} is uniformly distributed between $[10, 20]$. Similarly, the coefficient c^g_{ijn} follows a uniform distribution between $[0.85, 1]$. In the supply-demand market constraint, the demand D_{in} is uniformly distributed between $[50, 150]$, reflecting the variability in the electricity demand. The parameter ϖ_{in} is also uniformly distributed between $[50, 90]$, representing the variability in the water inflow for hydroelectric power generation. Furthermore, the parameter T_{ijn} is fixed at 1. In the *Type II* and *Type III* market constraints, the parameter γ_{in} is fixed at 10^{-2}, implying a predetermined level of risk tolerance.

4.3.2.2 Numerical results

In these sections, we are testing the performance of following methods:

- PATH: Solving the problem by utilizing the PATH solver with an arbitrary initial point and the default setting.
- PATH-RN: Solving the problem by employing the PATH solver, utilizing the solution obtained from the risk-neutral case and the default setting.
- PD: Solving the problem by utilizing the primal-MOPEC-dual-risk decomposition algorithm.
- PD-PATH: Solving the problem by employing the primal-MOPEC-dual-risk decomposition algorithm, followed by empolying the PATH solver.

The subsequent experiments are designed to evaluate the performance of the aforementioned four methods in three distinct examples with varying market constraints. These evaluations utilize the risk measure $\overline{CVaR}(\lambda, \varphi)(\cdot)$. To systematically investigate the impact of two key parameters, namely ϵ and λ, on the success rate of the algorithms, controlled variations of these parameters are employed in the experiments. Specifically, for each fixed value of ϵ and λ, the experiments are conducted using two different values of the parameter φ and employing 16 distinct random seeds.

Before we talk more about the numerical results. We want to first give the defintion of a mixed solution of a stochastic MOPEC with risk-averse players using an *equilibrium reformulation*.

Definition 4.1. The solution of the stochastic MOPEC with risk-averse players using an *equilibrium reformulation* (4.4) with polyhedron risk sets $\{\mathcal{D}_{an} | n \in \mathcal{N} \backslash \mathcal{L}\}$ is called a mixed solution if for some $a \in \mathcal{A}$ and $n \in \mathcal{N} \backslash \mathcal{L}$ the probability vector μ_{an_+} is not an extreme point of the polyhedral risk set $\mathcal{D}_{an}$.

Tables 4.1 to 4.3 present the performance of the four methods across the economic dispatch example, which exhibits diverse market constraints. Additionally, these tables depict the relationship between problem difficulty and the percentage of solutions containing a probability vector $\boldsymbol{\mu}_{an_+}$ that is not an extreme point of the corresponding risk set $\mathcal{D}_{an}$ for certain values of $a \in \mathcal{A}$ and $n \in \mathcal{N}$. In the tables, the first two columns represent the fixed values of ϵ and λ utilized in the numerical experiments. Subsequently, the third to eighth columns display the successful rates (SR) and worst elapse time (seconds) of each

approach across a total of 32 experiments. Finally, the last column indicates the percentage of solutions that are not a mixed solution.

Let's first focus on the performances of the PATH and PATH-RN approaches, which are shown in the column 3 to column 6 in Tables 4.1. The analysis of this table reveals that both the PATH and PATH-RN approaches exhibit poor performance across all the problems considered. Additionally, for both approaches, the difficulty of the problem generally increases as the parameter λ becomes larger while keeping ϵ constant. This observation aligns with expectations, as larger values of λ result in larger risk sets $\mathcal{D}_{an}$ and greater risk aversion among the players, consequently rendering the problem more challenging. It is also observed that when λ is fixed, the problem will be harder for the PATH and PATH-RN when ϵ becomes smaller and closer to 0. Especially when $\epsilon = 0$, each player's optimization problem will be a linear programming problem, which doesn't guarantee an unique solution and thus makes the problem more difficult.

The performance of PD approach shows a very different trend compared to the performance of the PATH and PATH-RN approaches on these experiments. Although the PD approach doesn't have a good performance when $\epsilon \leq 10^{-2}$ and performs worse than PATH and PATH-RN when $\epsilon = 10^{-1}$ and $\lambda \leq 0.3$, it outperforms the PATH and PATH-RN in most cases when $\epsilon \geq 1$, even if λ is really large and both PATH and PATH-RN performs really poorly, which shows the PD approach has very different approach compared to the PATH solver.

In fact, we observed that the performance of the PD approach is really connected to one special property of the problem, which is the percentage of solutions in which all probability vectors $\boldsymbol{\mu}_{an_+}$ are extreme points of the corresponding risk set $\mathcal{D}_{an}$ for certain values of $a \in \mathcal{A}$ and $n \in \mathcal{N}$. It is observed that when $\epsilon \geq 10^{-2}$, then the succesful rate of PD is equivalent to the percentage of solutions in which all probability vectors $\boldsymbol{\mu}_{an_+}$ are extreme points of the corresponding risk set $\mathcal{D}_{an}$ for certain values of $a \in \mathcal{A}$ and $n \in \mathcal{N}$. According to this phenomenon we can see that it is highly possible that the PD approach could find the solution if the solution isn't a mixed one. It is reasonable to explain why it is difficult for the PD approach to find a none mixed one. Since in the dual-risk problem, we are solving a series of linear programming problem, the simplex solver like CPLEX or others will only give an extreme point solution even if the solution set is not a singleton in linear programming. Thus it is impossible for the PD approach to find the solution in this situation.

However, the PD approach not only solves the majority of problems when all the included probability vectors $\boldsymbol{\mu}_{an_+}$ are extreme points of the corresponding risk sets $\mathcal{D}_{an}$, but it also provides an excellent initial point for the PATH solver. This is evident from the

results presented in Table 4.1, where the PD-PATH approach consistently exhibits the best performance among the four approaches. Furthermore, the numerical results demonstrate that the PD-PATH approach solves all problem instances in economic dispatch problem with *Type III* market constraint. The column of time shows that the PD-PATH approach although takes more time to solve the problem but the elapsed time is not a big increase compared to the PATH solver.

ϵ	λ	PATH		PATH-RN		PD	PD-PATH		No mixed solution
		SR(%)	Time(s)	SR(%)	Time(s)	SR(%)	SR(%)	Time(s)	percentage(%)
0	0.1	100.0	1.4	100.0	0.3	0.0	96.9	7.8	12.5
0	0.3	37.5	1.9	68.8	1.9	0.0	78.1	8.6	6.2
0	0.5	0.0	1.9	9.4	2.2	3.1	59.4	7.6	6.5
0	0.7	0.0	1.8	0.0	2.1	0.0	18.8	6.4	0.0
0	0.9	0.0	2.4	0.0	2.4	0.0	3.1	9.0	0.0
1e-2	0.1	100.0	0.4	100.0	0.9	34.4	100.0	6.8	34.4
1e-2	0.3	68.8	1.5	96.9	2.1	6.2	100.0	7.5	6.2
1e-2	0.5	9.4	1.9	68.8	3.7	3.1	90.6	7.6	3.1
1e-2	0.7	3.1	2.7	3.1	2.3	0.0	71.9	6.7	0.0
1e-2	0.9	0.0	2.4	0.0	2.3	0.0	37.5	6.8	0.0
1e-1	0.1	100.0	1.0	100.0	0.6	78.1	100.0	6.6	78.1
1e-1	0.3	71.9	0.4	100.0	1.9	68.8	100.0	6.8	68.8
1e-1	0.5	31.2	1.9	90.6	2.0	46.9	100.0	6.9	46.9
1e-1	0.7	9.4	2.2	87.5	2.2	37.5	100.0	7.2	37.5
1e-1	0.9	0.0	2.5	18.8	2.2	34.4	100.0	6.7	34.4
1	0.1	100.0	0.3	100.0	0.7	100.0	100.0	4.0	100.0
1	0.3	100.0	0.3	100.0	0.8	100.0	100.0	3.7	100.0
1	0.5	50.0	0.5	100.0	2.0	100.0	100.0	4.1	100.0
1	0.7	25.0	2.5	46.9	2.3	96.9	100.0	6.8	96.9
1	0.9	3.1	2.4	9.4	2.4	100.0	100.0	4.0	100.0
10	0.1	100.0	0.4	100.0	0.7	100.0	100.0	3.1	100.0
10	0.3	100.0	1.7	28.1	0.8	100.0	100.0	3.9	100.0
10	0.5	100.0	2.2	18.8	0.9	96.9	100.0	6.2	96.9
10	0.7	56.2	2.5	3.1	2.8	100.0	100.0	3.8	100.0
10	0.9	3.1	2.4	0.0	2.6	96.9	100.0	6.2	96.9

Table 4.1: Performance of PATH, PATH-RN, PD and PD-PATH over economic dispatch example with *Type I* market constraint on scenario tree 2

Table 4.2 shows the results of the above four algorithms when solving the problems with *Type II* market constraints. When the problem has *Type II* market constraints, the above conclusions achieved from the results of *Type I* market constraints also hold. It could be observed that in most cases the above four methods have higher successful rates when solving problems with *Type II* market constraints compared to problems with *Type I* market constraints.

ϵ	λ	PATH		PATH-RN		PD	PD-PATH		No mixed solution
		SR(%)	Time(s)	SR(%)	Time(s)	SR(%)	SR(%)	Time(s)	percentage(%)
0	0.1	100.0	0.5	100.0	0.3	0.0	96.9	9.2	9.4
0	0.3	75.0	1.8	87.5	1.7	0.0	84.4	9.3	6.2
0	0.5	9.4	2.3	15.6	2.1	0.0	62.5	8.0	3.1
0	0.7	0.0	2.2	0.0	2.3	0.0	25.0	7.5	0.0
0	0.9	0.0	2.0	0.0	2.1	0.0	3.1	6.1	0.0
1e-2	0.1	100.0	0.4	100.0	0.3	28.1	100.0	7.0	28.1
1e-2	0.3	100.0	0.8	100.0	0.8	6.2	100.0	7.5	6.2
1e-2	0.5	65.6	1.8	78.1	2.4	3.1	96.9	7.7	3.1
1e-2	0.7	6.2	2.3	15.6	2.8	0.0	84.4	6.7	0.0
1e-2	0.9	0.0	2.1	0.0	2.8	0.0	43.8	8.6	0.0
1e-1	0.1	100.0	0.4	100.0	0.4	78.1	100.0	6.7	78.1
1e-1	0.3	100.0	0.4	100.0	1.6	68.8	100.0	6.7	68.8
1e-1	0.5	87.5	0.8	100.0	1.8	46.9	100.0	6.8	46.9
1e-1	0.7	59.4	1.5	96.9	2.3	34.4	100.0	6.5	34.4
1e-1	0.9	9.4	2.6	46.9	2.8	34.4	100.0	6.6	34.4
1	0.1	100.0	0.4	100.0	0.7	100.0	100.0	2.9	100.0
1	0.3	100.0	0.3	100.0	0.9	100.0	100.0	3.0	100.0
1	0.5	96.9	0.4	100.0	1.7	100.0	100.0	3.9	100.0
1	0.7	78.1	2.0	84.4	2.1	96.9	100.0	6.1	96.9
1	0.9	25.0	2.4	68.8	2.4	100.0	100.0	3.0	100.0
10	0.1	100.0	0.3	100.0	0.9	100.0	100.0	3.0	100.0
10	0.3	100.0	2.0	6.2	1.0	100.0	100.0	3.1	100.0
10	0.5	100.0	2.0	6.2	1.0	90.6	100.0	5.9	90.6
10	0.7	96.9	2.2	0.0	2.1	100.0	100.0	2.9	100.0
10	0.9	59.4	2.4	3.1	2.8	96.9	100.0	6.1	96.9

Table 4.2: Performance of PATH, PATH-RN, PD and PD-PATH over economic dispatch example with *Type II* market constraint on scenario tree 2

Table 4.3 shows the results of the above four algorithms when solving the problems with *Type III* market constraints. When the problem has *Type III* market constraints, not only the above conclusions achieved from the previous market constraints hold, but also the all four approaches have better results compared to the results when having *Type I* and *Type II* market constraints. In fact, the PD-PATH approach is able to solve all problem instances with *Type III* market constraints.

ϵ	λ	PATH		PATH-RN		PD	PD-PATH		No mixed solution
		SR(%)	Time(s)	SR(%)	Time(s)	SR(%)	SR(%)	Time(s)	percentage(%)
0	0.1	100.0	0.5	100.0	0.2	87.5	100.0	6.3	87.5
0	0.3	100.0	0.4	100.0	0.2	78.1	100.0	6.4	78.1
0	0.5	100.0	1.4	96.9	0.8	59.4	100.0	6.9	59.4
0	0.7	53.1	1.7	78.1	3.0	56.2	100.0	6.6	56.2
0	0.9	0.0	2.5	21.9	3.0	28.1	100.0	7.5	31.2
1e-2	0.1	100.0	0.3	100.0	0.5	96.9	100.0	6.4	96.9
1e-2	0.3	100.0	0.3	100.0	0.4	81.2	100.0	6.7	81.2
1e-2	0.5	100.0	0.5	100.0	1.0	56.2	100.0	6.6	56.2
1e-2	0.7	65.6	1.7	90.6	4.5	40.6	100.0	6.8	40.6
1e-2	0.9	6.2	2.3	43.8	3.4	31.2	100.0	7.5	31.2
1e-1	0.1	100.0	0.4	100.0	0.2	96.9	100.0	5.4	96.9
1e-1	0.3	100.0	0.3	100.0	0.4	87.5	100.0	6.7	87.5
1e-1	0.5	100.0	0.8	100.0	0.5	84.4	100.0	6.6	84.4
1e-1	0.7	100.0	2.1	90.6	1.0	84.4	100.0	7.0	84.4
1e-1	0.9	31.2	2.3	78.1	3.8	78.1	100.0	6.5	78.1
1	0.1	100.0	0.3	100.0	0.7	100.0	100.0	3.2	100.0
1	0.3	100.0	0.4	100.0	0.7	100.0	100.0	3.1	100.0
1	0.5	100.0	0.3	100.0	0.7	100.0	100.0	3.1	100.0
1	0.7	93.8	1.6	90.6	1.9	100.0	100.0	3.0	100.0
1	0.9	50.0	2.3	68.8	2.5	96.9	100.0	5.5	96.9
10	0.1	100.0	0.3	100.0	0.2	100.0	100.0	3.0	100.0
10	0.3	100.0	1.5	21.9	0.6	96.9	100.0	5.2	96.9
10	0.5	100.0	1.8	15.6	0.8	93.8	100.0	5.6	93.8
10	0.7	100.0	2.0	34.4	0.9	100.0	100.0	3.0	100.0
10	0.9	96.9	2.1	6.2	1.9	100.0	100.0	3.0	100.0

Table 4.3: Performance of PATH, PATH-RN, PD and PD-PATH over economic dispatch example with *Type III* market constraint on scenario tree 2

For the capacity expansion and hydroelectricity example, we are also running 32 independent results for each $\epsilon \in \{0, 10^{-2}, 10^{-1}, 1, 10\}$ and $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The detailed results are included in the appendix chapter 8. Here we only show the summary tables 4.4 to 4.5 for all experiments. For each market type there are all together $32 \times 25 = 800$ independent experiments.

The examination of tables 4.4 through 4.5 reveals consistent patterns in the numerical outcomes for the capacity expansion example and hydroelectricity example, akin to those observed in the economic dispatch example. Both the PD approach and PD-PATH approach exhibit significantly superior results compared to the PATH and PATH-RN approaches. Furthermore, it is worth noting that the efficacy of the PD approach exhibits a strong correlation with the absence of mixed solutions. Moreover, there is a high likelihood that the PD approach can successfully ascertain a solution for a stochastic MOPEC with risk-averse participants by employing an *equilibrium reformulation*.

Market Type	PATH SR(%)	PATH-RN SR(%)	PD SR(%)	PD-PATH SR(%)	No mixed solution percentage(%)
Type I	56.4	78.0	98.2	100.0	98.4
Type II	64.8	82.4	97.9	100.0	97.9
Type III	89.9	93.2	77.4	100.0	78.0

Table 4.4: Summary table of performance of PATH, PATH-RN, PD and PD-PATH over capacity expansion example on scenario tree 2

Market Type	PATH SR(%)	PATH-RN SR(%)	PD SR(%)	PD-PATH SR(%)	No mixed solution percentage(%)
Type I	50.9	42.1	95.5	99.5	96.8
Type II	77.9	43.5	95.1	99.8	99.9
Type III	58.9	26.6	97.5	100.0	97.9

Table 4.5: Summary table of performance of PATH, PATH-RN, PD and PD-PATH over hydroelectricity example on scenario tree 2

4.4 Computational Enhancements

4.4.1 Primal-MOPEC-dual-risk decomposition algorithm with proximal-term

The preceding section highlights the observation that the primal-MOPEC-dual-risk decomposition Algorithm 5 may exhibit a cyclic behavior, thereby lacking an assurance of producing a series of convergent points. This limitation arises due to the Algorithm 5 employing an excessively aggressive strategy during the resolution of subproblems (4.5) and (4.6). To address this issue effectively, a viable approach involves incorporating the proximal operator while solving these subproblems, thereby facilitating a constrained movement.

Definition 4.2. Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function, the *proximal operator* $prox_f : \mathbb{R}^n \to \mathbb{R}^n$ of f is defined by

$$prox_f(v) = \arg\min_x (f(x) + \frac{1}{2}\|x - v\|_2^2) \tag{4.9}$$

where $\|\cdot\|_2$ is the usual Euclidean norm.

It is noteworthy that the function minimized on the right-hand side of equation (4.9) exhibits strong convexity, and for every $x \in \text{dom}(f) \subseteq \mathbb{R}^n$, the inequality $f(x) + \frac{1}{2}\|x - v\|_2^2 <$

$+\infty$ holds, ensuring the existence of a unique minimizer for each $v \in \mathbb{R}^n$. Consequently, $prox_f(v)$ invariably comprises a solitary point. The definition explicitly states that $prox_f(v)$ represents a point that strikes a balance between minimizing f and maintaining proximity to v. This point $prox_f(v)$ is also referred to as a *proximal point* of v with respect to f.

Moreover, the proximal term can be equivalently expressed in a scaled form with $\omega > 0$, adhering to the subsequent representation:

$$prox_{\omega f}(v) = \arg\min_x (f(x) + \frac{1}{2\omega}\|x - v\|_2^2)$$

The parameter ω can be interpreted as a relative weight or trade-off parameter governing the balance between the objective function f and the least norm term $\frac{1}{2\omega}||\cdot - v||_2^2$ in the minimization of $f(x) + \frac{1}{2\omega}||x - v||_2^2$. Notably, the choice of ω influences the emphasis placed on $\frac{1}{2\omega}||x - v||_2^2$ during the minimization process. Specifically, when ω assumes a small value, greater significance is attributed to the term $\frac{1}{2\omega}||x - v||_2^2$. Conversely, a larger value of ω results in a shift of emphasis.

Example (f as the Indicator function) When f is the indicator function

$$I_{\mathcal{C}}(x) = \begin{cases} 0 & x \in \mathcal{C} \\ +\infty & x \notin \mathcal{C}, \end{cases}$$

where $\mathcal{C}$ is a closed nonempty convex set, the proximal operator of f is equivalent to Euclidean projection onto set $\mathcal{C}$, which we denote

$$prox_{I_{\mathcal{C}}}(v) = \Pi_{\mathcal{C}}(v) = \arg\min_{x \in \mathcal{C}} ||x - c||_2$$

The proximal operators have been used commonly in optimization since there is a close connection between proximal operators and gradient methods. In fact, the proximal operator associated with function f can be seen as a form of gradient step specific to f. Consequently, it is possible to obtain

$$prox_{\omega f}(v) \approx v - \omega \nabla f(v)$$

when f is continuously differentiable and ω is small. It also indicates that ω plays a role similar to that of a step size in a gradient method.

Given the significance of the proximal term $\frac{1}{2\omega}||x - v||_2^2$ when ω assumes a small value, we have incorporated the proximal term into the Risk decomposition algorithm. Specifically, we augment the objective function of the problem $R_{an}(\boldsymbol{x}^r, \boldsymbol{p}^r, \mu^{r+1}a, \bar{\bar{\mathcal{S}}}(n))$ with the proximal

term $\sum_{m\in n_+}\frac{1}{2\omega_{am}}||\cdot-\mu^r_{am}||_2^2$. By introducing this additional term, the problems encountered during the backward step can be formulated in the following manner:

$$R^{\mathcal{P}}_{an}(\boldsymbol{x}^r,\boldsymbol{p}^r,\mu^{r+1}_{a,\bar{\bar{\mathcal{S}}}(n)},\mu^r_{an_+},\omega_{an_+}):\max_{\mu_{an_+}\in\mathcal{D}_{an}}\quad\sum_{m\in n_+}\mu_{am}\cdot\Big[f_{am}(x^r_{am};x^r_{-am},p^r_m)+\sum_{l\in m_+}\mu^{r+1}_{al}\cdot\big[\cdots\big]\Big]+\sum_{m\in n+}\frac{1}{2\omega_{am}}||\mu_{am}-\mu^r_{am}||_2^2 \tag{4.10}$$

where $\omega_{an_+}=(\omega_{am})_{m\in n_+}$. It is apparent that the inclusion of such a proximal term, with each component representing a small positive real number, enables us to enforce proximity between μ^{r+1}_{an} and μ^r_{an} for all $a\in\mathcal{A}$ and $n\in\mathcal{N}$. This particular mechanism significantly enhances the stability of the algorithm. Notably, we refrain from incorporating proximal terms on players' strategies $\boldsymbol{x}$ and market price $\boldsymbol{p}$. This decision stems from the fact that including such proximal terms would impede the convergence speed of the primal-MOPEC-dual-risk decomposition Algorithm 5 in practical scenarios, particularly when the risk-measure sets $\mathcal{D}_{an}$ are polyhedrons.

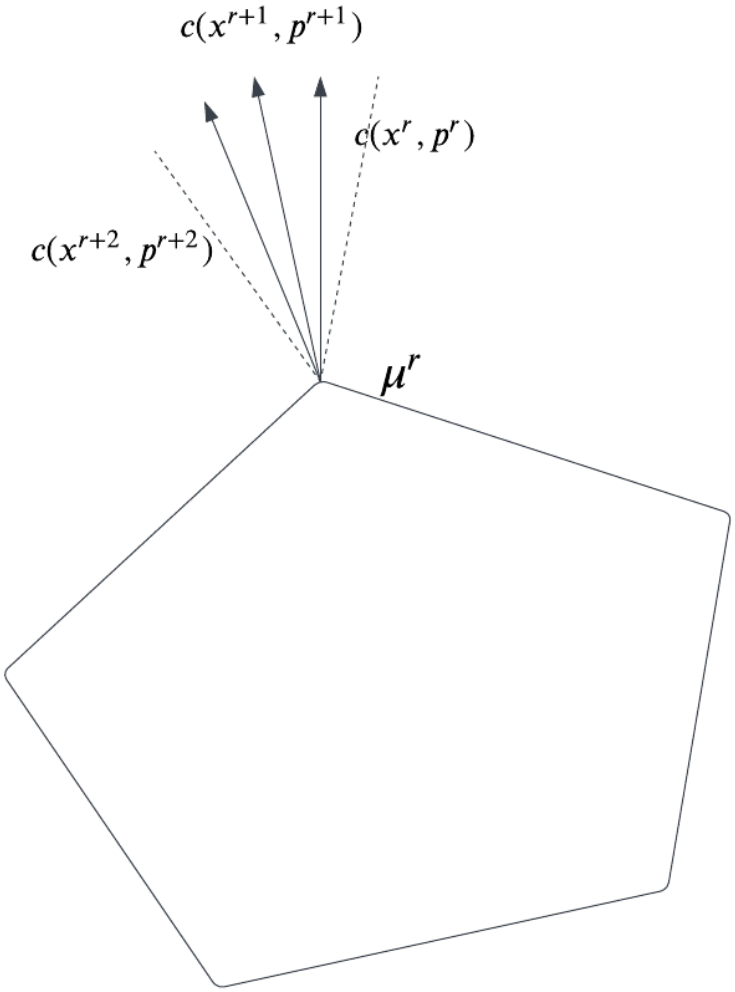


Figure 4.1: Proximal term on probability μ^r

Figure (4.1) illustrates the reason behind exclusively introducing a proximal term to the risk probabilities $\boldsymbol{\mu}$. In this context, let us consider addressing the following risk-averse problem while given $\boldsymbol{x}$ and $\boldsymbol{p}$:

$$\max_{\boldsymbol{\mu}\in\mathcal{D}} \boldsymbol{\mu}^T c(\boldsymbol{x}, \boldsymbol{p}), \tag{4.11}$$

where $c(\boldsymbol{x}, \boldsymbol{p})$ is the cost vector and $c(\cdot, \cdot)$ is continuously differentiable. We also assume that the risk set $\mathcal{D}$ is a polyhedron, which is a reasonable assumption since both the risk sets of $CVaR$ and mean-deviation risk-measures are polydedrons. At iteration r, we denote $(\boldsymbol{x}^r, \boldsymbol{p}^r)$ as the solution of the primal-MOPEC with risk probability $\boldsymbol{\mu}^r$, and the cost vector $c(\boldsymbol{x}^r, \boldsymbol{p}^r) \in relint\mathcal{N}_{\mathcal{D}}(\boldsymbol{\mu}^r)$, where $relint\mathcal{N}_{\mathcal{D}}(\boldsymbol{\mu}^r)$ represents the relative interior of $\mathcal{N}_{\mathcal{D}}(\boldsymbol{\mu}^r)$. If we introduce proximal terms to the objective of the primal-MOPEC, we obtain a new

solution $(\boldsymbol{x}^{r+1}, \boldsymbol{p}^{r+1})$ that is very close to the previous iterate solution $(\boldsymbol{x}^{r}, \boldsymbol{p}^{r})$. As a result, the cost vector $c(\boldsymbol{x}^{r+1}, \boldsymbol{p}^{r+1})$ is also very close to the cost vector $c(\boldsymbol{x}^{r}, \boldsymbol{p}^{r})$ and lies within the relative interior of $\mathcal{N}_{\mathcal{D}}(\boldsymbol{\mu}^{r})$. In this situation, the new risk probability $\boldsymbol{\mu}^{r+1}$ will be equal to the previous risk probability $\boldsymbol{\mu}^{r}$. Futhermore, if each component of the proximal scale ω_{an_+} is chosen to be too small, the risk probability will remain stuck at one extreme point for many iterations, which will significantly slow down the convergence speed of the algorithm in practice. Based on this phenomenon, we choose to utilize the proximal terms only on the dual-risk. The modified algorithm is as follows:

Algorithm 6 Primal-MOPEC-dual-risk decomposition algorithm with proximal term on dual-risk problem

Input and Initialization Choose $\boldsymbol{\mu}^1 = \mathbb{P}$, where $\mathbb{P}$ is vector for risk-neutral probabilities for each scenario node $n \in \mathcal{N}$ and $\mathbb{P}_n = \prod_{m \in \mathcal{P}(n)} \phi_m$, positive integers J and M, and the proximal parameters $\{\omega_{an}\}_{a \in \mathcal{A}, n \in \mathcal{N}}$. Set $r = 1$.
while $r \leq J$ **do**
  Step 1: Primal-step Solve the primal-MOPEC $\mathcal{F}(\boldsymbol{\mu}^r)$, get $(\boldsymbol{x}^r, \boldsymbol{p}^r)$ such that

$$\begin{array}{lll} \boldsymbol{x}_a^r & \text{is an optimal solution of} \quad P_a^{\mathcal{F}}(\boldsymbol{x}_{-a}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^r) & \text{for all } a \in \mathcal{A} \\ \boldsymbol{p}^r & \text{belongs to set} \quad Q^{\mathcal{M}}(\boldsymbol{x}^r) & \end{array} \tag{4.12}$$

  Step 2: Dual-step.
  for $t = |\mathcal{T}| - 1, \ldots, 1$ **do**
    for $a \in \mathcal{A}$ **do**
      for $n \in \mathcal{N}(t)$ **do**
        Compute $\mu_{an_+}^{r+1}$ that is the solution of risk-averse problem $R_{an}(\boldsymbol{x}^r, \boldsymbol{p}^r, \mu_{a,\bar{\mathcal{S}}(n)}^{r+1})$.
      end for
    end for
  end for
  Step 3: Stopping Test. If stopping criteria is met, break and go to **Step 5**.
  Step 4: Loop. Set $r := r + 1$.
end while
Step 5: PATH solve. Use the PATH solver to solve the whole problem with initial point $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$. Stop the algorithm and get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*)$ if the solution is found, else continue.
Step 6. Set $\boldsymbol{\mu}^1 = \boldsymbol{\mu}^{r+1}$ and $r = 1$.
while $r \leq M$ **do**
  Step 7: Primal-step Solve the primal-MOPEC $\mathcal{F}(\boldsymbol{\mu}^r)$, get $(\boldsymbol{x}^r, \boldsymbol{p}^r)$ such that

$$\begin{array}{lll} \boldsymbol{x}_a^r & \text{is an optimal solution of} \quad P_a^{\mathcal{F}}(\boldsymbol{x}_{-a}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^r) & \text{for all } a \in \mathcal{A} \\ \boldsymbol{p}^r & \text{belongs to set} \quad Q^{\mathcal{M}}(\boldsymbol{x}^r) & \end{array} \tag{4.13}$$

  Step 8: Dual-step.
  for $t = |\mathcal{T}| - 1, \ldots, 1$ **do**
    for $a \in \mathcal{A}$ **do**
      for $n \in \mathcal{N}(t)$ **do**
        Compute $\mu_{an_+}^{r+1}$ that is the solution of risk-averse problem $R_{an}^{\mathcal{P}}(\boldsymbol{x}^r, \boldsymbol{p}^r, \mu_{a,\bar{\mathcal{S}}(n)}^{r+1}, \mu_{an_+}^r, \omega_{an_+})$.
      end for
    end for
  end for
  Step 9: Stopping Test. If stopping criteria is met, break and go to **Step 11**.
  Step 10: Loop. Set $r := r + 1$.
end while
Step 11: PATH solve. Use the PATH solver to solve the whole problem with initial point $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$. Stop the algorithm and get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*)$ if the solution is found.

Although Algorithm 5 demonstrates excellent practical performance in most problems of this type, its convergence cannot be guaranteed. To address this issue, we have developed a modified version of the algorithm, referred to as Algorithm 6, which incorporates classical proximal terms. These terms help ensure that the risk probability vector $\boldsymbol{\mu}$ takes a more controlled step at each iteration, preventing overly aggressive updates.

The proximal parameters ω_{an} are employed to regulate the step size of the probability vector. A larger value of ω_{an} forces a smaller step size, while a smaller value of ω_{an} encourages a larger step size. However, this introduces a new question: how do we determine an appropriate value for ω_{an} to achieve the best practical performance? In reality, the answer depends on various specific properties of each problem, making it unrealistic to find a universally optimal value for each ω_{an}. If the value of ω_{an} is too small, the step size becomes too large, leading to algorithm divergence. Conversely, if the value of ω_{an} is too large, the convergence speed slows down due to excessively small step sizes at each iteration.

In fact, we have also tested the primal-MOPEC-dual-risk decomposition algorithm with proximal term on dual-risk problem and also got improved results compared to PD approach or PD-PATH approach. However, just like mentioned before, it is very difficult to choose an appropriate parameter ω in practice. Thus, we mainly focus on the PD-CC-PATH approach when doing the numerical experiments. The results of primal-MOPEC-dual-risk decomposition algorithm with proximal term on dual-risk problem is attached in the appendix chapter 8.

4.4.2 Primal-MOPEC-dual-risk decomposition algorithm with convex combination heuristics

To address the issue of slow convergence while maintaining the convergence properties of the algorithm, we propose a heuristic approach that relies on a convex combination of risk probabilities generated during the iteration process of Algorithm 5. The underlying idea is as follows: in Algorithm 5, if the algorithm fails to converge, we repeat the algorithm and generate a series of risk probability vectors $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \ldots, \boldsymbol{\mu}^r, \ldots$. Suppose we have completed r iterations, including the primal step and dual step, and obtained a set of risk probability vectors $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \ldots, \boldsymbol{\mu}^{r+1}$. Instead of using $\boldsymbol{\mu}^{r+1}$ as the given parameter in the primal-MOPEC at the start of iteration $r+1$, we generate a new risk probability vector $\hat{\boldsymbol{\mu}}^{r+1}$. This vector is a convex combination of the previous v risk probability vectors, specifically $\boldsymbol{\mu}^{r+2-v}, \ldots, \boldsymbol{\mu}^{r+1}$,

and can be computed as follows:

$$\hat{\boldsymbol{\mu}}^{r+1} = \sum_{q=0}^{v-1} w_q \cdot \boldsymbol{\mu}^{r-v+q} \tag{4.14}$$

where $\sum_{q=0}^{v-1} w_q = 1$ and $w_q \geq 0$ for each q represent fixed parameters that can be chosen by the user. In essence, this convex combination trick can be viewed as an averaging strategy applied to the previous risk probability vectors. The purpose of this strategy is to address the issue of the algorithm generating a divergent series of risk probability vectors, which occurs when the $\boldsymbol{\mu}$ moves too aggressively during iterations. By employing the averaging trick, we constrain the step sizes taken by the risk probability vector and utilize more information from the previous steps, particularly if we select a parameter $v \geq 3$.

After obtaining the average risk probability vector $\hat{\boldsymbol{\mu}}^{r+1}$, we follow the same framework as in Algorithm 5 to generate $(\boldsymbol{x}^{r+1}, \boldsymbol{p}^{r+1})$ through a *primal-step*. Subsequently, we generate $\boldsymbol{\mu}^{r+2}$ through a *dual-step*. As previously mentioned, this type of algorithm can achieve good practical performance with the assistance of the PATH solver. There are two ways to combine the PATH solver with the average steps plus primal-MOPEC-dual-risk iterations. The first option is to repeat the average steps plus primal-MOPEC-dual-risk iterations and use the solution achieved as the initial point for the PATH solver to solve. The second option is to incorporate a PATH solve after the *dual-step* in Algorithm 5.

Algorithm 7 Primal-MOPEC-dual-risk decomposition algorithm with convex combination on risk probability vector

Input and Initialization Choose $\boldsymbol{\mu}^1 = \mathbb{P}$, where $\mathbb{P}$ is vector for risk-neutral probabilities for each scenario node $n \in \mathcal{N}$ and $\mathbb{P}_n = \prod_{m \in \mathcal{P}(n)} \phi_m$, positive integer v, positive integer $J \geq v$ and positive integer $M \geq J$. Set $r = 1$.
while $r \leq J$ **do**
  Step 1: Primal-step Solve the primal-MOPEC $\mathcal{F}(\boldsymbol{\mu}^r)$, get $(\boldsymbol{x}^r, \boldsymbol{p}^r)$ such that

$$\begin{array}{lll} \boldsymbol{x}_a^r & \text{is an optimal solution of} \quad P_a^{\mathcal{F}}(\boldsymbol{x}_{-a}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^r) & \text{for all } a \in \mathcal{A} \\ \boldsymbol{p}^r & \text{belongs to set} \quad Q^{\mathcal{M}}(\boldsymbol{x}^r) & \end{array} \tag{4.15}$$

  Step 2: Dual-step.
  for $t = |\mathcal{T}| - 1, \ldots, 1$ **do**
    for $a \in \mathcal{A}$ **do**
      for $n \in \mathcal{N}(t)$ **do**
        Compute $\mu_{an_+}^{r+1}$ that is the solution of risk-averse problem $R_{an}(\boldsymbol{x}^r, \boldsymbol{p}^r, \mu_{a,\bar{\bar{\mathcal{S}}}(n)}^{r+1})$.
      end for
    end for
  end for
  Step 3: Stopping Test. If stopping criteria is met, break and go to **Step 5**.
  Step 4: Loop. Set $r := r + 1$.
end while
Step 5: PATH solve. Use the PATH solver to solve the whole problem with initial point $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$. Stop the algorithm and get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*)$ if the solution is found, else continue.
while $r \leq M$ **do**
  Set $\hat{\boldsymbol{\mu}}^r = \sum_{q=0}^{v-1} w_q \cdot \boldsymbol{\mu}^{r+1-v+q}$.
  Step 6: Primal-step Solve the primal-MOPEC $\mathcal{F}(\hat{\boldsymbol{\mu}}^r)$, get $(\boldsymbol{x}^r, \boldsymbol{p}^r)$ such that

$$\begin{array}{lll} \boldsymbol{x}_a^r & \text{is an optimal solution of} \quad P_a^{\mathcal{F}}(\boldsymbol{x}_{-a}^r, \boldsymbol{p}^r, \hat{\boldsymbol{\mu}}^r) & \text{for all } a \in \mathcal{A} \\ \boldsymbol{p}^r & \text{belongs to set} \quad Q^{\mathcal{M}}(\boldsymbol{x}^r) & \end{array} \tag{4.16}$$

  Step 7: Dual-step.
  for $t = |\mathcal{T}| - 1, \ldots, 1$ **do**
    for $a \in \mathcal{A}$ **do**
      for $n \in \mathcal{N}(t)$ **do**
        Compute $\mu_{an_+}^{r+1}$ that is the solution of risk-averse problem $R_{an}(\boldsymbol{x}^r, \boldsymbol{p}^r, \mu_{a,\bar{\bar{\mathcal{S}}}(n)}^{r+1})$.
      end for
    end for
  end for
  Step 8: Stopping Test. If stopping criteria is met, break and go to **Step 10**.
  Step 9: Loop. Set $r := r + 1$.
end while
Step 10: PATH solve. Use the PATH solver to solve the whole problem with initial point $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$. Stop the algorithm and get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*)$ if the solution is found.

4.4.3 Numerical experiments

We are conducting a series of computational experiments in order to evaluate the effectiveness of the variants of primal-MOPEC-dual-risk decompositionn approach. The experiments are implemented in the same computational setting in section 4.3.2.

4.4.3.1 Problem parameter setting

In the following experiments, we are using the same parameters setting as in subsection 4.3.2.1. The only difference is that we will also testing our algorithms on these type of problems on scenario tree 3 defined in chapter 2. The Table lists the problem size of the reformulated mixed complementarity problem of all of these problems based on scenario tree 3.

4.4.3.2 Numerical results

In this subsection, we are testing the performance of following methods:

- PD-CC-PATH: Solving the problem using primal-MOPEC-dual-risk decomposition algorithm with convex combination on risk probability vector.

In the following experiments, we first show the results of the PD-CC-PATH approach on all the previous problem instances on scenario tree 2 that cannot be solved by all the previous four methods. Table 4.6 shows that the PD-CC-PATH appraoch have better performance compared to PD-PATH, especially this approach achieves 100% succesful rate when $\epsilon \geq 10^{-2}$, which helps to solve more cases for this type of problem. The PD-CC-PATH also improves the succesful rates a lot compared to PD-PATH when $\epsilon = 0$. It is also shown in this table that the computational time of PD-CC-PATH is just increased a little bit compared to PD-PATH.

ϵ	λ	*Type I* market				*Type II* market			
		PD-PATH		PD-CC-PATH		PD-PATH		PD-CC-PATH	
		SR(%)	Time(s)	SR(%)	Time(s)	SR(%)	Time(s)	SR(%)	Time(s)
0	0.1	96.9	7.8	100.0	8.9	96.9	9.2	100.0	10.2
0	0.3	78.1	8.6	100.0	12.8	84.4	9.3	100.0	10.7
0	0.5	59.4	7.6	96.9	15.2	62.5	8.0	100.0	12.8
0	0.7	18.8	6.4	96.9	30.9	25.0	7.5	100.0	35.6
0	0.9	3.1	9.0	65.6	32.6	3.1	6.1	68.8	33.6
1e-2	0.1	100.0	6.8	100.0	7.6	100.0	7.0	100.0	7.4
1e-2	0.3	100.0	7.5	100.0	8.6	100.0	7.5	100.0	8.5
1e-2	0.5	90.6	7.6	100.0	9.0	96.9	7.7	100.0	8.7
1e-2	0.7	71.9	6.7	100.0	9.8	84.4	6.7	100.0	10.0
1e-2	0.9	37.5	6.8	100.0	18.7	43.8	8.6	100.0	10.1

Table 4.6: Performance of PD-PATH and PD-CC-PATH over economic dispatch example with *Type I* and *Type II* market constraints on scenario tree 2

Following the presentation of results for the five aforementioned approaches on all problem instances associated with scenario tree 2, we extend the analysis by including the results obtained from these methods on scenario tree 3, which features a larger number of scenarios. Tables 4.7 to 4.9 present a comparison of the performance of PATH, PATH-RN, PD, PD-PATH, and PD-CC-PATH on an economic dispatch example with three distinct types of market constraints, using scenario tree 3.

From the findings in Table 4.7, it is evident that the difficulty of the problem increases as the value of λ grows and the quadratic coefficient ϵ diminishes. This observation aligns with the numerical results obtained from scenario tree 2. Notably, due to the larger size of the tree, PATH and PATH-RN exhibit inferior performance compared to their performance on scenario tree 2. When the quadratic coefficient ϵ is less than or equal to 10^{-2}, PATH and PATH-RN can only solve a subset of problems with $\lambda = 0.1$ and fail to solve any cases with $\lambda > 0.1$. When ϵ exceeds 10^{-1}, PATH and PATH-RN cannot solve any cases.

Conversely, the performance of the PD approach remains relatively stable as the scenario tree expands, particularly when $\epsilon \geq 1$. This stability highlights the robustness of our algorithm when confronted with an increasing tree size. The PD-PATH approach consistently outperforms the previous three methods, indicating consistent results even with smaller scenario trees. However, the performance of the PD-PATH approach is influenced by the size of the scenario tree, with a lower success rate observed when ϵ is small. The novel heuristic approach, PD-CC-PATH, demonstrates superior performance compared to PD-PATH, providing evidence of the effectiveness of the proposed heuristics. We have tried many different implementation methods and found that the best option is choosing $v = 3$ and $w_1 = 0.1, w_2 = 0.3, w_3 = 0.6$.

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	0.0	37.5	0.0	59.4	100.0
0	0.3	0.0	0.0	0.0	12.5	96.9
0	0.5	0.0	0.0	0.0	9.4	71.9
0	0.7	0.0	0.0	0.0	3.1	18.8
0	0.9	0.0	0.0	0.0	0.0	9.4
1e-2	0.1	28.1	90.6	15.6	100.0	100.0
1e-2	0.3	0.0	0.0	0.0	90.6	100.0
1e-2	0.5	0.0	0.0	0.0	40.6	100.0
1e-2	0.7	0.0	0.0	0.0	21.9	84.4
1e-2	0.9	0.0	0.0	0.0	6.2	53.1
1e-1	0.1	0.0	100.0	59.4	100.0	100.0
1e-1	0.3	0.0	68.8	43.8	100.0	100.0
1e-1	0.5	0.0	3.1	18.8	96.9	100.0
1e-1	0.7	0.0	0.0	12.5	100.0	100.0
1e-1	0.9	0.0	0.0	15.6	93.8	100.0
1	0.1	71.9	100.0	100.0	100.0	100.0
1	0.3	12.5	100.0	100.0	100.0	100.0
1	0.5	3.1	15.6	93.8	100.0	100.0
1	0.7	0.0	0.0	93.8	100.0	100.0
1	0.9	0.0	0.0	96.9	100.0	100.0
10	0.1	100.0	100.0	100.0	100.0	100.0
10	0.3	96.9	96.9	93.8	100.0	100.0
10	0.5	65.6	56.2	100.0	100.0	100.0
10	0.7	0.0	3.1	93.8	100.0	100.0
10	0.9	0.0	0.0	100.0	100.0	100.0

Table 4.7: Performance of PATH, PATH-RN, PD, PD-PATH and PD-CC-PATH over economic dispatch example with *Type I* market constraint on scenario tree 3

Table 4.8 shows similar result as Table 4.7. The difference is that from the numerical results the *Type II* problem is easier than *Type I* problem.

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	62.5	71.9	0.0	96.9	100.0
0	0.3	0.0	0.0	0.0	43.8	100.0
0	0.5	0.0	0.0	0.0	9.4	71.9
0	0.7	0.0	0.0	0.0	0.0	31.2
0	0.9	0.0	0.0	0.0	0.0	9.4
1e-2	0.1	96.9	100.0	15.6	100.0	100.0
1e-2	0.3	9.4	31.2	0.0	96.9	100.0
1e-2	0.5	0.0	0.0	0.0	71.9	100.0
1e-2	0.7	0.0	0.0	0.0	40.6	96.9
1e-2	0.9	0.0	0.0	0.0	9.4	65.6
1e-1	0.1	96.9	100.0	53.1	100.0	100.0
1e-1	0.3	40.6	100.0	46.9	100.0	100.0
1e-1	0.5	3.1	46.9	21.9	100.0	100.0
1e-1	0.7	0.0	0.0	18.8	100.0	100.0
1e-1	0.9	0.0	0.0	15.6	93.8	100.0
1	0.1	100.0	100.0	100.0	100.0	100.0
1	0.3	90.6	100.0	96.9	100.0	100.0
1	0.5	71.9	96.9	90.6	100.0	100.0
1	0.7	18.8	56.2	90.6	100.0	100.0
1	0.9	0.0	15.6	93.8	100.0	100.0
10	0.1	100.0	100.0	100.0	100.0	100.0
10	0.3	100.0	96.9	96.9	100.0	100.0
10	0.5	90.6	59.4	96.9	100.0	100.0
10	0.7	62.5	12.5	100.0	100.0	100.0
10	0.9	21.9	6.2	96.9	100.0	100.0

Table 4.8: Performance of PATH, PATH-RN, PD, PD-PATH and PD-CC-PATH over economic dispatch example with *Type II* market constraint on scenario tree 3

Numerical results for the economic dispatch problem with *Type III* market constraints on scenario tree 3 are presented in Table 4.9. The findings indicate that problems with *Type III* market constraints are considerably easier compared to those with *Type I* and *Type II* constraints. However, PATH and PATH-RN still exhibit a very low success rate when ϵ is small and λ is large. In contrast, the PD-PATH approach demonstrates a near-complete success rate in solving instances of problems with *Type III* market constraints. Furthermore, the PD-CC-PATH approach successfully solves every single instance of this problem type.

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	100.0	100.0	81.2	100.0	100.0
0	0.3	100.0	100.0	50.0	100.0	100.0
0	0.5	50.0	93.8	15.6	100.0	100.0
0	0.7	3.1	15.6	21.9	100.0	100.0
0	0.9	0.0	0.0	18.8	90.6	100.0
1e-2	0.1	100.0	100.0	65.6	100.0	100.0
1e-2	0.3	100.0	100.0	46.9	100.0	100.0
1e-2	0.5	59.4	93.8	18.8	100.0	100.0
1e-2	0.7	9.4	28.1	9.4	96.9	100.0
1e-2	0.9	0.0	0.0	6.2	96.9	100.0
1e-1	0.1	100.0	100.0	81.2	100.0	100.0
1e-1	0.3	93.8	100.0	53.1	100.0	100.0
1e-1	0.5	62.5	100.0	46.9	100.0	100.0
1e-1	0.7	9.4	90.6	40.6	100.0	100.0
1e-1	0.9	12.5	37.5	25.0	100.0	100.0
1	0.1	100.0	100.0	96.9	100.0	100.0
1	0.3	100.0	100.0	100.0	100.0	100.0
1	0.5	100.0	100.0	96.9	100.0	100.0
1	0.7	87.5	100.0	90.6	100.0	100.0
1	0.9	46.9	90.6	93.8	100.0	100.0
10	0.1	100.0	100.0	96.9	100.0	100.0
10	0.3	100.0	100.0	100.0	100.0	100.0
10	0.5	100.0	100.0	100.0	100.0	100.0
10	0.7	100.0	100.0	100.0	100.0	100.0
10	0.9	71.9	96.9	93.8	100.0	100.0

Table 4.9: Performance of PATH, PATH-RN, PD, PD-PATH and PD-CC-PATH over economic dispatch example with *Type III* market constraint on scenario tree 3

4.4.4 Homotopy approach

Building upon the introduced convex-combination approach, we can enhance its effectiveness by incorporating a homotopy approach. In Algorithm 7, which employs primal-MOPEC-dual-risk decomposition, we always initiate the process with $\boldsymbol{\mu}^1 = \mathbb{P}$, generating $(\boldsymbol{x}^1, \boldsymbol{p}^1)$ based on the primal-MOPEC $\mathcal{F}(\boldsymbol{\mu}^1)$ even for risk measures such as $\overline{CVaR}(\lambda, \varphi)$ with significantly large values of λ. Notably, the risk-neutral case can be considered as a problem with the risk measure $\overline{CVaR}(0, \varphi)$. Therefore, the original Algorithm 7 can be treated as a homotopy method with $\Delta\lambda = \lambda - 0 = \lambda$. Given the continuity of the risk measure operator $\overline{CVaR}(\lambda, \varphi)$ with respect to λ, our objective is to reduce the magnitude of $\Delta\lambda$.

The underlying concept of the devised homotopy approach is as follows: when addressing a problem with the risk measure $\overline{CVaR}(\lambda, \varphi)$, rather than commencing from a solution in the risk-neutral case, we assume that an equilibrium point $(\boldsymbol{x}^*(\lambda - \Delta\lambda), \boldsymbol{p}^*(\lambda - \Delta\lambda), \boldsymbol{\mu}^*(\lambda - \Delta\lambda))$ already exists, representing the solution to the problem with the risk measure $\overline{CVaR}(\lambda - \Delta\lambda, \varphi)$. Since the risk set undergoes changes as λ increases, while considering the solution $(\boldsymbol{x}^*(\lambda), \boldsymbol{p}^*(\lambda))$ as a continuous solution mapping with respect to λ, we initiate the process with the solution $(\boldsymbol{x}^*(\lambda - \Delta\lambda), \boldsymbol{p}^*(\lambda - \Delta\lambda))$ and obtain an updated risk probability vector $\boldsymbol{p}^*$ that satisfies $\mathcal{B}(\boldsymbol{x}^*(\lambda - \Delta\lambda), \boldsymbol{p}^*(\lambda - \Delta\lambda))$. Subsequently, we apply the primal-MOPEC-dual-risk decomposition algorithm, incorporating a convex combination on the risk probability vector until we attain a satisfactory solution or reach the initial point. The complete algorithm is outlined as follows:

Algorithm 8 Homotopy primal-MOPEC-dual-risk decomposition algorithm with convex combination on risk probability vector to solve risk averse problem with risk measure $\overline{CVaR}(\lambda, \varphi)$

Input and Initialization Assume at the beginning we have the equilibrium point $(\boldsymbol{x}^{**}, \boldsymbol{p}^{**}, \boldsymbol{\mu}^{**})$, which is the solution of risk-averse problem with risk measure $\overline{CVaR}(\lambda - \Delta\lambda, \varphi)$. Choose positive integer v, positive integer $J \geq v$ and positive integer $M \geq J$.. Set $r = 1$.
Initial Dual-step.
for $t = |\mathcal{T}| - 1, \ldots, 1$ **do**
  for $a \in \mathcal{A}$ **do**
    for $n \in \mathcal{N}(t)$ **do**
      Compute $\mu^1_{an_+}$ that is the solution of risk-averse problem $R_{an}(\boldsymbol{x}^{**}_{-a}, \boldsymbol{p}^*, \mu^1_{a,\bar{\mathcal{S}}(n)})$.
    end for
  end for
end for
while $r \leq J$ **do**
  Step 1: Primal-step Solve the primal-MOPEC $\mathcal{F}(\boldsymbol{\mu}^r)$, get $(\boldsymbol{x}^r, \boldsymbol{p}^r)$ such that

$$\begin{array}{lll} \boldsymbol{x}^r_a & \text{is an optimal solution of} \quad P^{\mathcal{F}}_a(\boldsymbol{x}^r_{-a}, \boldsymbol{p}^r, \boldsymbol{\mu}^r) & \text{for all } a \in \mathcal{A} \\ \boldsymbol{p}^r & \text{belongs to set} \quad Q^{\mathcal{M}}(\boldsymbol{x}^r) & \end{array} \tag{4.17}$$

  Step 2: Dual-step.
  for $t = |\mathcal{T}| - 1, \ldots, 1$ **do**
    for $a \in \mathcal{A}$ **do**
      for $n \in \mathcal{N}(t)$ **do**
        Compute $\mu^{r+1}_{an_+}$ that is the solution of risk-averse problem $R_{an}(\boldsymbol{x}^r, \boldsymbol{p}^r, \mu^{r+1}_{a,\bar{\mathcal{S}}(n)})$.
      end for
    end for
  end for
  Step 3: Stopping Test. If stopping criteria is met, break and go to **Step 5**.
  Step 4: Loop. Set $r := r + 1$.
end while
Step 5: PATH solve. Use the PATH solver to solve the whole problem with initial point $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$. Stop the algorithm and get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*)$ if the solution is found, else continue.
while $r \leq M$ **do**
  Set $\hat{\boldsymbol{\mu}}^r = \sum_{q=0}^{v-1} w_q \cdot \boldsymbol{\mu}^{r+1-v+q}$.
  Step 6: Primal-step Solve the primal-MOPEC $\mathcal{F}(\hat{\boldsymbol{\mu}}^r)$, get $(\boldsymbol{x}^r, \boldsymbol{p}^r)$ such that

$$\begin{array}{lll} \boldsymbol{x}^r_a & \text{is an optimal solution of} \quad P^{\mathcal{F}}_a(\boldsymbol{x}^r_{-a}, \boldsymbol{p}^r, \hat{\boldsymbol{\mu}}^r) & \text{for all } a \in \mathcal{A} \\ \boldsymbol{p}^r & \text{belongs to set} \quad Q^{\mathcal{M}}(\boldsymbol{x}^r) & \end{array} \tag{4.18}$$

  Step 7: Dual-step.
  for $t = |\mathcal{T}| - 1, \ldots, 1$ **do**
    for $a \in \mathcal{A}$ **do**
      for $n \in \mathcal{N}(t)$ **do**
        Compute $\mu^{r+1}_{an_+}$ that is the solution of risk-averse problem $R_{an}(\boldsymbol{x}^r, \boldsymbol{p}^r, \mu^{r+1}_{a,\bar{\mathcal{S}}(n)})$.
      end for
    end for
  end for
  Step 8: Stopping Test. If stopping criteria is met, break and go to **Step 11**.
  Step 9: Loop. Set $r := r + 1$.
end while
Step 10: PATH solve. Use the PATH solver to solve the whole problem with initial point $(\boldsymbol{x}^r, \boldsymbol{p}^r, \boldsymbol{\mu}^{r+1})$. Stop the algorithm and get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*, \boldsymbol{\mu}^*)$ if the solution is found.

4.4.5 Numerical experiments

We are conducting a series of computational experiments in order to evaluate the effectiveness of the variants of primal-MOPEC-dual-risk decomposition approach. The experiments are implemented in the same computational setting as section 4.3.2.

4.4.5.1 Problem parameter setting

In the following experiments, we are using the same parameters setting as in subsection 4.3.2.1. The only difference is that we will also testing our algorithms on these type of problems on scenario tree 3 defined in chapter 2.

4.4.5.2 Numerical results

In this subsection, we are testing the performance of following method:

- Homot: Solving the problem using homotopy primal-MOPEC-dual-risk decomposition algorithm with convex combination on risk probability vector to solve risk averse problem with risk measure $\overline{CVaR}(\lambda, \varphi)$

It is observed that in table 4.10 the Homotopy approach performs much better than all the previous results.

ϵ	λ	*Type I* market		*Type II* market	
		successful rate (%)			
		PD-CC-PATH	Homot	PD-CC-PATH	Homot
0	0.1	100.0	100.0	100.0	100.0
0	0.3	96.9	100.0	100.0	100.0
0	0.5	71.9	90.6	71.9	87.5
0	0.7	18.8	53.1	31.2	50.0
0	0.9	9.4	21.9	9.4	12.5
1e-2	0.1	100.0	100.0	100.0	100.0
1e-2	0.3	100.0	100.0	100.0	100.0
1e-2	0.5	100.0	100.0	100.0	100.0
1e-2	0.7	84.4	93.8	96.9	100.0
1e-2	0.9	53.1	68.8	65.6	81.2

Table 4.10: Performance of PD-CC-PATH and Homot on economic dispatch example with *Type I* and *Type II* market constraints on scenario tree 3

4.5 Conclusion

This chapter primarily focused on addressing the challenge of solving the general stochastic MOPEC with risk-averse players by employing an "equilibrium reformulation" approach. Our main contribution involves the introduction of a novel primal-MOPEC-dual-risk decomposition algorithm, which leverages the inherent structure of the problem. Additionally, we propose several variants of computational enhancement techniques, including those based on proximal term manipulation, convex combination strategies, and homotopy methods. To assess the efficacy of the proposed algorithms, we conducted performance evaluations on all three problem instances described in chapter 2. These instances encompass various types of market constraints, with the classical $\overline{CVaR}(\lambda, \varphi)(\cdot)$ risk measure. Similar results are obtained also for the K-Deviance coherent risk measure defined in chapter 1.

Through the computational experiments, we observe that the difficulty of solving the stochastic MOPEC with risk-averse players escalates as the parameter λ increases and the quadratic coefficient ϵ diminishes. We note that the primal-MOPEC-dual-risk algorithm exhibits a high likelihood of obtaining a solution for the problem when the solution is a non-mixed one, as per our defined criteria. Moreover, we find that the primal-MOPEC-dual-risk algorithm, when combined with the PATH solver, demonstrates notably improved performance in tackling such problem instances compared to the original PATH solver. Furthermore, we continually enhance the algorithms by incorporating proximal terms, convex combination approaches, and homotopy techniques. These enhancements contribute to effectively addressing a broader range of problem instances that were previously unsolvable.

5 TIME STAGE DECOMPOSITION METHOD FOR MULTISTAGE STOCHASTIC EQUILIBRIUM WITH FIXED RISK PROBABILITY

This chapter is dedicated to the study of the stage decomposition method that aims to solve the stochastic MOPEC with risk-averse players using an *equilibrium reformulation* but with a fixed risk probability vector $\boldsymbol{\mu}$. In the previous chapter 4, we introduced the primal-MOPEC-dual-risk algorithm that decomposes the entire system into two subproblems and solves the stochastic MOPEC with risk-averse players using the *equilibrium reformulation* iteratively. However, it has been observed that the primal-MOPEC subproblem becomes large-scale and computationally intractable when the number of time stages is large. This is due to the exponential growth of the scenario tree size with the number of stages. The motivation behind this chapter is to present a new stage-based decomposition algorithm that further decomposes the primal-MOPEC subproblem into a collection of smaller subproblems indexed by stages, which helps to solve this type of problem more effectively when the original subproblem size becomes too large to handle. The smaller subproblems can then be solved iteratively, contributing to the solution of the primal-MOPEC subproblem even in cases where the computational challenges arise from the large size of the scenario tree resulting from multiple stages.

The chapter is organized as follows: Section 5.1 briefly recalls the problem introduced in chapter 1. Furthermore, this section establishes the satisfaction of the time consistency property by the problem, which serves as the foundation for the stage decomposition algorithm proposed in this chapter. In section 5.2, a new stage-based decomposition algorithm is proposed, which is based on the time consistency property in section 5.1 and is a Gauss-Seidel iterative method on the equivalent KKT system of the original problem. Building upon the main framework of the stage-based algorithm, the section 5.3 aims to demonstrate the practical effectiveness of the stage decomposition algorithm and effect of algorithimic enhancements based on examples from chapter 2. The final section 5.4 will present the main findings and conclusions drawn from the research conducted in this chapter.

5.1 Model and structure

This chapter presents a stage-based algorithm designed to address the multistage stochastic MOPEC, which has been reformulated as an equilibrium problem and characterized by fixed risk probabilities denoted as $\boldsymbol{\mu} = (\mu_{an})_{a\in\mathcal{A},n\in\mathcal{N}}$. In this context, the set $\mathcal{A}$ represents

the collection of all players involved, while $\mathcal{N}$ denotes the set of scenario nodes. For each player $a \in \mathcal{A}$ and scenario tree node $n \in \mathcal{N}$, the strategy employed by player a is denoted by the vector $x_{an} \in \mathbb{R}^{d_{an}}$, where d_{an} signifies the dimensionality of the strategy and is considered a positive integer. The complete set of strategies for player a is represented by $\boldsymbol{x}_a = (x_{an})_{n\in\mathcal{N}}$. Moreover, the vector $x_{-an} \in \mathbb{R}^{d_n - d_{an}}$ captures the strategies of all other players, excluding player a, at node n, where d_n corresponds to the sum of d_{an} over all players in $\mathcal{A}$. Collectively, these vectors are represented as $\boldsymbol{x}_a = (x_{an})_{n\in\mathcal{N}}$, $\boldsymbol{x}_{-a} = (x_{-an})_{n\in\mathcal{N}}$, and $\boldsymbol{p} = (p_n)_{n\in\mathcal{N}}$. The market price at node n is denoted by the vector $p_n \in \mathbb{R}^{\alpha_n}$, where α_n represents a positive integer. Given the fixed risk probability $\boldsymbol{\mu}$, each player $a \in \mathcal{A}$ solves an optimization problem using the variables $(\boldsymbol{x}_{-a}, \boldsymbol{p}, \boldsymbol{\mu})$.

$$
\begin{aligned}
P_a^{\mathcal{F}}(\boldsymbol{x}_{-a}, \boldsymbol{p}, \boldsymbol{\mu}): \quad \min_{(x_{am})_{m\in\mathcal{N}}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \sum_{m\in 1_+} \mu_{am} \cdot \Big[f_{am}(x_{am}; x_{-am}, p_m) \\
& \qquad + \sum_{l\in m_+} \mu_{al} \cdot \big[f_{al}(x_{al}; x_{-al}, p_l) + \cdots \big] \Big] \\
\text{s.t.} \quad & \mathcal{G}_{an}(x_{a,n_-}, x_{an}; x_{-an}, p_n) \in \mathcal{K}_{an}, \qquad \forall n \in \mathcal{N} \\
& x_{an} \in \mathcal{X}_{an}, \quad \forall n \in \mathcal{N}
\end{aligned}
\tag{5.1}
$$

where $\mathcal{X}_{an} \subseteq \mathbb{R}^{d_{an}}$ is a closed, convex set and $\mathcal{K}_{an} \subseteq \mathbb{R}^{m_{an}}$ is a closed, convex cone for each $n \in \mathcal{N}$, $f_{an} : \mathbb{R}^{d_n+\alpha_n+\beta_n} \to \mathbb{R}^{m_{an}}, \mathcal{G}_{an} : \mathbb{R}^{d_{an_-}+d_n+\alpha_n} \to \mathbb{R}^{m_{an}}$ are continuously differentiable functions and $f_{an}(\cdot; x_{-an}, p_n), \mathcal{G}_{an}(\cdot; x_{-an}, p_n)$ are convex functions with fixed (x_{-an}, p_n) for each player $a \in \mathcal{A}$ and $n \in \mathcal{N}$.

For each scenario node $n \in \mathcal{N}$, there is a market constraint:

$$
0 \in F_n(p_n; x_n) + \mathcal{N}_{K_n}(p_n), \tag{5.2}
$$

where $K_n \subseteq \mathbb{R}^{\alpha_n}$ is a closed, convex set, $\mathcal{N}_{K_n} : \mathbb{R}^{\alpha_n} \to \mathbb{R}^{\alpha_n}$ is a normal cone mapping of the set K_n and $F_n : \mathbb{R}^{\alpha_n+d_n} \to \mathbb{R}^{\alpha_n}$ is a smooth function for each $n \in \mathcal{N}$.

Let us define the set $Q_n^{VI}(\boldsymbol{x}) \coloneqq \{p_n | p_n \in K_n, p_n \text{ satisfies } (5.2)\}$. With the aforementioned formulations as a foundation, we can proceed to define the multistage stochastic mathematical program with equilibrium constraints (SMOPEC) under the assumption of a fixed risk probability vector $\boldsymbol{\mu}$.

Definition 5.1. $(\mathcal{F}(\boldsymbol{\mu}))$ $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \Big((x_{an}^*)_{a\in\mathcal{A}, n\in\mathcal{N}}, (p_n^*)_{n\in\mathcal{N}} \Big)$ is an equilibrium point of the

multistage stochastic MOPEC with fixed $\boldsymbol{\mu}$ if and only if

$$\begin{aligned} &\boldsymbol{x}_a^* \quad \text{is an optimal solution of} \quad P_a^{\mathcal{F}}(\boldsymbol{x}_{-a}^*, \boldsymbol{p}^*, \boldsymbol{\mu}) \qquad \text{for all } a \in \mathcal{A}, \\ &p_n^* \quad \text{belongs to set} \quad Q_n^{VI}(\boldsymbol{x}^*) \end{aligned} \tag{5.3}$$

5.1.1 Time consistency property

The stagewise decomposition approach employed for the solution of the stochastic MOPEC is founded upon the principle of *time consistency*. This inherent property stipulates that in the case of a truncated stochastic process, the solution to the corresponding truncated SMOPEC must constitute an equilibrium point. Specifically, when the stochastic variables assume discrete values from a countable set, this implies that at each node within the scenario tree, the solution to the stochastic MOPEC pertaining to the subtree rooted at that node, with the initial value determined by the decision variable from the preceding stage, must be an equilibrium.

To elaborate further, consider an optimization problem commencing at node n, where the variables are provided as $(x_{a,n_-}, (x_{-am})_{m\in\mathcal{S}(n)}, (p_m)_{m\in\mathcal{S}(n)}, (\mu_{am})_{a\in\mathcal{A},m\in\bar{\mathcal{S}}(n)})$, where the successor trees $\mathcal{S}(n)$ and $\bar{\mathcal{S}}(n)$ are defined in chapter 1. With this premise, we define the subsequent problem that each player $a \in \mathcal{A}$ must address when initiating from scenario node n:

$$\begin{aligned} P_a^{\mathcal{SF}}(n, x_{a,n_-}, \mu_{a,\mathcal{S}(n)}): \quad \min_{(x_{am})_{m\in\mathcal{S}(n)}} \quad & f_{an}(x_{an}; x_{-an}, p_n) + \sum_{m\in n_+} \mu_{am} \cdot \Big[f_{am}(x_{am}; x_{-am}, p_m) \\ & \qquad + \sum_{l\in m_+} \mu_{al} \cdot \big[f_{al}(x_{al}; x_{-al}, p_l) + \cdots \big] \Big] \\ \text{s.t.} \quad & \mathcal{G}_{am}(x_{a,m_-}, x_{am}; x_{-am}, p_m) \in \mathcal{K}_{am}, \qquad \forall m \in \mathcal{S}(n) \\ & x_{am} \in \mathcal{X}_{am}, \quad \forall m \in \mathcal{S}(n) \end{aligned} \tag{5.4}$$

Here $\mu_{a,\mathcal{S}(n)}$ represents the vector $(\mu_{am})_{m\in\mathcal{S}(n)}$. Combining the subsequent optimization problem of each player $a \in \mathcal{A}$ and all equilibrium constraints starting at scenario node n, we could have the definition of the conditional multistage stochastic MOPEC given $\boldsymbol{\mu}_{\bar{\mathcal{S}}(n)} = (\mu_{am})_{a\in\mathcal{A},m\in\bar{\mathcal{S}}(n)}$ and initial state $\boldsymbol{x}_{n_-} = (x_{a,n_-})_{a\in\mathcal{A}}$:

Definition 5.2. ($\mathcal{SF}(n, \boldsymbol{x}_{n_-}, \boldsymbol{\mu}_{\bar{\mathcal{S}}(n)})$ $\Big((x_{am}^*)_{a\in\mathcal{A},m\in\mathcal{S}(n)}, (p_m^*)_{m\in\mathcal{S}(n)} \Big)$ is an equilibrium point of

the conditional multistage SMOPEC given $\boldsymbol{\mu}_{\bar{S}(n)}$ and initial state $\boldsymbol{x}_{n_-}$ if and only if

$$\begin{aligned} (x^*_{am})_{m\in\mathcal{S}(n)} \quad &\text{is an optimal solution of} \quad P_a^{\mathcal{SF}}(n, x_{a,n_-}, \mu_{a,\mathcal{S}(n)}) \\ &\text{for all } a \in \mathcal{A} \\ p^*_m \quad &\text{belongs to set} \quad Q_m^{VI}(\boldsymbol{x}^*) \ \text{ for any } \ m \in S(n) \end{aligned} \tag{5.5}$$

Definition 5.3. The SMOPEC $\mathcal{F}(\boldsymbol{\mu})$ with given $\boldsymbol{\mu}$ based on a scenario tree is called time consistent, if for each its equilibrium point $(\boldsymbol{x}^*, \boldsymbol{p}^*)$, the subset vector $\Big((x^*_{am})_{a\in\mathcal{A},m\in\mathcal{S}(n)}, (p^*_m)_{m\in\mathcal{S}(n)}\Big)$ is an equilibrium point of the conditional problem $\mathcal{SF}(n, \boldsymbol{x}_{n_-}, \mu_{a,\mathcal{S}(n)})$ for any $n \in \mathcal{N}$.

Theorem 5.4. *Assume $\mathcal{X}_{an} \subseteq \mathbb{R}^{d_{an}}$ is a closed, convex set and $\mathcal{K}_{an} \subseteq \mathbb{R}^{m_{an}}$ is a closed, convex cone for each $n \in \mathcal{N}$. $f_{an} : \mathbb{R}^{d_n+\alpha_n+\beta_n} \to \mathbb{R}^{m_{an}}, \mathcal{G}_{an} : \mathbb{R}^{d_{an_-}+d_n+\alpha_n} \to \mathbb{R}^{m_{an}}$ are continuously differentiable functions and $f_{an}(\cdot; x_{-an}, p_n), \mathcal{G}_{an}(\cdot; x_{-an}, p_n)$ are convex function with fixed (x_{-an}, p_n) for each player $a \in \mathcal{A}$ and $n \in \mathcal{N}$. $K_n \subseteq \mathbb{R}^{\alpha_n}$ is a closed, convex set, $\mathcal{N}_{K_n} : \mathbb{R}^{\alpha_n} \to \mathbb{R}^{\alpha_n}$ is a normal cone mapping of set K_n and $F_n : \mathbb{R}^{\alpha_n+d_n} \to \mathbb{R}^{\alpha_n}$ is a smooth function for each $n \in \mathcal{N}$, then $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ is the equilibrium point of the SMOPEC (5.1)-(5.2) with fixed $\boldsymbol{\mu}$ $(\mathcal{F}(\boldsymbol{\mu}))$ if and only if it satisfies the KKT condition:*

$$\begin{aligned} 0 \in \quad & (\prod_{m\in\mathcal{P}(n)} \mu_{am})\nabla_{x_{an}} f_{an}(x^*_{an}; x^*_{-an}, p^*_n) + \nabla_{x_{an}} \mathcal{G}_{an}(x^*_{a,n_-}, x^*_{an}; x^*_{-an}, p^*_n)\lambda^*_{an} \\ & + \sum_{m\in n_+} \nabla_{x_{an}} \mathcal{G}_{am}(x^*_{an}, x^*_{am}; x^*_{-am}, p^*_m)\lambda^*_{am} + \mathcal{N}_{\mathcal{X}_{an}}(x^*_{an}), && \forall n \in \mathcal{N} \\ 0 \in \quad & F_n(p^*_n; \boldsymbol{x}^*_n) + \mathcal{N}_{K_n}(p^*_n), && \forall n \in \mathcal{N} \\ \mathcal{K}_{an} \ni \quad & \mathcal{G}_{an}(x^*_{a,n_-}, x^*_{an}; x^*_{-an}, p^*_n) \perp \lambda^*_{an} \in \mathcal{K}^*_{an}, && \forall n \in \mathcal{N} \end{aligned} \tag{5.6}$$

Here λ^*_{an} is the dual multipliers of the constraint $\mathcal{G}_{an}(x^*_{a,n_-}, x^*_{an}; x^*_{-an}, p^*_n) \in \mathcal{K}_{an}$ for any $n \in \mathcal{N}$.

Proof. This can be proved using the first-order conditions that are necessary and sufficient in this case. □

Theorem 5.5. *Assume $\mathcal{X}_{an} \subseteq \mathbb{R}^{d_{an}}$ is a closed, convex set and $\mathcal{K}_{an} \subseteq \mathbb{R}^{m_{an}}$ is a closed, convex cone for each $n \in \mathcal{N}$. $f_{an} : \mathbb{R}^{d_n+\alpha_n+\beta_n} \to \mathbb{R}^{m_{an}}, \mathcal{G}_{an} : \mathbb{R}^{d_{an_-}+d_n+\alpha_n} \to \mathbb{R}^{m_{an}}$ are continuously differentiable functions and $f_{an}(\cdot; x_{-an}, p_n), \mathcal{G}_{an}(\cdot; x_{-an}, p_n)$ are convex function with fixed (x_{-an}, p_n) for each player $a \in \mathcal{A}$ and $n \in \mathcal{N}$. $K_n \subseteq \mathbb{R}^{\alpha_n}$ is a closed, convex set, $\mathcal{N}_{K_n} : \mathbb{R}^{\alpha_n} \to \mathbb{R}^{\alpha_n}$ is a normal cone mapping of set K_n and $F_n : \mathbb{R}^{\alpha_n+d_n} \to \mathbb{R}^{\alpha_n}$ is a smooth function for each $n \in \mathcal{N}$, then the SMOPEC (5.1)-(5.2) with fixed $\boldsymbol{\mu}$ $(\mathcal{F}(\boldsymbol{\mu}))$ satisfies the time consistent property.*

Proof. If $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ is an equilibrium point of the forward SMOPEC $\mathcal{F}(\boldsymbol{\mu})$, then it satisfies the

KKT condition:

$$\begin{aligned} 0 \in \quad & (\prod_{m\in\mathcal{P}(n)} \mu_{am})\nabla_{x_{an}} f_{an}(x^*_{an}; x^*_{-an}, p^*_n) + \nabla_{x_{an}} \mathcal{G}_{an}(x^*_{a,n_-}, x^*_{an}; x^*_{-an}, p^*_n)\lambda^*_{an} \\ & + \sum_{m\in n_+} \nabla_{x_{an}} \mathcal{G}_{am}(x^*_{an}, x^*_{am}; x^*_{-am}, p^*_m)\lambda^*_{am} + \mathcal{N}_{\mathcal{X}_{an}}(x^*_{an}), \qquad \forall n \in \mathcal{N} \end{aligned} \tag{5.7}$$

$$0 \in \quad F_n(p^*_n; \boldsymbol{x}^*_n) + \mathcal{N}_{K_n}(p^*_n), \qquad \forall n \in \mathcal{N} \tag{5.8}$$

$$\mathcal{K}_{an} \ni \quad \mathcal{G}_{an}(x^*_{a,n_-}, x^*_{an}; x^*_{-an}, p^*_n) \perp \lambda^*_{an} \in \mathcal{K}^*_{an}, \qquad \forall n \in \mathcal{N} \tag{5.9}$$

Here λ^*_{an} is the dual multipliers of the constraint $\mathcal{G}_{an}(x^*_{a,n_-}, x^*_{an}; x^*_{-an}, p^*_n) \in \mathcal{K}_{an}$ for any $n \in \mathcal{N}$.

Since the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*, \lambda^*)$ satisfies the KKT condition (5.6), with any fixed n and any $m \in \mathcal{S}(n)$, we would have

$$\begin{aligned} 0 \in \quad & (\prod_{l\in\mathcal{P}(m)} \mu_{al})\nabla_{x_{am}} f_{am}(x^*_{am}; x^*_{-am}, p^*_m) + \nabla_{x_{am}} \mathcal{G}_{am}(x^*_{a,m_-}, x^*_{am}; x^*_{-am}, p^*_m)\lambda^*_{am} \\ & + \sum_{l\in m_+} \nabla_{x_{am}} \mathcal{G}_{al}(x^*_{am}, x^*_{al}; x^*_{-al}, p^*_l)\lambda^*_{al} + \mathcal{N}_{\mathcal{X}_{am}}(x^*_{am}) \end{aligned} \tag{5.10}$$

$$\begin{aligned} (\Rightarrow) \quad 0 \in \quad & (\prod_{j\in\mathcal{P}(n)} \mu_{aj})(\prod_{l\in\mathcal{P}(n,m)} \mu_{al})\nabla_{x_{am}} f_{am}(x^*_{am}; x^*_{-am}, p^*_m) \end{aligned} \tag{5.11}$$

$$\begin{aligned} & + \nabla_{x_{am}} \mathcal{G}_{am}(x^*_{a,m_-}, x^*_{am}; x^*_{-am}, p^*_m)\lambda^*_{am} \\ & + \sum_{l\in m_+} \nabla_{x_{am}} \mathcal{G}_{al}(x^*_{am}, x^*_{al}; x^*_{-al}, p^*_l)\lambda^*_{al} + \mathcal{N}_{\mathcal{X}_{am}}(x^*_{am}) \end{aligned} \tag{5.12}$$

$$\begin{aligned} (\Rightarrow) \quad 0 \in \quad & (\prod_{l\in\mathcal{P}(n,m)} \mu_{al})\nabla_{x_{am}} f_{am}(x^*_{am}; x^*_{-am}, p^*_m) \end{aligned} \tag{5.13}$$

$$\begin{aligned} & + \nabla_{x_{am}} \mathcal{G}_{am}(x^*_{a,m_-}, x^*_{am}; x^*_{-am}, p^*_m)\Big(\lambda^*_{am}/(\prod_{j\in\mathcal{P}(n)} \mu_{aj})\Big) \\ & + \sum_{l\in m_+} \nabla_{x_{am}} \mathcal{G}_{al}(x^*_{am}, x^*_{al}; x^*_{-al}, p^*_l)\Big(\lambda^*_{al}/(\prod_{j\in\mathcal{P}(n)} \mu_{aj})\Big) + \mathcal{N}_{\mathcal{X}_{am}}(x^*_{am}) \end{aligned} \tag{5.14}$$

Denote $\bar{\lambda}_{am} = \lambda^*_{am}/(\prod_{j\in\mathcal{P}(n)} \mu_{aj})$ for any $m \in \mathcal{S}(n)$. Then we can establish the equivalence of (5.14) with the following generalized equation. This equivalence is achieved by dividing

both sides of (5.14) by $\prod_{j\in\mathcal{P}(n)}\mu_{aj}$:

$$\begin{aligned} 0 \in \quad & (\prod_{l\in\mathcal{P}(n,m)}\mu_{al})\nabla_{x_{am}} f_{am}(x^*_{am}; x^*_{-am}, p^*_m) + \nabla_{x_{am}}\mathcal{G}_{am}(x^*_{a,m_-}, x^*_{am}; x^*_{-am}, p^*_m)\bar{\lambda}_{am} \\ & + \sum_{l\in m_+}\nabla_{x_{am}}\mathcal{G}_{al}(x^*_{am}, x^*_{al}; x^*_{-al}, p^*_l)\bar{\lambda}_{al} + \mathcal{N}_{\mathcal{X}_{am}}(x^*_{am}) \end{aligned} \tag{5.15}$$

We can also divide term $\prod_{l\in S(n)}\mu^*_{al}$ on both sides of equations (5.9) and get

$$\mathcal{K}_{an} \ni \quad \mathcal{G}_{an}(x^*_{a,n_-}, x^*_{an}; x^*_{-an}, p^*_n) \perp \bar{\lambda}_{an} \in \mathcal{K}^*_{an}, \qquad \forall n \in \mathcal{N} \tag{5.16}$$

The new generalized equations (5.8), (5.15) - (5.2) shows that the point

$$\Big((x^*_{am})_{a\in\mathcal{A},m\in\mathcal{S}(n)}, (p^*_m)_{m\in\mathcal{S}(n)}, (\bar{\lambda}_{am})_{a\in\mathcal{A},m\in\mathcal{S}(n)}\Big)$$

is satisfying the KKT condition of conditional forward problem $\mathcal{SF}(n, \boldsymbol{x}_{n_-}, (\mu_{am})_{a\in\mathcal{A},m\in\bar{\mathcal{S}}(n)})$.

The above analysis demonstrates that the SMOPEC (5.1)-(5.2), with a fixed risk probability vector denoted as $\boldsymbol{\mu}$ or $\mathcal{F}(\boldsymbol{\mu})$, satisfies the property of time consistency. □

5.2 Stagewise decomposition algorithm

In Chapter 1, we established the existence of an equivalent variational formulation for any equilibrium problem. Building upon this foundation, the current chapter aims to derive the corresponding variational format for the problem $\mathcal{F}(\mu^*)$ and develop a stagewise decomposition algorithm based on it.

In Theorem 5.4, we have already demonstrated that if $(\boldsymbol{x}^*, \boldsymbol{p}^*)$ represents an equilibrium point of the problem $\mathcal{F}(\mu^*)$, then there exists a set of variables $\{\lambda^*_{an} | a \in \mathcal{A}, n \in \mathcal{N}\}$ such that the vector $(\boldsymbol{x}^*, \boldsymbol{p}^*, \lambda^*)$ serves as a solution to the subsequent variational inequality problem.

$$\begin{aligned} 0 \in \quad & (\prod_{m\in\mathcal{P}(n)}\mu_{am})\nabla_{x_{an}} f_{an}(x_{an}; x_{-an}, p_n) + \nabla_{x_{an}}\mathcal{G}_{an}(x_{a,n_-}, x_{an}; x_{-an}, p_n)\lambda_{an} \\ & + \sum_{m\in n_+}\nabla_{x_{an}}\mathcal{G}_{am}(x_{an}, x_{am}; x_{-am}, p_m)\lambda_{am} + \mathcal{N}_{\mathcal{X}_{an}}(x_{an}), && \forall n \in \mathcal{N} \\ 0 \in \quad & F_n(p_n; \boldsymbol{x}_n) + \mathcal{N}_{K_n}(p_n), && \forall n \in \mathcal{N} \\ \mathcal{K}_{an} \ni \quad & \mathcal{G}_{an}(x_{a,n_-}, x_{an}; x_{-an}, p_n) \perp \lambda_{an} \in \mathcal{K}^*_{an}, && \forall n \in \mathcal{N} \end{aligned} \tag{5.17}$$

Here λ_{an} are the dual multipliers of the constraint $\mathcal{G}_{an}$ for any $n \in \mathcal{N}$.

The key to our approach is to solve these conditions iteratively via decomposition.

Two-stage decomposition

To begin, we shall focus on the two-stage case. In this particular scenario, the set $\mathcal{N}$ representing the scenario tree is defined as $\{1\} \cup \mathcal{L}$, where $\mathcal{L}$ denotes the set of leaf nodes. The original variational inequality problem (5.17) can be decomposed into the following constituent parts.

For $n \in \mathcal{L}$, The algorithm starts by fixomg $\boldsymbol{x}_1$ and having the sub variational inequality problem given $\boldsymbol{x}_1$ as follows:

$$\begin{aligned} ST(n, \boldsymbol{x}_1): \quad 0 \in & \prod_{m \in \mathcal{P}(n)} \mu_{am} \cdot \nabla_{x_{an}} f_{an}(x_{an}; x_{-an}, p_n) \\ & + \nabla_{x_{an}} \mathcal{G}_{an}(x_{a1}, x_{an}; x_{-an}, p_n) \lambda_{an} + \mathcal{N}_{\mathcal{X}_{an}}(x_{an}), \quad \forall a \in \mathcal{A} \\ 0 \in & \quad F_n(p_n; x_n) + \mathcal{N}_{K_n}(p_n), \\ \mathcal{K}_{an} \ni & \quad \mathcal{G}_{an}(x_{a1}, x_{an}; x_{-an}, p_n) \perp \lambda_{an} \in \mathcal{K}^*_{an}, \qquad \forall a \in \mathcal{A} \end{aligned} \tag{5.18}$$

As we can see, the second stage problem is decomposed into subproblems for each node $n \in \mathcal{L}$ once the first stage variable $\boldsymbol{x}_1$ is fixed.

Then by fixing the values of $\boldsymbol{x}$ and λ at all nodes $j \in 1_+ = \mathcal{L}$, consequently, for node 1, we can define the following sub variational inequality problem given $\left(\boldsymbol{x}_{1_+}, \lambda_{1_+}\right)$ as follows:

$$\begin{aligned} ST(1, \boldsymbol{x}_{1_+}, \lambda_{1_+}): \quad 0 \in & \quad \nabla_{x_{a1}} f_{a1}(x_{a1}; x_{-a1}, p_1) + \nabla_{x_{a1}} \mathcal{G}_{a1}(x_{a0}, x_{a1}; x_{-a1}, p_1) \lambda_{a1} \\ & + \sum_{m \in 1_+} \nabla_{x_{a1}} \mathcal{G}_{am}(x_{a1}, x_{am}; x_{-am}, p_m) \lambda_{am} + \mathcal{N}_{\mathcal{X}_{a1}}(x_{a1}), \quad \forall a \in \mathcal{A} \\ 0 \in & \quad F_1(p_1; \boldsymbol{x}_1) + \mathcal{N}_{K_1}(p_1), \\ \mathcal{K}_{a1} \ni & \quad \mathcal{G}_{a1}(x_{a0}, x_{a1}; x_{-a1}, p_1) \perp \lambda_{a1} \in \mathcal{K}^*_{a1}, \qquad \forall a \in \mathcal{A} \end{aligned} \tag{5.19}$$

and solve this to generate new values for $\boldsymbol{x}_1, \boldsymbol{p}_1, (\text{and} \lambda_1)$.

Thus, like previously, the main framework of the decomposition algorithm can be depicted by the following Figure 5.1.

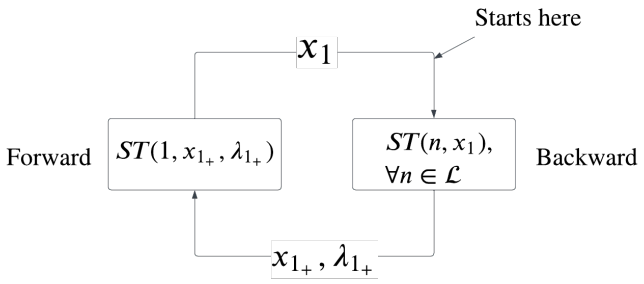


Figure 5.1: Stagewise decomposition for two-stage primal-MOPEC $\mathcal{F}(\boldsymbol{\mu})$

Note that problems ST are all MOPEC problems involving all players. The decomposition is not by player but by tree.

The main flow of the algorithm can be thought as iterating between the first-stage node problem and second-stage children node problems. At iteration r, we first go through a **backward step**: we solve the second-stage node problems as parameterized by given first-stage variable x_1^r to get an updated second-stage solution $\left((x_{an}^{r+1})_{a\in\mathcal{A}}, p_n^{r+1}, (\lambda_{an}^{r+1})_{a\in\mathcal{A}}\right)$. Then we go through a **forward step**: we solve the first-stage node problem as parameterized by given second-stage variable $(\boldsymbol{x}_{1_+}^{r+1}, \lambda_{1_+}^{r+1})$ to get updated first-stage solution $\left((x_{a1}^{r+1})_{a\in\mathcal{A}}, p_1^{r+1}\right)$. Repeat the above steps until the stopping criteria is achieved.

Following the above idea, we could have the detailed algorithm as follows:

Algorithm 9 Stagewise decomposition for two-stage primal-MOPEC $\mathcal{F}(\boldsymbol{\mu})$

1: **Input and Initialization** Set $r = 0$. Choose initial $\boldsymbol{x}_1^0$ in the solution space.
2: **while** not satisfying the stopping criterion **do**
3: **Backward step:** For $n \in 1_+$, solve each second stage problem $ST(n, x_1^r)$ to get the solutions $\left((x_{an}^{r+1})_{a\in\mathcal{A}}, p_n^{r+1}, (\lambda_{an}^{r+1})_{a\in\mathcal{A}}\right)$.
4: **Forward step:** Solve the first stage approximated equilibrium problem $ST(1, \boldsymbol{x}_{1_+}^{r+1}, \lambda_{1_+}^{r+1})$ to get $\left((x_{a1}^{r+1})_{a\in\mathcal{A}}, p_1^{r+1}\right)$.
5: Set $r = r + 1$.
6: **end while**
7: Get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \left((x_{an}^r)_{a\in\mathcal{A},n\in\mathcal{N}}, (p_n^r)_{n\in\mathcal{N}}\right)$.

Example

To illustrate the basic idea behind the approach, we consider a 2-stage dynamic NEP with 2 players. In this example $\mathcal{A} = \{1, 2\}, \mathcal{N} = \{1, 2\}, \mathcal{L} = \{2\}, 1_+ = \{2\}$. Note that x_{12} is the first players variable in the second node for example. The equilibrium problem is as

follows:

$$\text{(Player 1)} \quad \min_{x_{11},x_{12}\geq x_{11}} \frac{7}{2}(x_{11})^2 - 6x_{11} + \frac{1}{2}(x_{12})^2 + 3x_{12}x_{22} - 4x_{12} \tag{5.20}$$

$$\text{(Player 2)} \quad \min_{x_{21}=0,x_{22}} \frac{1}{2}(x_{22})^2 + x_{12}x_{22} - 3x_{22} \tag{5.21}$$

The above problem has the following equivalent reformulated KKT system (see (5.17)):

$$\begin{aligned} 0 &= 7x_{11} - 6 + \lambda_{12} \\ 0 &= x_{12} + 3x_{22} - 4 - \lambda_{12} \\ 0 &\leq \lambda_{12} \perp x_{12} - x_{11} \geq 0 \\ 0 &= x_{22} + x_{12} - 3 \end{aligned} \tag{5.22}$$

For this problem, we will have

$$\begin{aligned} ST(1, \boldsymbol{x}_{1_+}, \lambda_{1_+}) \text{ :Find } & x_{11} \text{ s.t.} \\ & 0 = 7x_{11} - 6 + \lambda_{12} \end{aligned} \tag{5.23}$$

and

$$\begin{aligned} ST(2, \boldsymbol{x}_1) \text{ :Find } & x_{11}, x_{22}, \lambda_{12} \text{ s.t.} \\ & 0 = x_{12} + 3x_{22} - 4 - \lambda_{12} \\ & 0 \leq \lambda_{12} \perp x_{12} - x_{11} \geq 0 \\ & 0 = x_{22} + x_{12} - 3 \end{aligned} \tag{5.24}$$

In the Algorithm 9, at iteration r given incoming state x_{11}^r, we could compute the solution to the NEP at stage 2. With the conditions

$$\begin{aligned} 0 = & \ x_{12}^{r+1} + 3x_{22}^{r+1} - 4 - \lambda_{12}^{r+1} \\ 0 = & \ x_{22}^{r+1} + x_{12}^{r+1} - 3 \end{aligned} \tag{5.25}$$

we could have

$$\lambda_{12}^{r+1} = 1 - 2x_{22}^{r+1}$$

- **Case 1:** $\lambda_{12}^{r+1} = 0$. We could have $x_{22}^{r+1} = \frac{1}{2}$. $x_{12}^{r+1} = \frac{5}{2}$ and $x_{11}^{r+1} = \frac{7}{6}$. In this case we get the solution $(x_{11}^*, x_{12}^*, x_{22}^*) = (\frac{7}{6}, \frac{5}{2}, \frac{1}{2})$.
- **Case 2:** $\lambda_{12}^{r+1} > 0$. We could have $x_{12}^{r+1} = x_{11}^r$. $x_{22}^{r+1} = 3 - x_{12}^{r+1} = 3 - x_{11}^r$ and $\lambda_{12}^{r+1} = x_{12}^{r+1} + 3x_{22}^{r+1} - 4 = 2x_{11}^r - 5$. In this case, we continue solve the problem

$ST(1, x_{1_+}^{r+1}, \lambda_{1_+}^{r+1})$ and could get

$$x_{11}^{r+1} = \frac{6 - \lambda_{12}^{r+1}}{7} = \frac{11 - 2x_{11}^{r}}{7}$$

This equation shows the series $\{x_{11}^r\}_{r=1,\ldots,\infty}$ is a convergent series and the limit of x_{11}^r is $x_{11}^* = \lim_{r\to\infty} x_{11}^r = \frac{11}{9}$. In this case we get the solution $(x_{11}^*, x_{12}^*, x_{22}^*) = (\frac{11}{9}, \frac{11}{9}, \frac{16}{9})$.

Note the non-uniqueness of the solution.

This could also be extended to more stable version based on proximal terms. Let's introduce the sub variational inequality problem with proximal terms. For node 1, we can define the following sub variational inequality problem given $\left(\boldsymbol{x}_{1_+}, \lambda_{1_+}\right)$, previous variables' values $\left(\boldsymbol{x}_1^r, \lambda_1^r\right)$ and proximal parameters $(\omega_{an}^x)_{a\in\mathcal{A},n\in\mathcal{N}}$:

$$\begin{aligned}
ST^{\rho}(1, \boldsymbol{x}_{1_+}, \lambda_{1_+}, \boldsymbol{x}_1^r): \quad 0 \in \quad & \nabla_{x_{a1}} f_{a1}(x_{a1}; x_{-a1}, p_1) + \nabla_{x_{a1}} \mathcal{G}_{a1}(x_{a0}, x_{a1}; x_{-a1}, p_1)\lambda_{a1} \\
& + \sum_{m\in 1_+} \nabla_{x_{a1}} \mathcal{G}_{am}(x_{a1}, x_{am}; x_{-am}, p_m)\lambda_{am} \\
& + \omega_{a1}^x \cdot (x_{a1} - x_{a1}^r) + \mathcal{N}_{\mathcal{X}_{an}}(x_{an}), \qquad \forall a \in \mathcal{A} \\
0 \in \quad & F_1(p_1; \boldsymbol{x}_1) + \mathcal{N}_{K_1}(p_1), \\
\mathcal{K}_{a1} \ni \quad & \mathcal{G}_{a1}(x_{a0}, x_{a1}; x_{-a1}, p_1) \perp \lambda_{a1} \in \mathcal{K}_{a1}^*, \qquad \forall a \in \mathcal{A}
\end{aligned} \tag{5.26}$$

For $n \in \mathcal{L}$, we will have the sub variational inequality problem given $\boldsymbol{x}_1$, previous variables' values $\left(\boldsymbol{x}_n^r, \lambda_n^r\right)$, and proximal parameters $\left((\omega_{an}^x)_{a\in\mathcal{A},n\in\mathcal{N}}, (\omega_{an}^\lambda)_{a\in\mathcal{A},n\in\mathcal{N}}\right)$:

$$\begin{aligned}
ST^{\rho}(n, \boldsymbol{x}_1, \boldsymbol{x}_n^r, \lambda_n^r): \quad 0 \in \quad & \prod_{m\in\mathcal{P}(n)} \mu_{am} \cdot \nabla_{x_{an}} f_{an}(x_{an}; x_{-an}, p_n) + \nabla_{x_{an}} \mathcal{G}_{an}(x_{a1}, x_{an}; x_{-an}, p_n)\lambda_{an} \\
& + \omega_{an}^x \cdot (x_{an} - x_{an}^r) + \mathcal{N}_{\mathcal{X}_{an}}(x_{an}), \qquad \forall a \in \mathcal{A} \\
0 \in \quad & F_n(p_n; x_n) + \mathcal{N}_{K_n}(p_n), \\
\mathcal{K}_{an} \ni \quad & \mathcal{G}_{an}(x_{a1}, x_{an}; x_{-an}, p_n) + \omega_{an}^\lambda \cdot (\lambda_{an} - \lambda_{an}^r) \perp \lambda_{an} \in \mathcal{K}_{an}^*, \qquad \forall a \in \mathcal{A}
\end{aligned} \tag{5.27}$$

The modified algorithm with proximal terms will be following:

Algorithm 10 Stagewise decomposition for two-stage primal-MOPEC $\mathcal{F}(\boldsymbol{\mu})$ with proximal terms

1: **Input and Initialization** Set $r = 0$. Choose initial $\boldsymbol{x}_1^0$ in the solution space and the appropriate proximal parameters $\left((\omega_{an}^x)_{a\in\mathcal{A},n\in\mathcal{N}}, (\omega_{an}^\lambda)_{a\in\mathcal{A},n\in\mathcal{N}}\right)$.
2: **while** not satisfying the stopping criterion **do**
3: **Backward step:** For $n \in 1_+$, solve each second stage problem $ST^\rho(n, \boldsymbol{x}_1^r, \boldsymbol{x}_n^r, \lambda_n^r)$ to get the solutions $\left((x_{an}^{r+1})_{a\in\mathcal{A}}, p_n^{r+1}, (\lambda_{an}^{r+1})_{a\in\mathcal{A}}\right)$.
4: **Forward step:** Solve the first stage approximated equilibrium problem $ST^\rho(1, \boldsymbol{x}_{1_+}, \lambda_{1_+}, \boldsymbol{x}_1^r)$ to get $\left((x_{a1}^{r+1})_{a\in\mathcal{A}}, p_1^{r+1}\right)$.
5: Set $r = r + 1$.
6: **end while**
7: Get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \left((x_{an}^r)_{a\in\mathcal{A},n\in\mathcal{N}}, (p_n^r)_{n\in\mathcal{N}}\right)$.

In the multistage problem, for the scenario node $n \in \mathcal{N}\backslash(\{1\} \cup \mathcal{L})$, we define the following sub variational inequality problem given $\left(\boldsymbol{x}_{n_-}, \boldsymbol{x}_{n_+}, \lambda_{n_+}\right)$, previous variables' values $\left(\boldsymbol{x}_n^r, \lambda_n^r\right)$:

$$
\begin{aligned}
ST(n, \boldsymbol{x}_{n_-}, \boldsymbol{x}_{n_+}, \lambda_{n_+}): \quad 0 \in & \prod_{m\in\mathcal{P}(n)} \mu_{am} \cdot \nabla_{x_{an}} f_{an}(x_{an}; x_{-an}, p_n) + \nabla_{x_{an}} \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n)\lambda_{an} \\
& + \sum_{m\in n_+} \nabla_{x_{an}} \mathcal{G}_{am}(x_{an}, x_{am}; x_{-am}, p_m)\lambda_{am} + \mathcal{N}_{\mathcal{X}_{an}}(x_{an}), \qquad \forall a \in \mathcal{A} \\
0 \in & \quad F_n(p_n; x_n) + \mathcal{N}_{K_n}(p_n), \\
\mathcal{K}_{an} \ni & \quad \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) + \omega_{an}^\lambda \cdot (\lambda_{an} - \lambda_{an}^r) \perp \lambda_{an} \in \mathcal{K}_{an}^*, \qquad \forall a \in \mathcal{A}
\end{aligned}
\tag{5.28}
$$

Thus at node n we fix all variables except those associated with the node specifically. The multistage decomposition follows the same idea as the two stage decomposition and could be interpreted as Figure 5.2.

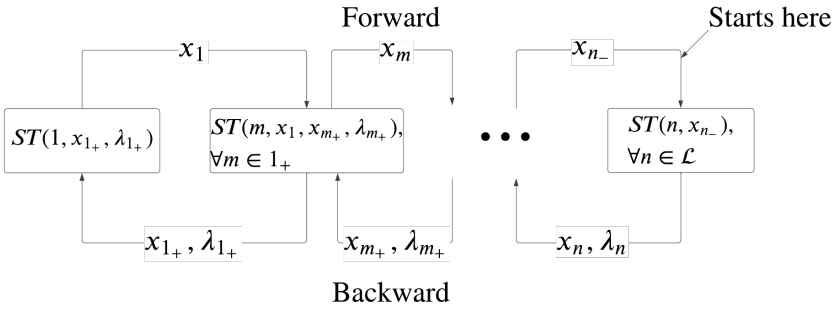


Figure 5.2: Stagewise decomposition for multistage primal-MOPEC $\mathcal{F}(\boldsymbol{\mu})$

The underlying concept illustrated in Figure 5.2 can be summarized as follows: Commencing at iteration $r \geq 1$ with an initial point $\boldsymbol{x}_n^r$ for all $n \in \mathcal{N}\backslash\mathcal{L}$, the process ini-

tially traverses through the various stage problems in a **backward** manner. This entails solving the subproblem $ST(n, \boldsymbol{x}^r_{n_-})$ for all $n \in \mathcal{N}(|\mathcal{T}|)$, yielding the updated solution $\Big((\hat{x}^r_{an})_{a\in\mathcal{A}}, \hat{p}^r_n, (\hat{\lambda}^r_{an})_{a\in\mathcal{A}}\Big)$. Subsequently, the subproblems $ST(n, \boldsymbol{x}^r_{n_-}, \hat{\boldsymbol{x}}^r_{n_+}, \hat{\lambda}^r_{n_+})$ are addressed to obtain the updated values of $\Big((\hat{x}^r_{an})_{a\in\mathcal{A}}, \hat{p}^r_n, (\hat{\lambda}^r_{an})_{a\in\mathcal{A}}\Big)$ for each $n \in \mathcal{N}(t), t = |\mathcal{T}| - 1$, whereby the time stage t is visited in a reverse order based on the sequence of time stages. Following the resolution of the root node problem, the procedure then advances in a **forward** direction over t to derive the updated solution $\Big((x^{r+1}_{an})_{a\in\mathcal{A},n\in\mathcal{N}}, (p^{r+1}_n)_{n\in\mathcal{N}}\Big)$. Note that this exploits the time and tree structure of our problem, specifically via the use of multipliers if $\mathcal{G}$.

The detailed algorithm is as follows:

Algorithm 11 Stagewise decomposition for multistage primal-MOPEC $\mathcal{F}(\boldsymbol{\mu})$

Input and Initialization Set $r = 0$. Choose initial $\boldsymbol{x}^0_{n_-}$ in the solution space for any $n \in \mathcal{N}\backslash\{1\}$.
while not satisfying the stopping criterion **do**
  Backward step:
  for $n \in \mathcal{L}$ **do**
    Solve each last stage problem $ST(n, \boldsymbol{x}^r_{n_-})$ to get the solutions $\Big((\hat{x}^r_{an})_{a\in\mathcal{A}}, \hat{p}^r_n, (\hat{\lambda}^r_{an})_{a\in\mathcal{A}}\Big)$.
  end for
  for $t = T-1, \ldots, 2$ **do**
    for $n \in \mathcal{N}(t)$ **do**
      Solve each problem $ST(n, \boldsymbol{x}^r_{n_-}, \hat{\boldsymbol{x}}^r_{n_+}, \hat{\lambda}^r_{n_+})$ to get the solutions $\Big((\hat{x}^r_{an})_{a\in\mathcal{A}}, \hat{p}^r_n, (\hat{\lambda}^r_{an})_{a\in\mathcal{A}}\Big)$.
    end for
  end for
  Forward step:
  Solve the first stage approximated equilibrium problem $ST(1, \hat{\boldsymbol{x}}^r_{1_+}, \hat{\lambda}^r_{1_+})$ to get $\Big((x^{r+1}_{a1})_{a\in\mathcal{A}}, p^{r+1}_1\Big)$.
  for $t = 2, \ldots, T-1$ **do**
    for $n \in \mathcal{N}(t)$ **do**
      Solve each problem $ST(n, \boldsymbol{x}^{r+1}_{n_-}, \hat{\boldsymbol{x}}^r_{n_+}, \hat{\lambda}^r_{n_+})$ to get $\Big((x^{r+1}_{an})_{a\in\mathcal{A}}, p^{r+1}_n, (\lambda^{r+1}_{an})_{a\in\mathcal{A}}\Big)$.
    end for
  end for
  for $n \in \mathcal{L}$ **do**
    Set $\Big((x^{r+1}_{an})_{a\in\mathcal{A}}, p^{r+1}_n\Big) = \Big((\hat{x}^r_{an})_{a\in\mathcal{A}}, \hat{p}^r_n\Big)$.
  end for
  Set $r = r + 1$.
end while
Get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \Big((x^r_{an})_{a\in\mathcal{A},n\in\mathcal{N}}, (p^r_n)_{n\in\mathcal{N}}\Big)$.

To enhance the stability of the algorithm within a multistage setting, the utilization of proximal terms becomes advantageous. In the context of the multistage problem, when considering the scenario node $n \in \mathcal{N}\backslash(\{1\} \cup \mathcal{L})$, we define a corresponding sub-problem involving the subsequent variational inequality. This sub-problem takes into account various elements, such as $\left(\boldsymbol{x}_{n_-}, \boldsymbol{x}_{n_+}, \lambda_{n_+}\right)$ (representing the current values of the variables), previous variables' values $\left(\boldsymbol{x}_n^r, \lambda_n^r\right)$, and proximal parameters $\left((\omega_{an}^x)_{a\in\mathcal{A},n\in\mathcal{N}}, (\omega_{an}^\lambda)_{a\in\mathcal{A},n\in\mathcal{N}}\right)$:

$$
\begin{array}{lll}
ST^\rho(n, \boldsymbol{x}_{n_-}, \boldsymbol{x}_{n_+}, \lambda_{n_+}, \boldsymbol{x}_n^r, \lambda_n^r): & 0 \in & \prod_{m\in\mathcal{P}(n)} \mu_{am} \cdot \nabla_{x_{an}} f_{an}(x_{an}; x_{-an}, p_n) + \nabla_{x_{an}} \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n)\lambda_{an} \\
& & + \sum_{m\in n_+} \nabla_{x_{an}} \mathcal{G}_{am}(x_{an}, x_{am}; x_{-am}, p_m)\lambda_{am} \\
& & + \omega_{an}^x \cdot (x_{an} - x_{an}^r) + \mathcal{N}_{\mathcal{X}_{an}}(x_{an}), \qquad \forall a \in \mathcal{A} \\
& 0 \in & F_n(p_n; x_n) + \mathcal{N}_{K_n}(p_n), \\
& \mathcal{K}_{an} \ni & \mathcal{G}_{an}(x_{an_-}, x_{an}; x_{-an}, p_n) + \omega_{an}^\lambda \cdot (\lambda_{an} - \lambda_{an}^r) \perp \lambda_{an} \in \mathcal{K}_{an}^*, \qquad \forall a \in \mathcal{A}
\end{array}
\tag{5.29}
$$

Algorithm 12 Stagewise decomposition for multistage primal-MOPEC $\mathcal{F}(\boldsymbol{\mu})$ with proximal terms

Input and Initialization Set $r = 0$. Choose initial $\boldsymbol{x}_{n_-}^0$ in the solution space for any $n \in \mathcal{N}\backslash\{1\}$.
while not satisfying the stopping criterion **do**
  Backward step:
  for $n \in \mathcal{L}$ **do**
    Solve each last stage problem $ST^{\rho}(n, \boldsymbol{x}_{n_-}^r, \boldsymbol{x}_n^r, \lambda_n^r)$ to get the solutions $\Big((\hat{x}_{an}^r)_{a\in\mathcal{A}}, \hat{p}_n^r, (\hat{\lambda}_{an}^r)_{a\in\mathcal{A}}\Big)$.
  end for
  for $t = T-1, \ldots, 2$ **do**
    for $n \in \mathcal{N}(t)$ **do**
      Solve each problem $ST^{\rho}(n, \boldsymbol{x}_{n_-}^r, \hat{\boldsymbol{x}}_{n_+}^r, \hat{\lambda}_{n_+}^r, \boldsymbol{x}_n^r, \lambda_n^r)$ to get the solutions $\Big((\hat{x}_{an}^r)_{a\in\mathcal{A}}, \hat{p}_n^r, (\hat{\lambda}_{an}^r)_{a\in\mathcal{A}}\Big)$.
    end for
  end for
  Forward step:
  Solve the first stage approximated equilibrium problem $ST^{\rho}(1, \hat{\boldsymbol{x}}_{1_+}^r, \hat{\lambda}_{1_+}^r, \boldsymbol{x}_1^r)$ to get $\Big((x_{a1}^{r+1})_{a\in\mathcal{A}}, p_1^{r+1}\Big)$.
  for $t = 2, \ldots, T-1$ **do**
    for $n \in \mathcal{N}(t)$ **do**
      Solve each problem $ST^{\rho}(n, \boldsymbol{x}_{n_-}^{r+1}, \hat{\boldsymbol{x}}_{n_+}^r, \hat{\lambda}_{n_+}^r, \hat{\boldsymbol{x}}_n^r, \hat{\lambda}_n^r)$ to get $\Big((x_{an}^{r+1})_{a\in\mathcal{A}}, p_n^{r+1}, (\lambda_{an}^{r+1})_{a\in\mathcal{A}}\Big)$.
    end for
  end for
  for $n \in \mathcal{L}$ **do**
    Set $\Big((x_{an}^{r+1})_{a\in\mathcal{A}}, p_n^{r+1}\Big) = \Big((\hat{x}_{an}^r)_{a\in\mathcal{A}}, \hat{p}_n^r\Big)$.
  end for
  Set $r = r + 1$.
end while
Get the solution $(\boldsymbol{x}^*, \boldsymbol{p}^*) = \Big((x_{an}^r)_{a\in\mathcal{A}, n\in\mathcal{N}}, (p_n^r)_{n\in\mathcal{N}}\Big)$.

This algorithm is identical to the previous one except the subproblems involve proximal terms.

5.3 Numerical experiments

This section presents numerical results to demonstrate the algorithm's performance, utilizing the identical parameter setting employed in the numerical experiments described in

chapter 4. The experiments are conducted on the primal-MOPEC $\mathcal{F}(\boldsymbol{\mu})$ of an economic dispatch example, incorporating *Type II* market constraint and employing a risk probability vector $\boldsymbol{\mu}$ based on scenario tree 2. The objective is to assess the effectiveness of this algorithm and examine the influence of the proximal term size ω_{an} on its performance.

Throughout the experiments, a uniform value is assigned to each ω_{an}, denoted as ω for simplicity. Figure 5.3 illustrates the impact of the proximal term size on the algorithm's performance. The x-axis represents the iteration number, while the y-axis represents the logarithm (base 10) of the FB residual at each iteration of Algorithm 12. In these experiments, the algorithm's performance is tested for various values of ω, specifically $\omega = 0, 0.1, 0.5, 1, 1.5, 5, 10$. The results indicate that the algorithm fails to converge to an equilibrium point in the absence of proximal terms. Conversely, excessively large proximal term sizes yield an overly conservative algorithm that exhibits slower convergence. Furthermore, Figure 5.3 reveals that Algorithm 12 exhibits a linear convergence rate.

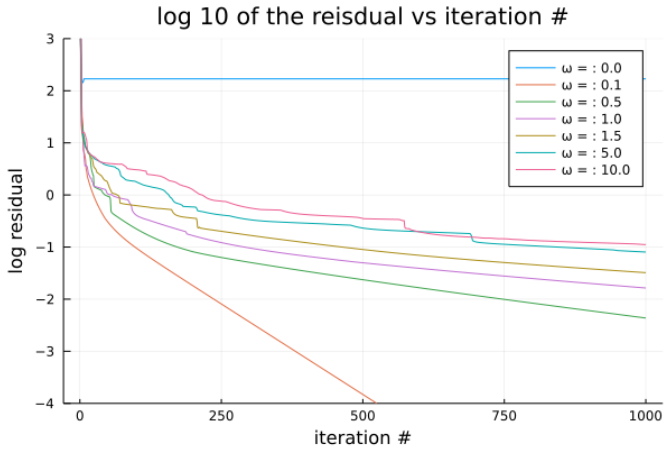


Figure 5.3: Plot of FB-Residual vs iteration # based on different size of ω

5.4 Conclusion

In this chapter, we introduce a novel stage-based decomposition algorithm designed to solve the primal-MOPEC problem under a risk probability vector. Additionally, we propose a computational enhancement technique by incorporating proximal terms. To evaluate the effectiveness of these algorithms, we conduct a performance analysis using the economic dispatch problem with a *Type II* market constraint.

The computational experiments provide valuable insights into the impact of proximal terms on algorithm convergence. Specifically, we observe that the inclusion of proximal terms can facilitate convergence in cases where the algorithm would otherwise diverge.

Moreover, the convergence speed is significantly influenced by the size of the proximal terms, with larger sizes resulting in slower convergence rates. Furthermore, the experiments reveal a distinct linear convergence rate exhibited by the algorithm.

These experimental findings demonstrate the efficacy of the proposed stage-decomposition algorithm in addressing the primal-MOPEC problem. This approach offers a significant improvement over the primal-MOPEC-dual-risk decomposition algorithm presented in chapter 4, particularly when faced with computational intractability due to the exponential growth in problem size as a function of the number of stages.

However, tuning of the approach is important since while the subproblems are much smaller MOPECs, we must endeavor to limit the number of these that must be processed in order to have a fast overall algorithm.

6 MULTISTAGE STOCHASTIC EQUILIBRIUM PROBLEM IN GENERAL FORMAT

This chapter is devoted to investigating stochastic MOPECs with stochastic processes that follow general random distributions. Previous chapters have provided a formal discussion on the formulation of a general stochastic MOPEC with risk-averse players based on a discrete scenario tree, as outlined in chapter 1. Various approaches to solving this problem in the discrete setting have been introduced in preceding chapters, including chapters 3, 4, and 5. However, these approaches are not directly applicable to scenarios where stochastic processes follow general random distributions. To address this limitation, we present a sample average approximation method in this chapter. This method allows us to approximate a general problem using an appropriate problem based on a scenario tree, which aids in achieving a relatively good equilibrium point of the original problem. Furthermore, we develop a two-level graph representation of the stochastic MOPEC problem, offering a concise representation of stochastic information and avoiding the need to input the entire scenario tree when defining the stochastic problem.

The chapter is structured as follows: We begin with section 6.1 to first introduce a mathematical definition of the dynamic MOPEC, as presented in chapter 1. With certain parameters of the dynamic MOPEC following a general stochastic process, we naturally introduce the mathematical formulation of the stochastic version of the problem. In section 6.3, we present a practical framework for representing any general stochastic process through a one-level graph representation. This framework allows researchers to input information from each stage instead of the entire scenario tree. Expanding on the framework introduced in section 6.3, section 6.4 presents a two-level graph representation framework for the entire stochastic MOPEC with general stochastic processes. This framework facilitates the input of information for the entire stochastic MOPEC from each stage, eliminating the need to provide the entire scenario tree. Finally, section 6.6 provides the main findings and conclusions derived from the research conducted in this chapter.

6.1 Multistage stochastic equilibrium problem with general random distributions

This chapter presents the mathematical concepts of general stochastic MOPEC with risk-averse players and a new framework called two-level graph representation for the general stochastic problem.

6.1.1 Preliminaries using dynamic equilibrium

In this section, we want to show how to generate a stochastic MOPEC under general uncertainty process from a deterministic MOPEC in the two stage setting. Let's first show a Walras barter dynamic equilibrium model with two stages $t = 1, 2$ and a set of players $a \in \mathcal{A}$ trading a finite number (L) of goods in each stage at the prices $p_1, p_2 \in \mathbb{R}^L$. These goods can be used in activities that depend on the input-output technologies (S_a, R_a) available to them. $e_{a1}, e_{a2} \in \mathbb{R}^L$ are the endowments for each player a at stage 1 and 2. $c_{a1}, c_{a2} \in \mathbb{R}^L$ are the consumptions by each player a in stage 1 and 2. $y_{a1}, y_{a2} \in \mathbb{R}^{m_a}_+$ represents the transfer activity levels. Let $x_{a1} = (c_{a1}, y_{a1})$, $x_{a2} = (c_{a2}, y_{a2})$, $\mathcal{X}_{a1} = \mathbb{R}^L_+ \times \mathbb{R}^{m_a}_+$ and $\mathcal{X}_{a2} \in \mathbb{R}^L_+ \times \mathbb{R}^{m_a}_+$.

The dynamic optimization problem for each player $a \in \mathcal{A}$ with fixed price $p = (p_1, p_2)$ is:

$$\begin{aligned} \min_{c_{a1}, c_{a2}, y_{a1}, y_{a2}} \quad & u_{a1}(c_{a1}) + u_{a2}(c_{a2}) \\ \text{s.t.} \quad & (p_1)^T c_{a1} + (p_1)^T S_a y_{a1} \leq (p_1)^T e_{a1} \\ & (p_2)^T c_{a2} - (p_2)^T R_a y_{a2} \leq (p_2)^T e_{a2}, \\ & y_{a1} = y_{a2}, \\ & c_{a1}, c_{a2} \in \mathbb{R}^L_+, \quad y_{a1}, y_{a2} \in \mathbb{R}^{m_a}_+ \end{aligned} \tag{6.1}$$

For each player, constraints $(p_1)^T c_{a1} + (p_1)^T S_a y_{a1} \leq (p_1)^T e_{a1}$ and $(p_2)^T c_{a2} - (p_2)^T R_a y_{a2} \leq (p_2)^T e_{a2}$ restrict each player to limit its market value of consumptions to its market value of endowments in both stages. $y_{a1} = y_{a2}$ are the goods transformed from stage 1 to stage 2. $u_{at}(\cdot), t = 1, 2$ are the cost function and the total cost is the summation of them. Also one thing to be noticed is that S_a and e_{a1} will be available to the players before they make decisions (c_{a1}, y_{a1}). But R_a and e_{a2} are not available to the player at that time until he make decisions (c_{a2}, y_{a2}).

The market constraint in this problem is that in each time stage the total demand doesn't exceed total supply, i.e.

$$\begin{aligned} 0 \leq \sum_{a \in \mathcal{A}} (e_{a1} - c_{a1} - S_a y_{a1}) \perp p_1 \geq 0 \\ 0 \leq \sum_{a \in \mathcal{A}} (e_{a2} + R_a y_{a2} - c_{a2}) \perp p_2 \geq 0 \end{aligned} \tag{6.2}$$

equivalently, this is

$$\begin{aligned} 0 \in \sum_{a \in \mathcal{A}} (e_{a1} - c_{a1} - S_a y_{a1}) + \mathcal{N}_{\mathbb{R}^L_+}(p_1) \\ 0 \in \sum_{a \in \mathcal{A}} (e_{a2} + R_a y_{a2} - c_{a2}) + \mathcal{N}_{\mathbb{R}^L_+}(p_2) \end{aligned} \tag{6.3}$$

Here we can see in this problem

$$\begin{aligned} F_1(p_1; x_1) &= \sum_{a\in\mathcal{A}}(e_{a1} - c_{a1} - S_a y_{a1}) \\ F_2(p_2; x_2) &= \sum_{a\in\mathcal{A}}(e_{a2} + R_a y_{a2} - c_{a2}) \\ K_1 &= \mathbb{R}^L_+ \\ K_2 &= \mathbb{R}^L_+ \end{aligned} \tag{6.4}$$

As we can see, if all the parameters in the problem are just deterministic parameters, we can use the methods for solving equilibrium problem to solve the dynamic equilibrium problem. However, if the parameters involve uncertainty and are stochastic processes, the equilibrium problem will become a stochastic equilibrium problem and needs more specific methods to formulate it.

In this example we have already shown that the strategy of each player is following a sequential pattern: although $c_{a1}, y_{a1}, c_{a2}, y_{a2}$ all belong to the strategy of player a, player will first decide the optimal value for c_{a1} and y_{a1} at stage 1, then he will decide the optimal value for c_{a2} and y_{a2} until stage 2. Now consider the case when the output technology R_a and second-stage endowments e_{a2} follow known distributions. In practical situation the distribution of this random vector can be estimated, say from historical data. Then, the players will decide their strategy $(c_{a2}, y_{a2})_{a\in\mathcal{A}}$ as a recourse action corresponding to the realization of $(R_a, e_{a2})_{a\in\mathcal{A}}$.

Suppose for the moment the random vector $(R_a, e_{a2})_{a\in\mathcal{A}}$ has a finitely supported distribution, i.e., it takes values $(R^1_a, e^1_{a2}), (R^2_a, e^2_{a2}), \ldots, (R^K_a, e^K_{a2})$ (called scenarios) with respective probabilities $\pi_1, \ldots, \pi_K$. For each scenario $(R^k_a, e^k_{a2}), k = 1, \ldots, K$, each player would have a corresponding second stage strategy $(c^k_{a2}, y^k_{a2})_{a\in\mathcal{A}}$, then the second stage market will also have a recourse action p^k_2 for each scenario $k = 1, \ldots, K$. Thus, we would have different feasible regions of each player under scenario $k = 1, \ldots, K$, which is

$$\begin{aligned} (p^k_2)^T c^k_{a2} - (p^k_2)^T R^k_a y^k_{a2} \le (p^k_2)^T e^k_{a2}, \quad \text{for } k = 1, \ldots, K \\ y^k_{a2} = y_{a1}, \quad \text{for } k = 1, \ldots, K \end{aligned} \tag{6.5}$$

and will have different market constraint

$$0 \le \sum_{a\in\mathcal{A}}(e^k_{a2} + R^k_a y^k_{a2} - c^k_{a2}) \perp p^k_2 \ge 0, \quad \text{for } k = 1, \ldots, K \tag{6.6}$$

Now let's transfer our focus on the first stage. At first stage, even the players know

the distribution of random vector $(R_a, e_{a2})_{a\in\mathcal{A}}$, they still don't know which scenario will happen. To make a reasonable decision about which actions should be taken at first stage, they need to find a way to measure the effect of all possible scenarios that could happen. A natural way here is to consider the expected value, denote $\mathbb{E}[u_{a2}(c_{a2})]$ in the first stage. In the case of finitely many scenarios we could write the expected value $\mathbb{E}[u_{a2}(c_{a2})]$ as the weighted sum:

$$\mathbb{E}[u_{a2}(c_{a2})] = \sum_{k=1}^{K} \phi_k \cdot u_{a2}(c_{a2}^k) \tag{6.7}$$

In this way it is possible to model the two-stage stochastic equilibrium problem as a deterministic equilibrium problem:

$$\begin{aligned}
\min_{c_{a1},(c_{a2}^k)_{k=1}^K, y_{a1},(y_{a2}^k)_{k=1}^K} \quad & u_{a1}(c_{a1}) + \sum_{k=1}^{K} \phi_k \cdot u_{a2}(c_{a2}^k) \\
\text{s.t.} \quad & (p_1)^T c_{a1} + (p_1)^T S_a y_{a1} \le (p_1)^T e_{a1} \\
& (p_2^k)^T c_{a2}^k - (p_2^k)^T R_a^k y_{a2}^k \le (p_2^k)^T e_{a2}^k, \qquad \text{for } k = 1, \dots, K \\
& y_{a2}^k = y_{a1}, \qquad \text{for } k = 1, \dots, K \\
& c_{a1} \in \mathbb{R}_+^L, \quad y_{a1} \in \mathbb{R}_+^{m_a} \\
& c_{a2}^k \in \mathbb{R}_+^L, \quad y_{a2}^k \in \mathbb{R}_+^{m_a}, \qquad \text{for } k = 1, \dots, K
\end{aligned} \tag{6.8}$$

with the mixed complementarity constraints

$$\begin{aligned}
0 &\le \sum_{a\in\mathcal{A}} (e_{a1} - c_{a1} - S_a y_{a1}) \perp p_1 \ge 0, \\
0 &\le \sum_{a\in\mathcal{A}} (e_{a2}^k + R_a^k y_{a2}^k - c_{a2}^k) \perp p_2^k \ge 0, \quad \text{for } k = 1, \dots, K
\end{aligned} \tag{6.9}$$

Based on the formulations of finitely many scenarios, we could extend and achieve the formulations of general situations. Denote $\xi : \Omega \to \mathbb{R}^d$ be a random vector in probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with support set $\Xi \subset \mathbb{R}^d$. For any $\xi \in \Omega$ there is a realization $(R_a(\xi))_{a\in\mathcal{A}}$ and $(e_{a2}(\xi))_{a\in\mathcal{A}}$. Each player $a \in \mathcal{A}$ will have stratety $(c_{a2}(\xi), y_{a2}(\xi))$. The market at second stage will also have a corresponding price $p_2(\xi)$. On the contrary, the player's first stage strategy (c_{a1}, y_{a1}) and first stage market price p_1 will not be affected by the random vector ξ. In another words, we know that the actions, prices and uncertainty are revealed in a

sequential way as follows:

$$\begin{array}{ccc} & \xi & \\ & \downarrow & \\ (\{c_{a1}, y_{a1}\}_{a\in\mathcal{A}}, p_1) \rightarrow & & (\{c_{a2}(\xi), y_{a2}(\xi)\}_{a\in\mathcal{A}}, p_2(\xi)) \end{array} \tag{6.10}$$

And at the second stage, just like in discrete situation, the player needs to adjust their second stage strategy. So only the second stage actions and prices will be affected by ξ_2. Let $c_{a2}(\xi_2^i), p_2(\xi_2^i)$ be the actions and prices under realization $\xi_2^i, \quad i = 1, 2, 3$. To be a feasible action, under realization $\xi_2^i, i = 1, 2, 3$, they must all satisfy the player's feasible constraint

$$p_2(\xi_2^i)^T c_{a2}(\xi_2^i) - p_2(\xi_2^i)^T R_a(\xi_2^i) y_a \leq p_2(\xi_2^i)^T e_{a2}(\xi_2^i)$$

and the market constraint

$$0 \leq \sum_{a\in\mathcal{A}} (e_{a2}(\xi_2^i) + R_a(\xi_2^i) y_a - c_{a2}(\xi_2^i)) \perp p_2(\xi_2^i) \geq 0$$

Also, under realization $\xi_2^i, i = 1, 2, 3$, the cost of player a at stage 2 is $u_{a2}(c_{a2}(\xi_2^i), \xi_2^i)$. The above derivation from a dynamic equilibrium to a stochastic equilibrium can also be shown by Figure 6.1, where each node in the right hand side of the Figure 6.1 represents a realization of ξ_2. Each realization of ξ_2 generates the arrow between node from two nodes.

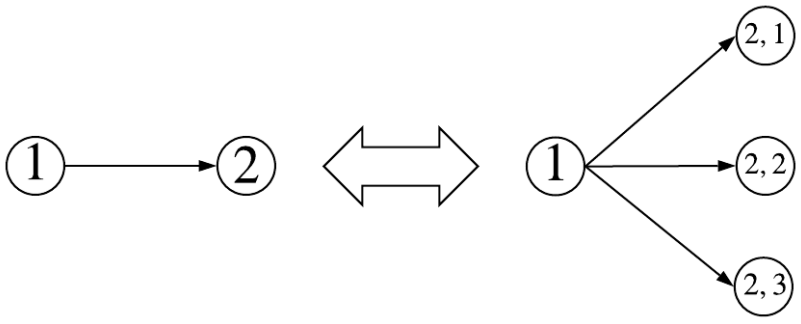


Figure 6.1: Transformation from a two stage dynamic equilibrium to a two stage stochastic equilibrium

Assume $\phi(\xi_2^i), i = 1, 2, 3$ are the probabilities of realization ξ_2. The stochastic equilibrium

problem will be

$$\begin{aligned}
\min_{x_{a\cdot}} \quad & u_{a1}(c_{a1}) + \sum_{i=1}^{3} \phi(\xi_2^i)\Big[u_{a2}(c_{a2}(\xi_2^i), \xi_2^i)\Big] \\
\text{s.t.} \quad & (p_1)^T c_{a1} + (p_1)^T S_a y_a \leq (p_1)^T e_{a1}, \\
& p_2(\xi_2^i)^T c_{a2}(\xi_2^i) - p_2(\xi_2^i)^T R_a(\xi_2^i) y_a \leq p_2(\xi_2^i)^T e_{a2}(\xi_2^i), \qquad \forall i = 1,2,3 \\
& c_{a1} \in \mathbb{R}_+^L, \quad y_a \in \mathbb{R}_+^{m_a}, \\
& c_{a2}(\xi_2^i) \in \mathbb{R}_+^L, \quad i = 1,2,3
\end{aligned} \tag{6.11}$$

with market constraint:

$$\begin{aligned}
0 \leq \sum_{a\in\mathcal{A}} (e_{a1} - c_{a1} - S_a y_a) \perp p_1 \geq 0 \\
0 \leq \sum_{a\in\mathcal{A}} \Big(e_{a2}(\xi_2^i) + R_a(\xi_2^i) y_a - c_{a2}(\xi_2^i)\Big) \perp p_2(\xi_2^i) \geq 0, \quad \forall i = 1,2,3
\end{aligned} \tag{6.12}$$

Here $\sum_{i=1}^{3} \phi(\xi_2^i)\Big[u_{a2}(c_{a2}(\xi_2^i), \xi_2^i)\Big]$ is the expectation of the cost in second stage, and note that prices clear in each stage and each realization.

Based on the above example, we will introduce the general two stage stochastic equilibrium problem. Consider the dynamic equilibrium (1.35) with $\mathcal{T} = \{1, 2\}$. In this situation, we will only have ξ_1, ξ_2. We always assume ξ_1 is constant and ξ_2 is a random vector in the probability space $(\Xi_2, \mathcal{F}, \mathbb{P})$. Like (6.10), the elements of the stochastic equilibrium are revealed in a sequential way as follows:

$$\begin{array}{ccc}
 & \xi_2 & \\
 & \downarrow & \\
(\{x_{a1}\}_{a\in\mathcal{A}}, p_1) & \rightarrow & (\{x_{a2}\}_{a\in\mathcal{A}}, p_2)
\end{array} \tag{6.13}$$

Therefore, for any $\xi_2 \in \Xi_2$, the players' actions and the market price at second stage will be affected by it. Generally, we can build the mapping of player actions x_a and market price p of any scenario in $\xi_2 \in \Xi_2$:

$$\begin{aligned}
x_a(\cdot) : \xi_2 \rightarrow x_a(\xi_2) = (x_{a1}, x_{a2}(\xi_2)), \quad \forall a \in \mathcal{A} \\
p(\cdot) : \xi_2 \rightarrow p(\xi_2) = (p_1, p_2(\xi_2))
\end{aligned} \tag{6.14}$$

In fact, in the two stage problem, the property of $x_a(\cdot), p(\cdot)$ that only $(\{x_{a2}\}_{a\in\mathcal{A}}, p_2)$ will depend on ξ_2 is called nonanticipativity. The property is also equivalent to the measurability

of $(\{x_{a2}\}_{a\in\mathcal{A}}, p_2)$ with respect to the $\sigma-$field $\mathcal{F}_2 \subseteq \mathcal{F}$, which is generated by ξ_2. At $t = 1$ the input is known, we just assume that $\mathcal{F}_1 = \{\emptyset, \Xi_2\}$. Clearly, we will have $\mathcal{F}_1 \subseteq \mathcal{F}_2$.

The two-stage stochastic equilibrium problem with risk-neutral players then will be of the following form:

$$\begin{aligned}
\min_{x_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1, \xi_1) + \mathbb{E}_{\mathbb{P}}\Big[f_{a2}(x_{a2}(\xi_2); x_{-a2}(\xi_2), p_2(\xi_2), \xi_2)\Big] \\
\text{s.t.} \quad & \mathcal{G}_{a1}(x_{a0}, x_{a1}; x_{-a1}, p_1, \xi_1) \in \mathcal{K}_{a1}, \\
& \mathcal{G}_{a2}(x_{a1}, x_{a2}(\xi_2); x_{-a2}(\xi_2), p_2(\xi_2), \xi_2) \in \mathcal{K}_{a2}, \qquad \forall \xi_2 \in \Xi_2 \\
& x_{a1} \in \mathcal{X}_{a1}, \quad x_{a2}(\xi_2) \in \mathcal{X}_{a2}, \quad \forall \xi_2 \in \Xi_2
\end{aligned} \tag{6.15}$$

with equilibrium constraint:

$$\begin{aligned}
& 0 \in F_1(p_1; x_1, \xi_1) + \mathcal{N}_{K_1}(p_1), \\
& 0 \in F_2(p_2(\xi_2); x_2(\xi_2), \xi_2) + \mathcal{N}_{K_2}(p_2(\xi_2)), \quad \forall \xi_2 \in \Xi_2
\end{aligned} \tag{6.16}$$

Here $\mathbb{E}_{\mathbb{P}}(\cdot)$ is the expectation operator on the random variable $f_{a2}(x_{a2}(\xi_2); x_{-a2}(\xi_2), p_2(\xi_2), \xi_2)$ with corresponding probability $\mathbb{P}$, which is the most commonly used risk measure in the literature. We could replace expectation by coherent risk measure ρ_{a1} to have the following risk-averse problem for each player:

$$\begin{aligned}
\min_{x_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1, \xi_1) + \rho_{a1}\Big[f_{a2}(x_{a2}(\xi_2); x_{-a2}(\xi_2), p_2(\xi_2), \xi_2)\Big] \\
\text{s.t.} \quad & \mathcal{G}_{a1}(x_{a0}, x_{a1}; x_{-a1}, p_1, \xi_1) \in \mathcal{K}_{a1}, \\
& \mathcal{G}_{a2}(x_{a1}, x_{a2}(\xi_2); x_{-a2}(\xi_2), p_2(\xi_2), \xi_2) \in \mathcal{K}_{a2}, \qquad \forall \xi_2 \in \Xi_2 \\
& x_{a1} \in \mathcal{X}_{a1}, \quad x_{a2}(\xi_2) \in \mathcal{X}_{a2}, \quad \forall \xi_2 \in \Xi_2
\end{aligned} \tag{6.17}$$

6.2 Multistage stochastic equilibrium problem

6.2.1 Preliminaries using dynamic equilibrium

In this section, we extend our derivation of stochastic equilibrium from two stages to $|\mathcal{T}|$ stages. To do this we consider the multistage dynamic equilibrium first. We defined a mathematical formulation of a dynamic equilibrium (1.35) over $|\mathcal{T}|$ stages parameterized with fixed parameter $\{\xi_t\}_{t\in\mathcal{T}}$.

A concrete example of a dynamic multistage equilibrium problem is the multistage dynamic Walras barter model with activities. Suppose $\mathcal{T} = \{1, 2, \ldots, |\mathcal{T}|\}$, where $|\mathcal{T}|$ can be any positive number. There are player $a \in \mathcal{A}$ and there are a finite number (L) of goods

in each stage, which causes $p_t \in \mathbb{R}^L$ for any $t \in \mathcal{T}$. $e_{at} \in \mathbb{R}^L_+$ is the endowment for each player $a \in \mathcal{A}$ at stage $t \in \mathcal{T}$.There is an input-output process in each time stage described by the sequence of matrices $\{(S_a^{t-1}, R_a^t), t \in \mathcal{T}\backslash\{1\}\}$. Here $S_a^{t-1}, R_a^t \in \mathbb{R}^{m_a \times L}$. The sequence of decisions for player a is

$$x_a = \Big(x_{a1}, x_{a2}, \ldots, x_{a|\mathcal{T}|}\Big)$$

where

$$x_{at} = \begin{cases} (c_{at}, y_{at}), & \text{if} \quad t \in \mathcal{T}\backslash\{|\mathcal{T}|\} \\ c_{at}, & \text{if} \quad t = |\mathcal{T}| \end{cases}$$

$c_{at} \in \mathbb{R}^L$ is the consumption of each player in stage t. $y_{at} \in \mathbb{R}^{m_a}_+$ is the transfer activity level of each player. This is used to define $\mathcal{G}_{at}$ as follows:

$$\mathcal{G}_{at}(x_{at-1}, x_{at}; p_t, \xi_t) := \begin{cases} p_1^T c_{a1} + p_1^T S_a^1(\xi_1) y_{a1} \le p_1^T e_{a1}, & \text{if } \ t = 1 \\ p_t^T c_{at} - p_t^T R_a^t(\xi_t) y_{a,t-1} + p_t^T S_a^t(\xi_t) y_{at} \le p_t^T e_{at}(\xi_t), & \text{if } \ t \in \mathcal{T}\backslash\{1, |\mathcal{T}|\} \\ p_t^T c_{at} - p_t^T R_a^t(\xi_t) y_{a,t-1} \le p_t^T e_{at}(\xi_t), & \text{if } \ t = |\mathcal{T}| \end{cases} \tag{6.18}$$

The market clearing is defined as follows:

$$F_{at}(p_t; x_{at}, \xi_t) := \begin{cases} \sum_{a\in\mathcal{A}}(e_{a1} - c_{a1} - S_a^1 y_{a1}), & \text{if } \ t = 1 \\ \sum_{a\in\mathcal{A}}(e_{at} + R_a^t y_{at-1} - c_{at} - S_a^t y_{at}), & \text{if } \ t \in \mathcal{T}\backslash\{1, |\mathcal{T}|\} \\ \sum_{a\in\mathcal{A}}(e_{at} + R_a^t y_{a,t-1} - c_{at}), & \text{if } \ t = |\mathcal{T}| \end{cases} \tag{6.19}$$

For simplicity here, we just write $S_a^t(\xi_t) = S_a^t, R_a^t(\xi_t) = R_a^t, e_{at}(\xi_t) = e_{at}, u_{at}(\cdot, \xi_t) = u_{at}$, then the dynamic optimization problem (defined using $x, \mathcal{G}$ and F in our general form (1.35) for each player $a \in \mathcal{A}$ with fixed price $p = (p_1, \ldots, p_{|\mathcal{T}|})$ becomes:

$$\begin{aligned} \min_{x_{a\cdot}} \quad & \sum_{t\in\mathcal{T}} u_{at}(c_{at}) \\ \text{s.t.} \quad & p_1^T c_{a1} + p_1^T S_a^1 y_{a1} \le p_1^T e_{a1}, \\ & p_t^T c_{at} - p_t^T R_a^t y_{at-1} + p_t^T S_a^t y_{at} \le p_t^T e_{at}, \quad \forall t \in \mathcal{T}\backslash\{1, |\mathcal{T}|\} \\ & p_{|\mathcal{T}|}^T c_{a|\mathcal{T}|} - p_{|\mathcal{T}|}^T R_a^{|\mathcal{T}|} y_{a|\mathcal{T}|-1} \le p_{|\mathcal{T}|}^T e_{a|\mathcal{T}|}, \\ & c_{at} \in \mathbb{R}^L_+, \quad y_{at} \in \mathbb{R}^{m_a}_+ \end{aligned} \tag{6.20}$$

with the market clearing constraints:

$$\begin{aligned} 0 \leq \sum_{a\in\mathcal{A}}(e_{a1} - c_{a1} - S_a^1 y_{a1}) \perp p_1 \geq 0 \\ 0 \leq \sum_{a\in\mathcal{A}}(e_{at} + R_a^t y_{at-1} - c_{at} - S_a^t y_{at}) \perp p_t \geq 0, \quad \forall t \in \mathcal{T}\backslash\{1, |\mathcal{T}|\} \\ 0 \leq \sum_{a\in\mathcal{A}}(e_{a|\mathcal{T}|} + R_a^{|\mathcal{T}|} y_{a|\mathcal{T}|-1} - c_{a|\mathcal{T}|}) \perp p_{|\mathcal{T}|} \geq 0 \end{aligned} \tag{6.21}$$

6.2.2 Extension to stochastic equilibrium

Define

$$\xi = (\xi_1, \ldots, \xi_t, \ldots, \xi_{|\mathcal{T}|}) \tag{6.22}$$

The player is making decisions in a sequential way:

$$\begin{array}{ccccccc} & \xi_1 & & \xi_2 & & \xi_{|\mathcal{T}|-1} & & \xi_{|\mathcal{T}|} & \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\ (\{x_{a0}\}_{a\in\mathcal{A}}) \rightarrow & (\{x_{a1}\}_{a\in\mathcal{A}}, p_1) \rightarrow & \cdots\cdots\cdots & \rightarrow & (\{x_{a|\mathcal{T}|-1}\}_{a\in\mathcal{A}}, p_{|\mathcal{T}|-1}) & \rightarrow & (\{x_{a|\mathcal{T}|}\}_{a\in\mathcal{A}}, p_{|\mathcal{T}|}) \end{array} \tag{6.23}$$

To satisfy (6.23), $x_t(\xi)$ and $p_t(\xi)$ at stage t must be nonanticipative, which means players' actions $x_t(\xi)$ and market price $p_t(\xi)$ depends only on a portion $(\xi_1, \ldots, \xi_t)$. Denote $\xi_{[t]} = (\xi_2, \ldots, \xi_t), \quad \forall t \in \mathcal{T}\backslash\{0\}$,

$$\begin{aligned} x_a(\cdot) : \xi \rightarrow x_a(\xi) = (x_{a1}, x_{a2}(\xi_2), x_{a3}(\xi_3), \ldots, x_{a|\mathcal{T}|}(\xi_{|\mathcal{T}|})), \quad \forall a \in \mathcal{A} \\ p(\cdot) : \xi \rightarrow p(\xi) = (p_1, p_2(\xi_2), p_3(\xi_3), \ldots, p_{|\mathcal{T}|}(\xi_{|\mathcal{T}|})) \end{aligned} \tag{6.24}$$

Note that $\{\xi_t\}_{t\in\mathcal{T}}$ is a stochastic process, where ξ_t lies in sets Ξ_t and Ξ is the probability space of the stochastic process ξ. There exists a probability measure $\mathbb{P}\{\xi \in \Xi\} = \mathbb{P}\{(\xi_2, \xi_3, \ldots, \xi_{|\mathcal{T}|}) \in \Xi\} = 1$. Let $\Xi_{[t]} = \{(\xi_1, \xi_2, \ldots, \xi_t) | \xi = (\xi_2, \ldots, \xi_t) \in \Xi\}$ be the probability space for $(\xi_2, \ldots, \xi_t)$ for $t \in [2, |\mathcal{T}|] \cap \mathbb{N}$. Then there will be uncertainty in the dynamic equilibrium problem (1.35), transforming the problem into the stochastic equilibrium problem. In another words, $x_{at}(\cdot)$ is the player a's decision process and is $\mathcal{F}_t-$ measurable, and

$$p(\xi) = (p_1, p_2(\xi_2), p_3(\xi_3), \ldots, p_{|\mathcal{T}|}(\xi_{|\mathcal{T}|})) \tag{6.25}$$

is the price process determined by the market and $p_t(\cdot)$ is $\mathcal{F}_t-$ measurable. Based on this, a stochastic equilibrium based on risk neutral players is defined in the following:

Definition 6.1. Multistage stochastic equilibrium problem with risk-neutral players:

$$
\begin{aligned}
\min_{x_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1, \xi_1) + \mathbb{E}_{\mathbb{P}}\Big[\sum_{t\in\mathcal{T}\setminus\{1\}} f_{at}(x_{at}(\xi_t); x_{-at}(\xi_t), p_t(\xi_t), \xi_t)\Big] \\
\text{s.t.} \quad & \mathcal{G}_{a1}(x_{a0}, x_{a1}; x_{-a1}, p_1, \xi_1) \in \mathcal{K}_{a1}, \\
& \mathcal{G}_{at}(x_{at-1}(\xi_{t-1}), x_{at}(\xi_t); x_{-at}(\xi_t), p_t(\xi_t), \xi_t) \in \mathcal{K}_{at}, \qquad \forall \xi_t \in \Xi_t, \forall t \in \mathcal{T}\setminus\{1\} \\
& x_{a1} \in \mathcal{X}_{a1}, \quad x_{at}(\xi_t) \in \mathcal{X}_{at}, \quad \forall \xi_t \in \Xi_t, \forall t \in \mathcal{T}\setminus\{1\}
\end{aligned}
\tag{6.26}
$$

with equilibrium constraint:

$$
\begin{aligned}
& 0 \in F_1(p_1; x_1, \xi_1) + \mathcal{N}_{K_1}(p_1), \\
& 0 \in F_t(p_t(\xi_t); x_t(\xi_t), \xi_t) + \mathcal{N}_{K_t}(p_t(\xi_t)), \quad \forall \xi_t \in \Xi_t, \forall t \in \mathcal{T}\setminus\{1\}
\end{aligned}
\tag{6.27}
$$

For concreteness we can now extend the previous multistage dynamic Walras barter model with activities into a multistage stochastic Walras barter model with activities. Involving the uncertainty $\xi = \{\xi_2, \xi_3, \ldots, \xi_{|\mathcal{T}|}\}$, the parameters $R_a^t(\xi_t), S_a^t(\xi_t), e_{at}(\xi_t)$ and the function $u_{at}(\cdot, \xi_t)$ will also be uncertain. We will use the notational equivalences defined above for $x, \mathcal{G}$ and F. The stochastic version of the problem will be

$$
\begin{aligned}
\min_{c_{a\cdot}, y_{a\cdot}} \quad & u_{a1}(c_{a1}) + \mathbb{E}_{\mathbb{P}}\Big[\sum_{t\in\mathcal{T}\setminus\{1\}} u_{at}(c_{at}(\xi_t), \xi_t)\Big] \\
\text{s.t.} \quad & p_1^T c_{a1} + p_1^T S_a^1 y_{a1} \le p_1^T e_{a1}, \\
& (p_t(\xi_t))^T c_{at}(\xi_t) - (p_t(\xi_t))^T R_a^t(\xi_t) y_{at-1}(\xi_{t-1}) \\
& + (p_t(\xi_t))^T S_a^t(\xi_t) y_{at}(\xi_t) \le (p_t(\xi_t))^T e_{at}(\xi_t), \qquad \forall \xi_t \in \Xi_t \\
& \qquad\qquad \forall t \in \mathcal{T}\setminus\{1, |\mathcal{T}|\} \\
& (p_t(\xi_t))^T c_{at}(\xi_t) - (p_t(\xi_t))^T R_a^t(\xi_t) y_{at-1}(\xi_{t-1}) \\
& \le (p_t(\xi_t))^T e_{at}(\xi_t), \qquad \forall \xi_t \in \Xi_t, t = |\mathcal{T}| \\
& c_{a1} \in \mathbb{R}_+^L, \quad y_{a1} \in \mathbb{R}_+^{m_a}, \quad c_{a|\mathcal{T}|}\big(\xi_{|\mathcal{T}|}\big) \in \mathbb{R}_+^L, \\
& c_{at}(\xi_t) \in \mathbb{R}_+^L, \quad y_{at}(\xi_t) \in \mathbb{R}_+^{m_a} \qquad \forall t \in \mathcal{T}\setminus\{1, |\mathcal{T}|\}
\end{aligned}
\tag{6.28}
$$

with a market constraint for each $t \in \mathcal{T}, \xi_t \in \Xi_t$:

$$\begin{aligned}
0 \leq & \sum_{a \in \mathcal{A}} (e_{a1} - c_{a1} - S_a^1 y_{a1}) && \perp p_1 \geq 0 \\
0 \leq & \sum_{a \in \mathcal{A}} \Big(e_{at}(\xi_t) + R_a^t(\xi_t) y_{at-1}(\xi_{t-1}) - c_{at}(\xi_t) - S_a^t(\xi_t) y_{at}(\xi_t) \Big) && \perp p_t(\xi_t) \geq 0, \quad \forall \xi_t \in \Xi_t, \forall t \in \mathcal{T} \backslash \{1, |\mathcal{T}|\} \\
0 \leq & \sum_{a \in \mathcal{A}} \Big(e_{at}(\xi_t) + R_a^t(\xi_t) y_{at-1}(\xi_{t-1}) - c_{at}(\xi_t) \Big) && \perp p_t(\xi_t) \geq 0, \quad \forall \xi_t \in \Xi_t, t = |\mathcal{T}|
\end{aligned} \tag{6.29}$$

6.3 One-level graph representation for stochastic process

The new graph representation framework can be expressed by the following Figure 6.2

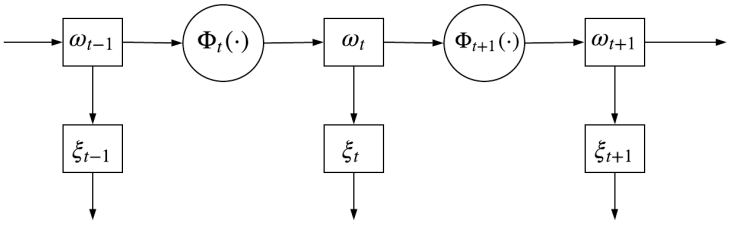


Figure 6.2: Stochastic process one-level graph representation

$$\begin{aligned}
\omega_t &= \zeta_{t|\Phi_t(\omega_{t-1})} \\
\xi_t &= \varphi_{t|\omega_t}
\end{aligned} \tag{6.30}$$

where the random variable $\xi_t : \Xi_t \to \mathbb{R}^n$ is the uncertainty that will come into the optimization problem, , $\omega_t : \Omega_t \to \mathbb{R}^m$ is the state uncertainty, $\zeta_{t|\omega_{t-1}}$ is a random vector that its distribution will depend on $\Phi_t(\omega_{t-1})$, $\varphi_{t|\omega_t}$ is a random vector whose distribution depending on ω_t, $\Phi_t(\cdot)$ is a mapping function that describes how random variable ω_t and ξ_t is dependent on the past state uncertainty ω_{t-1}.

In fact, our new graph representation and its corresponding mathematical representation is very similar to the well-known state-space equation in the control theory. Here we use the uncertainty ω_t to denote the state in our graph representation, which is linked by the line in the graph and has the dependence structure between stages. The uncertainty ξ_t will follow another distribution that depends on ω_t, which is the realization of the state uncertainty ω in stage t.

Among all the previous work, almost all of them just focus on the stagewise independent uncertainty, so what they need is just the basic information of the uncertainty ξ_t of every

stage t and the relationship between uncertainties in different stages have been overlooked. The policy graph structure introduced by [26] although doesn't ignore such important information, but its framework is still limited and can just represent a limited number of stochastic process, and for some type of stochastic process it will not have a very succinct way to represent, which will cause troubles for the user to input a problem.

After presenting the general graphical representation framework, we will introduce how several commonly used stochastic processes can be represented in our framework in the following section to show the breadth and applicability of our graph representation framework. These commonly used stochastic processes are:

- stagewise independent stochastic process
- time series
- discrete index markov chain stochastic process

6.3.1 Stagewise independent stochastic process

The stagewise independent stochastic process can be represented by (6.30) in the following format:

$$\begin{aligned} \omega_t &= \omega_{t-1} \\ \xi_t &= \varphi_t \end{aligned} \tag{6.31}$$

In this representation, we assume that ω_t always equals to ω_{t-1}, which means the state ω_t is always the same throughout the entire stage. So in this case, $\varphi_t = \varphi_{t|\omega_t}$ will have the same distribution in all t. Therefore $\xi_t = \varphi_t$ is stagewise independent across t.

6.3.2 Time Series

The general time series can be represented by (6.30) in the following format:

$$\begin{aligned} \omega_t &= \Phi_t(\omega_{t-1}) + \zeta_t \\ \xi_t &= \varphi_{t|\omega_t} \end{aligned} \tag{6.32}$$

In this representation, $\Phi_t(\cdot)$ is a function that will only use information from the past ω_{t-1}, ζ is a independent random variable for each time stage t. Here, we list several most commonly used time series and see how they can be represented in our framework. The commonly used time series are: (1) Autoregressive model(AR), (2) Moving-average

model(MA). (3) Autoregressive moving-average model(ARMA). (4) Autoregressive conditional heteroskedasticity model(ARCH) (5) Generalized autoregressive conditional heteroskedasticity (GARCH).

6.3.2.1 Different types of time series

The autoregressive model of order p can be represented by simplified notation AR(p), and its mathematical representation is:

$$\xi_t = \mu + \theta_1 \xi_{t-1} + \cdots + \theta_p \xi_{t-p} + \vartheta_t \tag{6.33}$$

here $\{\theta_i\}_{i=1,\ldots,p}, \mu$ are the constant parameters and ϑ_t is a stagewise independent random noise. In this case, we can assume

$$\omega_{t-1} = \begin{bmatrix} \xi_{t-p} \\ \vdots \\ \xi_{t-1} \end{bmatrix}, \quad \Phi_t(\omega_{t-1}) = A \cdot \omega_{t-1}, \quad \zeta_t = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mu + \vartheta_t \end{bmatrix}, \tag{6.34}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \ldots & 1 \\ \theta_p & \theta_{p-1} & \ldots & \ldots & \ldots & \theta_1 \end{bmatrix}$$

The moving-average model of order q can be represented by simplified notation MA(q), and its mathematical representation is:

$$\xi_t = \mu + \theta_1 \vartheta_{t-1} + \cdots + \theta_p \vartheta_{t-p} + \vartheta_t \tag{6.35}$$

here $\{\theta_i\}_{i=1,\ldots,p}, \mu$ are the constant parameters and ϑ_t is a stagewise independent random noise. In this case, we can assume

$$\omega_{t-1} = \begin{bmatrix} \vartheta_{t-p} \\ \vdots \\ \vartheta_{t-1} \end{bmatrix}, \quad \Phi_t(\omega_{t-1}) = A \cdot \omega_{t-1}, \quad \zeta_t = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mu + \vartheta_t \end{bmatrix}, \tag{6.36}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 \\ \theta_p & \theta_{p-1} & \dots & \dots & \dots & \theta_1 \end{bmatrix}$$

The autoregressive moving-average model of order p and q can be represented by simplified notation ARMA(p, q), and its mathematical representation is:

$$\xi_t = \mu + \sum_{i=1}^{p} \kappa_i \xi_{t-i} + \sum_{j=1}^{q} \theta_j \vartheta_{t-j} + \vartheta_t \tag{6.37}$$

Here $\{\theta_i\}_{i=1,\dots,p}$, $\{\kappa_i\}_{i=1,\dots,p}$, μ are the constant parameters and ϑ_t is a stagewise independent random noise. Then we will have

$$\omega_{t-1} = \begin{bmatrix} \vartheta_{t-q} \\ \vdots \\ \vartheta_{t-1} \\ \xi_{t-p} \\ \vdots \\ \xi_{t-1} \end{bmatrix}, \quad \Phi_t(\omega_{t-1}) = A \cdot \omega_{t-1}, \quad \zeta_t = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mu + \vartheta_t \end{bmatrix}, \tag{6.38}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 1 \\ \theta_q & \theta_{q-1} & \dots & \dots & \dots & \theta_1 & \kappa_p & \kappa_{p-1} & \kappa_{p-2} & \dots & \dots & \kappa_1 \end{bmatrix}$$

6.3.3 Markov Chain

The Markov chain stochastic process can be represented by (6.30) in the following format:

$$\begin{aligned} \omega_t &= \zeta_{t|\omega_{t-1}} \\ \xi_t &= \omega_t \end{aligned} \tag{6.39}$$

here the uncertainty $\zeta_{t|\omega_{t-1}}$ will be identical at every stage if the Markov chain is stationary.

6.3.4 Example

After introducing our new framework structure on several commonly used stochastic processes, in this section we will give an example to show how our framework is easy to use when we deal with a complicated stochastic process.

In our example, we will need to deal with a stochastic process where the uncertainty is the amount of daily rainfall $\{\xi_t\}_{t\in T}$. As we all know, the amount of daily rainfall will be related to the weather condition on that day. So it is natural not to use a stagewise independent stochastic process to model it. Previously, time series models like AR, MA, and ARMA are commonly used tools to model this process. And in our paper, we present a Markov chain-based model and illustrate how it can be represented succinctly and efficiently in our framework and how this structure fails to be represented in other frameworks.

In the example, the uncertainty $\{\xi_t\}_{t\in T}$ will be the amount of daily rainfall, and the state uncertainty $\{\omega_t\}_{t\in T}$ will be in one of three weather conditions: rainy, cloudy, sunny. And the relationship of the transition between weather condition is assumed to be described by a Markov chain matrix:

	sunny	cloudy	rainy
sunny	0.4	0.4	0.2
cloudy	0.3	0.2	0.5
rainy	0.1	0.4	0.5

Here the matrix element p_{ij} is the probability transferring from state i to state j. For example, the element $p_{12} = 0.4$ means that the probability transferring from sunny today to cloudy tomorrow is 0.4.

We assume that the amount of daily rainfall follows different distributions based on different weather conditions. Assume the rainfall will have five states of amount: 0, 100,

200, 400, 600, 1000. When the weather of the day is sunny, the φ_t follows the distribution:

$$\varphi_t = \begin{cases} 0 & \text{w.p. } 0.7 \\ 100 & \text{w.p. } 0.2 \\ 200 & \text{w.p. } 0.1 \end{cases}$$

When the weather of the day is cloudy, the φ_t follows the distribution:

$$\varphi_t = \begin{cases} 0 & \text{w.p. } 0.2 \\ 100 & \text{w.p. } 0.3 \\ 200 & \text{w.p. } 0.3 \\ 400 & \text{w.p. } 0.2 \end{cases}$$

When the weather of the day is rainy, the φ_t follows the distribution:

$$\varphi_t = \begin{cases} 200 & \text{w.p. } 0.2 \\ 400 & \text{w.p. } 0.2 \\ 600 & \text{w.p. } 0.4 \\ 1000 & \text{w.p. } 0.2 \end{cases}$$

This process can be represented by the following figure:

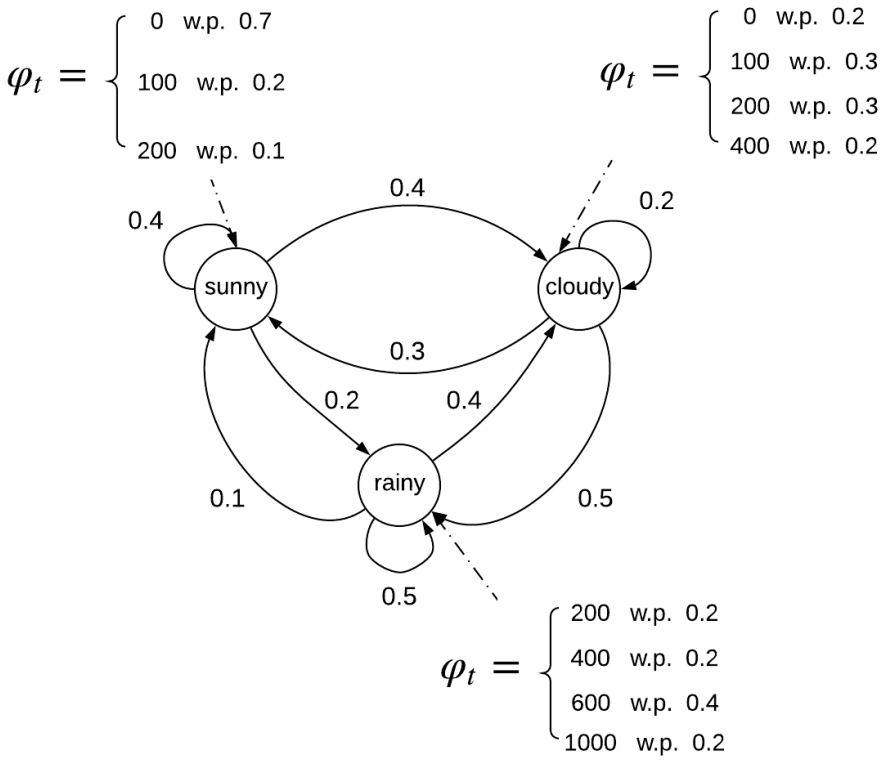


Figure 6.3: Stochastic process

To model this stochastic process in our framework, we define $\omega_t = 1$ when the weather is sunny, $\omega_t = 0$ when the weather is cloudy, and $\omega_t = -1$ when the weather is rainy. Let $\Phi_t(\cdot)$ be identity and the distribution of random variable $\zeta_{t|\omega_{t-1}}$ be defined by the following Markov chain matrix:

ω_{t-1} \ ζ_t	1	0	-1
1	0.4	0.4	0.2
0	0.3	0.2	0.5
-1	0.1	0.4	0.5

and let $\varphi_{t|\omega_t}$ be defined by the following probability transition matrix:

ω_t \ φ_t	0	100	200	400	600	1000
1	0.7	0.2	0.1	0	0	0
0	0.2	0.3	0.3	0.2	0	0
-1	0	0	0.2	0.2	0.4	0.2

Remark 6.2. For the above example, SMPS, GAMS EMPSP, and PySP fail to have an input format to load this kind of stochastic process directly. The Policy graph invented by [26] can have a way to load this stochastic process, but the problem is that it must have a data structure called node for each realization of the ω_t. In each node a stagewise subproblem needs to be defined, which will cause redundancy if ω_t has many realizations but has the same subproblems in each node. Our framework here will solve this kind of problem by separating the stochastic process and the optimization problem.

6.4 Two-level graph representation for stochastic programming and stochastic equilibrium problems

Based on our previous one-level graph representation for the stochastic process, in this section, we propose a framework for representing the stochastic programming problem, which is designed as a succinct way to input the optimization model into the solver. The two-level graph representation can be represented in the following figure:

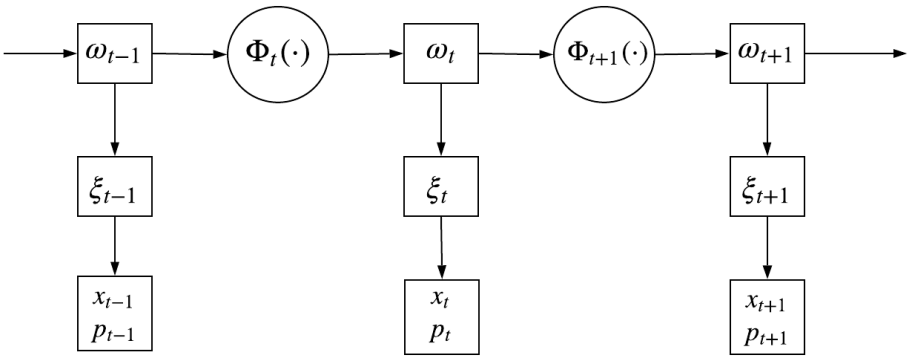


Figure 6.4: Two level graph representation for stochastic optimization/equilibrium problems

From the picture, we can see that our two-level representation consists of two main parts:

- an upper-level stochastic process
- a lower-level multistage optimization problem (or lower-level multistage equilibrium problem)

The upper-level is the stochastic process defined by our new one-level graph representation and new state transition equation representation. The lower part is a deterministic dynamic programming problem or a deterministic dynamic equilibrium problem.

6.4.1 Modeling two-level graph representation stochastic equilibrium problems using the existing EMP framework

Previously, how to specify stochastic optimization problems and equilibrium problems in the EMP framework has been outlined. In this section, we describe how to specify stochastic equilibrium problems in modeling languages using the EMP framework.

Based on the EMP framework of stochastic programming problems and equilibrium problems in the previous section, we have already known that: (1) In the equilibrium problems we need to assign each variable and each constraint into the players that it belongs to (2) In the stochastic programming problems we need to assign each variable into the time stage it belongs to. Then a very natural thought is that in the stochastic equilibrium problems, we just need to combine what we already have in the previous two works. The specification of a stochastic equilibrium problem needs to assign each variable into the corresponding player and stage, and to assign each constraint into the corresponding player. This specification process can be visualized by the following Figure 6.5.

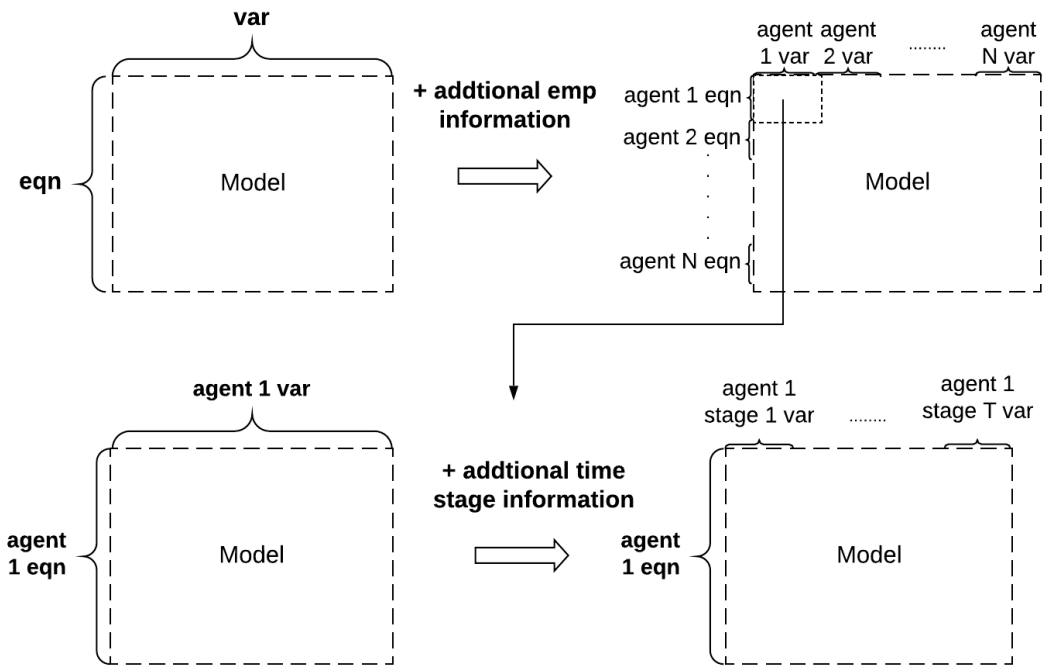


Figure 6.5: Discrete-time dynamic optimization model transformation

6.5 Sample average approximation method for multistage MOPEC

While the format described above is general, it is important to see how that might interact with the approaches outlined in the rest of this thesis. As such, we suggest a sample average approximation (SAA) approach.

6.5.1 Sample average approximation method for multistage MOPEC with risk-neutral players

We demonstrate our process for SAA using a forward tree sampling approach. To that end we iteratively define notation to specify a staged model for approximation.

If we denote by $Q_{a2}(\boldsymbol{x}_1, \boldsymbol{x}_2(\xi_2), p_2(\xi_2), \xi_2)$ the optimal value of player a of $(|\mathcal{T}|-1)$-stage MOPEC:

$$\begin{aligned} \min_{x_{a\cdot}} \quad & f_{a2}(x_{a2}(\xi_2); x_{-a2}(\xi_2), p_2(\xi_2), \xi_2) \\ & + \mathbb{E}_{\mathbb{P}}\Big[\sum_{t\in\mathcal{T}\setminus\{1,2\}} f_{at}(x_{at}(\xi_t); x_{-at}(\xi_t), p_t(\xi_t), \xi_t)\Big] \\ \text{s.t.} \quad & \mathcal{G}_{at}(x_{at-1}(\xi_{t-1}), x_{at}(\xi_t); x_{-at}(\xi_t), p_t(\xi_t), \xi_t) \in \mathcal{K}_{at}, \quad \forall \xi_t \in \Xi_t, \forall t \in \mathcal{T}\setminus\{1\} \\ & x_{at}(\xi_t) \in \mathcal{X}_{at}, \quad \forall \xi_t \in \Xi_t, \forall t \in \mathcal{T}\setminus\{1\} \end{aligned} \tag{6.40}$$

with equilibrium constraint:

$$0 \in F_t(p_t(\xi_t); x_t(\xi_t), \xi_t) + \mathcal{N}_{K_t}(p_t(\xi_t)), \quad \forall \xi_t \in \Xi_t, \forall t \in \mathcal{T}\setminus\{1\} \tag{6.41}$$

then we can write the $|\mathcal{T}|$-stage problem in the following form of a two-stage MOPEC:

$$\begin{aligned} \min_{x_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \mathbb{E}_{\mathbb{P}}\Big[Q_{a2}(\boldsymbol{x}_1, \boldsymbol{x}_2(\xi_2), p_2(\xi_2), \xi_2)\Big] \\ \text{s.t.} \quad & \mathcal{G}_{a1}(x_{a0}, x_{a1}; x_{-a1}, p_1, \xi_1) \in \mathcal{K}_{a1}, \\ & x_{a1} \in \mathcal{X}_{a1} \end{aligned} \tag{6.42}$$

with equilibrium constraint:

$$0 \in F_1(p_1; \boldsymbol{x}_1, \xi_1) + \mathcal{N}_{K_1}(p_1) \tag{6.43}$$

To facilitate the analysis, we restrict our derivations to the three-stage problem, where $\mathcal{T} = 1, 2, 3$ (it should be noted that the obtained results can be readily extended to scenarios

with $|\mathcal{T}| > 3$). In this particular case, the quantity $Q_{a2}(\boldsymbol{x}_1, \boldsymbol{x}_2(\xi_2), p_2(\xi_2), \xi_2)$ represents the optimal value attained by player a in the MOPEC:

$$\begin{aligned} \min_{x_{a\cdot}} \quad & f_{a2}(x_{a2}(\xi_2); x_{-a2}(\xi_2), p_2(\xi_2), \xi_2) \\ & + \mathbb{E}_{\mathbb{P}}\Big[Q_{a3}(\boldsymbol{x}_2(\xi_2), \boldsymbol{x}_3(\xi_3), p_3(\xi_3), \xi_3)|\xi_2\Big] \\ \text{s.t.} \quad & \mathcal{G}_{a2}(x_{a1}, x_{a2}(\xi_2); x_{-a2}(\xi_2), p_2(\xi_2), \xi_2) \in \mathcal{K}_{a2}, \\ & x_{a2}(\xi_2) \in \mathcal{X}_{a2} \end{aligned} \tag{6.44}$$

with equilibrium constraint:

$$0 \in F_2(p_2(\xi_2); x_2(\xi_2), \xi_2) + \mathcal{N}_{K_2}(p_2(\xi_2)) \tag{6.45}$$

where the expectation of each player a is taken with respect to the conditional distribution of ξ_3 given ξ_2.

Consider a set of random samples $\xi_2^i, i = 1, \ldots, N_2$, which are independent realizations of the random vector ξ_2. In order to approximate the optimization problem (6.42) for each player a, we can formulate the following Sample Average Approximation (SAA) problem:

$$\begin{aligned} \min_{x_{a\cdot}} \quad & f_{a1}(x_{a1}; x_{-a1}, p_1) + \frac{1}{N_2}\sum_{i=1}^{N_2} Q_{a2}(\boldsymbol{x}_1, \boldsymbol{x}_2(\xi_2^i), p_2(\xi_2^i), \xi_2^i) \\ \text{s.t.} \quad & \mathcal{G}_{a1}(x_{a0}, x_{a1}; x_{-a1}, p_1, \xi_1) \in \mathcal{K}_{a1}, \\ & x_{a1} \in \mathcal{X}_{a1} \end{aligned} \tag{6.46}$$

Given that the values $Q_{a2}(\boldsymbol{x}_1, \boldsymbol{x}_2(\xi_2^i), p_2(\xi_2^i), \xi_2^i)$ are not explicitly provided, it becomes necessary to estimate these quantities using a method known as *conditional sampling*. This involves generating a random sample $\xi_3^{ij}, j = 1, \ldots, N_{3i}$, consisting of N_{3i} independent realizations from the conditional distribution of ξ_3 given $\xi_2^i, i = 1, \ldots, N_2$. As a result, we can approximate $\hat{Q}_{a2}(\boldsymbol{x}_1, \boldsymbol{x}_2(\xi_2^i), p_2(\xi_2^i), \xi_2^i)$ by determining the optimal value attained by player a in the following approximate MOPEC:

$$\begin{aligned} \min_{x_{a\cdot}} \quad & f_{a2}(x_{a2}(\xi_2^i); x_{-a2}(\xi_2^i), p_2(\xi_2^i), \xi_2^i) \\ & + \frac{1}{N_3}\sum_{j=1}^{N_3} Q_{a3}(\boldsymbol{x}_2(\xi_2^i), \boldsymbol{x}_3(\xi_3^{ij}), p_3(\xi_3^{ij}), \xi_3^{ij}) \\ \text{s.t.} \quad & \mathcal{G}_{a2}(x_{a1}, x_{a2}(\xi_2^i); x_{-a2}(\xi_2^i), p_2(\xi_2^i), \xi_2^i) \in \mathcal{K}_{a2}, \\ & x_{a2}(\xi_2^i) \in \mathcal{X}_{a2} \end{aligned} \tag{6.47}$$

with equilibrium constraint:

$$0 \in F_2(p_2(\xi_2^i); x_2(\xi_2^i), \xi_2^i) + \mathcal{N}_{K_2}(p_2(\xi_2^i)) \tag{6.48}$$

where $Q_{a3}(\boldsymbol{x}_2(\xi_2^i), \boldsymbol{x}_3(\xi_3^{ij}), p_3(\xi_3^{ij}), \xi_3^{ij})$ is the optimal value of player a in the following MOPEC:

$$\begin{aligned} \min_{x_{a\cdot}} \quad & f_{a3}(x_{a3}(\xi_3^{ij}); x_{-a3}(\xi_3^{ij}), p_3(\xi_3^{ij}), \xi_3^{ij}) \\ \text{s.t.} \quad & \mathcal{G}_{a3}(x_{a2}(\xi_2^i), x_{a3}(\xi_3^{ij}); x_{-a3}(\xi_3^{ij}), p_3(\xi_3^{ij}), \xi_3^{ij}) \in \mathcal{K}_{a3}, \\ & x_{a3}(\xi_3^{ij}) \in \mathcal{X}_{a3} \end{aligned} \tag{6.49}$$

with equilibrium constraint:

$$0 \in F_3(p_3(\xi_3^{ij}); x_3(\xi_3^{ij}), \xi_3^{ij}) + \mathcal{N}_{K_3}(p_3(\xi_3^{ij})) \tag{6.50}$$

This process leads to an approximate scenario tree for the problem, specificed by the sampling structure (e.g., N_2, N_{3i}) and the realization data generated.

6.5.2 Sample average approximation method for the multistage MOPEC with risk-averse players employing the $\overline{CVaR}(\lambda, \varphi)$ measure

This section focuses on the utilization of the sample average approximation method for analyzing the multistage MOPEC with risk-averse players, utilizing the $\overline{CVaR}(\lambda, \varphi)$ measure. Specifically, the application of the sample average approximation method to the multistage MOPEC with risk-averse players employing the coherent risk measure can be divided into two primary tasks:

- Constructing the approximated scenario tree
- Building the corresponding risk set $\mathcal{D}_{an}$ for each player a and each scenario node $n \in \mathcal{N} \backslash \mathcal{L}$

The construction of the approximated scenario tree $(\mathcal{N}, \mathcal{E})$ can be achieved using a similar approach as described in section 6.5.1. Subsequently, the approximated risk set $\mathcal{D}_{an}$ can be constructed based on this approximated scenario tree. Assuming that each player $a \in \mathcal{A}$ and each scenario node $n \in \mathcal{N} \backslash \mathcal{L}$ adopts the $\overline{CVaR}(\lambda_{an}, \varphi_{an})$ risk profile, the corresponding dual representation can be established:

$$\overline{CVaR}(\lambda_{an}, \varphi_{an})(\cdot) = \sigma_{\mathcal{D}_{an}}(\cdot) = \sigma_{(1-\lambda_{an})\{\phi\}+\lambda_{an}\mathcal{D}(\varphi_{an})}(\cdot) \tag{6.51}$$

where $\phi_{n_+} = (\phi_m)_{m \in n_+}$ is the vector of children node conditional probability and

$$\mathcal{D}(\varphi_{an}) = \left\{ \mu \in \frac{1}{1-\varphi_{an}} \prod_{m \in n_+} [0, \phi_m] | \sum_{m \in n_+} \mu_m = 1 \right\} \tag{6.52}$$

Utilizing the aforementioned definition, the construction of the corresponding risk set $\mathcal{D}_{an}$ can be accomplished through a series of calculations involving ϕ_{n_+} as well as the parameters λ_{an} and φ_{an}.

With this in mind, three distinct types of input formats can be designed to accommodate the loading of the coherent risk measure:

- $\Big(CVaR, \lambda, \varphi\Big)$: For this input format it is assumed that all players at different scenario node n will employ the same coherent risk measure $\overline{CVaR}$, which means $\lambda_{an} = \lambda, \varphi_{an} = \varphi$ for each $a \in \mathcal{A}, n \in \mathcal{N} \backslash \mathcal{L}$.
- $\Big(CVaR, (\lambda_a, \varphi_a)_{a \in \mathcal{A}}\Big)$: For this input format it is assumed that each player a will employ the same coherent risk measure $\overline{CVaR}$, which means $\lambda_{an} = \lambda_a, \varphi_{an} = \varphi$ for each $a \in \mathcal{A}, n \in \mathcal{N} \backslash \mathcal{L}$.
- $\Big(CVaR, (\lambda_{an}, \varphi_{an})_{a \in \mathcal{A}, n \in \mathcal{N}}\Big)$: For this input format it is assumed that for each player a and each scenario node will have individual coherent risk measure $\overline{CVaR}$.

6.6 Conclusion

This chapter first introduced the one-level stochastic graph representation and two-level framework for stochastic programming problems and stochastic equilibrium problems. For the one-level stochastic graph representation, it's a new way to introduce the state-space equation from control theory into the representation of the stochastic process. Several commonly used stochastic processes were shown to be successfully represented by the new framework, and some of them were found hard to be represented by previous methods. This chapter also discussed the two-level graph representation for stochastic programming and stochastic equilibrium problems, and also presented a way to represent the multistage stochastic equilibrium problems efficiently in the EMP framework.

7 CONCLUSION

The stochastic MOPEC with risk-averse players has garnered increasing significance within the domains of game theory and equilibrium analysis. This prominence is primarily attributed to its extensive practical applications in various real-world scenarios, including power system markets and modern economics. The advancement in addressing this particular problem has consequently spurred advancements in policy frameworks and operational standards within the power industry and economics community. Drawing upon meticulous analyses, concrete illustrations, and comprehensive numerical findings, this doctoral thesis has made contributions by proposing several original ideas and solution methodologies for tackling stochastic MOPECs with risk-averse players.

In summary, this dissertation has the following main contributions: This thesis examines the formulation of stochastic MOPECs involving risk-averse players through the utilization of scenario trees, with an emphasis on exploiting the inherent problem structures. The research focuses on designing various decomposition algorithms based on different problem structures. Firstly, a player-based decomposition approach is explored, aiming to enhance the successful rate of stochastic PENP; however, its practical effectiveness is limited. Secondly, the Primal-MOPEC-dual-risk decomposition technique, in conjunction with PATH, demonstrates promising practical performance and proves capable of solving the majority of problems through several enhancements. Additionally, stage-based enhancement strategies are investigated to facilitate the creation of an inner loop, enabling the resolution of large-scale primal-MOPEC problems. Lastly, sample-based extension and large-scale implementation are considered to further extend the applicability of the proposed framework.

This dissertation also presents conclusive responses to the following inquiry.

- What is a standard mathematical definition of stochastic MOPEC with risk-averse players based on scenario tree and how to formulate this type of problem? Chapter 1 provides a overview of the stochastic MOPECs with risk-averse players, encompassing a comprehensive exposition of the fundamental mathematical constructs pertinent to this domain. The mathematical essence of the Nash equilibrium problem, variational inequality problem, stochastic problem utilizing a scenario tree, and the adoption of a coherent risk measure are expounded upon in detail. Additionally, diverse formulations of stochastic MOPECs with risk-averse players are outlined, thereby offering a comprehensive understanding of the various approaches adopted in addressing this class of problems.

- What are the representative numerical test instances for the stochastic MOPEC with risk-averse players? Within Chapter 2, three practical problem exemplifications, namely the economic dispatch example, capacity expansion example, and hydroelectricity example, are comprehensively presented as the most commonly employed scenarios. Each example is accompanied by an intricate algebraic formulation, meticulously delineating the mathematical representation of the respective problem instance. Furthermore, this chapter examines three distinct categories of market constraints, each holding paramount significance within the realm of modern economics and possessing substantial application domains. The inclusion of these three diverse problem types and the consideration of various market constraints collectively facilitate an extensive coverage of the prevailing and representative stochastic MOPECs with risk-averse players encountered in real-world problem.

- Can we solve the problem by the decomposition approach depending on the inner structure of the problem? How is the performance of these algorithms compared to the classical PATH solver? We tried to develop three decomposition approaches that depend on different properties of the problem.

  - Decomposition by players: Chapter 3 investigated the player-based inner stucture of the problem and mainly focused on the stochastic PNEP with risk-averse players using a *conjugate-based reformulation*, which will make each players' subproblem independent if the market price $\boldsymbol{p}$ is fixed. Based on this property, many existing player-based algorithms have been designed to solve this type of problem. However, most of them cannot be applied to stochastic MOPECs with risk-averse players. Chapter 3 first discussed two existing methods and then proposed an ADMM-based algorithm to solve the stochastic PNEP with risk-averse players, and showed its advantages over previous two methods. Chapter 3 also used numerical results to show various properties of the ADMM-based algorithm and how the performance of this algorithm will be changed by the algorithmic parameters.
  - Decomposition by primal-MOPEC-dual-risk structure: In Chapter 4, a novel primal-MOPEC-dual-risk algorithm was introduced, presenting an *equilibrium reformulation* as a means to solve the stochastic MOPEC with risk-averse players. This approach utilized a decomposition strategy, which is motivated by the inherent nonlinearity introduced by the additional probability vector $\boldsymbol{\mu}$ within the multistage problem. Consequently, when tackled by the PATH solver, the problem manifests as a highly nonlinear complementarity problem. However,

by fixing the probability vector $\boldsymbol{\mu}$, the problem undergoes simplification, and in the case of quadratic player objective functions, it can even be formulated as a linear complementarity problem. The remaining subproblem can be efficiently addressed through a series of linear programming problems. Compared to the ADMM-based algorithm discussed in Chapter 3, the proposed primal-MOPEC-dual-risk algorithm offers enhanced versatility in addressing general stochastic MOPECs with risk-averse players, particularly in scenarios where players exhibit diverse Nash behavior during their interactions. We are using all three examples with all three market constraints to test the efficiency of the algorithm and showed its advantage compared to the classical PATH solver.

  - Decomposition by stage structure: In Chapter 4 we proposed a primal-MOPEC-dual-risk decomposition algorithm. Nonetheless, a significant challenge emerges when confronted with large-sized scenario trees, resulting in a considerable increase in the dimensions of the primal-MOPEC subproblem. To overcome this challenge, a stage-based decomposition method is developed, enabling the subdivision of the extensive problem into smaller subproblems indexed by time stages. This division enhances computational efficiency and enables the effective resolution of large-scale stochastic MOPECs with risk-averse players. The subsequent Chapter 5 provides comprehensive details on the intricacies and implementation of this stage-based decomposition approach.

- What is the mathematical formulation of stochastic MOPECs with risk-averse players under general uncertainty process? Is there a simplied representation of stochastic problem such that the size of input of these problems could avoid exponential growth? Chapter 6 introduces how the multistage stochastic MOPEC with risk-averse player is formulated in mathematics and presents a new two-level graph representation for the general stochastic problem. Moreover, a sample average approximation (SAA) method is given to approximate any general stochastic MOPEC based a scenario tree, which then can be solved by the previous decomposition methods introduced.

8 APPENDIX

8.1 Numerical results of primal-MOPEC-dual-risk decomposition approach with proximal-terms on economic dispatch example

Tables 8.1 - 8.5 present the numerical outcomes obtained from Algorithm 6, along with a comparison to the original PD+PATH approach, concerning an economic dispatch example on scenario tree 2. The parameter configuration employed in this algorithm corresponds to that discussed in chapter 4. Within the algorithm, the proximal parameters ω_{an} are set to unity for all $a \in \mathcal{A}$ and $n \in \mathcal{N}$. Furthermore, a maximum iteration count is specified for the proximal steps.

The first two columns of the tables denote the values of the parameters ϵ and λ, respectively. For each fixed value of these parameters, a series of experiments is conducted using two distinct values of φ and 16 independent random seeds to generate random parameters. Consequently, each row represents a collection of 32 independent experiments.

The third to fifth columns of the tables provide the success rates of the PD+PATH and PD+PATH+proximal approaches, with a maximum of 20 and 50 iterations, respectively. Finally, the last two columns illustrate the enhancements observed in the algorithms following the inclusion of the proximal terms.

ϵ	λ	successful rate (%)			Improvement on PD+PATH (%)	
		PD+PATH	PD+PATH+proximal(20)	PD+PATH+proximal(50)	PD+PATH+proximal(20)	PD+PATH+proximal(50)
0	0.1	96.9	100.0	100.0	3.2	3.2
0	0.3	78.1	90.6	100.0	16.0	28.0
0	0.5	59.4	84.4	96.9	42.1	63.2
0	0.7	18.8	37.5	50.0	100.0	166.7
0	0.9	3.1	15.6	34.4	400.0	1000.0
1e-2	0.1	100.0	100.0	100.0	0.0	0.0
1e-2	0.3	100.0	100.0	100.0	0.0	0.0
1e-2	0.5	90.6	100.0	100.0	10.3	10.3
1e-2	0.7	71.9	93.8	100.0	30.4	39.1
1e-2	0.9	37.5	62.5	78.1	66.7	108.3
1e-1	0.1	100.0	100.0	100.0	0.0	0.0
1e-1	0.3	100.0	100.0	100.0	0.0	0.0
1e-1	0.5	100.0	100.0	100.0	0.0	0.0
1e-1	0.7	100.0	100.0	100.0	0.0	0.0
1e-1	0.9	100.0	100.0	100.0	0.0	0.0
1	0.1	100.0	100.0	100.0	0.0	0.0
1	0.3	100.0	100.0	100.0	0.0	0.0
1	0.5	100.0	100.0	100.0	0.0	0.0
1	0.7	100.0	100.0	100.0	0.0	0.0
1	0.9	100.0	100.0	100.0	0.0	0.0
10	0.1	100.0	100.0	100.0	0.0	0.0
10	0.3	100.0	100.0	100.0	0.0	0.0
10	0.5	100.0	100.0	100.0	0.0	0.0
10	0.7	100.0	100.0	100.0	0.0	0.0
10	0.9	100.0	100.0	100.0	0.0	0.0

Table 8.1: Improvement of PD+PATH with proximal terms over Type I problems on scenario tree 2

ϵ	λ	successful rate (%)			Improvement on PD+PATH (%)	
		PD+PATH	PD+PATH+proximal(20)	PD+PATH+proximal(50)	PD+PATH+proximal(20)	PD+PATH+proximal(50)
0	0.1	96.9	100.0	100.0	3.2	3.2
0	0.3	84.4	100.0	100.0	18.5	18.5
0	0.5	56.2	78.1	90.6	38.9	61.1
0	0.7	25.0	43.8	59.4	75.0	137.5
0	0.9	3.1	28.1	40.6	800.0	1200.0
1e-2	0.1	100.0	100.0	100.0	0.0	0.0
1e-2	0.3	100.0	100.0	100.0	0.0	0.0
1e-2	0.5	96.9	100.0	100.0	3.2	3.2
1e-2	0.7	84.4	96.9	96.9	14.8	14.8
1e-2	0.9	43.8	71.9	78.1	64.3	78.6
1e-1	0.1	100.0	100.0	100.0	0.0	0.0
1e-1	0.3	100.0	100.0	100.0	0.0	0.0
1e-1	0.5	100.0	100.0	100.0	0.0	0.0
1e-1	0.7	100.0	100.0	100.0	0.0	0.0
1e-1	0.9	100.0	100.0	100.0	0.0	0.0
1	0.1	100.0	100.0	100.0	0.0	0.0
1	0.3	100.0	100.0	100.0	0.0	0.0
1	0.5	100.0	100.0	100.0	0.0	0.0
1	0.7	100.0	100.0	100.0	0.0	0.0
1	0.9	100.0	100.0	100.0	0.0	0.0
10	0.1	100.0	100.0	100.0	0.0	0.0
10	0.3	100.0	100.0	100.0	0.0	0.0
10	0.5	100.0	100.0	100.0	0.0	0.0
10	0.7	100.0	100.0	100.0	0.0	0.0
10	0.9	100.0	100.0	100.0	0.0	0.0

Table 8.2: Improvement of PD+PATH with proximal terms over Type II problems on scenario tree 2

ϵ	λ	successful rate (%)			Improvement on PD+PATH (%)	
		PD+PATH	PD+PATH+proximal(20)	PD+PATH+proximal(50)	PD+PATH+proximal(20)	PD+PATH+proximal(50)
0	0.1	59.4	84.4	87.5	42.1	47.4
0	0.3	12.5	31.2	40.6	150.0	225.0
0	0.5	9.4	18.8	18.8	100.0	100.0
0	0.7	3.1	6.2	18.8	100.0	500.0
0	0.9	0.0	0.0	6.2	-	-
1e-2	0.1	100.0	100.0	100.0	0.0	0.0
1e-2	0.3	90.6	93.8	93.8	3.4	3.4
1e-2	0.5	40.6	59.4	75.0	46.2	84.6
1e-2	0.7	21.9	40.6	53.1	85.7	142.9
1e-2	0.9	6.2	12.5	21.9	100.0	250.0
1e-1	0.1	100.0	100.0	100.0	0.0	0.0
1e-1	0.3	100.0	100.0	100.0	0.0	0.0
1e-1	0.5	96.9	100.0	100.0	3.2	3.2
1e-1	0.7	100.0	100.0	100.0	0.0	0.0
1e-1	0.9	93.8	93.8	93.8	0.0	0.0
1	0.1	100.0	100.0	100.0	0.0	0.0
1	0.3	100.0	100.0	100.0	0.0	0.0
1	0.5	100.0	100.0	100.0	0.0	0.0
1	0.7	100.0	100.0	100.0	0.0	0.0
1	0.9	100.0	100.0	100.0	0.0	0.0
10	0.1	100.0	100.0	100.0	0.0	0.0
10	0.3	100.0	100.0	100.0	0.0	0.0
10	0.5	100.0	100.0	100.0	0.0	0.0
10	0.7	100.0	100.0	100.0	0.0	0.0
10	0.9	100.0	100.0	100.0	0.0	0.0

Table 8.3: Improvement of PD+PATH with proximal terms over Type I problems on scenario tree 3

ϵ	λ	successful rate (%)			Improvement on PD+PATH (%)	
		PD+PATH	PD+PATH+proximal(20)	PD+PATH+proximal(50)	PD+PATH+proximal(20)	PD+PATH+proximal(50)
0	0.1	96.9	96.9	96.9	0.0	0.0
0	0.3	40.6	62.5	71.9	53.8	76.9
0	0.5	12.5	18.8	28.1	50.0	125.0
0	0.7	3.1	6.2	3.1	100.0	0.0
0	0.9	0.0	0.0	6.2	-	-
1e-2	0.1	100.0	100.0	100.0	0.0	0.0
1e-2	0.3	96.9	100.0	100.0	3.2	3.2
1e-2	0.5	71.9	75.0	93.8	4.3	30.4
1e-2	0.7	40.6	46.9	59.4	15.4	46.2
1e-2	0.9	9.4	15.6	31.2	66.7	233.3
1e-1	0.1	100.0	100.0	100.0	0.0	0.0
1e-1	0.3	100.0	100.0	100.0	0.0	0.0
1e-1	0.5	100.0	100.0	100.0	0.0	0.0
1e-1	0.7	100.0	100.0	100.0	0.0	0.0
1e-1	0.9	93.8	100.0	96.9	6.7	3.3
1	0.1	100.0	100.0	100.0	0.0	0.0
1	0.3	100.0	100.0	100.0	0.0	0.0
1	0.5	100.0	100.0	100.0	0.0	0.0
1	0.7	100.0	100.0	100.0	0.0	0.0
1	0.9	100.0	100.0	100.0	0.0	0.0
10	0.1	100.0	100.0	100.0	0.0	0.0
10	0.3	100.0	100.0	100.0	0.0	0.0
10	0.5	100.0	100.0	100.0	0.0	0.0
10	0.7	100.0	100.0	100.0	0.0	0.0
10	0.9	100.0	100.0	100.0	0.0	0.0

Table 8.4: Improvement of PD+PATH with proximal terms over Type II problems on scenario tree 3

ϵ	λ	successful rate (%)			Improvement on PD+PATH (%)	
		PD+PATH	PD+PATH+proximal(20)	PD+PATH+proximal(50)	PD+PATH+proximal(20)	PD+PATH+proximal(50)
0	0.1	100.0	100.0	100.0	0.0	0.0
0	0.3	100.0	100.0	100.0	0.0	0.0
0	0.5	100.0	100.0	100.0	0.0	0.0
0	0.7	100.0	100.0	100.0	0.0	0.0
0	0.9	90.6	100.0	100.0	10.3	10.3
1e-2	0.1	100.0	100.0	100.0	0.0	0.0
1e-2	0.3	100.0	100.0	100.0	0.0	0.0
1e-2	0.5	100.0	100.0	100.0	0.0	0.0
1e-2	0.7	96.9	96.9	100.0	0.0	3.2
1e-2	0.9	96.9	100.0	100.0	3.2	3.2
1e-1	0.1	100.0	100.0	100.0	0.0	0.0
1e-1	0.3	100.0	100.0	100.0	0.0	0.0
1e-1	0.5	100.0	100.0	100.0	0.0	0.0
1e-1	0.7	100.0	100.0	100.0	0.0	0.0
1e-1	0.9	100.0	100.0	100.0	0.0	0.0
1	0.1	100.0	100.0	100.0	0.0	0.0
1	0.3	100.0	100.0	100.0	0.0	0.0
1	0.5	100.0	100.0	100.0	0.0	0.0
1	0.7	100.0	100.0	100.0	0.0	0.0
1	0.9	100.0	100.0	100.0	0.0	0.0
10	0.1	100.0	100.0	100.0	0.0	0.0
10	0.3	100.0	100.0	100.0	0.0	0.0
10	0.5	100.0	100.0	100.0	0.0	0.0
10	0.7	100.0	100.0	100.0	0.0	0.0
10	0.9	100.0	100.0	100.0	0.0	0.0

Table 8.5: Improvement of PD+PATH with proximal terms over Type III problems on scenario tree 3

8.2 Numerical results of primal-MOPEC-dual-risk decomposition approach on capacity expansion example and hydroelectricity example

Tables 8.6 - 8.19 present the numerical outcomes from capacity expansion problem and hydroelectricity problem on scenario 2 and scenario tree 3.

The first two columns of the tables denote the values of the parameters ϵ and λ, respectively. For each fixed value of these parameters, a series of experiments is conducted using two distinct values of φ and 16 independent random seeds to generate random parameters. Consequently, each row represents a collection of 32 independent experiments.

The third to fifth columns of the tables provide the success rates of the PD+PATH and PD+PATH+proximal approaches, with a maximum of 20 and 50 iterations, respectively. Finally, the last two columns illustrate the enhancements observed in the algorithms following the inclusion of the proximal terms.

ϵ	λ	successful rate (%)				No mixed solution percentage (%)
		PATH	PATH-RN	PD	PD-PATH	
0	0.1	100.0	100.0	100.0	100.0	100.0%
0	0.3	34.4	100.0	100.0	100.0	100.0%
0	0.5	3.1	90.6	100.0	100.0	100.0%
0	0.7	0.0	53.1	100.0	100.0	100.0%
0	0.9	0.0	6.2	96.9	100.0	100.0%
1e-2	0.1	100.0	100.0	100.0	100.0	100.0%
1e-2	0.3	90.6	100.0	100.0	100.0	100.0%
1e-2	0.5	6.2	100.0	100.0	100.0	100.0%
1e-2	0.7	9.4	96.9	100.0	100.0	100.0%
1e-2	0.9	0.0	65.6	100.0	100.0	100.0%
1e-1	0.1	100.0	100.0	100.0	100.0	100.0%
1e-1	0.3	100.0	100.0	100.0	100.0	100.0%
1e-1	0.5	78.1	100.0	100.0	100.0	100.0%
1e-1	0.7	15.6	100.0	93.8	100.0	93.8%
1e-1	0.9	15.6	71.9	100.0	100.0	100.0%
1	0.1	100.0	100.0	100.0	100.0	100.0%
1	0.3	100.0	100.0	100.0	100.0	100.0%
1	0.5	90.6	100.0	100.0	100.0	100.0%
1	0.7	56.2	65.6	90.6	100.0	90.6%
1	0.9	0.0	34.4	93.8	100.0	93.8%
10	0.1	100.0	96.9	100.0	100.0	100.0%
10	0.3	100.0	75.0	100.0	100.0	100.0%
10	0.5	100.0	53.1	96.9	100.0	96.9%
10	0.7	87.5	31.2	90.6	100.0	90.6%
10	0.9	21.9	9.4	93.8	100.0	93.8%

Table 8.6: Performance of PATH, PATH-RN, PD and PD-PATH over hydroelectricity example with *Type I* market constraint on scenario tree 2

ϵ	λ	successful rate (%)				No mixed solution percentage (%)
		PATH	PATH-RN	PD	PD-PATH	
0	0.1	100.0	100.0	100.0	100.0	100.0%
0	0.3	59.4	100.0	100.0	100.0	100.0%
0	0.5	3.1	87.5	100.0	100.0	100.0%
0	0.7	0.0	40.6	100.0	100.0	100.0%
0	0.9	0.0	0.0	100.0	100.0	100.0%
1e-2	0.1	100.0	100.0	100.0	100.0	100.0%
1e-2	0.3	96.9	100.0	100.0	100.0	100.0%
1e-2	0.5	21.9	100.0	100.0	100.0	100.0%
1e-2	0.7	3.1	100.0	100.0	100.0	100.0%
1e-2	0.9	0.0	56.2	100.0	100.0	100.0%
1e-1	0.1	100.0	100.0	100.0	100.0	100.0%
1e-1	0.3	100.0	100.0	96.9	100.0	96.9%
1e-1	0.5	87.5	100.0	100.0	100.0	100.0%
1e-1	0.7	31.2	96.9	100.0	100.0	100.0%
1e-1	0.9	9.4	71.9	100.0	100.0	100.0%
1	0.1	100.0	100.0	100.0	100.0	100.0%
1	0.3	100.0	100.0	93.8	100.0	93.8%
1	0.5	96.9	100.0	93.8	100.0	93.8%
1	0.7	84.4	96.9	96.9	100.0	96.9%
1	0.9	31.2	71.9	96.9	100.0	96.9%
10	0.1	100.0	100.0	96.9	100.0	96.9%
10	0.3	100.0	84.4	96.9	100.0	96.9%
10	0.5	100.0	53.1	93.8	100.0	93.8%
10	0.7	100.0	56.2	93.8	100.0	93.8%
10	0.9	93.8	43.8	87.5	100.0	87.5%

Table 8.7: Performance of PATH, PATH-RN, PD and PD-PATH over hydroelectricity example with *Type II* market constraint on scenario tree 2

ϵ	λ	successful rate (%)				No mixed solution percentage (%)
		PATH	PATH-RN	PD	PD-PATH	
0	0.1	100.0	100.0	81.2	100.0	81.2%
0	0.3	100.0	100.0	78.1	100.0	81.2%
0	0.5	100.0	100.0	56.2	100.0	62.5%
0	0.7	87.5	96.9	37.5	100.0	40.6%
0	0.9	28.1	87.5	28.1	96.9	34.4%
1e-2	0.1	100.0	100.0	90.6	100.0	90.6%
1e-2	0.3	100.0	100.0	81.2	100.0	81.2%
1e-2	0.5	100.0	100.0	65.6	100.0	65.6%
1e-2	0.7	81.2	100.0	50.0	100.0	50.0%
1e-2	0.9	25.0	87.5	31.2	100.0	31.2%
1e-1	0.1	100.0	100.0	84.4	100.0	84.4%
1e-1	0.3	100.0	100.0	75.0	100.0	75.0%
1e-1	0.5	100.0	100.0	71.9	100.0	71.9%
1e-1	0.7	93.8	100.0	68.8	100.0	68.8%
1e-1	0.9	43.8	90.6	68.8	100.0	68.8%
1	0.1	100.0	100.0	100.0	100.0	100.0%
1	0.3	100.0	100.0	100.0	100.0	100.0%
1	0.5	100.0	100.0	93.8	100.0	93.8%
1	0.7	100.0	100.0	93.8	100.0	93.8%
1	0.9	87.5	100.0	93.8	100.0	93.8%
10	0.1	100.0	100.0	100.0	100.0	100.0%
10	0.3	100.0	87.5	100.0	100.0	100.0%
10	0.5	100.0	62.5	100.0	100.0	100.0%
10	0.7	100.0	68.8	93.8	100.0	93.8%
10	0.9	100.0	50.0	90.6	100.0	90.6%

Table 8.8: Performance of PATH, PATH-RN, PD and PD-PATH over hydroelectricity example with *Type III* market constraint on scenario tree 2

ϵ	λ	successful rate (%)				No mixed solution percentage (%)
		PATH	PATH-RN	PD	PD-PATH	
0	0.1	96.9	90.6	100.0	100.0	100.0
0	0.3	53.1	90.6	100.0	100.0	100.0
0	0.5	6.2	28.1	100.0	100.0	100.0
0	0.7	0.0	0.0	100.0	100.0	100.0
0	0.9	0.0	0.0	90.6	96.9	100.0
1e-2	0.1	100.0	100.0	100.0	100.0	100.0
1e-2	0.3	100.0	100.0	100.0	100.0	100.0
1e-2	0.5	100.0	90.6	100.0	100.0	100.0
1e-2	0.7	100.0	96.9	100.0	100.0	100.0
1e-2	0.9	93.8	90.6	100.0	100.0	100.0
1e-1	0.1	100.0	100.0	100.0	100.0	100.0
1e-1	0.3	100.0	84.4	100.0	100.0	100.0
1e-1	0.5	81.2	84.4	100.0	100.0	100.0
1e-1	0.7	100.0	50.0	100.0	100.0	100.0
1e-1	0.9	75.0	46.9	100.0	100.0	100.0
1	0.1	40.6	0.0	100.0	100.0	100.0
1	0.3	43.8	0.0	100.0	100.0	100.0
1	0.5	34.4	0.0	96.9	100.0	100.0
1	0.7	34.4	0.0	96.9	100.0	100.0
1	0.9	12.5	0.0	100.0	100.0	100.0
10	0.1	0.0	0.0	87.5	96.9	100.0
10	0.3	0.0	0.0	84.4	100.0	100.0
10	0.5	0.0	0.0	78.1	96.9	100.0
10	0.7	0.0	0.0	75.0	100.0	100.0
10	0.9	0.0	0.0	78.1	96.9	100.0

Table 8.9: Performance of PATH, PATH-RN, PD and PD-PATH over capacity expansion example with *Type I* market constraint on scenario tree 2

ϵ	λ	successful rate (%)				No mixed solution percentage (%)
		PATH	PATH-RN	PD	PD-PATH	
0	0.1	96.9	100.0	100.0	100.0	100.0
0	0.3	59.4	93.8	93.8	100.0	100.0
0	0.5	28.1	43.8	100.0	100.0	100.0
0	0.7	3.1	9.4	90.6	100.0	100.0
0	0.9	0.0	0.0	96.9	100.0	100.0
1e-2	0.1	100.0	100.0	100.0	100.0	100.0
1e-2	0.3	100.0	100.0	100.0	100.0	100.0
1e-2	0.5	100.0	93.8	100.0	100.0	100.0
1e-2	0.7	100.0	93.8	100.0	100.0	100.0
1e-2	0.9	100.0	81.2	100.0	100.0	100.0
1e-1	0.1	100.0	100.0	100.0	100.0	100.0
1e-1	0.3	100.0	84.4	100.0	100.0	100.0
1e-1	0.5	100.0	84.4	100.0	100.0	100.0
1e-1	0.7	100.0	50.0	100.0	100.0	100.0
1e-1	0.9	100.0	50.0	100.0	100.0	100.0
1	0.1	100.0	3.1	100.0	100.0	100.0
1	0.3	96.9	0.0	100.0	100.0	100.0
1	0.5	78.1	0.0	96.9	100.0	100.0
1	0.7	90.6	0.0	96.9	100.0	100.0
1	0.9	87.5	0.0	100.0	100.0	100.0
10	0.1	100.0	0.0	87.5	96.9	100.0
10	0.3	87.5	0.0	84.4	100.0	100.0
10	0.5	65.6	0.0	78.1	96.9	100.0
10	0.7	50.0	0.0	75.0	100.0	100.0
10	0.9	3.1	0.0	78.1	100.0	100.0

Table 8.10: Performance of PATH, PATH-RN, PD and PD-PATH over capacity expansion example with *Type II* market constraint on scenario tree 2

ϵ	λ	successful rate (%)				No mixed solution percentage (%)
		PATH	PATH-RN	PD	PD-PATH	
0	0.1	100.0	100.0	96.9	100.0	100.0
0	0.3	100.0	100.0	87.5	100.0	100.0
0	0.5	100.0	100.0	93.8	100.0	100.0
0	0.7	96.9	84.4	96.9	100.0	100.0
0	0.9	62.5	50.0	96.9	100.0	100.0
1e-2	0.1	100.0	100.0	100.0	100.0	100.0
1e-2	0.3	100.0	46.9	100.0	100.0	100.0
1e-2	0.5	93.8	34.4	96.9	100.0	100.0
1e-2	0.7	62.5	18.8	100.0	100.0	100.0
1e-2	0.9	25.0	3.1	96.9	100.0	100.0
1e-1	0.1	78.1	28.1	100.0	100.0	100.0
1e-1	0.3	71.9	0.0	96.9	100.0	100.0
1e-1	0.5	43.8	0.0	90.6	100.0	100.0
1e-1	0.7	25.0	0.0	93.8	100.0	100.0
1e-1	0.9	9.4	0.0	93.8	100.0	100.0
1	0.1	62.5	0.0	100.0	100.0	100.0
1	0.3	53.1	0.0	100.0	100.0	100.0
1	0.5	62.5	0.0	100.0	100.0	100.0
1	0.7	43.8	0.0	100.0	100.0	100.0
1	0.9	31.2	0.0	100.0	100.0	100.0
10	0.1	15.6	0.0	100.0	100.0	100.0
10	0.3	3.1	0.0	96.9	100.0	100.0
10	0.5	18.8	0.0	100.0	100.0	100.0
10	0.7	62.5	0.0	100.0	100.0	100.0
10	0.9	50.0	0.0	100.0	100.0	100.0

Table 8.11: Performance of PATH, PATH-RN, PD and PD-PATH over capacity expansion example with *Type III* market constraint on scenario tree 2

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	100.0	100.0	81.2	100.0	100.0
0	0.3	100.0	100.0	78.1	100.0	100.0
0	0.5	100.0	100.0	56.2	100.0	100.0
0	0.7	87.5	96.9	37.5	100.0	100.0
0	0.9	28.1	87.5	28.1	96.9	100.0
1e-2	0.1	100.0	100.0	90.6	100.0	100.0
1e-2	0.3	100.0	100.0	81.2	100.0	100.0
1e-2	0.5	100.0	100.0	65.6	100.0	100.0
1e-2	0.7	81.2	100.0	50.0	100.0	100.0
1e-2	0.9	25.0	87.5	31.2	100.0	100.0
1e-1	0.1	100.0	100.0	84.4	100.0	100.0
1e-1	0.3	100.0	100.0	75.0	100.0	100.0
1e-1	0.5	100.0	100.0	71.9	100.0	100.0
1e-1	0.7	93.8	100.0	68.8	100.0	100.0
1e-1	0.9	43.8	90.6	68.8	100.0	100.0
1	0.1	100.0	100.0	100.0	100.0	100.0
1	0.3	100.0	100.0	100.0	100.0	100.0
1	0.5	100.0	100.0	93.8	100.0	100.0
1	0.7	100.0	100.0	93.8	100.0	100.0
1	0.9	87.5	100.0	93.8	100.0	100.0
10	0.1	100.0	100.0	100.0	100.0	100.0
10	0.3	100.0	87.5	100.0	100.0	100.0
10	0.5	100.0	62.5	100.0	100.0	100.0
10	0.7	100.0	68.8	93.8	100.0	100.0
10	0.9	100.0	50.0	90.6	100.0	100.0

Table 8.12: Performance of PATH, PATH-RN, PD, PD-PATH and PD-CC-PATH over hydro-electricity example with *Type III* market constraint on scenario tree 2

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	96.9	90.6	100.0	100.0	100.0
0	0.3	53.1	90.6	100.0	100.0	100.0
0	0.5	6.2	28.1	100.0	100.0	100.0
0	0.7	0.0	0.0	100.0	100.0	100.0
0	0.9	0.0	0.0	90.6	96.9	100.0
1e-2	0.1	100.0	100.0	100.0	100.0	100.0
1e-2	0.3	100.0	100.0	100.0	100.0	100.0
1e-2	0.5	100.0	90.6	100.0	100.0	100.0
1e-2	0.7	100.0	96.9	100.0	100.0	100.0
1e-2	0.9	93.8	90.6	100.0	100.0	100.0
1e-1	0.1	100.0	100.0	100.0	100.0	100.0
1e-1	0.3	100.0	84.4	100.0	100.0	100.0
1e-1	0.5	81.2	84.4	100.0	100.0	100.0
1e-1	0.7	100.0	50.0	100.0	100.0	100.0
1e-1	0.9	75.0	46.9	100.0	100.0	100.0
1	0.1	40.6	0.0	100.0	100.0	100.0
1	0.3	43.8	0.0	100.0	100.0	100.0
1	0.5	34.4	0.0	96.9	100.0	100.0
1	0.7	34.4	0.0	96.9	100.0	100.0
1	0.9	12.5	0.0	100.0	100.0	100.0
10	0.1	0.0	0.0	87.5	96.9	100.0
10	0.3	0.0	0.0	84.4	100.0	100.0
10	0.5	0.0	0.0	78.1	96.9	100.0
10	0.7	0.0	0.0	75.0	100.0	100.0
10	0.9	0.0	0.0	78.1	96.9	100.0

Table 8.13: Performance of PATH, PATH-RN, PD, PD-PATH and PD-CC-PATH over capacity expansion example with *Type I* market constraint on scenario tree 2

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	6.2	100.0	96.9	100.0	100.0
0	0.3	0.0	71.9	96.9	100.0	100.0
0	0.5	0.0	9.4	93.8	93.8	100.0
0	0.7	0.0	0.0	93.8	100.0	100.0
0	0.9	0.0	0.0	34.4	56.2	100.0
1e-2	0.1	71.9	100.0	100.0	100.0	100.0
1e-2	0.3	18.8	100.0	100.0	100.0	100.0
1e-2	0.5	0.0	59.4	100.0	100.0	100.0
1e-2	0.7	0.0	12.5	100.0	100.0	100.0
1e-2	0.9	0.0	0.0	96.9	100.0	100.0
1e-1	0.1	100.0	100.0	100.0	100.0	100.0
1e-1	0.3	34.4	100.0	100.0	100.0	100.0
1e-1	0.5	12.5	90.6	100.0	100.0	100.0
1e-1	0.7	0.0	62.5	96.9	100.0	100.0
1e-1	0.9	0.0	0.0	96.9	100.0	100.0
1	0.1	100.0	100.0	96.9	100.0	100.0
1	0.3	93.8	100.0	96.9	100.0	100.0
1	0.5	71.9	84.4	96.9	100.0	100.0
1	0.7	0.0	40.6	96.9	100.0	100.0
1	0.9	0.0	0.0	87.5	100.0	100.0
10	0.1	100.0	100.0	96.9	100.0	100.0
10	0.3	100.0	81.2	96.9	100.0	100.0
10	0.5	81.2	75.0	96.9	100.0	100.0
10	0.7	56.2	43.8	93.8	100.0	100.0
10	0.9	0.0	0.0	90.6	100.0	100.0

Table 8.14: Performance of PATH and Primal-dual+PATH over Hydroeletricity example with *Type I* market constraint on scenario tree 3

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	15.6	100.0	96.9	100.0	100.0
0	0.3	0.0	65.6	100.0	100.0	100.0
0	0.5	0.0	9.4	100.0	100.0	100.0
0	0.7	0.0	3.1	96.9	96.9	100.0
0	0.9	0.0	0.0	78.1	90.6	100.0
1e-2	0.1	78.1	100.0	100.0	100.0	100.0
1e-2	0.3	12.5	100.0	100.0	100.0	100.0
1e-2	0.5	0.0	62.5	96.9	100.0	100.0
1e-2	0.7	0.0	15.6	96.9	100.0	100.0
1e-2	0.9	0.0	0.0	96.9	100.0	100.0
1e-1	0.1	100.0	100.0	100.0	100.0	100.0
1e-1	0.3	59.4	100.0	96.9	100.0	100.0
1e-1	0.5	31.2	93.8	90.6	100.0	100.0
1e-1	0.7	9.4	75.0	93.8	100.0	100.0
1e-1	0.9	0.0	40.6	93.8	100.0	100.0
1	0.1	100.0	100.0	100.0	100.0	100.0
1	0.3	100.0	100.0	100.0	100.0	100.0
1	0.5	87.5	100.0	100.0	100.0	100.0
1	0.7	46.9	90.6	96.9	100.0	100.0
1	0.9	3.1	56.2	87.5	100.0	100.0
10	0.1	100.0	100.0	100.0	100.0	100.0
10	0.3	100.0	84.4	100.0	100.0	100.0
10	0.5	100.0	75.0	96.9	100.0	100.0
10	0.7	93.8	65.6	93.8	100.0	100.0
10	0.9	59.4	37.5	93.8	100.0	100.0

Table 8.15: Performance of PATH and Primal-dual+PATH over Hydroeletricity example with *Type II* market constraint on scenario tree 3

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	100.0	100.0	81.2	100.0	100.0
0	0.3	100.0	100.0	59.4	100.0	100.0
0	0.5	81.2	100.0	50.0	100.0	100.0
0	0.7	12.5	96.9	25.0	100.0	100.0
0	0.9	0.0	65.6	9.4	100.0	100.0
1e-2	0.1	100.0	100.0	78.1	100.0	100.0
1e-2	0.3	100.0	100.0	65.6	100.0	100.0
1e-2	0.5	78.1	100.0	40.6	100.0	100.0
1e-2	0.7	59.4	96.9	25.0	100.0	100.0
1e-2	0.9	6.2	68.8	9.4	100.0	100.0
1e-1	0.1	96.9	100.0	84.4	96.9	100.0
1e-1	0.3	100.0	100.0	43.8	100.0	100.0
1e-1	0.5	90.6	100.0	37.5	100.0	100.0
1e-1	0.7	71.9	96.9	15.6	100.0	100.0
1e-1	0.9	18.8	78.1	18.8	96.9	100.0
1	0.1	100.0	100.0	100.0	100.0	100.0
1	0.3	100.0	100.0	93.8	100.0	100.0
1	0.5	100.0	100.0	87.5	100.0	100.0
1	0.7	93.8	100.0	90.6	100.0	100.0
1	0.9	43.8	78.1	93.8	100.0	100.0
10	0.1	100.0	100.0	100.0	100.0	100.0
10	0.3	100.0	84.4	96.9	100.0	100.0
10	0.5	100.0	81.2	100.0	100.0	100.0
10	0.7	100.0	75.0	93.8	100.0	100.0
10	0.9	93.8	65.6	93.8	100.0	100.0

Table 8.16: Performance of PATH and Primal-dual+PATH over Hydroeletricity example with *Type III* market constraint on scenario tree 3

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	18.8	75.0	68.8	68.8	100.0
0	0.3	0.0	3.1	68.8	68.8	100.0
0	0.5	0.0	0.0	65.6	65.6	100.0
0	0.7	0.0	0.0	53.1	56.2	96.9
0	0.9	0.0	0.0	46.9	56.2	90.6
1e-2	0.1	31.2	46.9	25.0	37.5	100.0
1e-2	0.3	0.0	6.2	21.9	25.0	100.0
1e-2	0.5	0.0	0.0	21.9	25.0	100.0
1e-2	0.7	0.0	0.0	18.8	25.0	100.0
1e-2	0.9	0.0	0.0	12.5	21.9	100.0
1e-1	0.1	56.2	84.4	43.8	75.0	100.0
1e-1	0.3	0.0	21.9	43.8	43.8	100.0
1e-1	0.5	0.0	0.0	37.5	43.8	100.0
1e-1	0.7	0.0	0.0	40.6	43.8	100.0
1e-1	0.9	0.0	0.0	31.2	43.8	100.0
1	0.1	93.8	100.0	59.4	84.4	100.0
1	0.3	50.0	84.4	62.5	81.2	100.0
1	0.5	0.0	46.9	59.4	62.5	100.0
1	0.7	0.0	0.0	56.2	62.5	100.0
1	0.9	0.0	0.0	59.4	62.5	100.0
10	0.1	100.0	100.0	50.0	100.0	100.0
10	0.3	100.0	62.5	50.0	96.9	100.0
10	0.5	96.9	87.5	50.0	100.0	100.0
10	0.7	37.5	12.5	46.9	59.4	100.0
10	0.9	0.0	0.0	46.9	50.0	100.0

Table 8.17: Performance of PATH and Primal-dual+PATH over capacity expansion example with *Type I* market constraint on scenario tree 3

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	62.5	81.2	84.4	100.0	100.0
0	0.3	6.2	0.0	81.2	96.9	100.0
0	0.5	0.0	0.0	78.1	93.8	100.0
0	0.7	0.0	0.0	53.1	71.9	100.0
0	0.9	0.0	0.0	59.4	75.0	96.9
1e-2	0.1	84.4	100.0	90.6	100.0	100.0
1e-2	0.3	6.2	3.1	81.2	100.0	100.0
1e-2	0.5	0.0	0.0	71.9	90.6	100.0
1e-2	0.7	0.0	0.0	65.6	78.1	100.0
1e-2	0.9	0.0	0.0	56.2	84.4	100.0
1e-1	0.1	100.0	100.0	90.6	100.0	100.0
1e-1	0.3	43.8	40.6	81.2	100.0	100.0
1e-1	0.5	9.4	6.2	81.2	100.0	100.0
1e-1	0.7	0.0	0.0	78.1	100.0	100.0
1e-1	0.9	0.0	0.0	78.1	93.8	100.0
1	0.1	100.0	100.0	90.6	100.0	100.0
1	0.3	100.0	96.9	93.8	100.0	100.0
1	0.5	40.6	18.8	93.8	100.0	100.0
1	0.7	3.1	0.0	93.8	100.0	100.0
1	0.9	0.0	0.0	71.9	96.9	100.0
10	0.1	100.0	100.0	100.0	100.0	100.0
10	0.3	100.0	100.0	100.0	100.0	100.0
10	0.5	100.0	65.6	100.0	100.0	100.0
10	0.7	65.6	0.0	96.9	100.0	100.0
10	0.9	3.1	0.0	93.8	100.0	100.0

Table 8.18: Performance of PATH and Primal-dual+PATH over Capacity example with *Type II* market constraint on scenario tree 3

ϵ	λ	successful rate (%)				
		PATH	PATH-RN	PD	PD-PATH	PD-CC-PATH
0	0.1	100.0	100.0	90.6	100.0	100.0
0	0.3	100.0	100.0	71.9	100.0	100.0
0	0.5	75.0	71.9	56.2	100.0	100.0
0	0.7	18.8	9.4	68.8	100.0	100.0
0	0.9	0.0	0.0	90.6	100.0	100.0
1e-2	0.1	100.0	100.0	96.9	100.0	100.0
1e-2	0.3	100.0	100.0	84.4	100.0	100.0
1e-2	0.5	65.6	59.4	68.8	100.0	100.0
1e-2	0.7	15.6	6.2	65.6	100.0	100.0
1e-2	0.9	0.0	0.0	53.1	100.0	100.0
1e-1	0.1	100.0	100.0	84.4	100.0	100.0
1e-1	0.3	96.9	96.9	75.0	100.0	100.0
1e-1	0.5	59.4	68.8	53.1	100.0	100.0
1e-1	0.7	6.2	6.2	56.2	96.9	100.0
1e-1	0.9	0.0	0.0	78.1	100.0	100.0
1	0.1	100.0	100.0	100.0	100.0	100.0
1	0.3	100.0	100.0	93.8	100.0	100.0
1	0.5	90.6	96.9	78.1	100.0	100.0
1	0.7	40.6	18.8	59.4	100.0	100.0
1	0.9	3.1	0.0	81.2	100.0	100.0
10	0.1	100.0	100.0	100.0	100.0	100.0
10	0.3	100.0	100.0	100.0	100.0	100.0
10	0.5	100.0	75.0	100.0	100.0	100.0
10	0.7	50.0	3.1	100.0	100.0	100.0
10	0.9	12.5	0.0	90.6	100.0	100.0

Table 8.19: Performance of PATH and Primal-dual+PATH over Capacity example with *Type III* market constraint on scenario tree 3

REFERENCES

[1] Abada, Ibrahim, Steven Gabriel, Vincent Briat, and Olivier Massol. 2013. A generalized nash–cournot model for the northwestern european natural gas markets with a fuel substitution demand function: The gammes model. *Networks and Spatial Economics* 13: 1–42.

[2] Afifah, Fatima, and Zhaomiao Guo. 2022. Spatial pricing of ride-sourcing services in a congested transportation network. *Transportation Research Part C: Emerging Technologies* 142:103777.

[3] Agdeppa, Rhoda P, Nobuo Yamashita, and Masao Fukushima. 2007. The traffic equilibrium problem with nonadditive costs and its monotone mixed complementarity problem formulation. *Transportation Research Part B: Methodological* 41(8):862–874.

[4] Arrow, Kenneth J, and Gerard Debreu. 1954. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society* 265–290.

[5] Bagloee, Saeed Asadi, Majid Sarvi, Michael Patriksson, and Abbas Rajabifard. 2017. A mixed user-equilibrium and system-optimal traffic flow for connected vehicles stated as a complementarity problem. *Computer-Aided Civil and Infrastructure Engineering* 32(7):562–580.

[6] Barquín, Julián, and Miguel Vázquez. 2005. Cournot equilibrium in power networks. *Instituto de Investigacin Tecnolgica, Universidad Pontificia Comillas, Madrid*.

[7] ———. 2008. Cournot equilibrium calculation in power networks: An optimization approach with price response computation. *IEEE transactions on power systems* 23(2): 317–326.

[8] Belknap, Margaret H, Chun-Hung Chen, and Patrick T Harker. 2000. A gradient-based method for analyzing stochastic variational inequalities with one uncertain parameter. In *Opim working paper 00–03–13*. Department of Operations and Information Management, Wharton School.

[9] Billups, Stephen C. 1995. Algorithms for complementarity problems and generalized equations. Tech. Rep.

[10] Billups, Stephen C, and Michael C Ferris. 1997. Qpcomp: A quadratic programming based solver for mixed complementarity problems. *Mathematical Programming* 76(3): 533–562.

[11] Böhringer, Christoph, and Thomos F Rutherford. 2009. Integrated assessment of energy policies: Decomposing top-down and bottom-up. *Journal of Economic Dynamics and Control* 33(9):1648–1661.

[12] Borges, Pedro, Claudia Sagastizábal, and Mikhail Solodov. Decomposition algorithms for two-stage stochastic hierarchical optimization.

[13] Bushnell, James. 2003. A mixed complementarity model of hydrothermal electricity competition in the western united states. *Operations research* 51(1):80–93.

[14] Campos, FA, J Villar, and J Barquín. 2010. Solving cournot equilibriums with variational inequalities algorithms. *IET generation, transmission & distribution* 4(2):268–280.

[15] Cavazzuti, Ennio, Massimo Pappalardo, Mauro Passacantando, et al. 2002. Nash equilibria, variational inequalities, and dynamical systems. *Journal of optimization theory and applications* 114(3):491–506.

[16] Censor, Yair, Alfredo N Iusem, and Stavros A Zenios. 1998. An interior point method with bregman functions for the variational inequality problem with paramonotone operators. *Mathematical Programming* 81(3):373–400.

[17] Chen, Chunhui, and Olvi L Mangasarian. 1996. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications* 5(2):97–138.

[18] Chen, Xi, and Xiaotie Deng. 2006. Settling the complexity of two-player nash equilibrium. In *2006 47th annual ieee symposium on foundations of computer science (focs'06)*, 261–272. IEEE.

[19] Cournot, Antoine Augustin. 1838. *Recherches sur les principes mathématiques de la théorie des richesses*.

[20] Daskalakis, Constantinos, Paul W Goldberg, and Christos H Papadimitriou. 2009. The complexity of computing a nash equilibrium. *SIAM Journal on Computing* 39(1): 195–259.

[21] De Wolf, Daniel, and Yves Smeers. 1997. A stochastic version of a stackelberg-nash-cournot equilibrium model. *Management Science* 43(2):190–197.

[22] Debreu, Gerard. 1952. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences* 38(10):886–893.

[23] Deride, Julio, Alejandro Jofré, and Roger JB Wets. 2019. Solving deterministic and stochastic equilibrium problems via augmented walrasian. *Computational Economics* 53:315–342.

[24] Devine, Mel T, Steven A Gabriel, and Seksun Moryadee. 2016. A rolling horizon approach for stochastic mixed complementarity problems with endogenous learning: Application to natural gas markets. *Computers & Operations Research* 68:1–15.

[25] Dirkse, Steven P, and Michael C Ferris. 1995. The path solver: a nommonotone stabilization scheme for mixed complementarity problems. *Optimization methods and software* 5(2):123–156.

[26] Dowson, Oscar. 2018. The policy graph decomposition of multistage stochastic optimization problems. *Optimization Online*.

[27] Dreves, Axel, and Christian Kanzow. 2011. Nonsmooth optimization reformulations characterizing all solutions of jointly convex generalized nash equilibrium problems. *Computational Optimization and Applications* 50(1):23–48.

[28] Egging, Rudolf G, and Steven A Gabriel. 2006. Examining market power in the european natural gas market. *Energy policy* 34(17):2762–2778.

[29] Egging, Ruud. 2013. Benders decomposition for multi-stage stochastic mixed complementarity problems–applied to a global natural gas market model. *European Journal of Operational Research* 226(2):341–353.

[30] Egging, Ruud, Steven A Gabriel, Franziska Holz, and Jifang Zhuang. 2008. A complementarity model for the european natural gas market. *Energy policy* 36(7):2385–2414.

[31] Egging, Ruud, Franziska Holz, and Steven A Gabriel. 2010. The world gas model: A multi-period mixed complementarity model for the global natural gas market. *Energy* 35(10):4016–4029.

[32] Facchinei, Francisco, Andreas Fischer, and Veronica Piccialli. 2009. Generalized nash equilibrium problems and newton methods. *Mathematical Programming* 117(1-2): 163–194.

[33] Facchinei, Francisco, Christian Kanzow, and Simone Sagratella. 2014. Solving quasi-variational inequalities via their kkt conditions. *Mathematical Programming* 144(1-2): 369–412.

[34] Facchinei, Francisco, and Jong-Shi Pang. 2006. Exact penalty functions for generalized nash problems. In *Large-scale nonlinear optimization*, 115–126. Springer.

[35] ———. 2007. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media.

[36] ———. 2010. nash equilibria: the variational approach. *Convex optimization in signal processing and communications* 443.

[37] Farrell, Joseph. 1988. Communication, coordination and nash equilibrium. *Economics Letters* 27(3):209–214.

[38] Ferris, Michael, and Andy Philpott. 2022. Dynamic risked equilibrium. *Operations Research* 70(3):1933–1952.

[39] Ferris, Michael C, and Jong-Shi Pang. 1997. Engineering and economic applications of complementarity problems. *Siam Review* 39(4):669–713.

[40] Ferris, Michael C, and Andy Philpott. 2022. Renewable electricity capacity planning with uncertainty at multiple scales.

[41] Fischer, Andreas. 1992. A special newton-type optimization method. *Optimization* 24(3-4):269–284.

[42] Flåm, Sjur Didrik. 1993. Paths to constrained nash equilibria. *Applied Mathematics and Optimization* 27(3):275–289.

[43] Gabriel, Steven, and Yves Smeers. 2006. *Complementarity problems in restructured natural gas markets*. Springer.

[44] Gabriel, Steven A, and JD Fuller. 2010. A benders decomposition method for solving stochastic complementarity problems with an application in energy. *Computational economics* 35(4):301–329.

[45] Gabriel, Steven A, Supat Kiet, and Jifang Zhuang. 2005. A mixed complementarity-based equilibrium model of natural gas markets. *Operations Research* 53(5):799–818.

[46] Gabriel, Steven A, Jifang Zhuang, and Ruud Egging. 2009. Solving stochastic complementarity problems in energy market modeling using scenario reduction. *European Journal of Operational Research* 197(3):1028–1040.

[47] Gabriel, Steven A, Jifang Zhuang, and Supat Kiet. 2005. A large-scale linear complementarity model of the north american natural gas market. *Energy economics* 27(4): 639–665.

[48] Grippo, Luigi, and Marco Sciandrone. 2007. Nonmonotone derivative-free methods for nonlinear equations. *Computational Optimization and Applications* 37(3):297–328.

[49] Grübel, Julia, Thomas Kleinert, Vanessa Krebs, Galina Orlinskaya, Lars Schewe, Martin Schmidt, and Johannes Thürauf. 2019. On electricity market equilibria with storages: Modeling, uniqueness, and a distributed admm.

[50] Gürkan, Gül, and Jong-Shi Pang. 2009. Approximations of nash equilibria. *Mathematical Programming* 117(1-2):223–253.

[51] Gürkan, Gül, A Yonca Özge, and Stephen M Robinson. 1999. Sample-path solution of stochastic variational inequalities. *Mathematical Programming* 84(2):313–333.

[52] Harker, Patrick T. 1991. Generalized nash games and quasi-variational inequalities. *European journal of Operational research* 54(1):81–94.

[53] Haurie, Alain, and Québec). Groupe d'études et de recherche en analyse des décisions École des hautes études commerciales (Montréal. 1987. *A stochastic dynamic nash-cournot model for the european gas market*. Montréal: École des hautes études commerciales.

[54] von Heusinger, Anna, and Christian Kanzow. 2009. Optimization reformulations of the generalized nash equilibrium problem using nikaido-isoda-type functions. *Computational Optimization and Applications* 43(3):353–377.

[55] Hildebrand, Francis Begnaud. 1987. *Introduction to numerical analysis*. Courier Corporation.

[56] Huber, Olivier, and Michael C Ferris. 2023. Reformulations for convex composite functions and their nested compositions.

[57] Kalai, Ehud, and Ehud Lehrer. 1993. Rational learning leads to nash equilibrium. *Econometrica: Journal of the Econometric Society* 1019–1045.

[58] Kanzow, Christian, and Stefania Petra. 2004. On a semismooth least squares formulation of complementarity problems with gap reduction. *Optimization Methods and Software* 19(5):507–525.

[59] Liu, Nian, Xinghuo Yu, Cheng Wang, and Jinjian Wang. 2017. Energy sharing management for microgrids with pv prosumers: A stackelberg game approach. *IEEE Transactions on Industrial Informatics* 13(3):1088–1098.

[60] Luna, Juan Pablo, Claudia Sagastizábal, and Mikhail Solodov. 2016. An approximation scheme for a class of risk-averse stochastic equilibrium problems. *Mathematical Programming* 157(2):451–481.

[61] Ma, Jie, Min Xu, Qiang Meng, and Lin Cheng. 2020. Ridesharing user equilibrium problem under od-based surge pricing strategy. *Transportation Research Part B: Methodological* 134:1–24.

[62] Maskin, Eric. 1999. Nash equilibrium and welfare optimality. *The Review of Economic Studies* 66(1):23–38.

[63] Mays, Jacob, David Morton, and Richard P O'Neill. 2019. Asymmetric risk and fuel neutrality in capacity markets.

[64] Molina, Juan Pablo, Juan Manuel Zolezzi, Javier Contreras, Hugh Rudnick, and María José Reveco. 2010. Nash-cournot equilibria in hydrothermal electricity markets. *IEEE Transactions on Power Systems* 26(3):1089–1101.

[65] ———. 2011. Nash-cournot equilibria in hydrothermal electricity markets. *IEEE Transactions on Power Systems* 26(3):1089–1101.

[66] Munson, Todd S, Francisco Facchinei, Michael C Ferris, Andreas Fischer, and Christian Kanzow. 2001. The semismooth algorithm for large scale complementarity problems. *INFORMS Journal on Computing* 13(4):294–311.

[67] Nash, John. 1951. Non-cooperative games. *Annals of mathematics* 286–295.

[68] Nash, John F, et al. 1950. Equilibrium points in n-person games. *Proceedings of the national academy of sciences* 36(1):48–49.

[69] Neumann, J v. 1928. Zur theorie der gesellschaftsspiele. *Mathematische annalen* 100(1): 295–320.

[70] Pang, Jong-Shi, and Masao Fukushima. 2005. Quasi-variational inequalities, generalized nash equilibria, and multi-leader-follower games. *Computational Management Science* 2(1):21–56.

[71] Pang, Jong-Shi, and Steven A Gabriel. 1993. Ne/sqp: A robust algorithm for the nonlinear complementarity problem. *Mathematical Programming* 60(1-3):295–337.

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

[73] Ramos, Andrés, Mariano Ventosa, and Michel Rivier. 1999. Modeling competition in electric energy markets by equilibrium constraints. *Utilities Policy* 7(4):233–242.

[74] Riedel, Frank. 2004. Dynamic coherent risk measures. *Stochastic processes and their applications* 112(2):185–200.

[75] Rivier, Michel, Mariano Ventosa, Andrés Ramos, Francisco Martínez-Córcoles, and Ángel Chiarri Toscano. 2001. A generation operation planning model in deregulated electricity markets based on the complementarity problem. *Complementarity: Applications, Algorithms and Extensions* 273–295.

[76] Rockafellar, R Tyrrell, and Jie Sun. 2019. Solving monotone stochastic variational inequalities and complementarity problems by progressive hedging. *Mathematical Programming* 174(1-2):453–471.

[77] Ruszczyński, Andrzej. 2010. Risk-averse dynamic programming for markov decision processes. *Mathematical programming* 125(2):235–261.

[78] Rutherford, Thomas. 1993. Miles: A mixed inequality and nonlinear equation solver.

[79] Rutherford, Thomas F, and Renger van Nieuwkoop. 2011. An integrated transport network-computable general equilibrium models for zurich. In *Swiss transport research conference*, vol. 29.

[80] Schiro, Dane A, Jong-Shi Pang, and Uday V Shanbhag. 2013. On the solution of affine generalized nash equilibrium problems with shared constraints by lemke's method. *Mathematical Programming* 142(1-2):1–46.

[81] Shanbhag, Uday V. 2013. Stochastic variational inequality problems: Applications, analysis, and algorithms. In *Theory driven by influential applications*, 71–107. INFORMS.

[82] Shanbhag, Uday V, Gerd Infanger, and Peter W Glynn. 2011. A complementarity framework for forward contracting under uncertainty. *Operations Research* 59(4):810–834.

[83] Siggerud, Emily. 2014. A benders decomposition method for a multi-stage stochastic energy market equilibrium problem. Master's thesis, Institutt for matematiske fag.

[84] Ventosa, Mariano, Michel Rivier, Andrés Ramos, and Antonio García-Alcalde. 2000. An mcp approach for hydrothermal coordination in deregulated power markets. In *2000 power engineering society summer meeting (cat. no. 00ch37134)*, vol. 4, 2272–2277. IEEE.

[85] Von Heusinger, Anna, and Christian Kanzow. 2008. Sc 1 optimization reformulations of the generalized nash equilibrium problem. *Optimisation Methods & Software* 23(6): 953–973.

[86] Von Neumann, John, and Oskar Morgenstern. 2007. *Theory of games and economic behavior (commemorative edition)*. Princeton university press.

[87] Xu, Huayu, Jong-Shi Pang, Fernando Ordóñez, and Maged Dessouky. 2015. Complementarity models for traffic equilibrium with ridesharing. *Transportation Research Part B: Methodological* 81:161–182.

[88] Yang, Xia, Xuegang Jeff Ban, and Rui Ma. 2017. Mixed equilibria with common constraints on transportation networks. *Networks and Spatial Economics* 17:547–579.

[89] Yao, Jian, Ilan Adler, and Shmuel S Oren. 2008. Modeling and computing two-settlement oligopolistic equilibrium in a congested electricity network. *Operations Research* 56(1):34–47.