RISK-AVERSE MULTI-STAGE MIXED-INTEGER STOCHASTIC PROGRAMMING PROBLEMS

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

By
Ali İrfan Mahmutoğulları
January 2019

RISK-AVERSE MULTI-STAGE MIXED-INTEGER STOCHASTIC PROGRAMMING PROBLEMS

By Ali İrfan Mahmutoğulları

January 2019

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Özlem Çavuş İyigün(Advisor)

Mehmet Selim Aktürk(Co-Advisor)

Oya Karaşan

Seçil Savaşaneril Tüfekci

Emre Nadar

Sakine Batun

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

RISK-AVERSE MULTI-STAGE MIXED-INTEGER STOCHASTIC PROGRAMMING PROBLEMS

Ali İrfan Mahmutoğulları
Ph.D. in Industrial Engineering
Advisor: Özlem Çavuş İyigün
Co-Advisor: Mehmet Selim Aktürk
January 2019

Risk-averse multi-stage mixed-integer stochastic programming problems form a class of extremely challenging problems since the problem size grows exponentially with the number of stages, they are non-convex due to integrality restrictions, and their objective functions are nonlinear in general. In this thesis, we first focus on such problems with an objective of dynamic mean conditional value-at-risk. We propose a scenario tree decomposition approach to obtain lower and upper bounds for their optimal values and then use these bounds in an evaluate-and-cut procedure which serves as an exact solution algorithm for such problems with integer first-stage decisions. Later, we consider a risk-averse day-ahead scheduling of electricity generation or unit commitment problem where the objective is a dynamic coherent risk measure. We consider two different versions of the problem: adaptive and non-adaptive. In the adaptive model, the commitment decisions are updated in each stage, whereas in the non-adaptive model, the commitment decisions are fixed in the first-stage. We provide theoretical and empirical analyses on the benefit of using an adaptive multi-stage stochastic model. Finally, we investigate the trade off between the adaptivity of the model and the computational effort to solve it for risk-averse multi-stage production planning problems with an objective of dynamic coherent risk measure. We also conduct computational experiments in order to verify the theoretical findings and discuss the results of these experiments.

Keywords: Risk-averse optimization, Multi-stage stochastic programming, Mixed-integer programming, Dynamic coherent risk measures.

ÖZET

RİSKTEN KAÇINAN ÇOK AŞAMALI KARMA TAM SAYILI RASSAL PROGRAMLAMA PROBLEMLERİ

Ali İrfan Mahmutoğulları
Endüstri Mühendisliği, Doktora
Tez Danışmanı: Özlem Çavuş İyigün
İkinci Tez Danışmanı: Mehmet Selim Aktürk
Ocak 2019

Riskten kaçınan çok aşamalı karma tam sayılı rassal programlama problemleri, problem büyüklüğünün aşama sayısıyla hızla artması, tam sayı kısıtlamaları nedeniyle konveks olmamaları ve amaç fonksiyonlarının genellikle doğrusal olmaması yüzünden çok zor problemlerdir. Bu tezde, öncelikle amaç fonksiyonları dinamik ortalama şartlı riske maruz değer olan problemleri ele alıyoruz. Bu problemlerin en iyi değerleri için alt ve üst sınırlar veren bir senaryo gruplama yöntemi öneriyoruz ve daha sonra bu sınırları, ilk aşamasındaki değişkenleri tam sayı olan problemler için bir kesin çözüm yöntemi olan hesapla-ve-kes prosedüründe kullanıyoruz. Daha sonra, amaç fonksiyonu dinamik tutarlı risk ölçütü olan riskten kaçınan gün öncesi elektrik üretimi olarak da bilinen birim taahhütü problemini ele alıyoruz. Bu problemin iki farklı türünü göz önünde bulunduruyoruz. Uyarlanabilir modelde, taahhüt kararları her aşamada güncellenirken uyarlanamaz modelde bu kararlar ilk aşamada veriliyor. Uyarlanabilir modeli kullanmanın getirisini teorik ve deneysel olarak gösteriyoruz. Son olarak da amaç fonksiyonları dinamik tutarlı risk ölçütleri olan riskten kaçınan çok aşamalı üretim planlama problemleri için uyarlanabilirlik ve hesaplama emeği arasındaki dengeyi inceliyoruz. Ayrıca, teorik bulgularımızı doğrulamak için hesaplamalı deneyler düzenleyip, deney sonuçlarını tartışıyoruz.

Anahtar sözcükler: Riskten kaçınan optimizasyon, Çok aşamalı rassal programlama, Karma tam sayılı programlama, Dinamik tutarlı risk ölçütleri.

Acknowledgement

I would like to express my deepest gratitude to my advisors Asst. Prof. Özlem Çavuş İyigün and Prof. Mehmet Selim Aktürk for their endless support and guidance throughout my study. Özlem Çavuş İyigün has spent an enormous amount of effort for this thesis and Mehmet Selim Aktürk is not only an advisor but also a life-long mentor to me. It was a great privilege for me to have such great advisors. I cannot thank them enough.

During the last year of my study, I was a visiting student at H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology. I would like to express my gratitude to Prof. Shabbir Ahmed for his guidance and inspiration during my visit at Georgia Tech.

I am grateful to Assoc. Prof. Seçil Savaşaneril Tüfekci and Prof. Hande Yaman Paternotte for reading each part of this thesis and providing precious suggestions during my study. I also want to express my gratitude to Prof. Oya Karaşan, Asst. Prof. Emre Nadar and Asst. Prof. Sakine Batun for accepting to be a member of my examination committee and for their valuable suggestions.

It was an honor and a privilege to be a member of Bilkent IE family, and I would like to thank each member of the department.

I am grateful to so many people for their unlimited support to me. First, I would like to thank my dear friends Haşim Özlü, Cemal İlhan and Nur Timurlenk for their endless trust and encouragement. I also would like to thank Kamyar Kargar, Nihal Berktaş, Halil İbrahim Bayrak, Cansu Gülcan, Başak Yazar and Oğuzhan Efe Şakrak for being great friends. I have also spent great time with my dear friends Fırat Kılcı, Işıl Koyuncu and Ömer Kerem Bekteş during my visit to Atlanta. I would also like to thank Okan Dükkancı, Meltem Peker, Gizem Özbaygın and Özge Şafak for their helps during my study.

During my study, I have been financially supported by TÜBİTAK programs 2211 and 2214. I am grateful to TÜBİTAK for providing this opportunity to me.

Finally, I would like to thank my wife, colleague and friend Halenur; my sister, colleague and friend Özlem and my mother Zehra. I could not succeed without their endless love, encouragement and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The vast majority of operations research literature is devoted to *deterministic optimization models* where problem parameters are deterministic and known when the decision is made. A deterministic optimization model can be written as

$$\min_{x \in \mathcal{X}} \quad f(x),$$

where $x \in \mathbb{R}^m$ is the vector of decision variables, $f : \mathbb{R}^m \to \mathbb{R}$ is the cost function to be minimized and $\mathcal{X} \subseteq \mathbb{R}^m$ is the set of feasible solutions.

However, parameters of many real life problems are not deterministic and not exactly known when the decisions are made. Demand of a product or return of an asset are examples of such parameters. *Stochastic programming* deals with optimization problems where some (or all) problem parameters are subject to uncertainty and follow known probability distributions. Theory of stochastic programming dates back to Dantzig's pioneering work [1] in 1955. Recent applications in finance, energy, health care and production systems revealed that stochastic programming is a powerful tool to elaborate uncertainty in these systems. The modern theory and applications in various areas of operation research are discussed in [2], [3], [4] and references therein.

Stochastic optimization models are divided into two groups depending on their attitude towards risk: risk-neutral and risk-averse models. In *risk-neutral* (stochastic optimization) models, the objective is to minimize expected total cost. The generic risk-neutral model is

$$\begin{aligned} \min \quad & \mathbb{E}_\xi[f(x,\xi)], \\ \text{s.t.} \quad & x \in \mathcal{X}(\xi), \end{aligned}$$

where $f(x,\xi)$ is the random cost depending on the decision vector x and random problem parameters ξ. The set of all feasible decisions $\mathcal{X}(\xi)$ is also defined by ξ. The expectation $\mathbb{E}_\xi$ is taken with respect to ξ and assumed to be well-defined (see, for example, [5]). The risk-neutral model minimizes the cost "on the average" as a corollary of the Law of Large Numbers. However, risk-neutral models are reasonable only if the long term performance is considered irrespective of specific realizations. The following portfolio optimization example shows a shortcoming of risk-neutral models in existence of risky realizations.

Example 1. *([3], pg. 13) An investor wishes to maximize the total expected return by distributing an initial capital W onto m different risky assets. Let x_i be the amount invested in asset i and R_i be the (random) return of asset i for $i = 1, \ldots, m$. The corresponding optimization model is*

$$\begin{aligned} min \quad & -\mathbb{E}_\xi \left[\sum_{i=1}^{m} \xi_i x_i \right], \\ s.t. \quad & \sum_{i=1}^{m} x_i = W, \\ & x = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^m_+, \end{aligned} \tag{1.1}$$

where $\xi_i = 1 + R_i$ and $\mathbb{R}^m_+$ is the nonnegative orthant in $\mathbb{R}^m$. Note that, the objective function (1.1) can alternatively be written as:

$$\mathbb{E}_\xi \left[\sum_{i=1}^{m} \xi_i x_i \right] = \sum_{i=1}^{m} \mathbb{E}_\xi \left[\xi_i \right] x_i = \sum_{i=1}^{m} \mu_i x_i,$$

where $\mu_i = 1 + \mathbb{E}\left[R_i\right]$. An optimal solution of the problem is to invest all capital

into the asset which has the highest expected return. Therefore, optimal value is $-\mu^* W$ *where* $\mu^* := \max_{1 \leq i \leq m} \mu_i$.

Obviously, an optimal solution of the risk-neutral portfolio optimization problem may be subject to high level of risk. The total return depends on realization of the return of one asset only.

Example 1 illustrates that risk-neutral models cannot elaborate the risk due to specific realizations. This result motivates the *risk-averse* (stochastic optimization) models which reflect preferences of a decision maker who avoids risk. Different types of risk-averse models are available in the literature (see, Chapter 2, for a detailed discussion on these models). In this thesis, we consider the risk-averse models where a risk measure is used to reflect the risk-aversion of the decision maker.

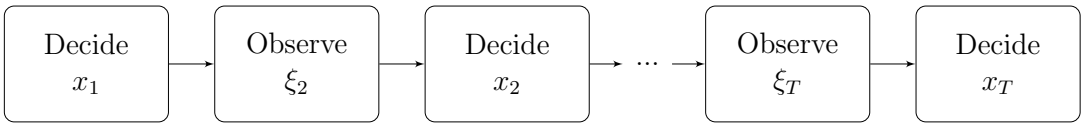
A *risk measure* is defined as a function $\rho : \mathcal{Z} \to \mathbb{R}$ which "quantifies" the risk involved in a random outcome. Here, $\mathcal{Z}$ is a set of random variables for which lower values are preferable, that is, they define a random cost. An interpretation of the risk measure is given in [6] as: $\rho(Z)$ is a fair one-time charge a risk-averse decision maker would be willing to pay instead of random cost $Z \in \mathcal{Z}$. In other words, $\rho(Z)$ is the "risk-adjusted deterministic cost" of the random cost Z. Risk measures are useful to model risk-averse attitudes of decision makers. Moreover, they are easier to interpret and have practical meaning. Therefore, they are widely used in the literature. A canonical risk-averse model is given by

$$\begin{aligned} \min \quad & \rho(f(x, \xi)), \\ \text{s.t.} \quad & x \in \mathcal{X}(\xi), \end{aligned}$$

where the expectation in the risk-neutral model is replaced by a risk measure.

In many stochastic optimization models, as in Example 1, the realization of random problem parameters occurs once and for all. In that case, decision variables can be grouped into two sets: the first and second stage variables. The first stage variables correspond to decisions which are made before the realization of

Figure 1.1: The decision process in multi-stage models.



random problem parameters. On the other hand, the second stage variables correspond to decisions which are made after observing the realization of random problem parameters. Thus, these model are called as *two-stage models*. However, in *multi-stage models*, the randomness unveils gradually a over fixed length decision horizon and decisions are made between two consecutive realizations. These decision epochs are called as *stages*.

In multi-stage models, the first stage decision x_1 is made based on the first stage deterministic problem parameters ξ_1. Then, a realization of the second stage problem parameters ξ_2 occurs and the second stage decisions x_2 are made. This decision process continues through a $T-$stage decision horizon. If the random parameters evolve as a discrete-time stochastic process with finite support, then the whole process can be represented by a $T-$stage *scenario tree*. The decision process in multi-stage models is depicted in Figure 1.1.

Multi-stage models provide high adaptability of decisions in a dynamic environment. However, they are more challenging than their two-stage counterparts due to their increasing problem size which, in general, increases exponentially with respect to the number of stages T.

In this thesis, we consider solution methods for risk-averse multi-stage mixed-integer stochastic optimization models. These models are large scale non-convex optimization problems for which no efficient solution method is available, to the best of our knowledge. Difficulty of these problems is due to three main reasons. The first reason is the risk measures used in the objective function. Unlike their risk-neutral counterparts, risk-averse models do not enjoy decomposition directly due to structure of risk measures used in the objective. The second reason is scalability. For a non-trivial problem, size of a multi-stage model grows exponentially with the number of stages. The third reason is the mixed-integer nature of

decision variables. Even if the multi-stage problem is deterministic, the resulting problem is still a large scale mixed-integer model.

Therefore, in this thesis, we provide approximate and exact solution methods for risk-averse multi-stage mixed-integer stochastic optimization models. We provide theoretical discussions on these methods such as guaranteed bounds for approximate methods, and convergence proofs for exact methods. We also provide a set of computational experiments on different problems in order to test the proposed methods. In these experiments, we use problems from different areas of operations research such as location, production, expansion and power system optimization. Computational results of these experiments indicate that the proposed methods are powerful tools for risk-averse multi-stage mixed-integer stochastic optimization models.

The rest of the thesis is organized as follows. In Chapter 2, we present the related literature on risk-averse stochastic optimization problems. In Chapter 3, we propose a scenario tree decomposition approach, namely group subproblem approach, to obtain bounds for risk-averse multi-stage mixed-integer stochastic optimization problems with an objective of dynamic mean conditional value-at-risk (mean-CVaR). Our approach does not require any special problem structure such as convexity and linearity, therefore it can be applied to a wide range of problems. We obtain lower bounds by using different convolution of mean-CVaR risk measures and different scenario partition strategies. The upper bounds are obtained through the use of optimal solutions of group subproblems. Using these lower and upper bounds, we propose a solution algorithm for risk-averse mixed-integer multi-stage stochastic problems with mean-CVaR risk measures. We test the performance of the proposed algorithm on a multi-stage stochastic lot sizing problem and compare different choices of lower bounds and partition strategies. Comparison of the proposed algorithm to a commercial solver reveals that, on the average, the proposed algorithm yields 1.13% stronger bounds. The commercial solver, on the average, requires more than a factor of five additional running time to reach the same optimality gap obtained by the proposed algorithm.

In Chapter 4, we propose an exact solution algorithm for risk-averse multi-stage mixed-integer stochastic optimization problems with an objective of dynamic mean-CVaR risk measure and binary first stage decision variables. The proposed algorithm is based on an evaluate-and-cut procedure and it uses lower bounds obtained from a scenario tree decomposition method presented in Chapter 3. We also show that, under the assumption that the first stage integer variables are bounded, our algorithm solves problems with mixed-integer variables in all stages. Computational experiments on risk-averse multi-stage stochastic server location and generation expansion problems reveal that the proposed algorithm is able to solve problem instances with more than one million binary variables within a reasonable time under a modest computational setting.

In Chapter 5, we consider day-ahead scheduling of electricity generation or unit commitment which is an important and challenging optimization problem in power systems. Variability in net load arising from the increasing penetration of renewable technologies have motivated study of various classes of stochastic unit commitment models. In the models with non-adaptive commitment, the generation schedule for the entire day is fixed while the dispatch is adapted to the uncertainty, whereas in the models with adaptive commitment, the generation schedule is also allowed to dynamically adapt to the uncertainty realization. The latter one provides more flexibility in the generation schedule, however, it requires significantly higher computational effort. To justify this additional computational effort, we provide theoretical and empirical analyses of the *value of adaptive commitment* for risk-averse multi-stage stochastic unit commitment models. The value of adaptive commitment measures the relative advantage of adaptive commitment over its non-adaptive counterpart. Our results indicate that, for unit commitment models, value of adaptive commitment increases with the level of uncertainty and number of periods, and decreases with the degree of risk-aversion of the decision maker.

In Chapter 6, we consider risk-averse multi-stage production planning problems as a generalization of the unit commitment problem discussed in Chapter 5. In these problems, we make set-up decisions of a set of generators and production amount decisions of each generator in order to satisfy random demand

of a single product through a multi-stage decision horizon. For these problems, we consider two types of models with respect to their adaptivity to the demand uncertainty. In fully adaptive models, both set-up and production decisions are given in on-line fashion, that is, they are adapted to demand uncertainty. However, in the models with non-adaptive set-up decisions, the set-up decisions are off-line, that is, they are fixed at the beginning of the decision horizon whereas the production decisions are on-line. We discuss the trade off between flexibility of the adaptive set-up decisions and computational convenience of the non-adaptive set-up decisions. As an intermediate case between the models with adaptive and non-adaptive set-up decisions, we also consider a model with partially adaptive set-up decisions. Moreover, we propose a rolling horizon solution algorithm for the fully adaptive model where the model with non-adaptive set-up decisions is used as an approximation. In order to reduce computational difficulty of the proposed models and the rolling horizon method, we consider restricting the production amounts as affine functions of demand realizations. We propose analytical results on the relation among the optimal solutions of the models with fully adaptive, partially adaptive, non-adaptive set-up decisions and the solution obtained from the rolling horizon algorithm. Finally, we conduct a set of computational experiments on a risk-averse multi-stage lot sizing problem to investigate the computational efficiency of the proposed rolling horizon method and verify the analytical results.

Chapters 3, 5 and 6 correspond to approximate solution methods for risk-averse multi-stage mixed-integer stochastic programming problems. In Chapter 4, we provide an exact solution approach for these problems where the objective is a dynamic mean-CVaR risk measure. The concluding remarks and a discussion on the contribution of this thesis to the literature are given in Chapter 7. We also provide a detailed discussion on future research directions and possible extensions of current work in Chapter 7.

Chapter 2

Literature Review

In stochastic optimization problems, risk-averse attitude of decision makers can be represented in several ways. In this chapter, we present a brief survey of existing approaches in risk-averse optimization such as expected utility theory, stochastic dominance, chance constraints and mean-risk models. Then, we present definitions of coherent, conditional and dynamic risk measures. Finally, we present the existing literature on decomposition methods for multi-stage stochastic programming problems.

2.1 Expected Utility Theory

In *expected utility theory*, two random outcomes are compared based on their expected (dis-)utilities. A random variable Z is preferred to another random variable W if $\mathbb{E}[u(Z)] \leq \mathbb{E}[u(W)]$ where $u(\cdot)$ is some dis-utility function. Let $f(x, \xi)$ be the random cost due to decision x and random problem parameters ξ. Also let $\mathcal{X}(\xi)$ be the set of feasible decisions with respect to ξ. Then, the

optimization problem is given as

$$\min_{x \in \mathcal{X}(\xi)} \quad \mathbb{E}_{\xi}[u(f(x,\xi))]. \tag{2.1}$$

If $u(\cdot)$ is convex, then Jensen's inequality

$$u(\mathbb{E}_{\xi}[f(x,\xi)]) \leq \mathbb{E}_{\xi}[u(f(x,\xi))],$$

implies that the certain outcome $\mathbb{E}_{\xi}[f(x,\xi)]$ is at least as preferable as the random outcome $f(x,\xi)$. Therefore, (2.1) is a risk-averse formulation for any convex and non-decreasing $u(\cdot)$. Moreover, if $u(\cdot)$ is affine, (2.1) is a risk-neutral formulation and if $u(\cdot)$ is concave, it is a risk-seeking formulation. If the random outcome represents profit instead of cost, (2.1) is replaced with a maximization problem where $u(\cdot)$ is assumed to be non-decreasing and concave. In that case, $u(\cdot)$ is called as a *utility function*.

An important measure of risk-aversion is *coefficient of absolute risk-aversion* $A(x,u) = -u''(x)/u'(x)$ proposed in [7] and [8] to control the degree of risk-aversion. However, interpreting (dis-)utility functions is not straightforward. A more detailed discussion on usage of (dis-)utility functions in risk-averse problems can be found in [3].

2.2 Stochastic Dominance Constraints

Another way to deal with risk is to use *stochastic dominance constraints* in stochastic optimization models. Stochastic dominance constraints enable the decision maker to compare two different random variables with respect to their involved risk. A random variable Z dominates another random variable W in the first order, i.e. $Z \succcurlyeq_{(1)} W$, if $F(a) \leqslant G(a)$ for all $a \in \mathbb{R}$ and in the second order, i.e. $Z \succcurlyeq_{(2)} W$, if $\int_{-\infty}^{a} F(x)dx \leq \int_{-\infty}^{a} G(x)dx$ for all $a \in \mathbb{R}$ where $F(\cdot)$ and $G(\cdot)$ are distribution functions of Z and W, respectively. If a reference random

outcome Z is known, then a set of constraints can be included in stochastic optimization problems which ensure that an optimal solution is at least as preferable as Z. However, determining the reference random outcome is another issue that the decision maker should address before adding these constraints to the model.

Stochastic dominance constraints date back to pioneering works [9] and [10]. A more detailed discussion about using stochastic dominance constraints in risk-averse problems can be found in [3] and [4].

2.3 Chance Constraints

A canonical formulation of chance constrained stochastic programming models is given as

$$\min \quad \mathbb{E}_{\xi}[f(x,\xi)], \tag{2.2}$$

$$\text{s.t.} \quad \Pr\{g_j(x,\xi) \leq 0, \forall j \in \mathcal{J}\} \geq p, \tag{2.3}$$

$$x \in \mathcal{X}(\xi), \tag{2.4}$$

where constraint (2.3) ensures that the set of constraints $g_j(x,\xi) \leq 0, \forall j \in \mathcal{J}$ hold with probability of at least $p \in (0,1)$. Chance constrained models are introduced in [11] in 1959.

Even under simplest settings, these models are challenging. This challenge follows from the fact that the set of feasible solutions of the problem (2.2)-(2.4) can be non-convex even if the function $g_j(\cdot)$ is convex in x for all $j \in \mathcal{J}$ and $\mathcal{X}(\xi)$ is a convex set (see, for example, [12] for a detailed discussion).

A detailed survey on chance constrained stochastic programming models can be found in [2] and [3].

2.4 Mean-risk Models

The model

$$\min_{x \in \mathcal{X}(\xi)} \quad \mathbb{E}_\xi[f(x,\xi)] + \lambda \mathbb{D}[f(x,\xi)] \tag{2.5}$$

is called as a *mean-risk* model where $\mathbb{E}_\xi[f(x,\xi)]$ and $\mathbb{D}[f(x,\xi)]$ are expected value of the random cost and some dispersion measure, respectively. Here λ is the price of risk which controls the degree of risk-aversion. In his seminal work [13], Markowitz presents the first mean-risk model for a portfolio optimization problem.

The most natural choice for the dispersion measure in problem (2.5) is variance, that is, $\mathbb{D}(\cdot) = \mathbb{V}(\cdot)$ as in [13]. In that case, both upper and lower deviations from the mean cost are penalized in (2.5). However, we prefer lower cost values, therefore penalization of lower deviations is not desirable. The following example highlights this shortcoming of using variance as a dispersion measure in problem (2.5).

Example 2. *([14]) Assume that the random variable ξ takes values ξ^1 and ξ^2 with probabilities p and $1-p$, respectively for some $p \in (0,1)$. Let Z take values $Z(\xi^1) = -a$ for some $a > 0$ and $Z(\xi^2) = 0$ depending on the realization of ξ. Similarly, let W be another random variable with $W(\xi^1) = W(\xi^2) = 0$. The mean-risk function $F(\cdot) := \mathbb{E}_\xi[\cdot] + \lambda \mathbb{V}[\cdot]$ is calculated for Z and W as $F(Z) = -ap + \lambda a^2 p(1-p)$ and $F(W) = 0$. Thus, if $\lambda a(1-p) > 1$, we have $F(Z) > F(W)$. However, W dominates Z, that is, $W(\xi^1) \geq Z(\xi^1)$ and $W(\xi^2) \geq Z(\xi^2)$.*

Consider mean-risk model (2.5) where $f(x,\xi) := xZ + (1-x)W$ and $\mathcal{X} := [0,1]$. Note that $f(x,\xi) = xZ$ and $f(1,\xi)$ is dominated by $f(x,\xi)$ for $x \in \mathcal{X}$. However, $x = 1$ is not an optimal solution of (2.5) since $f(1,\xi) = F(Z)$ is strictly greater than $f(0,\xi) = F(W)$.

In Example 2, the optimal solution of the mean-risk model (2.5) is W even though Z always takes lower values than W. This example reveals that a optimal solution of the mean-risk model (2.5) may not be the most preferable solution.

Hence, we need to specify some axioms in order to measure the risk involved in a random cost properly. A more detailed discussion about using mean-risk models in risk-aversion can be found in [3] and [15].

2.5 Coherent Risk Measures

Let Ω be a *sample space* equipped with a *sigma algebra* $\mathscr{F}$ and $\mathcal{Z} := \mathcal{L}_p(\Omega, \mathscr{F}, P)$ be the space of all random variables that have finite moment of order p with respect to *probability distribution* P for some $p \in [1, \infty)$. An element ω of the sample space Ω is called as a *scenario*.

A *risk measure* is a function $\rho : \mathcal{Z} \to \mathbb{R} \cup \{\infty\} \cup \{-\infty\}$ which assigns a random variable to risk involved in that random variable. Then, the risk-averse stochastic optimization problem is

$$\min_{x \in \mathcal{X}(\xi)} \quad \rho(f(x, \xi)). \tag{2.6}$$

In order to guarantee that the problem (2.6) has meaningful interpretations, we follow the concept of *coherent risk measure* defined in [16].

Let $Z, W \in \mathcal{Z}$ be uncertain outcomes defined on the probability space $(\Omega, \mathscr{F}, P)$ for which lower realizations are preferable. A risk measure $\rho : \mathcal{Z} \to \mathbb{R}$ is called *coherent* if it satisfies:

- (A1) Convexity: $\rho(\alpha Z + (1-\alpha)W) \leq \alpha\rho(Z) + (1-\alpha)\rho(W)$ for all $Z, W \in \mathcal{Z}$ and $\alpha \in [0, 1]$,
- (A2) Monotonicity: $Z \preceq W$ implies $\rho(Z) \leq \rho(W)$ for all $Z, W \in \mathcal{Z}$,
- (A3) Translational Equivariance: $\rho(Z + t) = \rho(Z) + t$ for all $t \in \mathbb{R}$ and $Z \in \mathcal{Z}$,
- (A4) Positive Homogeneity: $\rho(tZ) = t\rho(Z)$ for all $t > 0$ and $Z \in \mathcal{Z}$,

where $Z \preceq W$ indicates component-wise partial ordering such that $Z_\omega \leq W_\omega$ for almost every $\omega \in \Omega$. Here, Z_ω is the value that Z takes in scenario $\omega \in \Omega$.

The above axioms have practical interpretations. Axioms (A1) and (A4) imply that diversification does not create extra risk or equivalently, $\rho(\cdot)$ is sub-additive in a sense that

$$\rho(Z + W) \leq \rho(Z) + \rho(W) \text{ for all } Z, W \in \mathcal{Z}.$$

Axiom (A2) implies that higher cost yields higher risk. Axiom (A3) implies that increasing the cost for a fixed t units increases risk by the same amount. Finally, axiom (A4) implies that risk remains same regardless of currency type. A more detailed discussion on interpretations of these axioms can be found in [16].

Important examples of coherent risk measures are quantile and deviation based risk measures. One example of the former one is conditional value-at-risk (CVaR) (see, for example, [17])

$$\text{CVaR}_\alpha(Z) := \inf_{\eta \in \mathbb{R}} \Big\{ \eta + \frac{1}{1-\alpha} \mathbb{E}[(Z - \eta)_+] \Big\}. \tag{2.7}$$

where $(a)_+ = \max\{a, 0\}$, $a \in \mathbb{R}$ and $\alpha \in [0, 1)$ is the level parameter. The infimum on the right hand side of (2.7) is attained at $\eta^* = \text{VaR}_\alpha(Z)$ where value-at-risk (VaR) corresponds to left-side quantile of Z. The interpretation of VaR at level α is that "costs larger than $\text{VaR}_\alpha(Z)$ occur with probability of at most α" (see, for example, [3]). Similarly, $\text{CVaR}_\alpha(Z)$ corresponds to "the expected cost in most pessimistic 100α percent of all scenarios". CVaR is a coherent risk measure, but VaR violates sub-additivity.

Another quantile-based risk measure, namely mean-CVaR, is defined as a convex combination of mean cost and CVaR, that is,

$$\rho(Z) := (1 - \epsilon)\mathbb{E}[Z] + \epsilon \text{CVaR}_\alpha(Z), \tag{2.8}$$

for some weight parameter $\epsilon \in [0, 1]$. Definition of CVaR only represents the quantile information, however, (2.8) conveys both expected cost and the quantile

information and thus generalizes CVaR.

An example of deviation based coherent risk measures is mean-upper semi deviation (MUSD). The mean-upper semi deviation is defined as the sum of expected cost and a penalty term for expected upper deviation from the mean cost, that is,

$$\text{MUSD}(Z) := \mathbb{E}[Z] + \lambda \mathbb{E}[(Z - \mathbb{E}[Z])_+], \tag{2.9}$$

for some penalty parameter $\lambda \in [0, 1]$.

More examples for coherent risk measures and a detailed discussion on their theoretical properties can be found in [3], [18] and references therein.

In this thesis, we consider a multi-stage decision horizon, therefore we consider extension of coherent risk measures in a dynamic setting.

2.6 Conditional and Dynamic Risk Measures

In order to measure risk in a dynamic setting, we use conditional risk measures as an extension of coherent risk measures. Consider sigma algebras $\mathscr{F}_t \subseteq \mathscr{F}_{t+1}$ and spaces $\mathcal{Z}_t := \mathcal{L}_p(\Omega, \mathscr{F}_t, P)$ and $\mathcal{Z}_{t+1} := \mathcal{L}_p(\Omega, \mathscr{F}_{t+1}, P)$ defined using these sigma algebras. A mapping $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t} : \mathcal{Z}_{t+1} \to \mathcal{Z}_t$ is called *one-step conditional risk measure* if:

- (B1) Convexity: $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(\alpha Z + (1-\alpha)W) \preceq \alpha \rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(Z) + (1-\alpha)\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(W)$ for all $Z, W \in \mathcal{Z}_{t+1}$ and $\alpha \in [0, 1]$,
- (B2) Monotonicity: $Z \preceq W$ implies $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(Z) \preceq \rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(W)$ for all $Z, W \in \mathcal{Z}_{t+1}$,
- (B3) Translational Equivariance: $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(Z + W) = \rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(Z) + W$ for all $W \in \mathcal{Z}_t$ and $Z \in \mathcal{Z}_{t+1}$,
- (B4) Positive Homogeneity: $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(tZ) = t\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(Z)$ for all $t > 0$ and $Z \in \mathcal{Z}_{t+1}$.

Axioms (B1)-(B4) have similar interpretations with axioms (A1)-(A4). Moreover, $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(\cdot)$ has a similar interpretation with the coherent risk measure $\rho(\cdot)$. $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(Z)$ is a fair one-time $\mathscr{F}_t-$measurable charge a risk-averse decision maker would be willing to pay instead of $\mathscr{F}_{t+1}-$measurable cost $Z \in \mathcal{Z}_{t+1}$. Alternatively, it can be said that $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(Z)$ is the "risk-adjusted $\mathscr{F}_t-$measurable cost" of cost Z.

Thus, one-step conditional counterparts of (2.7) and (2.9) can be defined as

$$\inf_{\eta \in \mathcal{Z}_t} \Big\{ \eta + \frac{1}{1-\alpha} \mathbb{E}[(Z-\eta)_+ | \mathcal{F}_t] \Big\}, \tag{2.10}$$

and

$$\mathbb{E}[Z|\mathcal{F}_t] + \lambda \mathbb{E}[(Z - \mathbb{E}[Z|\mathcal{F}_t])_+ | \mathcal{F}_t], \tag{2.11}$$

respectively. Theoretical properties of one-step conditional risk measures are extensively discussed in [19].

In a $T-$stage decision environment, we consider the risk involved in a random cost sequence instead of a single random cost. Let the nested sequence of sigma algebras $\{\emptyset, \Omega\} = \mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots \subset \mathcal{F}_T = \mathcal{F}$ be called as a *filtration* which represents our gradually increasing information through a $T-$stage planning horizon. The set of all $\mathcal{F}_t-$measurable and $p-$integrable random variables are denoted by $\mathcal{Z}_t := \mathcal{L}_p(\Omega, \mathcal{F}_t, P)$ for $t \in \{1, 2, \ldots, T\}$ for some $p \in [1, \infty)$. Note that since $\mathcal{F}_1 = \{\emptyset, \Omega\}$, $\mathcal{Z}_1 = \mathbb{R}$.

Consider the composition

$$\rho_{\mathscr{F}_2|\mathscr{F}_1} \circ \rho_{\mathscr{F}_3|\mathscr{F}_2} \circ \cdots \circ \rho_{\mathscr{F}_T|\mathscr{F}_{T-1}} : \mathcal{Z}_T \xrightarrow{\rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}} \mathcal{Z}_{T-1} \cdots \mathcal{Z}_3 \xrightarrow{\rho_{\mathscr{F}_3|\mathscr{F}_2}} \mathcal{Z}_2 \xrightarrow{\rho_{\mathscr{F}_2|\mathscr{F}_1}} \mathbb{R},$$

and let $\varrho_{1,T} : \mathcal{Z}_T \to \mathbb{R}$ be a *dynamic risk measure* such that

$$\varrho_{1,T} := \rho_{\mathscr{F}_2|\mathscr{F}_1} \circ \rho_{\mathscr{F}_3|\mathscr{F}_2} \circ \cdots \circ \rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}, \tag{2.12}$$

where each $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}$ is a conditional risk measure for all $t = 1, \cdots, T-1$. By the

translational equivariance axiom (B3), $\varrho_{1,T}(\cdot)$ can equivalently be written as:

$$\varrho_{1,T}(Z_1, Z_2, \ldots, Z_T) = Z_1 + \rho_{\mathscr{F}_2|\mathscr{F}_1}(Z_2 + \rho_{\mathscr{F}_3|\mathscr{F}_2}(Z_3 + \cdots + \rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}(Z_T)\cdots)), \tag{2.13}$$

where $Z_t \in \mathcal{Z}_t$ is the cost incurred at stage $t = 1, \cdots, T$.

We can interpret $\varrho_{1,T}(Z_1, Z_2, \ldots, Z_T)$ as a fair one-time deterministic charge a risk-averse decision maker would be willing to pay instead of random cost sequence $\{Z_t\}_{t=1}^T$. Another interpretation of $\varrho_{1,T}(Z_1, Z_2, \ldots, Z_T)$ is the "risk-adjusted deterministic cost" of random cost sequence $\{Z_t\}_{t=1}^T$.

2.7 Risk-averse Multi-stage Mixed-integer Stochastic Programming Problem

In this thesis, our main interest is a *risk-averse multi-stage mixed-integer stochastic programming problem*. We use ξ_t and x_t to denote the vector of problem parameters and decisions at stage $t \in \{1, \ldots, T\}$, respectively. Note that, for each t, ξ_t and x_t are $\mathscr{F}_t-$measurable. The collection of all decisions through the decision horizon $x := (x_1, x_2, \ldots, x_T)$ is called as a *policy*. At the first stage, the vector of problem parameters ξ_1 and decisions x_1 are deterministic since $\mathscr{F}_1 = \{0, \emptyset\}$. At stage $t \in \{2, \ldots, T\}$, some or all problem parameters are random.

The risk-averse multi-stage mixed-integer stochastic programming problem can be defined as

$$\min_{x \in \mathcal{X}} \quad \varrho_{1,T}(f_1(x_1), f_2(x_2, \xi_2), \ldots, f_T(x_T, \xi_T)), \tag{2.14}$$

where $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2(x_1, \xi_2) \times \cdots \times \mathcal{X}_T(x_{T-1}, \xi_T)$ is an abstract representation of (possibly non-linear) set of feasible polices. The first stage feasibility set $\mathcal{X}_1 \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{m_1}$ is a mixed-integer deterministic set and for $t \in \{2, \ldots, T\}$, $\mathcal{X}_t : \mathbb{R}^{n_{t-1}} \times \mathbb{Z}^{m_{t-1}} \times \Xi_t \rightrightarrows \mathbb{R}^{n_t} \times \mathbb{Z}^{m_t}$ are $\mathscr{F}_t-$measurable mixed-integer point-to-set mappings. The set Ξ_t is the support of ξ_t, for $t \in \{2, \ldots, T\}$. The first stage cost is deterministic and represented by a possibly nonlinear, real-valued

function $f_1 : \mathbb{R}^{n_1} \times \mathbb{Z}^{m_1} \to \mathbb{R}$. The cost functions $f_t : \mathbb{R}^{n_t} \times \mathbb{Z}^{m_t} \times \Xi_t \to \mathbb{R}$, $t \in \{2, \ldots, T\}$ are $\mathscr{F}_t-$measurable and may be nonlinear. If each one-step conditional risk measure in the definition of (2.12) is conditional expectation, that is, $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(\cdot) = \mathbb{E}[\cdot|\mathscr{F}_t]$ for each $t \in \{1, 2, \ldots, T-1\}$, the risk-averse multi-stage mixed-integer stochastic programming problem (2.14) reduces to its risk-neutral counterpart. The problem (2.14) can equivalently be written as

$$\begin{aligned} \min \quad & f_1(x_1) + \rho_{\mathscr{F}_2|\mathscr{F}_1}\left[f_2(x_2,\xi_2) + \rho_{\mathscr{F}_3|\mathscr{F}_2}\left(+\cdots+\rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}\{f_T(x_T,\xi_T)\}\right)\right], \\ \text{s.t.} \quad & x_1 \in \mathcal{X}_1, x_t \in \mathcal{X}_t(x_{t-1},\xi_t), \quad t \in \{2,\ldots,T\}, \end{aligned}$$

by using the relation between the dynamic risk measure and one-step conditional risk measures given in (2.13). Alternatively, a dynamic programming (DP) formulation of the risk-averse multi-stage mixed-integer stochastic programming problem can be written as

$$\begin{aligned} \min_{x_1\in\mathcal{X}_1} f_1(x_1) + \rho_{\mathscr{F}_2|\mathscr{F}_1}\Bigg[& \min_{x_2\in\mathcal{X}_2(x_1,\xi_2)} f_2(x_2,\xi_2) + \\ & \rho_{\mathscr{F}_3|\mathscr{F}_2}\left(\min_{x_3\in\mathcal{X}_3(x_2,\xi_3)} + \cdots + \rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}\left\{\min_{x_T\in\mathcal{X}_T(x_{T-1},\xi_T)} f_T(x_T,\xi_T)\right\}\right)\Bigg]. \end{aligned} \tag{2.15}$$

Two phenomena have an essential role in formulations of risk-averse multi-stage mixed-integer stochastic programming problems. Decisions at stage $t \in \{1, \ldots, T\}$ are made based on the available information up to stage t. This requirement is called as *non-anticipativity*. Moreover, for any state of the system at stage t, optimal decisions should not involve possible future realizations that cannot happen. This principle is called as *time consistency* (see, [20]).

If the random parameters' values $\xi_1, \xi_2, \ldots, \xi_T$ evolve as a discrete-time stochastic process with discrete support, that is $|\Xi_t| < \infty$ for all $t \in \{2, \ldots, T\}$, then the whole process can be represented by a $T-$stage scenario tree[1]. In this

[1]Even though the random parameters have continuous support, an empirical distribution obtained by sampling without replacement can be used to approximate the true distribution at any accuracy. A detailed discussion is given in Section 5 of [3].

scenario tree, each node at stage $t \in \{1, 2, \ldots, T\}$ represents a possible realization of random process $\xi_1, \xi_2, \ldots, \xi_t$. In this case, a deterministic optimization problem, namely *deterministic equivalent problem (DEP)*, can be used to solve the risk-averse multi-stage mixed-integer stochastic programming problem where each decision at each node of the scenario tree is represented by a variable in DEP. However, for any non-trivial scenario tree, size of DEP grows exponentially and problem gets computationally intractable for even moderate number of scenarios. Therefore, existing solution techniques for the risk-averse multi-stage stochastic programming problems are based on stage-wise and scenario-wise decomposition of scenario tree.

2.8 Decomposition Methods for Multi-stage Stochastic Programming Problems

Stochastic dual dynamic programming (SDDP) is a sampling-based stage-wise scenario tree decomposition technique for multi-stage stochastic programming problems. The method is first proposed in [21] for risk-neutral problems and later extended to risk-averse problems in [22], [23] and [24]. SDDP is based on approximation of cost-to-go functions in DP formulation (2.15) by piecewise linear functions. The convergence to an optimal solution is guaranteed under convexity assumption. Moreover, in SDDP, the random data process is assumed to be stage-wise independent, that is, ξ_{t+1} does not depend on $\xi_{[t]}$. This assumption on stage-wise independence enables us to write cost-to-go functions as functions of stages. In [25], an extension of SDDP is proposed for the risk-neutral problems with integer variables by relaxing the integrality requirements. Later, in [26], this approach is extended to risk-averse mixed-integer problems. Recently, in [27], an extension of SDDP is proposed to solve risk-neutral multi-stage mixed-integer problems with binary state variables. They prove that SDDP method provides an exact solution to the problem in finite number of iterations when the cuts satisfy some sufficient conditions.

In [28], a scenario-wise decomposition method is proposed for two-stage risk-averse mixed-integer stochastic programming problems. The proposed method is based on a branch-and-bound (BB) procedure where decomposition is achieved by dualizing the non-anticipativity constraints in a Lagrangian manner. Later, this BB procedure is extended to risk-neutral multi-stage problems in [29]. This procedure is also extended to two-stage risk-averse problems by exploiting the structure of specific risk measures in [30] and [31].

Another scenario-wise decomposition method for risk-neutral multi-stage convex optimization problems, namely progressive hedging algorithm (PHA), is proposed in [32]. PHA is based on iteratively solving saddle points of a proximal augmented Lagrangian function that decomposes for each scenario. Then, single scenario solutions are hedged to get a non-anticipative solution. In [33] and [34], PHA is extended to risk-neutral multi-stage mixed-integer problems where convergence to an optimal solution is not guaranteed. In [35], PHA is used to solve nodal relaxations within the framework of a branch-and-bound algorithm. A detailed discussion on PHA for risk-neutral multi-stage convex optimization problems can be found in [36].

A scenario-wise scenario tree decomposition method for risk-averse multi-stage stochastic programming problems is proposed in [37]. The decomposition is obtained by relaxing non-anticipativity constraints in a Lagrangian manner and solving the dual of the problem. In [37], the problem is assumed to be linear and hence convergence to the optimal solution is guaranteed.

A recent stream of research proposes an alternative way of obtaining bounds for mixed-integer multi-stage stochastic problems via a scenario tree decomposition. In that approach, the sample space is partitioned into subspaces called as groups, and the problem is solved for the scenarios in a group instead of the original sample space. These smaller problems are called as group subproblems. In [38], a group subproblem approach is proposed for risk-neutral mixed-integer two-stage stochastic problems. They show that the expected value of the optimal values of group subproblems gives a lower bound on the optimal value of the original problem. Later, this approach is extended to the risk-neutral multi-stage problems

in [39], [40], and [41]. Recently, in [42], group subproblem approach is applied to risk-averse mixed-integer multi-stage stochastic problems where the objective is a concave dis-utility function applied to the total cost over the planning horizon.

Recent studies reveal that it is possible to come up with exact solution methods for risk-neutral and risk-averse two-stage mixed-integer stochastic programming problems by exploiting the nature of binary variables. In [43], *no-good cuts* are used in an *evaluate-and-cut* procedure for risk-neutral two-stage mixed-integer models with binary first stage decisions. The proposed procedure is a scenario decomposition algorithm which iteratively evaluates the objective value for a set of binary first stage solutions and cuts these solutions from the feasible set. In [44], the procedure is extended to risk-averse two-stage problems with binary first stage variables. They consider three different exact solution algorithms using dual representations of coherent risk measures, scenario decomposition, cutting planes, subgradient method and no-good cuts. The computational experiments presented by [43] and [44] reveal that risk-neutral and risk-averse two-stage stochastic programing problems with binary first stage variables can be solved optimally within reasonable computation times.

2.9 Summary

Although the aforementioned solution methods are available for related problems, the complex structure of risk-averse multi-stage mixed-integer programming problems prohibits computationally tractable solution methods. For example, convergence of SDDP is based on convexity assumption which does not hold in existence of integer variables. Another challenge in risk-averse multi-stage mixed-integer programming problems is decomposition. Unlike their risk-neutral counterparts, when non-anticipative constraints are relaxed, the risk-averse problems do not decompose into each scenario.

We consider a group subproblem approach to obtain bounds for risk-averse

multi-stage mixed-integer programming problems in Chapter 3. Although similar scenario grouping based bounds have been obtained for risk-neural problems in [38], [39], [40], [41] and risk-averse models with dis-utility functions in [42], to the best of our knowledge, such bounds were not available for the risk-averse models with coherent risk measures. We later use the bounds obtained from group subproblems in an exact solution procedure in Chapter 4. To the best of out knowledge, there is no other exact solution method available for even moderate size instances of any class of risk-averse multi-stage mixed-integer programming problems. In Chapters 5 and 6, we investigate the trade off between the adaptability of these models and their computational performance. Although, the value of adaptivity has been investigated for some risk-neutral models (see, for example, [45]), similar results were missing for risk-averse models.

Chapter 3

Bounds on Risk-averse Multi-stage Mixed-integer Stochastic Programming Problems with Mean-CVaR

In this chapter, we propose a scenario tree decomposition algorithm for risk-averse multi-stage mixed-integer stochastic problems with a dynamic objective function defined via mean-CVaR. The suggested algorithm is based on group subproblem approach and it is used to find lower and upper bounds on the optimal value of the problem. We propose infinitely many valid lower bounds on mean-CVaR risk measure that can be used within the frame of the algorithm. We also investigate the effect of scenario partitioning strategies on the quality of the different lower bounds by considering different partitioning strategies based on the structure of the scenario tree and disparateness of scenario realizations.

The organization of the chapter is as follows: In Section 3.1, we present problem definition and scenario tree representation of the random process of problem parameters. Section 3.2 includes our main results on obtaining different lower

bounds for mean-CVaR via a scenario grouping approach. We consider the application of these lower bounds to a risk-averse mixed-integer multi-stage stochastic problem with a dynamic objective function defined via mean-CVaR. We also suggest a method to obtain an upper bound. The computational study conducted on a multi-stage lot sizing problem and related discussions are presented in Section 3.3. Section 3.4 is devoted to concluding remarks.

The results of this chapter are published in European Journal of Operational Research [46].

3.1 Problem Definition and Scenario Tree Representation

We first recall the risk-averse multi-stage mixed-integer stochastic programming problem

$$\min_{x \in \mathcal{X}} \quad \varrho_{1,T}(f_1(x_1), f_2(x_2, \xi_2), \dots, f_T(x_T, \xi_T)), \tag{3.1}$$

where $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2(x_1, \xi_2) \times \cdots \times \mathcal{X}_T(x_{T-1}, \xi_T)$ is the abstract representation of a (possibly non-linear) set of feasible polices. The first stage feasibility set $\mathcal{X}_1 \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{m_1}$ is a mixed-integer deterministic set and for $t \in \{2, \dots, T\}$, $\mathcal{X}_t : \mathbb{R}^{n_{t-1}} \times \mathbb{Z}^{m_{t-1}} \times \Xi_t \rightrightarrows \mathbb{R}^{n_t} \times \mathbb{Z}^{m_t}$ are $\mathscr{F}_t-$measurable mixed-integer point-to-set mappings. Here, Ξ_t is the support of ξ_t. The cost in the first stage is deterministic and represented by a possibly nonlinear, real-valued function $f_1 : \mathbb{R}^{n_1} \times \mathbb{Z}^{m_1} \to \mathbb{R}$. The cost functions $f_t : \mathbb{R}^{n_t} \times \mathbb{Z}^{m_t} \times \Xi_t \to \mathbb{R}$, $t \in \{2, \dots, T\}$ are $\mathscr{F}_t-$measurable and may be nonlinear.

When a multi-stage stochastic process is considered, all realizations of the process form a scenario tree in the finite distribution case. In this section, we follow the notation used in [37] to represent the scenario tree. Let Ω_t be the set of nodes at stage $t \in \{1, \dots, T\}$. At stage $t = 1$, there is only one node, called as root node and it is represented by v_1. The nodes at stages $t \in \{2, \dots, T\}$ represent elementary events in $\mathscr{F}_t$, that is $\mathscr{F}_t = \sigma(\Omega_t)$, a sigma algebra on Ω_t.

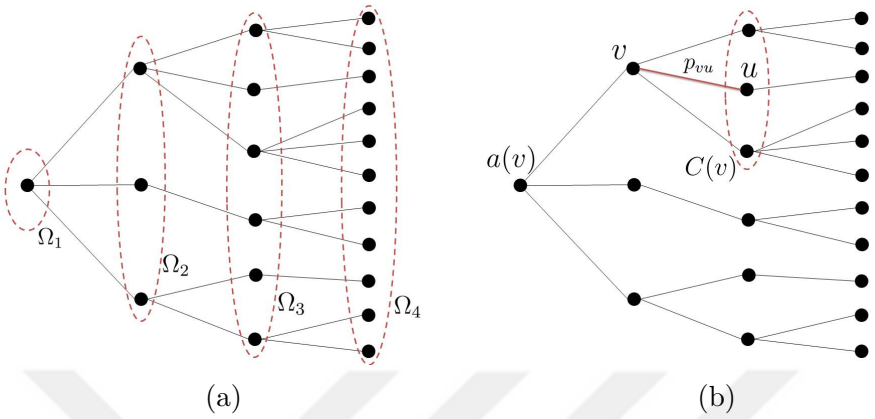


Figure 3.1: An example of four-stage scenario tree. (a) $\Omega_1, \Omega_2, \Omega_3$ and Ω_4 are the set of nodes at stages 1, 2, 3 and 4, respectively. (b) $C(v)$ is the set of children nodes of node v, $a(v)$ is the ancestor node of node v and p_{vu} is the conditional probability of node u given v.

The set Ω_T corresponds to all possible scenarios, that is $\Omega_T = \Omega$. Each node $v \in \Omega_t, t \in \{2, \ldots, T\}$ has a unique ancestor at stage $t-1$ and this ancestor node is called as $a(v)$. Also, each node $v \in \Omega_t, t \in \{1, \ldots, T-1\}$ has a set of children nodes $C(v)$ such that $C(v) = \{u \in \Omega_{t+1} : a(u) = v\}$. The probability measure P can be specified by conditional probabilities

$$p_{vu} := P[u|v], \quad v \in \Omega_t, u \in C(v), \ t \in \{1, \ldots, T-1\},$$

and probability of a scenario $\omega \in \Omega_T$ can be computed as

$$p_\omega = p_{v_1 v_2} p_{v_2 v_3} \cdots p_{v_{t-1}\omega},$$

where $v_1, v_2, \ldots, v_{t-1}, \omega$ is the unique path from root node v_1 to node ω.

The notation mentioned above is depicted in Figure 3.1 for a four-stage scenario tree.

The following fact is known as *dual representation of coherent measures of risk* (see, for example, [18]) and will be used in our results: if $\rho(\cdot)$ is a coherent measure of risk, then, under some mild assumptions, for every random variable

$Z \in \mathcal{Z}$,

$$\rho(Z) = \max_{\mu \in \mathcal{A}} \langle \mu, Z \rangle, \tag{3.2}$$

where $\mathcal{A} \subseteq \mathcal{Z}^*$ is a compact and convex set. We call this set as the dual set of the risk measure $\rho(\cdot)$. A coherent measure of risk can be characterized via its dual set.

In [37], the dual representation of coherent risk measures is extended to dynamic measures of risk. If $\varrho_{1,T}(\cdot)$ is a dynamic risk measure given as in (2.13), then for every sequence of random variables $\{Z_t \in \mathcal{Z}_t\}_{t=1}^T$,

$$\varrho_{1,T}(Z_1, Z_2, \ldots, Z_T) = \max_{q_T \in \mathcal{Q}_T} \langle q_T, Z_1 + Z_2 + \cdots + Z_T \rangle, \tag{3.3}$$

where

$$\mathcal{Q}_T = \mathcal{A}_{t-1} \circ \cdots \circ \mathcal{A}_2 \circ \mathcal{A}_1, \tag{3.4}$$

and $\mathcal{A}_t$, $t \in \{2, \ldots, T\}$ is a convex and compact set used in the dual representation of $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(\cdot)$. The operator "$\circ$" defines convolution of probability measures, that is,

$$(\mu_t \circ q_t)(u) = q_t(a(u))\mu_t(a(u), u), \forall u \in \Omega_{t+1},$$

and

$$\mathcal{A}_t \circ \mathcal{Q}_t = \{\mu_t \circ q_t : q_t \in Q_t, \mu_t \in \mathcal{A}_t\},$$

for all $t \in \{1, 2, \ldots, T-1\}$. Recall that $a(u)$ is the ancestor node of u.

As already mentioned in this section, we use conditional mean-CVaR as one-step conditional risk measure. We first recall the definitions of CVaR and mean-CVaR. Let

$$\mathrm{CVaR}_\alpha(Z) := \inf_{\eta \in \mathbb{R}} \left\{ \eta + \frac{1}{1-\alpha} \mathbb{E}[(Z - \eta)_+] \right\}, \tag{3.5}$$

where $(Z)_+ = \max\{Z, 0\}$ in a component-wise manner and $\alpha \in [0, 1)$ is the level parameter.

Given a level parameter $\alpha \in [0,1)$ and a weight parameter $\epsilon_1 \in [0,1]$, mean-CVaR of $Z \in \mathcal{Z}$ is defined as

$$\rho(Z) := (1-\epsilon_1)\mathbb{E}[Z] + \epsilon_1 \text{CVaR}_\alpha(Z). \tag{3.6}$$

As seen in (3.6), despite CVaR, mean-CVaR risk measure conveys the expected value information of a random variable, as well. As α or ϵ_1 increases, the decision-maker gets more risk-averse. Also note that, the expression in (3.6) can equivalently be represented as the following linear program for finite probability spaces.

$$\begin{aligned} \rho(Z) = \underset{\vartheta,\eta}{\text{minimize}} \quad & (1-\epsilon_1)\sum_{\omega\in\Omega} p_\omega Z_\omega + \epsilon_1\left(\eta + \frac{1}{1-\alpha}\sum_{\omega\in\Omega} p_\omega \vartheta_\omega\right) \\ \text{subject to} \quad & \vartheta_\omega \geq Z_\omega - \eta, \quad \forall \omega \in \Omega \\ & \vartheta_\omega \geq 0, \quad \forall \omega \in \Omega. \end{aligned}$$

When the sample space is finite, the dual representation (3.2) holds for mean-CVaR with the set $\mathcal{A}$ represented as (see, for example, [18]):

$$\mathcal{A} = \left\{\mu \in \mathcal{Z}^* : 1-\epsilon_1 \leq \mu_\omega \leq 1+\epsilon_2, \forall \omega \in \Omega \text{ and } \mathbb{E}[\mu] = 1\right\}, \tag{3.7}$$

where

$$\epsilon_2 := \frac{\alpha}{1-\alpha}\epsilon_1 \geq 0,$$

and $\mathbb{E}[\mu] = \sum_{\omega\in\Omega} p_\omega \mu_\omega$.

For any $Z_{t+1} \in \mathcal{Z}_{t+1}$, the one-step conditional mean-CVaR risk measure $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(Z_{t+1})$ with parameters $\alpha_t \in [0,1)$ and $\epsilon_{1t} \in [0,1]$ and its dual set $\mathcal{A}_t$ are defined similar to (3.6) and (3.7). However, in (3.5), the infimum is over $\eta_t \in \mathcal{Z}_t$ and the expectation operators in (3.5)-(3.7) are replaced with conditional expectations with respect to $\mathscr{F}_t$.

For the remainder of the chapter, we will focus on mean-CVaR risk measure. Hence, we will use $\rho(\cdot)$ to refer to mean-CVaR and $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(\cdot)$, $t \in \{1, 2 \ldots, T-1\}$ to refer to one-step conditional mean-CVaR.

3.2 Bounds

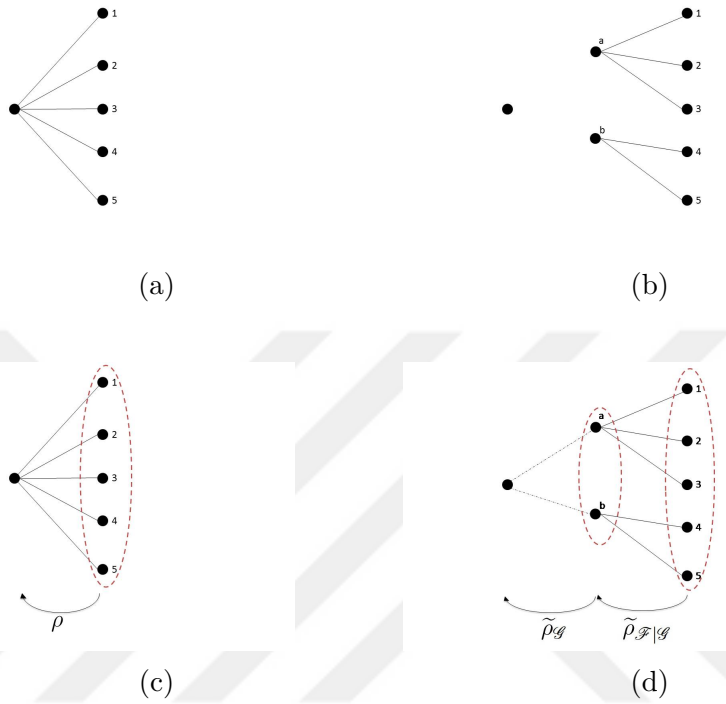
Let $\rho(\cdot)$ be a mean-CVaR risk measure with dual set $\mathcal{A}$. We would like to construct another coherent risk measure $\widetilde{\rho}(\cdot)$ which provides a time consistent lower bound for $\rho(\cdot)$. The risk measure $\widetilde{\rho}(\cdot)$, or equivalently its dual set $\widetilde{\mathcal{A}}$, can be constructed in different ways. When the cardinality of the sample space is large, due to computational concerns, one may think of dealing with subsets of sample space separately and then obtaining a lower bound for $\rho(\cdot)$. For such construction, we need the definition of scenario groups and partition.

A subset of scenarios $S \subseteq \Omega$ is called as a group. Let $\mathcal{S} = \{S_j\}_{j=1}^{J}$ be a collection of groups that forms a partition of Ω, that is, $\bigcup_{j=1}^{J} S_j = \Omega$ and $S_j \bigcap Sj' = \emptyset$ for all $j, j' \in \{1, 2, \ldots, J\}$ such that $j \neq j'$. Note that the groups may not be necessarily disjoint (see, [39]), i.e. $S_j \bigcap Sj' \neq \emptyset$, but for the ease of representation, we partition the sample space into disjoint groups. Let $\mathscr{G}$ be a $\sigma-$algebra generated by partition $\mathcal{S}$ where each group $S_j \in \mathcal{S}, j \in \{1, 2, \ldots, J\}$ corresponds to an elementary event j of $\mathscr{G}$. The probability of an elementary event j is $p_j = \sum_{\omega \in S_j} p_\omega$ which is the total probability of scenarios in S_j. We also define the adjusted probability of each scenario ω as $p_{j\omega} = p_\omega / p_j$ for all $\omega \in S_j$ and $j \in \{1, 2, \ldots, J\}$. Note that, $\mathscr{G}$ is a sub $\sigma-$algebra of $\mathscr{F}$.

Once a partition of the sample space Ω is given, one way to construct $\widetilde{\rho}(\cdot)$ is to define it as a convolution of a coherent risk measure $\widetilde{\rho}_{\mathscr{G}} : \mathcal{L}_\infty(\Omega, \mathscr{G}, P) \to \mathbb{R}$ with dual set $\widetilde{\mathcal{A}}_{\mathscr{G}}$ and a one-step conditional risk measure $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}} : \mathcal{Z} \to \mathcal{L}_\infty(\Omega, \mathscr{G}, P)$ with dual set $\widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}}$. That is, $\widetilde{\rho}(\cdot) = (\widetilde{\rho}_{\mathscr{G}} \circ \widetilde{\rho}_{\mathscr{F}|\mathscr{G}})(\cdot)$, and its dual set is the convolution of the sets $\widetilde{\mathcal{A}}_{\mathscr{G}}$ and $\widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}}$ such that $\widetilde{\mathcal{A}} = \widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}} \circ \widetilde{\mathcal{A}}_{\mathscr{G}}$.

Note that, $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)$ can be represented in terms of $\rho_{S_j}(\cdot), j \in \{1, 2 \ldots, J\}$, that is, $\left[\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)\right]_j = \rho_{S_j}(\cdot)$ (see, for example, [6]) where $\rho_{S_j} : \mathcal{L}_\infty(\Omega, \sigma(S_j), P) \to \mathbb{R}$ is a coherent risk measure and $\sigma(S_j)$ is the $\sigma-$algebra on S_j. Figure 3.2 depicts aforementioned notation for a given partition of a scenario tree with five scenarios.

For mean-CVaR, $\widetilde{\rho}(\cdot)$ or equivalently its dual set $\widetilde{\mathcal{A}}$, can be explicitly stated.



(a) (b) (c) (d)

Figure 3.2: (a) An example partition for a two-stage scenario tree: There are five scenarios $1, 2, 3, 4$, and 5 with probabilities p_1, p_2, p_3, p_4, and p_5, respectively. (b) $\mathcal{S} = \{S_a, S_b\}$ is a partition of Ω where $S_a = \{1, 2, 3\}$ and $S_b = \{4, 5\}$. Nodes a and b correspond to groups S_a and S_b with probabilities $p_a = p_1 + p_2 + p_3$ and $p_b = p_4 + p_5$, respectively. (c) $\rho : \mathcal{Z} \to \mathbb{R}$ is the original risk measure. (d) $\mathscr{G}$ is a sub $\sigma-$algebra of $\mathscr{F}$. $\widetilde{\rho}_{\mathscr{G}} : \mathcal{L}_\infty(\Omega, \mathscr{G}, P) \to \mathbb{R}$ is a coherent risk measure and $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}} : \mathcal{Z} \to \mathcal{L}_\infty(\Omega, \mathscr{G}, P)$ is a one-step conditional risk measure that can be represented via $\rho_{S_a} : \mathcal{L}_\infty(\Omega, \sigma(S_a), P) \to \mathbb{R}$ and $\rho_{S_b} : \mathcal{L}_\infty(\Omega, \sigma(S_b), P) \to \mathbb{R}$ as $[\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)]_a = \rho_{S_a}(\cdot)$ and $[\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)]_b = \rho_{S_b}(\cdot)$.

Let parameters of $\widetilde{\rho}_{\mathscr{G}}$ be $\alpha^1 \in [0, 1)$, $\epsilon_1^1 \in [0, 1]$, and $\epsilon_2^1 = \frac{\alpha^1}{1-\alpha^1}\epsilon_1^1$, and parameters of $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}$ be $\alpha^2 \in [0, 1)$, $\epsilon_1^2 \in [0, 1]$ and $\epsilon_2^2 = \frac{\alpha^2}{1-\alpha^2}\epsilon_1^2$. Consider the convolution $\widetilde{\rho} = \widetilde{\rho}_{\mathscr{G}} \circ \widetilde{\rho}_{\mathscr{F}|\mathscr{G}} : \mathscr{F} \to \mathbb{R}$ and its dual set

$$\begin{aligned}\widetilde{\mathcal{A}} &= \widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}} \circ \widetilde{\mathcal{A}}_{\mathscr{G}} = \{\mu \in \mathcal{Z}^* : \mu = \mu^1 \circ \mu^2, \mu_1 \in \widetilde{\mathcal{A}}_{\mathscr{G}}, \mu_2 \in \widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}}\} \\ &= \{\mu \in \mathcal{Z}^* : \mu = \mu^1 \circ \mu^2, 1 - \epsilon_1^1 \leq \mu_j^1 \leq 1 + \epsilon_2^1, \forall j \in 1, 2 \ldots, J \text{ and } \mathbb{E}[\mu^1] = 1, \\ &\qquad 1 - \epsilon_1^2 \leq \mu_\omega^2 \leq 1 + \epsilon_2^2, \forall \omega \in \Omega \text{ and } \mathbb{E}[\mu^2 | \mathscr{G}] = \mathbf{1}\}, \quad (3.8)\end{aligned}$$

where $\mathbb{E}[\mu^1] = \sum_{j \in \{1,\ldots,J\}} p_j \mu_j^1$, $[\mathbb{E}[\mu^2|\mathscr{G}]]_j = \sum_{\omega \in S_j} p_{j\omega} \mu_\omega^2$ for $j \in \{1, \ldots, J\}$, and $\mathbf{1}$ is a $\mathscr{G}$-measurable random variable that takes value of one in all realizations.

Construction of the set $\widetilde{\mathcal{A}}$ for the example in Figure 3.2 is as follows

$$\widetilde{\mathcal{A}}_{\mathscr{G}} = \Big\{(\mu_a, \mu_b) \in \mathbb{R}^2 : \\ 1 - \epsilon_1^1 \leq \mu_a \leq 1 + \epsilon_2^1, \\ 1 - \epsilon_1^1 \leq \mu_b \leq 1 + \epsilon_2^1, \\ (p_1 + p_2 + p_3)\mu_a + (p_4 + p_5)\mu_b = 1\Big\}.$$

$$\widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}} = \Big\{(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) \in \mathbb{R}^5 : \\ 1 - \epsilon_1^2 \leq \mu_1 \leq 1 + \epsilon_2^2, \\ 1 - \epsilon_1^2 \leq \mu_2 \leq 1 + \epsilon_2^2, \\ 1 - \epsilon_1^2 \leq \mu_3 \leq 1 + \epsilon_2^2, \\ 1 - \epsilon_1^2 \leq \mu_4 \leq 1 + \epsilon_2^2, \\ 1 - \epsilon_1^2 \leq \mu_5 \leq 1 + \epsilon_2^2, \\ \frac{p_1}{p_1 + p_2 + p_3}\mu_1 + \frac{p_2}{p_1 + p_2 + p_3}\mu_2 + \frac{p_3}{p_1 + p_2 + p_3}\mu_3 = 1, \\ \frac{p_4}{p_4 + p_5}\mu_4 + \frac{p_5}{p_4 + p_5}\mu_5 = 1\Big\}.$$

$$\widetilde{\mathcal{A}} = \widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}} \circ \widetilde{\mathcal{A}}_{\mathscr{G}} = \Big\{(\mu_a\mu_1, \mu_a\mu_2, \mu_a\mu_3, \mu_b\mu_4, \mu_b\mu_5) \in \mathbb{R}^5 : \\ (\mu_a, \mu_b) \in \widetilde{\mathcal{A}}_{\mathscr{G}}, (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) \in \widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}}\Big\}.$$

Now, we are ready to prove that a lower bound for mean-CVaR risk measure $\rho(\cdot)$ can be obtained by $\widetilde{\rho}(\cdot) = (\widetilde{\rho}_{\mathscr{G}} \circ \widetilde{\rho}_{\mathscr{F}|\mathscr{G}})(\cdot)$.

Proposition 1. *Let $\rho(\cdot)$ be a mean-CVaR risk measure with parameters $\alpha \in [0, 1)$, $\epsilon_1 \in [0, 1]$, $\epsilon_2 = \frac{\alpha}{1-\alpha}\epsilon_1 \geq 0$, and dual set $\mathcal{A}$. Also let $\widetilde{\rho}(\cdot) = (\widetilde{\rho}_{\mathscr{G}} \circ \widetilde{\rho}_{\mathscr{F}|\mathscr{G}})(\cdot)$ where $\widetilde{\rho}_{\mathscr{G}}$ is a mean-CVaR risk measure with parameters $\alpha^1 \in [0, 1)$, $\epsilon_1^1 \in [0, 1]$, $\epsilon_2^1 = \frac{\alpha^1}{1-\alpha^1}\epsilon_1^1$, and dual set $\widetilde{\mathcal{A}}_{\mathscr{G}}$; and $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}$ is a one-step conditional mean-CVaR risk measure with parameters $\alpha^2 \in [0, 1)$, $\epsilon_1^2 \in [0, 1]$, $\epsilon_2^2 = \frac{\alpha^2}{1-\alpha^2}\epsilon_1^2$, and dual set*

$\widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}}$. *Then,* $\widetilde{\rho}(Z) \leq \rho(Z)$ *for all* $Z \in \mathcal{Z}$ *if*

$$1-\epsilon_1 \leq (1-\epsilon_1^1)(1-\epsilon_1^2) \text{ and } \left(1+\frac{\alpha^1}{1-\alpha^1}\epsilon_1^1\right)\left(1+\frac{\alpha^2}{1-\alpha^2}\epsilon_1^2\right) \leq 1+\frac{\alpha}{1-\alpha}\epsilon_1. \tag{3.9}$$

Proof. Let $\mu \in \widetilde{\mathcal{A}} = \widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}} \circ \widetilde{\mathcal{A}}_{\mathscr{G}}$. Then, from (3.8), there exist $\mu^1 \in \widetilde{\mathcal{A}}_{\mathscr{G}}$ and $\mu^2 \in \widetilde{\mathcal{A}}_{\mathscr{F}|\mathscr{G}}$ such that $\mu = \mu^1 \circ \mu^2$ with $\mathbb{E}[\mu^1] = 1$ and $\mathbb{E}[\mu^2|\mathscr{G}] = \mathbf{1}$. Properties of conditional expectation implies that $\mathbb{E}[\mu] = \mathbb{E}\left[\mathbb{E}\left[\mu|\mathscr{G}\right]\right] = \mathbb{E}\left[\mathbb{E}\left[\mu^1 \circ \mu^2|\mathscr{G}\right]\right] = \mathbb{E}\left[\mu^1 \circ \mathbb{E}\left[\mu^2|\mathscr{G}\right]\right] = \mathbb{E}\left[\mu^1 \circ \mathbf{1}\right] = \mathbb{E}[\mu^1] = 1$.

From the definition of ϵ_2, ϵ_2^1 and ϵ_2^2, second part of (3.9) implies $(1+\epsilon_2^1)(1+\epsilon_2^2) \leq 1+\epsilon_2$. Moreover, by (3.8), $(1-\epsilon_1^1)(1-\epsilon_1^2) \leq \mu_\omega \leq (1+\epsilon_2^1)(1+\epsilon_2^2)$ for all $\omega \in \Omega$. If $1-\epsilon_1 \leq (1-\epsilon_1^1)(1-\epsilon_1^2)$ and $(1+\epsilon_2^1)(1+\epsilon_2^2) \leq 1+\epsilon_2$, then $1-\epsilon_1 \leq \mu_\omega \leq 1+\epsilon_2$, for all $\omega \in \Omega$ which implies, $\mu \in \mathcal{A}$. Since μ is arbitrary, $\widetilde{\mathcal{A}} \subseteq \mathcal{A}$.

For any $Z \in \mathcal{Z}$, let $\widetilde{\rho}(Z) = \max_{\mu\in\widetilde{\mathcal{A}}}\langle\mu, Z\rangle$ and $\mu^* \in \arg\max_{\mu\in\widetilde{\mathcal{A}}}\langle\mu, Z\rangle$. If $\widetilde{\mathcal{A}} \subseteq \mathcal{A}$, then $\mu^* \in \mathcal{A}$ and $\widetilde{\rho}(Z) = \langle\mu^*, Z\rangle \leq \max_{\mu\in\mathcal{A}}\langle\mu, Z\rangle = \rho(Z)$. Since Z is arbitrary, $\widetilde{\rho}(Z) \leq \rho(Z)$ for all $Z \in \mathcal{Z}$. □

Proposition 1 partially extends Theorem 8 and Corollary 6 of [47] to mean-CVaR risk measure. It implies that, under conditions (3.9), $\widetilde{\rho}(\cdot) = (\widetilde{\rho}_{\mathscr{G}} \circ \widetilde{\rho}_{\mathscr{F}|\mathscr{G}})(\cdot)$ is a valid lower bound for $\rho(\cdot)$ for any partition $\mathcal{S}$ of Ω. If $\rho(\cdot)$ is a conditional mean-CVaR risk measure, Proposition 1 still applies. In this case, the expectations in the proof are replaced with corresponding conditional expectations.

3.2.1 Possible Lower Bounds

We have shown that a lower bound for $\rho(\cdot)$ can be obtained by convolutions of mean-CVaR risk measures whose parameters satisfy condition (3.9). Due to Proposition 1, we can generate infinitely many lower bounds. Under the settings on Proposition 1, Table 3.1 presents some special cases of parameters of $\widetilde{\rho}_{\mathscr{G}}(\cdot)$ and $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)$ such that they can be used to obtain a lower bound for $\rho(\cdot)$.

	Parameters of $\widetilde{\rho}_{\mathscr{G}}$			Parameters of $\widetilde{\rho}_{\mathscr{F}\|\mathscr{G}}$		
$\widetilde{\rho}_{\mathscr{G}} \circ \widetilde{\rho}_{\mathscr{F}\|\mathscr{G}}$	ϵ_1^1	ϵ_2^1	α^1	ϵ_1^2	ϵ_2^2	α^2
$\rho_{\mathscr{G}} \circ \mathbb{E}_{\mathscr{F}\|\mathscr{G}}$	ϵ_1	ϵ_2	α	0	0	0
$\rho^s_{\mathscr{G}} \circ \rho^s_{\mathscr{F}\|\mathscr{G}}$	$1-\sqrt{1-\epsilon_1}$	$\sqrt{1+\epsilon_2}-1$	$\frac{\sqrt{1+\epsilon_2}-1}{\sqrt{1+\epsilon_2}-\sqrt{1-\epsilon_1}}$	$1-\sqrt{1-\epsilon_1}$	$\sqrt{1+\epsilon_2}-1$	$\frac{\sqrt{1+\epsilon_2}-1}{\sqrt{1+\epsilon_2}-\sqrt{1-\epsilon_1}}$
$\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}\|\mathscr{G}}$	0	0	0	ϵ_1	ϵ_2	α

Table 3.1: Possible choices of $\widetilde{\rho}_{\mathscr{G}}(\cdot)$ and $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)$ that can be used to obtain lower bound on mean-CVaR risk measure $\rho(\cdot)$.

LB Choice	$\mathcal{S}$	$\mathcal{S}'$
$\rho_{\mathscr{G}} \circ \mathbb{E}_{\mathscr{F}\|\mathscr{G}}$	*3.5*	*2.5*
$\rho^s_{\mathscr{G}} \circ \rho^s_{\mathscr{F}\|\mathscr{G}}$	*3.12*	*3*
$\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}\|\mathscr{G}}$	*3*	*3.5*

Table 3.2: Values of different lower bounds (LB's) for Example 3.

Bounds $\rho_{\mathscr{G}} \circ \mathbb{E}_{\mathscr{F}|\mathscr{G}}$ and $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$ represent the extreme cases where either $\widetilde{\rho}_{\mathscr{G}}(\cdot)$ or $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)$ is the expectation operator. Bound $\rho^s_{\mathscr{G}} \circ \rho^s_{\mathscr{F}|\mathscr{G}}$ is an intermediate case where both $\widetilde{\rho}_{\mathscr{G}}(\cdot)$ and $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)$ have the same parameters, that is, $\alpha^1 = \alpha^2$, $\epsilon_1^1 = \epsilon_1^2$ and $\epsilon_2^1 = \epsilon_2^2$. Under these conditions, in order to construct the largest set $\widetilde{\mathcal{A}}$, the inequalities in (3.9) are forced to hold at equality.

An interesting question is whether one of the possible lower bounds presented above is always preferable among others. Following example reveals that $\rho^s_{\mathscr{G}} \circ \rho^s_{\mathscr{F}|\mathscr{G}}$ is not necessarily the tightest bound among others.

Example 3. *Consider a random variable Z with sample space $\Omega = \{\omega_i\}_{i=1}^4$. All four realizations have equal probabilities, that is, $p_{\omega_i} = 1/4$ for all $i \in \{1,2,3,4\}$. The value that Z takes under scenario ω_i is i, that is, $Z_{\omega_i} = i$ for $i \in \{1,2,3,4\}$.*

Let $\epsilon_1 = 1$ and $\alpha = 0.5$, then (3.6) reduces to CVaR value at $\alpha = 0.5$ and then $\rho(Z) = 3.5$.

Two different partitions of scenarios are $\mathcal{S} = \{\{\omega_1, \omega_2\}, \{\omega_3, \omega_4\}\}$ and $\mathcal{S}' = \{\{\omega_1, \omega_4\}, \{\omega_2, \omega_3\}\}$. Values of the three bounds for partitions $\mathcal{S}$ and $\mathcal{S}'$ are given in Table 3.2.

As seen in Table 3.2, the tightest bounds for partitions $\mathcal{S}$ and $\mathcal{S}'$ are bounds $\rho_{\mathscr{G}} \circ \mathbb{E}_{\mathscr{F}|\mathscr{G}}$ and $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$, respectively. Another observation is the fact that $\rho^{\mathcal{S}}_{\mathscr{G}} \circ \rho^{\mathcal{S}}_{\mathscr{F}|\mathscr{G}}$ is not necessarily the tightest bound among others. In Example 3, although either $\rho_{\mathscr{G}} \circ \mathbb{E}_{\mathscr{F}|\mathscr{G}}$ or $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$ can be the tightest bound among others under different scenario partitions, the computational experiments in Section 3.3 reveal that $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$ is the most promising lower bound choice.

Although in [3] it is shown that, under some assumptions, the lower bound $\rho_{\mathscr{G}} \circ \mathbb{E}_{\mathscr{F}|\mathscr{G}}$ can be extended to any coherent risk measures, the other bounds provided in Table 3.1 may not be applicable for all coherent risk measures. Example 4 reveals that $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$ is not necessarily a valid lower bound for an arbitrary coherent risk measure.

Example 4. *Consider a random variable Z that takes values $Z_{\omega_1} = 100$, $Z_{\omega_2} = 0$, $Z_{\omega_3} = 1$ and $Z_{\omega_4} = 500$ with probabilities 0.3, 0.2, 0.4 and 0.1, respectively. Let $\rho_{\mathscr{F}|\mathscr{G}}(\cdot)$ be the one-step conditional first-order mean semi-deviation with $\lambda = 0.5$ in (2.9). For partition $\mathcal{S} = \{\{\omega_1, \omega_2\}, \{\omega_3, \omega_4\}\}$, $\rho(Z) = 104.32$ but $(\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}})(Z) = 106.36$.*

Therefore, $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$ is not necessarily a valid lower bound for all coherent risk measures.

3.2.2 Lower Bound for Optimization Problem

In this section, we extend the lower bound proposed in Proposition 1 to a risk-averse mixed-integer multi-stage stochastic problem with an objective of dynamic mean-CVaR risk measure. As discussed in Chapter 2, the problem (3.1) can be written as a DP problem,

$$\textbf{(P)} \quad \min_{x_1 \in \mathcal{X}_1} \quad f_1(x_1) + \rho(Q(x_1, \xi)), \tag{3.10}$$

where

$$Q(x_1, \xi) = \min_{x_t \in \mathcal{X}_t, t \in \{2,\dots,T\}} \varrho_{2,T}\left(f_2(x_2, \xi_2), \dots, f_T(x_T, \xi_T)\right), \tag{3.11}$$

$\xi = \{\xi_t\}_{t=2}^T$, $\rho(\cdot)$ is a mean-CVaR risk measure with parameters $\alpha \in [0,1)$ and $\epsilon_1 \in [0,1]$, and $\varrho_{2,T}(\cdot)$ is a dynamic mean-CVaR risk measure. Let x_1^* and z^* be an optimal first stage solution and the optimal value of **(P)**, respectively.

Recall the partition $\mathcal{S} = \{S_j\}_{j=1}^J$ of Ω and sigma algebra $\mathscr{G}$ induced by this partition. Then, the j^{th} group subproblem is just problem **(P)** with sample space S_j and adjusted probabilities $p_{j\omega}, \omega \in S_j$. Additionally, the risk measure $\rho(\cdot)$ in (3.10) is replaced by $\rho_{S_j}(\cdot)$. For $j \in \{1, 2, \ldots, J\}$, let z^j be the optimal value of j^{th} group subproblem. Also let Z_{LB} be a $\mathscr{G}$-measurable random variable that takes value of z^j with probability $p_j = \sum_{\omega \in S_j} p_\omega$.

In Theorem 1, we show that a lower bound for risk-averse mixed-integer multi-stage stochastic problem **(P)** can be obtained by using optimal values of group subproblems.

Theorem 1. *Let $\widetilde{\rho}_{\mathscr{G}} : \mathcal{L}_\infty(\Omega, \mathscr{G}, P) \to \mathbb{R}$ be a mean-CVaR risk measure with parameters $\alpha^1 \in [0,1)$ and $\epsilon_1^1 \in [0,1]$; and $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}} : \mathcal{L}_\infty(\Omega, \mathscr{F}, P) \to \mathcal{L}_\infty(\Omega, \mathscr{G}, P)$ be a conditional mean-CVaR risk measure with parameters $\alpha^2 \in [0,1)$ and $\epsilon_1^2 \in [0,1]$ satisfying $1 - \epsilon_1 \leq (1-\epsilon_1^1)(1-\epsilon_1^2)$ and $\left(1 + \frac{\alpha^1}{1-\alpha^1}\epsilon_1^1\right)\left(1 + \frac{\alpha^2}{1-\alpha^2}\epsilon_1^2\right) \leq 1 + \frac{\alpha}{1-\alpha}\epsilon_1$. Then, $z^* \geq \widetilde{\rho}_{\mathscr{G}}(Z_{LB})$.*

Proof. Recall that x_1^* is an optimal first stage solution of **(P)**. Note that, it is a feasible first stage solution for each group subproblem. By optimality of each group subproblem, we have

$$f_1(x_1^*) + \rho_{S_j}(Q(x_1^*, \xi)) \geq z^j, \quad \forall j \in \{1, \ldots, J\}$$

and

$$f_1(x_1^*) + \widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(Q(x_1^*, \xi)) \succeq Z_{LB}. \tag{3.12}$$

The values on both sides of inequality (3.12) are $\mathscr{G}-$measurable. Since, $\rho_{\mathscr{G}}(\cdot)$ is a coherent risk measure and it satisfies monotonicity axiom (A2), we get

$$\widetilde{\rho}_{\mathscr{G}}(f_1(x_1^*) + \widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(Q(x_1^*, \xi))) \geq \widetilde{\rho}_{\mathscr{G}}(Z_{LB}). \tag{3.13}$$

Note that, $f_1(x_1^*)$ is an $\mathscr{F}-$measurable cost. Since $\mathscr{G}$ is a sub $\sigma-$algebra of $\mathscr{F}$, $f_1(x_1^*)$ is $\mathscr{G}-$measurable, as well. Applying translational equivariance axiom (A3) to the left hand side of (3.13), we get

$$\widetilde{\rho}_{\mathscr{G}}(\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(f_1(x_1^*) + Q(x_1^*, \xi))) \geq \widetilde{\rho}_{\mathscr{G}}(Z_{LB}). \tag{3.14}$$

Since conditions in (3.9) are satisfied, we can apply Proposition 1 to the left hand side of inequality (3.14) and obtain:

$$\rho(f_1(x_1^*) + Q(x_1^*, \xi)) \geq \widetilde{\rho}_{\mathscr{G}}(Z_{LB}).$$

Finally, using translational equivariance axiom (A3), we get

$$z^* = f_1(x_1^*) + \rho(Q(x_1^*, \xi)) \geq \widetilde{\rho}_{\mathscr{G}}(Z_{LB}).$$

□

Theorem 1 implies that a lower bound on the optimal value of **(P)** can be obtained by solving group subproblems and then applying $\widetilde{\rho}_{\mathscr{G}}(\cdot)$ to the optimal values of these group subproblems. Since group subproblems include smaller number of scenarios compared to the original problem, they are computationally less challenging. Moreover, applying $\widetilde{\rho}_{\mathscr{G}}(\cdot)$ to the optimal values of group subproblems requires negligible computational effort, since it is only the calculation of value of a risk measure $\widetilde{\rho}_{\mathscr{G}}(\cdot)$ for a given random cost.

Although the dynamic risk measure (2.13) is widely used in the literature, there are other risk measures that can be used to evaluate the risk of a sequence of random variables. We show that our approach can also be applied to the risk-averse mixed-integer multi-stage stochastic problems with different dynamic extensions of mean-CVaR.

3.2.3 Extension to Other Dynamic Measures of Risk

Some examples of dynamic risk measures apart from (2.13) are multiperiod mean-CVaR and sum of mean-CVaR (see, [48] and [49], respectively). For a sequence of random variables $Z_t \in \mathcal{Z}_t, t \in \{1, \ldots, T\}$ adopted to the filtration $\mathscr{F}_t, t \in \{1, \ldots, T\}$, multiperiod mean-CVaR is defined as

$$\rho^{multi}(\{Z_t\}_{t=2}^T) = \sum_{t=2}^{T} \lambda_t \mathbb{E}[\rho_{\mathscr{F}_t|\mathscr{F}_{t-1}}(Z_t)], \tag{3.15}$$

and sum of mean-CVaR is represented as

$$\rho^{sum}(\{Z_t\}_{t=2}^T) = \sum_{t=2}^{T} \lambda_t \rho_t(Z_t), \tag{3.16}$$

with $\sum_{t=2}^{T} \lambda_t = 1, \lambda_t \geq 0$ for $t \in \{2, 3, \ldots, T\}$.

Our approach is also applicable for the case where the risk measure is applied to whole scenario cost as a time inconsistent objective function, that is,

$$\rho^{whole}(\{Z_t\}_{t=1}^T) = \rho(Z_1 + Z_2 + \ldots + Z_T). \tag{3.17}$$

Although the risk measure (3.17) can be applied to a sequence of random variables, it is not a dynamic measure of risk.

The risk measure defined in (3.15) is a time consistent dynamic measure of risk whereas the risk measures (3.16) and (3.17) are not time consistent.

In the following three propositions, we show that a lower bound for these three risk measures can be obtained by scenario grouping. Therefore, our approach is still valid for Problem **(P)** with an objective of one of these risk measures.

Consider an arbitrary sequence of random variables $Z_t \in \mathcal{Z}_t, t \in \{1, \ldots, T\}$ adopted to the filtration $\mathscr{F}_t, t \in \{1, \ldots, T\}$. To avoid notational ambiguity, expectation operators and risk measures are given without reference sigma algebras.

Proposition 2. *For a multiperiod mean-CVaR risk measure $\rho^{multi}(\cdot)$ as defined in (3.15), $\mathbb{E} \circ \rho^{multi}(\cdot)$ is a valid lower bound.*

Proof. If multiperiod mean-CVaR risk measure (3.15) is applied to the sequence $Z_t \in \mathcal{Z}_t, t \in \{1, \ldots, T\}$, then

$$\rho^{multi}(\{Z_t\}_{t=2}^T) = \sum_{t=2}^{T} \lambda_t \mathbb{E}\left[\rho_{\mathscr{F}_t|\mathscr{F}_{t-1}}(Z_t)\right].$$

Since $\rho_{\mathscr{F}_t|\mathscr{F}_{t-1}}(\cdot)$ is a conditional mean-CVaR risk measure, the lower bound $\mathbb{E} \circ \rho_{\mathscr{F}_t|\mathscr{F}_{t-1}}(\cdot)$ applies for $t \in \{2, 3, \ldots, T\}$. Then,

$$\rho^{multi}(\{Z_t\}_{t=2}^T) \geq \sum_{t=2}^{T} \lambda_t \mathbb{E}\left[\mathbb{E}\left[\rho_{\mathscr{F}_t|\mathscr{F}_{t-1}}(Z_t)\right]\right].$$

Since expectation is a linear operator, we get

$$\rho^{multi}(\{Z_t\}_{t=2}^T) \geq \mathbb{E}\left[\sum_{t=2}^{T} \lambda_t \mathbb{E}[\rho_{\mathscr{F}_t|\mathscr{F}_{t-1}}(Z_t)]\right],$$

or equivalently,

$$\rho^{multi}(\{Z_t\}_{t=2}^T) \geq \mathbb{E}\left[\rho^{multi}(\{Z_t\}_{t=2}^T)\right].$$

Since the sequence $Z_t \in \mathcal{Z}_t, t \in \{1, \ldots, T\}$ is arbitrary, the desired result follows. □

Proposition 3. *For a sum of mean-CVaR risk measure $\rho^{sum}(\cdot)$ as defined in (3.16), $\mathbb{E} \circ \rho^{sum}(\cdot)$ is a valid lower bound.*

Proof. If sum of mean-CVaR risk measure (3.16) is applied to the sequence $Z_t \in \mathcal{Z}_t, t \in \{1, \ldots, T\}$, then

$$\rho^{sum}(\{Z_t\}_{t=2}^T) = \sum_{t=2}^{T} \lambda_t \rho_t(Z_t).$$

Similarly, $\mathbb{E} \circ \rho_t(\cdot)$ applies for $t \in \{2, 3, \ldots, T\}$. Then,

$$\rho^{sum}(\{Z_t\}_{t=2}^T) \geq \sum_{t=2}^{T} \lambda_t \mathbb{E}\left[\rho_t(Z_t)\right],$$

and

$$\rho^{sum}(\{Z_t\}_{t=2}^T) \geq \mathbb{E}\left[\sum_{t=2}^{T} \lambda_t \rho_t(Z_t)\right],$$

or equivalently,

$$\rho^{sum}(\{Z_t\}_{t=2}^T) \geq \mathbb{E}\left[\rho^{sum}(\{Z_t\}_{t=2}^T)\right].$$

Since the sequence $Z_t \in \mathcal{Z}_t, t \in \{1, \ldots, T\}$ is arbitrary, the desired result follows. □

Proposition 4. *For the risk measure $\rho^{whole}(\cdot)$ as defined in (3.17), $\widetilde{\rho}_{\mathscr{G}} \circ \widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)$ is a valid lower bound if parameters of $\widetilde{\rho}_{\mathscr{G}}(\cdot)$ and $\widetilde{\rho}_{\mathscr{F}|\mathscr{G}}(\cdot)$ satisfy conditions in (3.9).*

Proof. Follows from Proposition 1. □

As shown above, our proposed lower bound is quite general and can be applied to other dynamic mean-CVaR measures.

3.2.4 Upper Bound for Optimization Problem

Obtaining an upper bound, or equivalently finding a feasible solution of a minimization problem, is crucial for the instances where an optimal solution is not available. A good quality feasible solution gives the decision maker an action to be taken and measures the quality of obtained lower bound when an optimal solution is not available.

An upper bound for the optimal value of **(P)** can be obtained by using optimal solutions of group subproblems. Once j^{th} group subproblem is solved, an optimal solution of it, namely x^j, is obtained. Let UB_j be the optimal value of **(P)** where (some of) the variables appearing in j^{th} group subproblem are set to x^j. We call

this problem as restricted problem. Since some of the problem variables are fixed, solving the restricted problem is easier than the original one and the resulting scenario tree can become decomposable.

If the restricted problem does not have a feasible solution, then corresponding upper bound UB_j is set to infinity. The best available upper bound UB is obtained by taking minimum of UB_j values over all $j \in \{1, \ldots, J\}$, that is,

$$UB = \min_{j \in \{1,\ldots,J\}} UB_j. \tag{3.18}$$

In Algorithm 1, we present how group subproblem approach can be used to obtain lower and upper bounds for the multi-stage risk-averse mixed-integer problem **(P)** with dynamic mean-CVaR objective. The algorithm can be easily adopted to the other risk measures given in Section 3.2.3.

3.3 Computational Experiments

In this section, we conduct our numerical experiments on a multi-stage lot sizing problem studied in [50]. All computational experiments are performed on an Intel(R) Core(TM) i7-4790 CPU@3.60 GHz computer with 8.00 GB of RAM with Java 1.8.0.31 and IBM ILOG CPLEX 12.6. We first introduce risk-averse multi-stage lot sizing problem (RAMLSP) with dynamic mean-CVaR defined in (2.13). Then, we compare the results obtained via usage of different scenario partition strategies and lower bound choices. We also compare the proposed algorithm and CPLEX in terms of solution quality and required CPU time.

Algorithm 1 Lower and upper bounds for **(P)**

Input: A risk-averse mixed-integer multi-stage stochastic problem **(P)** and a partition $\mathcal{S} = \{S_j\}_{j=1}^{J}$ of sample space Ω.
Initialize: $LB \leftarrow -\infty$ and $UB \leftarrow +\infty$

`Lower Bounding:`
for all $j \in \{1, 2, \ldots, J\}$ **do**
  Solve the j^{th} group subproblem.
  $x^j \leftarrow$ an optimal solution of j^{th} group subproblem
  $z^j \leftarrow$ optimal value of j^{th} group subproblem
end for
Let Z_{LB} be a random variable that takes value z^j with probability $p_j = \sum_{\omega \in S_j} p_\omega$
$LB \leftarrow \widetilde{\rho}_{\mathscr{G}}(Z_{LB})$

`Upper Bounding:`
for all $j \in \{1, 2, \ldots, J\}$ **do**
  $UB_j \leftarrow$ the optimal value of the original problem with the additional constraint where (some of) the variables appearing in j^{th} group subproblem are set to x^j.
end for
$UB \leftarrow \min_{j \in \{1,2\ldots,J\}} UB_j$

Return: LB and UB

3.3.1 Risk-averse Multi-stage Lot Sizing Problem with Mean-CVaR

The objective of RAMLSP is to minimize the dynamic mean-CVaR risk measure over T periods subject to demand satisfaction and capacity constraints. RAMLSP-T-r represents an RAMLSP instance with T stages in which random components can take r different values at each stage. Therefore, total number of scenarios in an RAMLSP-T-r instance is r^{T-1}. We generate random test instances as in [50]. The same setting of the parameters is also used by [39], that is, $h_{tu} \sim U[0, 10]$, $\alpha_{tu} \sim U[3.2, 4.8]E[h]$, $\beta_{tu} \sim U[320, 480]E[h]$, $d_{tu} \sim U[0, 100]$ and $M_{tu} \sim U[40T, 60T]$, where $U[a, b]$ represents uniform distribution between a and b.

Using the scenario tree representation given in Section 3.1 RAMLSP can be stated as follows:

$$\textbf{(RAMLSP)} \tag{3.19}$$

$$\text{minimize} \quad Z_1 + \rho_{\mathscr{F}_2|\mathscr{F}_1}\left(Z_2 + \rho_{\mathscr{F}_3|\mathscr{F}_2}\left(Z_3 + \ldots + \rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}(Z_T)\ldots\right)\right) \tag{3.20}$$

$$\text{subject to} \quad Z_{tu} = \alpha_{tu}x_{tu} + \beta_{tu}y_{tu} + h_{tu}s_{tu}, \quad \forall u \in \Omega_t,\ t \in \{1,\ldots,T\} \tag{3.21}$$

$$s_{(t-1)a(u)} + x_{tu} = d_{tu} + s_{tu}, \quad \forall u \in \Omega_t,\ t \in \{1,\ldots,T\} \tag{3.22}$$

$$x_{tu} \leq M_{tu}y_{tu}, \quad \forall u \in \Omega_t,\ t \in \{1,\ldots,T\} \tag{3.23}$$

$$x_{tu}, s_{tu} \geq 0 \text{ and integer},\ y_{tu} \in \{0,1\}, \quad \forall u \in \Omega_t,\ t \in \{1,\ldots,T\} \tag{3.24}$$

$$s_{0a(v_1)} = 0.$$

Here x_{tu} is the production level, y_{tu} is the setup indicator and s_{tu} is the inventory level variables at node $u \in \Omega_t$ in period $t \in \{1,\ldots,T\}$. $\alpha_{tu}, \beta_{tu}, h_{tu}, d_{tu}$ and M_{tu} denote unit production cost, setup cost, inventory holding cost, demand and production capacity parameters, respectively. Z_1 is the sum of deterministic production, setup and inventory holding costs incurred in the first stage. Similarly, Z_{tu} is the cost incurred at node $u \in \Omega_t$ at stage $t \in \{2,\ldots,T\}$. Z_t represents the random variable that takes values of $Z_{tu}, u \in \Omega_t$ with respective probabilities. The objective (3.20) is the dynamic risk value over the planning horizon. Constraints (3.21) calculate the cost incurred at each node of the scenario tree. Constraints (3.22) and (3.23) are inventory balance and capacity constraints, respectively. Constraints (3.24) are domain constraints. Unlike [50] and [39], we assume that production and inventory levels are required to be integer valued. Although this assumption increases the problem complexity, we have a more realistic representation to evaluate the performance of the algorithm. In order to linearize **RAMLSP**, the linearization of mean-CVaR presented in Section 3.1 is used.

For the computational experiments, we use three different values of weight parameter $\epsilon_1 \in \{0.8, 0.5, 0.3\}$ and level parameter $\alpha \in \{0.9, 0.8, 0.7\}$ of mean-CVaR. Therefore, we have nine different risk-aversion settings.

3.3.2 Choices of Scenario Partitions and Lower Bounds

As seen in Example 3, the value of each lower bound highly depends on chosen scenario partition. We consider four possible scenario partition strategies obtained by grouping the scenarios in different ways, namely *index1*, *index2*, *similar* and *different*. For each strategy, we can also specify the number of scenarios in each group as a function of the number of scenarios $|\Omega|$ and the number of groups J. Let $a\%b$ be the remainder after the division of $a \in \mathbb{R}$ by $b \in \mathbb{R}$, $\lceil \cdot \rceil$ be the ceiling function, and $\lfloor \cdot \rfloor$ be the floor function. Then, each scenario grouping strategy yields a scenario partition that has J groups, where $|\Omega|\%J$ groups have cardinality $\lceil |\Omega|/J \rceil$ and $J - (|\Omega|\%J)$ groups have cardinality $\lfloor |\Omega|/J \rfloor$. For example, if $|\Omega| = 32$ and $J = 5$ then the cardinality of two groups will be seven and the other three groups will have cardinality of six.

Partition strategies *index1* and *index2* are based on the structure of scenario tree. In *index1*, the last stage nodes sharing the highest number of common nodes are placed into the same group. On the other hand, *index2* is obtained by placing the last stage nodes sharing the least number of common nodes into the same group.

If a priori information on the cost of each single scenario under an optimal solution is available, the groups can also be obtained with respect to similarity and diversity of individual scenarios. Since this information is not available before solving the original problem, the deterministic version of the original problem apriori can be solved for each scenario separately, and the corresponding single scenario costs can be used to obtain two different scenario partition strategies named as *similar* and *different*. Note that, for both strategies, an additional computational effort is required to obtain single scenario costs.

In strategy *similar*, we assign $\lceil |\Omega|/J \rceil$ scenarios that have the largest single scenario costs to the first group. Then, depending on the cardinality of the second group, $\lceil |\Omega|/J \rceil$ or $\lfloor |\Omega|/J \rfloor$ scenarios that have the second largest single scenario costs are assigned to the second group, and so on.

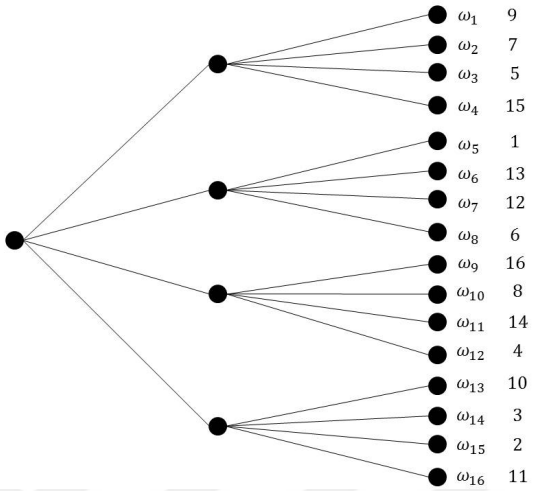


Figure 3.3: An example of three-stage scenario tree with 16 scenarios.

In partition strategy *different*, each scenario is assigned to one of J groups by assigning the scenario with the largest single scenario cost to the first group, the scenario with the second largest single scenario cost to the second group, and so on. This assignment process returns to the first group after assigning the first J scenarios and the process restarts. It is ended after all scenarios are placed in a group.

With respect to single scenario cost values, in strategy *similar*, the dispersion within each group is low, however, the dispersion between the groups is high. On the other hand, in partition strategy *different*, the dispersion within each group is high.

Example 5. *Figure 3.3 depicts the scenario tree for an RAMLSP-3-4 instance where the numbers near the scenarios indicate the cost of each individual scenario. The scenarios can be ordered as* ω_9, ω_4, ω_{11}, ω_6, ω_7, ω_{16}, ω_{13}, ω_1, ω_{10}, ω_2, ω_8, ω_3, ω_{12}, ω_{14}, ω_{15}, ω_5 *where the individual scenario costs decrease moving through from* ω_9 *to* ω_5. *Table 3.3 presents different scenario partition strategies for this scenario tree.*

In order to observe the quality of bounds obtained by different scenario partition strategies and lower bound choices, the proposed algorithm is applied to five

Partition Strategy	S_1	S_2	S_3	S_4
index1	$\{\omega_1, \omega_2, \omega_3, \omega_4\}$	$\{\omega_5, \omega_6, \omega_7, \omega_8\}$	$\{\omega_9, \omega_{10}, \omega_{11}, \omega_{12}\}$	$\{\omega_{13}, \omega_{14}, \omega_{15}, \omega_{16}\}$
index2	$\{\omega_1, \omega_5, \omega_9, \omega_{13}\}$	$\{\omega_2, \omega_6, \omega_{10}, \omega_{14}\}$	$\{\omega_3, \omega_7, \omega_{11}, \omega_{15}\}$	$\{\omega_4, \omega_8, \omega_{12}, \omega_{16}\}$
similar	$\{\omega_9, \omega_4, \omega_{11}, \omega_6\}$	$\{\omega_7, \omega_{16}, \omega_{13}, \omega_1\}$	$\{\omega_{10}, \omega_2, \omega_8, \omega_3\}$	$\{\omega_{12}, \omega_{14}, \omega_{15}, \omega_5\}$
different	$\{\omega_9, \omega_7, \omega_{10}, \omega_{12}\}$	$\{\omega_4, \omega_{16}, \omega_2, \omega_{14}\}$	$\{\omega_{11}, \omega_{13}, \omega_8, \omega_{15}\}$	$\{\omega_6, \omega_1, \omega_3, \omega_5\}$

Table 3.3: Different scenario partitions $\mathcal{S} = \{S_1, S_2, S_3, S_4\}$ for the example scenario tree in Figure 3.3.

RAMLSP-3-30 instances generated via different random seeds. Total number of scenarios is 900. We consider the number of groups as $J \in \{2, 4, 10\}$, and hence each group subproblem includes 450, 225, and 90 scenarios, for the respective value of J. While obtaining upper bounds, optimal production decisions of group subproblems are fixed in the restricted problems. As noted before, for strategies *similar* and *different*, single scenario costs are required. The CPU time needed to obtain these values are also included in the running time of the algorithm. In order to measure the quality of lower and upper bounds, an optimality gap information $Gap(\%) = 100\,((UB - LB)/UB)$ is used. All running times are reported in seconds. The results are presented in Table 3.4, where the *Gap* and *Time* values are the average values of five randomly generated instances.

			J = 2								J = 4								J = 10							
			index1		*index2*		*similar*		*different*		*index1*		*index2*		*similar*		*different*		*index1*		*index2*		*similar*		*different*	
LB	α	ϵ_1	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
$\rho_{\mathscr{G}} \circ \mathbb{E}_{\mathscr{F}\|\mathscr{G}}$	0.9	0.8	10.86%	7.1	12.45%	16.5	9.34%	11.1	12.54%	13.4	9.45%	2.0	13.07%	6.9	7.71%	2.1	13.71%	6.4	4.26%	7.8	18.54%	9.7	5.10%	6.6	16.99%	6.0
		0.5	7.64%	24.5	8.81%	24.8	7.92%	29.6	8.67%	23.9	6.85%	3.1	9.36%	8.6	7.97%	2.5	9.78%	8.5	3.63%	9.7	11.93%	8.6	7.01%	7.2	12.73%	5.4
		0.3	5.32%	18.2	5.79%	52.6	6.56%	95.3	5.69%	48.2	4.91%	4.1	6.33%	13.4	8.29%	3.7	6.73%	11.7	3.01%	11.4	8.34%	7.9	8.32%	7.3	8.28%	7.3
	0.8	0.8	9.61%	13.2	10.82%	21.9	7.16%	14.2	10.97%	24.0	8.14%	3.3	11.69%	9.6	5.32%	3.7	11.58%	10.1	3.46%	8.2	15.55%	10.3	6.89%	8.9	14.43%	5.5
		0.5	6.57%	22.0	7.20%	55.0	6.28%	116.0	7.40%	31.9	5.81%	4.0	7.80%	11.2	6.05%	3.3	8.15%	11.4	3.00%	9.8	9.89%	8.8	7.83%	7.2	10.51%	5.7
		0.3	4.44%	26.6	4.75%	88.6	5.46%	118.9	4.88%	99.2	4.12%	5.0	5.21%	14.9	7.01%	4.8	5.56%	13.8	2.59%	12.2	6.74%	8.0	8.68%	7.5	6.71%	7.5
	0.7	0.8	8.86%	21.2	10.11%	74.9	5.35%	24.2	10.10%	44.4	7.60%	4.0	10.49%	12.0	5.33%	4.5	10.51%	12.6	3.85%	7.7	13.09%	9.1	7.82%	7.6	12.70%	7.1
		0.5	5.90%	31.5	6.56%	73.7	5.11%	74.2	6.62%	89.7	5.24%	4.3	6.98%	16.3	6.31%	4.2	7.35%	14.9	3.08%	12.1	8.22%	8.5	8.45%	7.3	9.17%	7.5
		0.3	3.95%	36.4	4.27%	114.0	4.73%	144.6	4.29%	139.9	3.70%	5.6	4.50%	17.0	7.08%	5.4	4.80%	17.2	2.48%	14.5	5.50%	8.5	8.91%	7.8	5.81%	10.6
$\rho^s_{\mathscr{G}} \circ \rho^s_{\mathscr{F}\|\mathscr{G}}$	0.9	0.8	4.52%	7.2	5.52%	14.8	7.57%	8.3	5.10%	9.6	4.38%	1.6	5.99%	6.1	9.83%	1.8	6.03%	6.3	4.20%	6.7	8.95%	4.4	14.27%	6.8	8.76%	5.2
		0.5	3.99%	10.6	4.43%	18.5	7.80%	16.6	4.23%	15.5	3.86%	2.3	4.91%	6.9	10.97%	1.7	5.13%	6.2	4.04%	9.9	7.18%	4.6	13.95%	7.3	7.67%	4.1
		0.3	3.08%	20.4	3.14%	21.5	6.57%	66.6	3.02%	20.1	3.04%	4.0	3.56%	7.4	10.47%	3.2	3.79%	10.6	3.38%	10.5	5.31%	6.0	12.83%	7.4	5.18%	4.3
	0.8	0.8	3.67%	12.5	4.45%	18.9	6.16%	17.3	4.40%	19.9	3.66%	4.1	5.06%	8.5	10.85%	2.2	5.06%	8.9	3.35%	8.5	7.58%	5.5	14.81%	8.7	7.41%	3.8
		0.5	3.30%	12.9	3.52%	22.0	6.68%	34.7	3.40%	59.3	3.12%	3.7	4.01%	10.2	9.91%	2.8	4.09%	9.6	3.09%	9.2	5.94%	5.4	13.86%	7.4	5.98%	4.8
		0.3	2.46%	21.1	2.49%	60.9	5.83%	166.9	2.41%	44.2	2.44%	4.2	2.85%	12.3	9.28%	4.1	2.98%	12.8	2.67%	10.3	4.26%	7.0	12.41%	7.6	4.07%	5.6
	0.7	0.8	3.37%	21.0	4.40%	19.8	5.10%	22.0	4.22%	28.6	3.39%	4.5	4.68%	11.1	10.73%	4.7	4.90%	8.5	3.80%	7.8	6.18%	5.1	15.10%	7.7	6.53%	3.8
		0.5	2.94%	25.1	3.26%	46.0	6.09%	52.5	3.10%	47.1	2.82%	6.1	3.62%	14.7	10.17%	3.4	3.63%	13.7	3.06%	10.2	4.72%	5.0	13.80%	7.3	5.12%	4.9
		0.3	2.30%	21.9	2.19%	73.9	5.57%	169.8	2.10%	95.8	2.26%	7.1	2.49%	15.2	9.50%	3.5	2.57%	16.4	2.39%	12.4	3.36%	7.0	12.46%	7.9	3.50%	8.8
$\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}\|\mathscr{G}}$	0.9	0.8	1.15%	6.4	0.65%	5.6	11.78%	6.8	**0.36%**	5.8	1.97%	2.9	1.10%	3.8	20.35%	1.9	1.00%	3.1	6.81%	9.3	3.80%	4.0	25.67%	6.7	4.31%	3.4
		0.5	1.26%	4.8	0.42%	11.9	9.54%	5.6	**0.26%**	9.1	1.82%	2.4	0.82%	4.3	16.15%	1.6	0.94%	4.4	5.09%	7.4	2.99%	3.1	20.73%	7.3	2.92%	3.1
		0.3	1.16%	9.2	0.43%	14.4	7.67%	14.7	**0.29%**	14.1	1.51%	3.1	0.76%	6.8	13.31%	2.0	0.90%	6.8	3.88%	10.7	2.41%	4.1	17.00%	7.4	2.00%	3.3
	0.8	0.8	0.59%	10.0	0.25%	10.2	10.95%	8.9	**0.15%**	10.9	1.23%	4.5	0.59%	6.9	18.86%	2.8	0.56%	7.4	5.17%	10.9	2.54%	4.1	24.29%	8.8	2.10%	3.9
		0.5	0.64%	11.3	0.27%	15.3	8.74%	16.8	**0.12%**	12.3	1.07%	3.2	0.59%	6.2	14.79%	1.9	0.52%	7.0	3.78%	9.2	2.08%	3.6	19.49%	7.3	1.89%	3.2
		0.3	0.60%	16.5	0.27%	17.8	7.13%	58.0	**0.19%**	18.8	1.04%	3.5	0.53%	7.1	12.36%	2.5	0.47%	8.5	2.94%	10.4	1.69%	4.4	15.95%	7.7	1.36%	4.4
	0.7	0.8	0.74%	11.2	0.26%	14.9	10.70%	13.0	**0.23%**	13.2	1.04%	4.3	0.49%	7.6	17.62%	4.5	0.45%	7.8	4.09%	8.0	1.72%	4.2	22.96%	7.8	1.77%	3.7
		0.5	0.63%	16.9	0.26%	14.7	8.47%	22.0	**0.22%**	18.2	0.87%	3.4	0.51%	7.3	14.15%	3.0	0.47%	8.3	3.02%	9.9	1.35%	3.4	18.68%	7.2	1.52%	3.1
		0.3	0.58%	40.8	0.23%	28.2	7.02%	71.0	**0.13%**	29.0	0.93%	5.8	0.47%	10.1	11.91%	4.3	0.40%	11.6	2.37%	12.6	1.25%	5.4	15.47%	7.9	1.27%	7.7

Table 3.4: Average optimality gap and running time values of the proposed algorithm for five different RAMLSP-3-30 instances with different partition and lower bound choices.

The bold entries in Table 3.4 correspond to the smallest optimality gap values among all lower bound choices, partition strategies and number of groups. It can be observed that, the smallest optimality gap values are obtained with strategy *different*, lower bound choice $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$, and $J = 2$. Regarding to the optimality gap, $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$ is the best lower bound choice. In general, for $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$, the partition strategy *different* provides the smallest optimality gap for any J value, and the strategy *similar* is the worst one. This is a consequence of the fact that group subproblems with original dynamic risk measure reflect the risk-aversion behavior of the original problem better when the dispersion within groups is high.

Moreover, the running time of the algorithm decreases as moving through lower bound choices $\rho_{\mathscr{G}} \circ \mathbb{E}_{\mathscr{F}|\mathscr{G}}$, $\rho^s_{\mathscr{G}} \circ \rho^s_{\mathscr{F}|\mathscr{G}}$ and $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$, in general. For example, with partition strategy *different* and $J = 2$, the average running times for lower bound choices $\rho_{\mathscr{G}} \circ \mathbb{E}_{\mathscr{F}|\mathscr{G}}$, $\rho^s_{\mathscr{G}} \circ \rho^s_{\mathscr{F}|\mathscr{G}}$ and $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$ are 57.18, 37.79 and 14.6 seconds, respectively. On the other hand, no partition strategy is preferable among others for all lower bound choices with respect to the running time. The computational experiments summarized in Table 3.4 reveal that the partition strategy *different* and the lower bound choice $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$ are the most promising choices when the bound quality and the running time are considered.

Although the upper bounds obtained from different partitions are incomparable, a hierarchy of lower bounds can be obtained using refinement chains. Moreover, the lower bound values can be improved by relaxing the requirement that groups should be disjoint. If this requirement is relaxed, some fixed scenarios appear in each group. We call this as scenario fixing. In the next subsection, we will discuss refinement chains, scenario fixing, and their impact on the quality of the lower bound.

3.3.3 Refinement Chains and Scenario Fixing

In [42], refinement chains and scenario fixing are considered to improve the quality of lower bounds obtained via scenario grouping. We suggest Proposition 5 to construct a refinement chain. We show a relation between two lower bounds

obtained via two different special scenario partitioning with the lower bound choice $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$. Let $LB(\mathcal{S}^1)$ and $LB(\mathcal{S}^2)$ be the lower bounds $\mathbb{E}_{\mathscr{G}^1} \circ \rho_{\mathscr{F}|\mathscr{G}^1}$ and $\mathbb{E}_{\mathscr{G}^2} \circ \rho_{\mathscr{F}|\mathscr{G}^2}$ on the optimal value z^* of **(P)**, obtained by partitions $\mathcal{S}^1 = \{S_1^1, \ldots, S_J^1\}$ and $\mathcal{S}^2 = \{S_1^2, \ldots, S_M^2\}$ where $\mathscr{G}^1$ and $\mathscr{G}^2$ are the sigma algebras induced by $\mathcal{S}^1$ and $\mathcal{S}^2$, respectively.

Proposition 5. *Let $\mathcal{S}^1$ and $\mathcal{S}^2$ be two different partitions of Ω, where $S_j^1 \bigcap S_{j'}^1 = \emptyset$ for $j, j' \in \{1, \ldots, J\}$, $j \neq j'$ and $S_m^2 \bigcap S_{m'}^2 = \emptyset$ for $m, m' \in \{1, \ldots, M\}$, $m \neq m'$. If for all $S_j^1 \in \mathcal{S}^1$, $j \in \{1, \ldots, J\}$ there exists $S_k^2 \in \mathcal{S}^2, k \in \{j_1, \ldots, j_{K_j}\}$ such that $S_j^1 = \bigcup_{k \in \{j_1, \ldots, j_{K_j}\}} S_k^2$, then $LB(\mathcal{S}^1) \geq LB(\mathcal{S}^2)$.*

Proof. For $j \in \{1, \ldots, J\}$ and $k \in \{j_1, \ldots, j_{K_j}\}$, let z^j and $\tilde{z}^k$ be the optimal values of group subproblems defined by groups $S_j^1 \in \mathcal{S}^1$ and $S_k^2 \in \mathcal{S}^2$, respectively. Also let $p_j = \sum_{\omega \in S_j^1} p_\omega$ and $\tilde{p}_k = \sum_{\omega \in S_k^2} p_\omega$.

Since $S_k^2, k \in \{j_1, \ldots, j_{K_j}\}$ is a partition of S_j^1, Theorem 1 implies that

$$z^j \geq \sum_{k \in \{j_1, \ldots, j_{K_j}\}} \frac{\tilde{p}_k}{p_j} \tilde{z}^k, \quad \forall j \in \{1, \ldots, J\},$$

where $\frac{\tilde{p}_k}{p_j}$ is the total conditional probability of scenarios in $S_k^2, k \in \{j_1, \ldots, j_{K_j}\}$ given S_j^1. Then, we have

$$\sum_{j \in \{1, \ldots, J\}} p_j z^j \geq \sum_{j \in \{1, \ldots, J\}} p_j \left(\sum_{k \in \{j_1, \ldots, j_{K_j}\}} \frac{\tilde{p}_k}{p_j} \tilde{z}^k \right),$$

or equivalently,

$$\sum_{j \in \{1, \ldots, J\}} p_j z^j \geq \sum_{j \in \{1, \ldots, J\}} \sum_{k \in \{1, \ldots, K_j\}} \tilde{p}_k \tilde{z}^k.$$

Then, we get

$$\sum_{j \in \{1, \ldots, J\}} p_j z^j \geq \sum_{m \in \{1, \ldots, M\}} \tilde{p}_m \tilde{z}^m,$$

where $\tilde{p}_m = \sum_{\omega \in S_m^2} p_\omega$ and $\tilde{z}^m$ is the optimal value of the group subproblem S_m^2 for $m \in \{1, \ldots, M\}$. Hence, $LB(\mathcal{S}^1) \geq LB(\mathcal{S}^2)$ by definition of $LB(\mathcal{S}^1)$ and $LB(\mathcal{S}^2)$. $\square$

Partition	Groups
$\mathcal{S}_1$	Ω
$\mathcal{S}_2$	$\{\omega_9, \omega_{11}, \omega_7, \omega_{13}, \omega_{10}, \omega_8, \omega_{12}, \omega_{15}\}, \{\omega_4, \omega_6, \omega_{16}, \omega_1, \omega_2, \omega_3, \omega_3, \omega_{14}, \omega_5\}$
$\mathcal{S}_4$	$\{\omega_9, \omega_7, \omega_{10}, \omega_{12}\}$,$\{\omega_4, \omega_{16}, \omega_2, \omega_{14}\}$, $\{\omega_{11}, \omega_{13}, \omega_8, \omega_{15}\}$,$\{\omega_6, \omega_1, \omega_3, \omega_5\}$
$\mathcal{S}_8$	$\{\omega_9, \omega_{10}\}, \{\omega_4, \omega_2\}, \{\omega_{11}, \omega_8\}, \{\omega_6, \omega_3\}, \{\omega_7, \omega_{12}\}, \{\omega_{16}, \omega_{14}\}, \{\omega_{13}, \omega_{15}\}, \{\omega_1, \omega_5\}$
$\mathcal{S}_{16}$	$\{\omega_9\}, \{\omega_4\}, \{\omega_{11}\}, \{\omega_6\}, \{\omega_7\}, \{\omega_{16}\}, \{\omega_{13}\}, \{\omega_1\}, \{\omega_{10}\}, \{\omega_2\}, \{\omega_8\}, \{\omega_3\}, \{\omega_{12}\}, \{\omega_{14}\}, \{\omega_{15}\}, \{\omega_5\}$

Table 3.5: A refinement chain for the scenario tree in Example 5 where the partition strategy is *different*.

A sequence of partitions $\mathcal{S}^1, \mathcal{S}^2, \mathcal{S}^3, \ldots$, for which $LB(\mathcal{S}^1) \geq LB(\mathcal{S}^2) \geq LB(\mathcal{S}^3) \geq \cdots$, is called a refinement chain. Table 3.5 shows a refinement chain for the scenario tree in Example 5, where the partition strategy *different* is used and $\mathcal{S}_J$ denotes a partition with J groups. Note that, in partition $\mathcal{S}_{16}$, each group subproblem is a deterministic problem with only one scenario.

We conduct a computational experiment where five different RAMLSP-3-32 instances with 1024 scenarios are used with the refinement chain $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_4, \ldots, \mathcal{S}_{128}$ which is obtained with partition strategy *different*. Table 3.6 presents the number of scenarios in each group (# sce), the average lower bound gap (LB_Gap) and the average running time (Time) for each partition of the refinement chain. LB_Gap values are calculated as

$$\text{LB\_Gap}(\%) = 100\frac{z^* - LB}{z^*},$$

where z^* is the optimal value of our problem.

As expected, the quality of the lower bound obtained by scenario grouping increases when the number of scenarios in each group subproblem increases with a cost of longer running times. Another suggestion of [42] is relaxing the disjoint groups assumption to improve lower bound quality. Recall that $\omega \in \Omega$ is a scenario with probability p_ω and J is the number of groups. We can relax disjoint groups assumption by placing ω into $k \in \{2, \ldots, J\}$ different groups. In this case, ω is replaced with k identical scenarios each having a probability of p_ω/k and these new scenarios are placed into k different groups.

We conduct another set of computational experiments for five different RAMLSP-3-32 instances where eight scenarios with the largest single scenario cost appear at each group and we present the results in Table 3.7. It is observed that the quality of lower bounds improves when we allow some scenarios to appear in all groups. However, the running times may get larger since the number of scenarios in each group subproblem increases. For example, when $J = 2$, $\alpha = 0.7$, $\epsilon_1 = 0.3$, for the case where eight scenarios are fixed, the LB_Gap decreases by 0.01% but the required time to obtain lower bound increases to 2749.1 from 104 seconds.

α		0.9						0.8						0.7					
ϵ_1		0.8		0.5		0.3		0.8		0.5		0.3		0.8		0.5		0.3	
Partition	# sce	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time
$\mathcal{S}_1$	1024	-	81.3	-	3285.1	-	6503.9	-	90.1	-	4496.4	-	7199.0	-	169.3	-	3149.7	-	7272.5
$\mathcal{S}_2$	512	**0.20%**	3.6	**0.30%**	7.2	**0.28%**	11.5	**0.34%**	7.1	**0.27%**	12.3	**0.23%**	16.4	**0.28%**	8.7	**0.24%**	46.1	**0.26%**	104.0
$\mathcal{S}_4$	256	**0.82%**	1.3	**0.94%**	1.2	**0.80%**	2.3	**0.76%**	1.8	**0.64%**	2.3	**0.59%**	3.3	**0.58%**	1.9	**0.65%**	3.8	**0.63%**	5.7
$\mathcal{S}_8$	128	**2.60%**	0.4	**2.46%**	0.46	**1.99%**	0.6	**1.93%**	0.8	**1.75%**	0.6	**1.47%**	0.9	**1.66%**	0.6	**1.59%**	1.0	**1.44%**	1.5
$\mathcal{S}_{16}$	64	**4.90%**	0.2	**4.29%**	0.2	**3.47%**	0.3	**4.18%**	0.3	**3.53%**	0.2	**2.90%**	0.3	**3.44%**	0.3	**3.06%**	0.2	**2.66%**	0.3
$\mathcal{S}_{32}$	32	**6.81%**	0.1	**5.99%**	0.1	**4.99%**	0.1	**6.37%**	0.1	**5.39%**	0.1	**4.47%**	0.1	**5.84%**	0.1	**4.93%**	0.1	**4.14%**	0.1
$\mathcal{S}_{64}$	16	**9.10%**	0.0	**7.84%**	0.0	**6.58%**	0.0	**8.68%**	0.0	**7.25%**	0.0	**6.03%**	0.0	**8.36%**	0.0	**6.93%**	0.0	**5.78%**	0.0
$\mathcal{S}_{128}$	8	**11.25%**	0.0	**9.58%**	0.0	**8.10%**	0.0	**10.89%**	0.0	**9.03%**	0.0	**7.53%**	0.0	**10.38%**	0.0	**8.64%**	0.0	**7.23%**	0.0

Table 3.6: Average lower bound gap and running time for the refinement chain $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_4, \ldots, \mathcal{S}_{128}$ obtained with partition strategy *different* for five different RAMLSP-3-32 instances.

α		0.9						0.8						0.7					
ϵ_1		0.8		0.5		0.3		0.8		0.5		0.3		0.8		0.5		0.3	
Partition	# sce	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time	LB_Gap	Time
$\mathcal{S}_1$	1024	-	81.3	-	3285.1	-	6503.9	-	90.13	-	4496.4	-	7199.0	-	169.3	-	3149.7	-	7272.5
$\mathcal{S}_2$	516	**0.11%**	12.8	**0.23%**	27.6	**0.24%**	115.9	**0.22%**	9.9	**0.23%**	36.7	**0.23%**	691.3	**0.22%**	10.8	**0.23%**	98.0	**0.25%**	2749.1
$\mathcal{S}_4$	262	**0.44%**	1.2	**0.70%**	2.1	**0.66%**	3.3	**0.59%**	1.9	**0.56%**	3.1	**0.54%**	4.4	**0.49%**	2.3	**0.59%**	4.1	**0.59%**	6.7
$\mathcal{S}_8$	135	**1.59%**	0.4	**1.82%**	0.5	**1.67%**	0.8	**1.52%**	0.5	**1.48%**	0.6	**1.33%**	0.9	**1.38%**	0.5	**1.44%**	0.8	**1.35%**	1.3

Table 3.7: Average lower bound gap and running time for the refinement chain $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_4, \mathcal{S}_8$ obtained with partition strategy *different* and eight fixed scenarios for five different RAMLSP-3-32 instances.

3.3.4 Computational Study Results for Larger Number of Stages

Up to now, we have shown that the lower bound choice $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$, the partition strategy *different*, and considering disjoint groups is the most promising combination among all bound and partition combinations. Therefore, further computational experiments are conducted on the instances with more stages under this setting. We also conduct a set of computational experiments to compare the performance of the proposed algorithm with CPLEX in terms of optimality gap and solution time.

In the upper bounding phase of the proposed algorithm, the restricted problem is solved for each group. When the number of groups J in a partition is large, the upper bounding phase requires long CPU times. Therefore, one may solve the restricted problem for only a subset of groups. Another computational enhancement for the upper bounding phase is running the restricted problems with a prespecified time limit and reporting the objective value of current incumbent solution as UB_j. Since, the optimal value of the restricted problem is an upper bound for the original problem, the objective value of any incumbent solution is also a valid upper bound.

We solve RAMLSP-3-64, RAMLSP-4-8, and RAMLSP-5-4 problems with 3, 4, and 5 stages, respectively, and for each risk setting, we generate five instances using different random seeds. The algorithm is applied with lower bound choice $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$ and the partition strategy *different*, where number of groups, J, takes values of 4, 8, 16, and 32. The number of restricted problems to be solved is $\lceil J/5 \rceil$, which are selected randomly. The time limit for each restricted problem is set to 10 seconds. The results are presented in Table 3.8.

			J = 4		J = 8		J = 16		J = 32	
	α	ϵ_1	Gap	Time	Gap	Time	Gap	Time	Gap	Time
RAMLSP-3-64	0.9	0.8	1.02%	68.9	2.42%	55.8	5.02%	44.6	8.13%	82.0
		0.5	0.92%	132.6	2.07%	66.2	4.27%	60.9	6.99%	86.1
		0.3	0.74%	697.4	1.57%	101.4	3.33%	75.6	5.80%	93.1
	0.8	0.8	0.71%	27.5	1.67%	13.0	3.82%	26.3	7.10%	85.2
		0.5	0.66%	97.4	1.55%	25.4	3.19%	39.3	6.03%	82.4
		0.3	0.60%	376.1	1.20%	49.6	2.53%	55.6	4.76%	86.5
	0.7	0.8	0.54%	51.8	1.30%	14.5	2.82%	37.6	5.67%	82.9
		0.5	0.56%	129.2	1.12%	27.4	2.37%	48.2	4.74%	81.9
		0.3	0.48%	373.1	0.93%	63.7	1.96%	57.6	3.73%	84.9
RAMLSP-4-8	0.9	0.8	5.92%	10.4	6.05%	9.0	9.62%	17.6	11.23%	37.5
		0.5	5.49%	16.9	6.69%	19.3	7.28%	26.0	9.18%	59.1
		0.3	4.06%	21.3	5.87%	27.0	6.45%	37.8	7.76%	66.5
	0.8	0.8	3.59%	13.9	5.03%	11.2	6.69%	18.8	9.62%	50.0
		0.5	3.64%	18.6	5.53%	25.5	6.35%	33.2	8.26%	63.5
		0.3	3.40%	22.3	5.28%	29.3	5.86%	40.9	7.48%	62.0
	0.7	0.8	3.48%	16.0	4.95%	14.5	6.14%	22.0	8.38%	54.3
		0.5	3.19%	21.1	5.31%	24.4	5.92%	33.7	7.95%	61.8
		0.3	3.12%	28.3	4.86%	32.2	5.54%	41.1	7.09%	65.1
RAMLSP-5-4	0.9	0.8	6.55%	8.9	10.25%	4.8	13.24%	11.7	16.09%	29.9
		0.5	5.53%	13.7	9.09%	5.6	11.69%	16.0	13.27%	43.2
		0.3	5.12%	17.1	7.97%	8.0	10.02%	20.0	11.38%	48.2
	0.8	0.8	6.83%	9.9	10.78%	4.7	13.40%	11.3	16.16%	29.6
		0.5	5.70%	13.6	9.48%	6.2	11.81%	14.9	13.40%	43.7
		0.3	5.28%	16.4	8.27%	8.6	10.08%	17.4	11.53%	50.1
	0.7	0.8	6.04%	10.3	9.92%	6.0	13.04%	17.0	14.58%	40.5
		0.5	5.39%	11.7	8.96%	6.7	10.96%	20.9	12.49%	51.4
		0.3	4.93%	15.5	7.94%	9.8	9.53%	22.5	11.00%	56.9

Table 3.8: Average optimality gap and running time values of the proposed algorithm for five different RAMLSP-3-30, RAMLSP-4-8, and RAMLSP-5-4 instances with partition strategy *different* and lower bound choice $\mathbb{E}_{\mathscr{G}} \circ \rho_{\mathscr{F}|\mathscr{G}}$.

		Proposed Algorithm		CPLEX			
α	ϵ_1	Gap	Time	Gap_CPLEX	GAP Difference	Time_CPLEX	Delay
0.9	0.8	1.02%	68.9	2.76%	1.74%	248.3	3.60
	0.5	0.92%	132.6	3.47%	2.55%	2366.7	17.84
	0.3	0.74%	697.4	1.39%	0.65%	719.9	1.03
0.8	0.8	0.71%	27.5	2.04%	1.33%	241.9	8.79
	0.5	0.66%	97.4	1.48%	0.82%	403.9	4.15
	0.3	0.60%	376.1	0.89%	0.29%	603.0	1.60
0.7	0.8	0.54%	51.8	2.47%	1.93%	246.4	4.75
	0.5	0.56%	129.2	1.13%	0.57%	444.2	3.44
	0.3	0.48%	373.1	0.81%	0.33%	1445.6	3.87

Table 3.9: Comparison of optimality gaps and running times of the proposed algorithm with CPLEX.

As seen in Table 3.8, increasing the number of groups in the partition may not always yield CPU time saving. As J increases, the optimality gap increases, on the other hand, the CPU time may not always decrease. Specifically, when J is increased to 32 from 16, the CPU time increases in all of the instances. As the number of groups J increases, the subproblems get smaller in size. However, the number of group subproblems and the restricted problems to solve increases. Therefore, increasing the number of groups may not always result in a decrease in the running time of the algorithm.

An interesting question is the comparison of the proposed algorithm with CPLEX in terms of optimality gap and CPU time. To make a fair comparison, we use RAMLSP-3-64 instances where CPLEX is run as long as it reaches to the optimality gap or the CPU time of the proposed algorithm.

When CPLEX is allowed to run with one hour of time limit, it cannot solve any of the instances optimally. Table 3.9 presents the comparison of the proposed algorithm with CPLEX for $J = 4$.

In Table 3.9, the column "*Gap_CPLEX*" corresponds to the optimality gap value reported by CPLEX when it is allowed to run as long as the running time of the proposed algorithm. Moreover, the values in the column "*GAP Difference*"

is measured as the difference between CPLEX gap and the gap obtained by the proposed algorithm. When CPLEX is allowed to run as long as the solution time of the proposed algorithm, the algorithm yields 1.13% stronger bounds on the average. For example, when $\alpha = 0.7$ and $\epsilon_1 = 0.8$, our algorithm terminates with an optimality gap of 0.54% within 51.8 seconds. CPLEX stops with an optimality gap of 1.93% within the same time limit, that is, the bounds obtained by our algorithm is 1.39% better than the bounds obtained by CPLEX.

In Table 3.9, the column "*Time_CPLEX*" corresponds to the time in seconds that CPLEX takes to reduce its gap to the level of the gap obtained by the proposed algorithm. Also, the values in the column "*Delay*" are measured as the ratio of *Time_CPLEX* to the running time of the proposed algorithm (*Time*). CPLEX requires 5.45 times longer running time to achieve the optimality gap of the proposed algorithm, on the average. For $\alpha = 0.9$ and $\epsilon_1 = 0.5$, CPLEX requires 2366.7 seconds to achieve the optimality gap of the proposed algorithm, that means CPLEX needs to spend more than 17 times of the running time of the proposed algorithm in order to reach this optimality gap. These results show that the proposed algorithm outperforms CPLEX with respect to both optimality gap and running time.

3.4 Conclusion

In this chapter, we propose a group subproblem approach for risk-averse mixed-integer multi-stage stochastic problems with a dynamic risk measure defined by mean-CVaR. To the best of our knowledge, this is the first study where group subproblem approach is applied to a risk-averse problem with an objective of a dynamic risk measure. We show that infinitely many lower bounds on the optimal value of the problem can be obtained by using different convolution of mean-CVaR risk measures. An upper bound is obtained through the use of optimal solutions of group subproblems, as well. The results are tested by a computational study on a multi-stage lot sizing problem. The effect of partition strategies and lower bound choices on the optimality gap of the proposed algorithm is investigated. Possible

computational enhancements such as refinement chains and scenario fixing are also considered.

It is revealed that, on the average, the optimality gap of the proposed algorithm is 1.13% stronger than the optimality gap of CPLEX within the same running time. By solving the original problem with CPLEX, the optimality gaps of our algorithm can be achieved with additional running time more than a factor of five.

Chapter 4

An Exact Solution Approach for Risk-averse Mixed-integer Multi-stage Stochastic Programming Problems

In this chapter, we propose an exact solution algorithm for risk-averse mixed-integer multi-stage stochastic programming problems with an objective of dynamic mean-CVaR risk measure and binary first stage variables. The proposed method is based on an evaluate-and-cut procedure where the lower bounds are obtained from group subproblems. Moreover, we show that, under the assumption that the first stage integer variables are bounded, the problem with mixed-integer variables in all stages can be solved with the proposed algorithm. In order to observe the performance of the proposed algorithm, we conduct a set of computational experiments on risk-averse mixed-integer multi-stage problems. In our experiments, we consider large instances of risk-averse multi-stage stochastic server location and generation expansion problems. We also investigate some implementation details of the algorithm such as scenario partitioning choices and group sizes and then analyze their effects on the performance of the algorithm.

As the computational experiments reveal, under modest computational settings, the proposed algorithm is able to solve large problem instances with over one million binary variables within a reasonable time.

The organization of the chapter is as follows: In Section 4.1, we define risk-averse mixed-integer multi-stage stochastic programing problems. In Section 4.2, we present the bounds obtained by scenario grouping. The proposed algorithm is presented in Section 4.3. In Section 4.4, we show that our algorithm can also be used for the problems with mixed-integer variables in all stages. The computational experiments are given in Section 4.5. The concluding remarks are presented Section 4.6.

The results of this chapter are accepted for publication in Annals of Operations Research.

4.1 Problem Definition

We first recall the risk-averse multi-stage stochastic programming problem

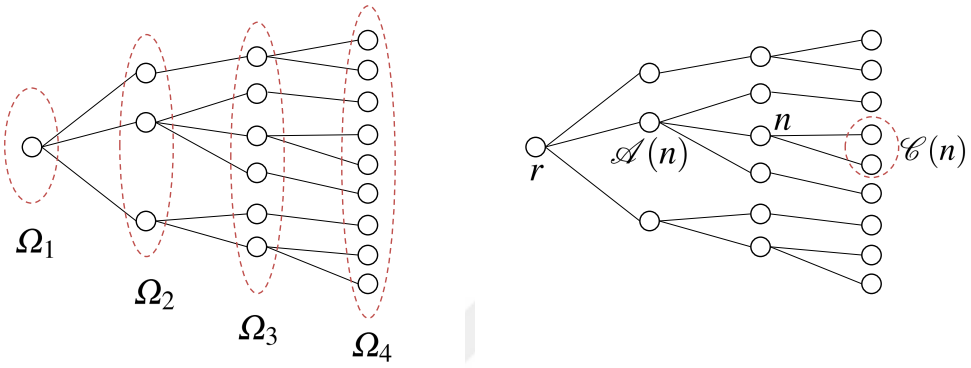
$$\min_{x \in \mathcal{X}} \quad \varrho_{1,T}(f_1(x_1), f_2(x_2, \xi_2), \ldots, f_T(x_T, \xi_T)), \tag{4.1}$$

where $\mathcal{X}_1 \subseteq \mathbb{R}_+^{K_1} \times \mathbb{Z}_+^{L_1} \times \{0,1\}^{M_1}$ is a mixed-integer deterministic set and $\mathcal{X}_t : \mathbb{R}_+^{K_{t-1}} \times \mathbb{Z}_+^{L_{t-1}} \times \{0,1\}^{M_{t-1}} \times \Xi_t \rightrightarrows \mathbb{R}_+^{K_t} \times \mathbb{Z}_+^{L_t} \times \{0,1\}^{M_t}$ is the point-to-set mapping representing mixed-integer $\mathscr{F}_t$-measurable decisions at stage $t \in \{2, \ldots, T\}$.

In this chapter, we use a scenario tree representation similar to the one in Chapter 3. Let $\mathcal{N}$ and $\mathcal{E}$ be the set of nodes and edges of the scenario tree, respectively. Nodes at stage $t \in \{1, \ldots, T\}$ correspond to possible realizations of the history until that stage. Let Ω_t be the set of nodes at stage $t \in \{1, \ldots, T\}$ and hence $\mathcal{N} = \bigcup_{t \in \{1,\ldots,T\}} \Omega_t$. At the first stage, there is only one node r called as the *root node*.

A scenario can also be defined as a unique path from the root node to a node

Figure 4.1: A four-stage scenario tree.



in Ω_T. Let p_ω be the probability of scenario $\omega \in \Omega$. Then, probability associated with node $n \in \mathcal{N}$ is $p_n = \sum_{\omega \in \mathcal{P}_n} p_\omega$ where $\mathcal{P}_n$ is the set of scenarios passing through node $n \in \mathcal{N}$. For the root node, we naturally have $p_r = 1$.

In the scenario tree representation, the sigma algebra $\mathcal{F}_T$ consists of all subsets of Ω_T. For every node $n \in \mathcal{N} \setminus \{r\}$ at stage t, there exists a unique *ancestor node* $\mathcal{A}(n) \in \Omega_{t-1}$ such that the edge $(\mathcal{A}(n), n)$ is in $\mathcal{E}$. For each node $n \in \mathcal{N} \setminus \Omega_T$, there exists a set of *children nodes* $\mathcal{C}(n) = \{m \in \Omega_{t+1} : \mathcal{A}(m) = n\}$ such that $\Omega_{t+1} = \bigcup_{n \in \Omega_t} \mathcal{C}(n)$. For $t \in \{1, \ldots, T-1\}$, let $\mathcal{F}_t$ be the subalgebra of $\mathcal{F}_{t+1}$ generated by sets $\mathcal{C}(n)$ for all $n \in \Omega_t$. Hence, there is a one-to-one correspondence between elementary events of $\mathcal{F}_t$ and nodes in the set Ω_t, for $t \in \{1, \ldots, T\}$. By construction, we get the filtration $\mathcal{F}_1 \subset \mathcal{F}_2 \cdots \subset \mathcal{F}_T$. The aforementioned notation is depicted on an example of four-stage scenario tree in Figure 4.1.

The number of decision variables in Problem (4.1) is proportional to the number of nodes $|\mathcal{N}|$ in the scenario tree. Since $|\mathcal{N}|$ grows exponentially with the number of stages for any non-trivial scenario tree, solving Problem (4.1) is computationally demanding for even small number of stages.

In this chapter, we focus on the case where $\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}$ is a *conditional mean-CVaR risk measure* for $t \in \{1, \ldots, T-1\}$. The conditional mean-CVaR is defined as

$$\rho_{\mathscr{F}_{t+1}|\mathscr{F}_t}(Z_{t+1}) = (1-\epsilon)\mathbb{E}[Z_{t+1}|\mathscr{F}_t] + \epsilon \mathrm{CVaR}_\alpha(Z_{t+1}|\mathscr{F}_t), \tag{4.2}$$

where

$$\mathrm{CVaR}_\alpha(Z_{t+1}|\mathscr{F}_t) = \inf_{\eta\in\mathcal{Z}_t}\left\{\eta + \frac{1}{1-\alpha}\mathbb{E}[(Z_{t+1}-\eta)_+|\mathscr{F}_t]\right\},$$

is *conditional CVaR* defined with parameter $\alpha \in [0,1)$ and $\epsilon \in [0,1]$. In (4.2), mean-CVaR value of a random variable $Z_{t+1} \in \mathcal{Z}_{t+1}$ can be interpreted as a convex combination of its conditional expectation and conditional CVaR values.

4.2 Lower Bounds via Scenario Grouping

In this section, we present a lower bound for our problem by using scenario grouping as discussed in Chapter 3. The proposed lower bound is used in the algorithm presented in the next section.

Recall the probability space for problem (4.1), $(\Omega, \mathscr{F}, P)$ where Ω is the sample space. A subset of scenarios $S \subseteq \Omega$ is called as a *group*. Let $\mathcal{S} = \{S_j\}_{j=1}^J$ be a collection of groups that forms a partition of Ω, that is, $\bigcup_{j=1}^J S_j = \Omega$ and $S_j \bigcap Sj' = \emptyset$ for all $j, j' \in \{1,2,\ldots,J\}$ such that $j \neq j'$. The empty intersection requirement can be relaxed (see, for example, [39]).

The total probability of scenarios in group S_j is $p_j = \sum_{\omega\in S_j} p_\omega$ for $j \in \{1,\ldots,J\}$. We also define conditional probability of scenario ω given that $\omega \in S_j$ as $\widetilde{p}_\omega = p_\omega/p_j$ and $\widetilde{P}$ as the probability distribution defined by these conditional probabilities on the sample space S_j.

We define j^{th} group subproblem as problem (4.1) which is defined on the probability space $(S_j, \widetilde{\mathscr{F}}, \widetilde{P})$ where $\widetilde{\mathscr{F}}$ is the sigma algebra generated by S_j.

Proposition 6. *Let z_j be the optimal value of j^{th} group subproblem. Then, $\sum_{j=1}^J p_j z_j$ is a lower bound for the optimal value of (4.1).*

Proof. See Theorem 1 in Chapter 3. □ □

In this chapter, we develop another lower bound obtained from scenario grouping.

Proposition 7. *Let $\tilde{z}_j$ be the optimal value of j^{th} group subproblem in which integrality requirements for the decision variables at stages $t \in \{2, 3, \ldots, T\}$ are relaxed. Then $\sum_{j=1}^{J} p_j \tilde{z}_j$ is a lower bound for the optimal value z of (4.1) .*

Proof. Due to Proposition 6, we have $z \geq \sum_{j=1}^{J} p_j z_j$. Since, $z_j \geq \tilde{z}_j$ for all $j \in \{1, 2, \ldots, J\}$, we get $z \geq \sum_{j=1}^{J} p_j z_j \geq \sum_{j=1}^{J} p_j \tilde{z}_j$. □ □

Note that the lower bound in Proposition 7 is no stronger than the lower bound in Proposition 6. However, calculating the lower bound in Proposition 7 is computationally easier since it requires solving group problems without integrality requirements for the decision variables at stages $t \in \{2, 3, \ldots, T\}$. Hence, the latter lower bound is used in the proposed algorithm.

Scenario grouping significantly reduces the computational effort required to solve the risk-averse mixed-integer multi-stage programming problem (4.1). Figure 4.2 illustrates the scenario trees of group subproblems obtained via scenario grouping. The scenarios of the original problem are placed in groups S_1, S_2 and S_3 and indicated by colors red, green and blue, respectively.

4.3 An Exact Solution Algorithm for the Problems with Binary First Stage Variables

In this section, extending the idea in [43], we propose an exact solution algorithm for the risk-averse mixed-integer multi-stage stochastic programming problem (4.1) with binary first stage variables, that is, $\mathcal{X}_1 \subseteq \{0, 1\}^{M_1}$ and $K_1 = L_1 = 0$.

The proposed algorithm stores a set $\mathcal{D} \subseteq \mathcal{X}_1$ of candidate first stage solutions and an incumbent first stage solution $x_1^* \in \mathcal{D}$ through execution. If a candidate

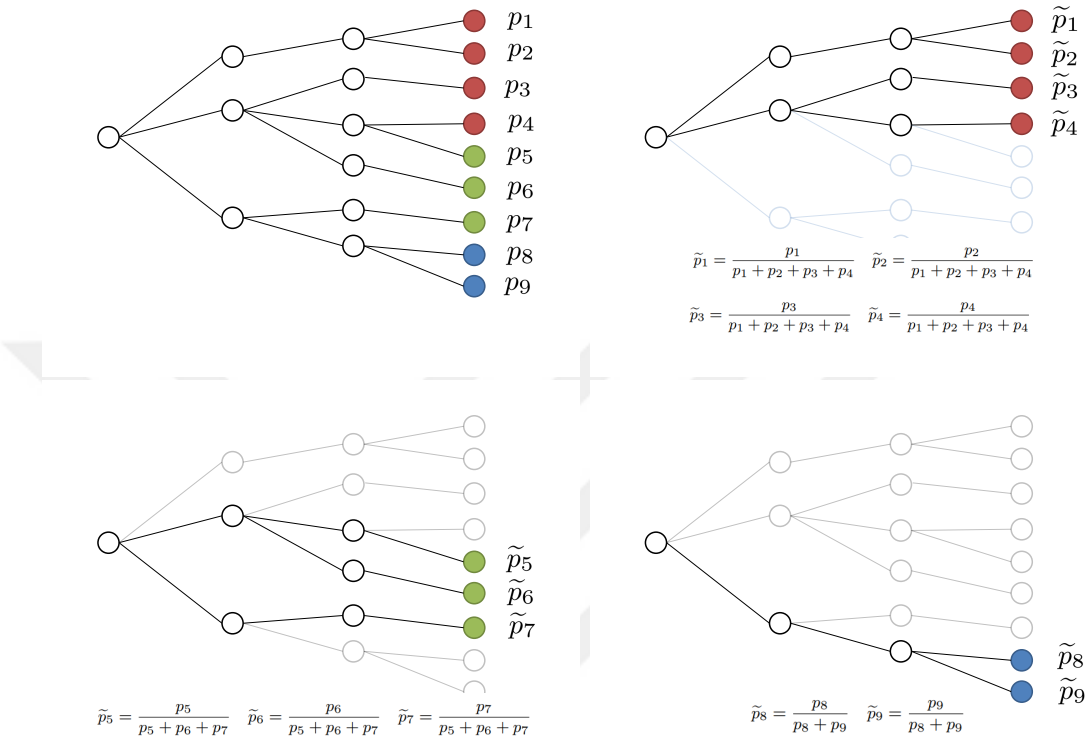


Figure 4.2: A four-stage scenario tree and the new scenario trees used in group subproblems.

first stage solution $x_1' \in \mathcal{D}$ is a feasible first stage solution for problem (4.1), then solving (4.1) with constraint $x_1 = x_1'$ gives an upper bound for the optimal value of (4.1). The incumbent solution x_1^* is the first stage solution which yields the smallest upper bound value among all candidate solutions in $\mathcal{D}$.

For a given scenario partition, a *socalled* lower bound is obtained on the optimal value of (4.1) by applying scenario grouping for (4.1) with first stage feasibility set $\mathcal{X}_1 \setminus \mathcal{D}$. Note that, this is not necessarily a lower bound for the original problem (4.1) since some feasible first stage solutions are eliminated from $\mathcal{X}_1$. At each iteration, if this lower bound is smaller than the objective value given by x_1^*, then the algorithm adds the new candidate solutions to set $\mathcal{D}$; otherwise, the algorithm terminates.

Since we assume that all first stage decision variables are binary, extraction of

the candidate first stage solutions in $\mathcal{D}$ from $\mathcal{X}_1$, that is obtaining the set $\mathcal{X}_1 \setminus \mathcal{D}$, can be done by using *no-good cuts*. A no-good cut separates a specific binary vector from a set. Specifically, a no-good cut for binary vector x_1' is of the form

$$\sum_{i:x_{1i}'=1} (1 - x_{1i}) + \sum_{i:x_{1i}'=0} x_{1i} \geq 1,$$

where x_{1i} and x_{1i}' correspond to the i^{th} component of vectors x_1 and x_1', respectively (see, for example, [51]). Then we have,

$$\mathcal{X}_1 \setminus \mathcal{D} = \left\{ x_1 \in \mathcal{X}_1 : \sum_{i:x_{1i}'=1} (1 - x_{1i}) + \sum_{i:x_{1i}'=0} x_{1i} \geq 1, \ \forall x_1' \in \mathcal{D} \right\}.$$

The overall scheme of the proposed algorithm is an *evaluate-and-cut* procedure presented in Algorithm 2.

Proposition 8. *Algorithm 2 terminates after a finite number of iterations finding an optimal solution if problem (4.1) is feasible and has finite optimal value or with declaration of infeasibility if (4.1) is infeasible.*

Proof. At the lower bounding phase of each iteration of the algorithm, at least one first stage solution is removed from $\mathcal{X}_1$ and added to $\mathcal{D}$ or the algorithm terminates directly. Also since $\mathcal{X}_1 \subseteq \{0,1\}^{M_1}$, the cardinality of $\mathcal{X}_1$ is finite. Therefore, the algorithm terminates after a finite number of iterations. Consider the following two cases:

(a) *Problem (4.1) is feasible and has finite optimal value*: Let $z^*(\mathcal{X}_1)$ be the optimal value of problem (4.1) as a function of the first stage feasibility set $\mathcal{X}_1$. Since, the cardinality of $\mathcal{X}_1$ is finite, we have

$$z^*(\mathcal{X}_1) = \min \{z^*(\mathcal{D}), z^*(\mathcal{X}_1 \setminus \mathcal{D})\} \tag{4.3}$$

at any iteration of the algorithm. For an incumbent solution $x_1^* \in \mathcal{D}$, we also have $UB = z^*(\{x_1^*\}) = z^*(\mathcal{D})$ and $LB \leq z^*(\mathcal{X}_1 \setminus \mathcal{D})$ due to definitions

of upper and lower bounds used in the algorithm. When the algorithm terminates with $UB \leq LB$, the incumbent first stage solution guarantees $z^*(\{x_1^*\}) = z^*(\mathcal{D}) = UB \leq LB \leq z^*(\mathcal{X}_1 \setminus \mathcal{D})$. Hence, using (4.3), we get $z^*(\mathcal{X}_1) = z^*(\{x_1^*\})$ and therefore x_1^* is an optimal first stage solution of problem (4.1).

(b) *Problem (4.1) is infeasible:* If the problem is infeasible, UB never takes a finite value since none of the problems in upper bounding phase is feasible. Then, at the end of the algorithm, the incumbent is null and UB is positive infinity.

□

If each group includes only one scenario, the proposed algorithm is a scenario decomposition algorithm where the non-anticipativity in all stages is completely relaxed. However, the proposed algorithm allows to obtain stronger lower bounds by partially maintaining non-anticipativity due to scenario grouping.

Another important property of Algorithm 2 is decomposition of the problem in the upper bounding phase. When the first stage decisions are fixed to x_1' in the original problem, the resulting problem decomposes into $|\Omega_2|$ smaller problems. These smaller problems are risk-averse mixed-integer multi-stage problems with $T-1$ stages. Therefore, we benefit from decomposition of the original problem in both lower and upper bounding phases of Algorithm 2.

In the next section, we show that the proposed algorithm can be used to solve the risk-averse mixed-integer multi-stage stochastic programming problems where the first stage decisions are mixed-integer as well.

4.4 Extension to the General Risk-averse Mixed-integer Multi-stage Stochastic Programming Problems

Although the proposed algorithm guarantees an exact solution for risk-averse mixed-integer multi-stage stochastic programming problems with a dynamic mean-CVaR objective function and binary first stage variables, it can be used for the general case, where the first stage decision variables are not necessarily pure binary. Let $Q(x_1)$ be the risk adjusted cost-to-go function depending on the first stage solution x_1, that is,

$$\begin{aligned} Q(x_1) :=& \rho_{\mathscr{F}_2|\mathscr{F}_1}\Big(\min_{x_2\in\mathcal{X}_2(x_1,\xi_2)} f_2(x_2,\xi_2) \\ &+ \cdots + \rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}\Big(\min_{x_T\in\mathcal{X}_T(x_{T-1},\xi_T)} f_T(x_T,\xi_T)\Big)\cdots\Big). \end{aligned} \tag{4.4}$$

Then, the general risk-averse mixed-integer multi-stage stochastic programming problem can be written as

$$\begin{aligned} &\min\ f_1(x_1) + Q(x_1) \\ &\text{s.t. } x_1 = (r_1; y_1; b_1) \in \mathcal{X}_1, \\ &\quad r_1 = (r_{11}, r_{12}, \ldots, r_{1K_1}) \in \boldsymbol{R} \subseteq \mathbb{R}_+^{K_1}, \\ &\quad y_1 = (y_{11}, y_{12}, \ldots, y_{1L_1}) \in \boldsymbol{Z} \subseteq \mathbb{Z}_+^{L_1}, \\ &\quad b_1 = (b_{11}, b_{12}, \ldots, b_{1M_1}) \in \boldsymbol{B} \subseteq \{0,1\}^{M_1}, \end{aligned} \tag{4.5}$$

where r_1, y_1, and b_1 represent continuous, integer and binary decision vectors in the first stage, respectively and x_1 represents the concatenate vector of first stage variables. Recall that we have already assumed the decisions at stages $2, 3, \ldots, T$ are mixed-integer.

Recently, [27] show that, under some reasonable assumptions, any risk-neutral mixed-integer stochastic problems with mixed-integer state variables can be approximated by binarizing the state variables. In our context, we only consider

binarizing the first stage integer variables. Following the similar arguments presented by [52] and [27], we can solve the general mixed-integer problem (4.5) by using binary representation of the integer first stage variables y_1.

Assumption 1. *The first stage integer variables in the general mixed-integer problem (4.5) are bounded.*

Assumption 1 holds for the most real life problems and ensures that there exists a non-negative real number U such that $y_{1l} \leq U$ for all $l \in \{1, 2, \ldots, L_1\}$. For each $l \in \{1, 2, \ldots, L_1\}$, $y_{1l} = \sum_{i=0}^{\tau} 2^i \gamma_{1li}$ is an exact representation of y_{1l} with binary variables γ_{1li}, $i \in \{0, 1, \ldots, \tau\}$ and $\tau = \lfloor \log_2 U \rfloor$. Thus, the general mixed-integer problem (4.5) can be written as

$$\begin{aligned} &\min f_1(x_1) + Q(x_1) \\ &\text{s.t. } x_1 = (r_1; A(\Gamma); b_1) \in \mathcal{X}_1, \\ &\qquad r_1 \in \boldsymbol{R}, A(\Gamma) \in \boldsymbol{Z}, \Gamma \in \{0,1\}^{L_1 \times (\tau+1)}, b_1 \in \boldsymbol{B}, \end{aligned} \tag{4.6}$$

where $A : \{0,1\}^{L_1 \times (\tau+1)} \to \mathbb{Z}_+^{L_1}$ is a linear mapping that restores y_1 from binary variables Γ.

We can employ Algorithm 2 to solve (4.6) where the evaluate-and-cut procedure is applied only to the first stage binary variables. Unlike the case where the first stage variables are pure binary, the continuous variables in the first stage prohibit decomposition in the upper bounding phase of the algorithm. However, the computational results in Section 4.5.2 reveal that the proposed algorithm efficiently solves the general mixed-integer problem though decomposition is possible only in the lower bounding phase.

4.5 Computational Experiments

In this section, we present the results of the computational experiments conducted on risk-averse multi-stage stochastic server location problem (SSLP) and

Table 4.1: The degrees of risk aversion used in computational experiments.

DRA	ϵ	α
I	0.4	0.7
II	0.6	0.8
III	0.8	0.9

generation expansion problem (GEP). In risk-averse multi-stage SSLP, the first stage decision variables are pure binary. Therefore, Algorithm 2 is directly applied to the problem. However, in risk-averse GEP, the first stage variables are mixed-integer and hence we use the extension presented in Section 4.4 to solve these problems.

In our experiments, we use different degrees of risk-aversion (DRA) by changing the values of the parameters of conditional mean-CVaR risk measure (4.2). These values are presented in Table 4.1.

The computational experiments are performed on an Intel(R) Core(TM) i7-4790 CPU@3.60 GHz computer with 8.00 GB of RAM. The algorithm is implemented on Java 1.8.0.31 where IBM ILOG CPLEX version 12.6 with default settings is used to solve optimization problems. For each test problem, five instances are generated randomly and average values are reported. The performance of the proposed algorithm is compared to deterministic equivalent problem (DEP) of same instances.

4.5.1 Stochastic Server Location Problem

SSLP is a popular two-stage risk-neutral stochastic programming problem in the literature. The motivation and detailed discussion on SSLP can be found in [53]. An instance of SSLP is available in SIPLIB library at `http://www2.isye.gatech.edu/~sahmed/siplib`. In our experiments, we use a risk-averse and multi-stage version of SSLP.

The statement of risk-averse multi-stage SSLP is as follows: In a T-stage

decision horizon, the objective is to determine the location of servers on a set of potential nodes of a given network so as to minimize the dynamic coherent risk defined in (2.13) where we use conditional mean-CVaR risk measure with parameters ϵ and α at each stage.

Servers can be located on M different potential server location nodes. There is a fixed cost c_m of locating a server at node $m \in \{1, 2, \ldots, M\}$. There are V different clients, whose demands at stage $t \in \{2, 3, \ldots, T\}$ should be satisfied by one of the located servers. Let d^t_{vm} be the demand of client v if it is served by the server located at m at stage t. One unit of profit is obtained for each unit of served demand. The service capacity of a server is u. If the capacity is not enough to serve all demands, a unit of demand can be served by the server located at m by an overcapacity serving cost q^t_m at stage t. A client may or may not appear under each scenario independently from other clients. Under each scenario, a client may appear at a stage with probability 0.5. Let $h^t_v(\omega) = 1$ if client v appears at stage t under scenario ω, and 0, otherwise. Let h^t be an $\mathscr{F}_t$-measurable random vector whose components are h^t_v with $P\{h^t_v = 1\} = P\{h^t_v = 0\} = 0.5$.

Location decisions are made at the first stage. Therefore, fixed cost due to server location decisions is incurred in the first stage. The allocation decisions are made at subsequent stages $t \in \{2, 3, \ldots, T\}$. At stage t, the random cost is the total service cost at stage t. The DEP of risk-averse multi-stage SSLP can be modeled by using parameters and decision variables for each node of the scenario tree. Moreover, linearization of mean-CVaR risk measure is possible by defining additional auxiliary variables and constraints as shown in Chapter 3.

As an example, the DEP of the risk-averse three-stage SSLP is given below which can be extended for larger number of stages easily. In the below model, a superscript n or n' indicates that the corresponding parameter or decision variable is defined for node $n \in \Omega_2$ or $n' \in \Omega_3$ of the scenario tree, respectively.

$$\min \sum_{m=1}^{M} c_m z_m + (1-\epsilon) \sum_{n \in \Omega_2} p_n \widetilde{Z}^n_2 + \epsilon \left(\eta_1 + \frac{1}{1-\alpha} \sum_{n \in \Omega_2} p_n \varphi^n_2 \right), \tag{4.7}$$

$$\text{s.t. } \widetilde{Z}_2^n = \sum_{m=1}^{M}\left(\sum_{v=1}^{V} -d_{vm}^{2n} y_{vm}^{2n} + q_m^{2n} o_m^{2n}\right) + (1-\epsilon)\sum_{n'\in\mathcal{C}(n)} \frac{p_{n'}}{p_n}\widetilde{Z}_3^{n'}$$

$$+ \epsilon\left(\eta_2^n + \frac{1}{1-\alpha}\sum_{n'\in\mathcal{C}(n)} \frac{p_{n'}}{p_n}\varphi_3^{n'}\right), \quad \forall n \in \Omega_2, \tag{4.8}$$

$$\widetilde{Z}_3^{n'} = \sum_{m=1}^{M}\left(\sum_{v=1}^{V} -d_{vm}^{3n'} y_{vm}^{3n'} + q_m^{3n'} o_m^{3n'}\right), \quad \forall n' \in \Omega_3, \tag{4.9}$$

$$\varphi_2^n \geq \widetilde{Z}_2^n - \eta_1, \quad \forall n \in \Omega_2, \tag{4.10}$$

$$\varphi_3^{n'} \geq \widetilde{Z}_3^{n'} - \eta_2^n, \quad \forall n \in \Omega_2, n' \in \mathcal{C}(n), \tag{4.11}$$

$$\sum_{v=1}^{V} d_{vm}^{tn} y_{vm}^{tn} \leq u z_m + o_m^{tn}, \quad \forall m \in \{1,2,\ldots,M\}, n \in \Omega_t, t \in \{2,3\}, \tag{4.12}$$

$$\sum_{m=1}^{M} y_{vm}^{tn} = h_v^{tn}, \quad \forall v \in \{1,2,\ldots,V\}, n \in \Omega_t, t \in \{2,3\}, \tag{4.13}$$

$$z_m \in \{0,1\}, y_{vm}^{tn} \in \{0,1\}, o_m^{tn} \geq 0,$$

$$\forall m \in \{1,2,\ldots,M\}, v \in \{1,2,\ldots,V\}, n \in \Omega_t, t \in \{2,3\}, \tag{4.14}$$

$$\widetilde{Z}_t^n, \varphi_t^n \geq 0, \quad \forall n \in \Omega_t, t \in \{2,3\}, \tag{4.15}$$

$$\eta_1, \eta_2^n \in \mathbb{R}, \quad \forall n \in \Omega_2, \tag{4.16}$$

where z_m takes value 1 if a server is located on node m and 0 otherwise, y_{vm}^{tn} takes value 1 if client v is served by a server located on node m for $n \in \Omega_t$ at stage t and 0 otherwise, and o_m^{tn} is the over capacity used by server m for the node $n \in \Omega_t$ at stage t.

The objective function (4.7) and constraints (4.8)-(4.11) linearize the dynamic coherent risk measure (2.13) defined by mean-CVaR at each stage. The constraints (4.12) and (4.13) are capacity and allocation constraints, respectively. The variables in (4.14) are the original problem variables. The auxiliary variables in (4.15) are used to linearize the mean-CVaR risk measure. Finally, variables in (4.16) are due to definition of mean-CVaR.

A test problem of risk-averse multi-stage SSLP is represented as T-SSLP-M-V-b where T is the number of stages, M is the number of potential server location nodes, V is the number of clients and b is the number of different values that

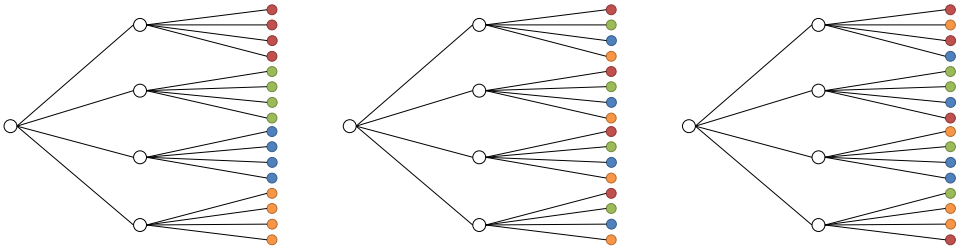
Table 4.2: Problem statistics for risk-averse multi-stage SSLP instances.

Number of stages	Test problem	Number of scenarios	Number of constraints	Number of binary variables
2	2-SSLP-5-25-50	50	1,601	6,255
	2-SSLP-5-25-100	100	3,201	12,505
	2-SSLP-10-50-50	50	3,101	25,010
	2-SSLP-10-50-100	100	6,201	50,010
	2-SSLP-10-50-500	500	31,001	250,010
	2-SSLP-10-50-1000	1,000	62,001	500,010
	2-SSLP-10-50-2000	2,000	124,001	1,000,010
3	3-SSLP-5-25-10	100	3,531	13,755
	3-SSLP-5-25-20	400	13,461	52,505
	3-SSLP-5-25-50	2,500	81,651	318,755
	3-SSLP-5-25-100	10,000	323,301	1,262,505
	3-SSLP-10-50-10	100	6,831	55,010
	3-SSLP-10-50-20	400	26,061	210,010
	3-SSLP-10-50-50	2,500	158,151	1,275,010
4	4-SSLP-5-25-10	1,000	35,631	138,755
	4-SSLP-5-25-20	8,000	269,861	1,052,505
	4-SSLP-10-50-10	1,000	68,931	555,010

the random vector h^t can take at stage t. Therefore, the number of scenarios in T-SSLP-M-V-b is b^{T-1}. Similar to [53], the problem parameters are selected as $c_m \sim \mathcal{U}[40, 80], d^t_{vm} \sim \mathcal{U}[0, 25], u = 2\frac{\sum_{v=1}^{V} \max_{m,n,t} d^{tn}_{vm}}{M}$, and $q^t_m = 1000$. Here, $\mathcal{U}[a, b]$ indicates that the respective value is sampled form the uniform distribution with range $[a, b]$. Problem statistics for risk-averse multi-stage SSLP instances used in the experiments are presented in Table 4.2.

In the proposed algorithm, for each instance of all problems presented in Table 4.2, we consider a partition with two groups with equal cardinalities, that is, $\mathcal{S} = \{S_1, S_2\}$ with $|S_1| = |S_2|$. The partitions are constructed in three different ways by using the scenario tree structure or randomly. In *similar* partition, the groups are generated by placing the scenarios with large number of common nodes in the scenario tree into the same group. However, in *different* partition, we place the scenarios with small number of common nodes into the same group. Finally, in *random* partition, the scenarios are assigned to the groups randomly. Note that for two-stage problems, all partitioning strategies are equivalent since

Figure 4.3: The partitions *similar*, *different* and *random* (from left to right, respectively) for a three-stage scenario tree where each color represents a group.



the locations of nodes in the second stage of the scenario tree are interchangeable. For a deeper discussion on the construction of partitions and their impact on the quality of bounds, we refer to [54] and [46] where various extensions on scenario grouping are considered. However, as seen in the results of our computational experiments, using a simple scenario grouping strategy is adequate for the proposed algorithm. Figure 4.3 depicts the three different scenario partitions we consider on a three-stage scenario tree.

In Table 4.3, we report the number of instances solved optimally (# opt out of five), average optimality gap (Gap) and solution time in seconds (Time) for DEP. Two hours of time limit is imposed when solving DEP of each instance. We also report the average number of iterations (# iter) and running time in seconds (Time) for the proposed algorithm with the three different partitions we consider. Note that, all reported values are averages for five instances of each problem.

The computational experiments reveal that as problem size grows, CPLEX fails to solve DEPs of the problem instances. For example, the problems 2-SSLP-10-50-2000 and 3-SSLP-10-50-50 have over one million binary variables, none of the DEPs are solved optimally within two hours of time limit for any value of DRA. Especially, 3-SSLP-10-50-50 instances terminate with large optimality gap values (59.11%, 85.64% and 27.91%, for DRA I, II, and III, respectively). However, the proposed algorithm solves all of these instances optimally in less than two hours with at least one of the three partitions we consider.

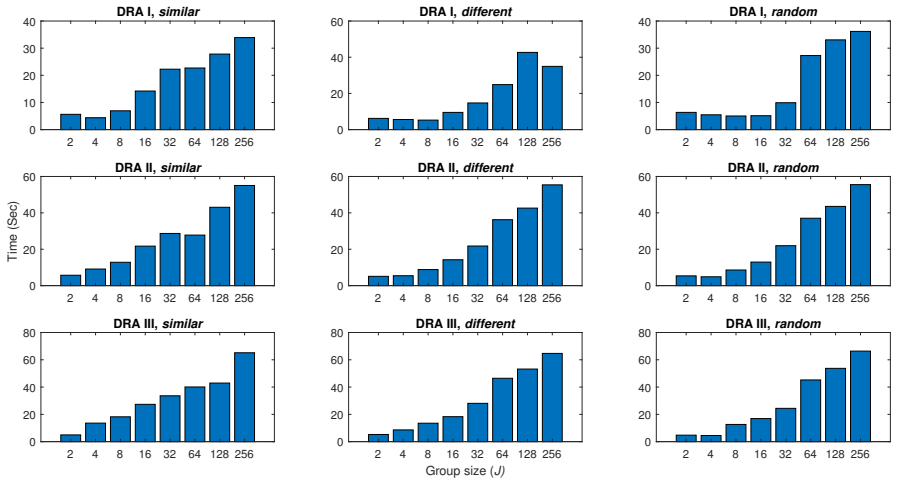
In Table 4.3, the bold entries are the smallest running times among the three different partitions for each problem. Moreover, in the last row of the table, for

each partitioning strategy, we present the percentage of test problems in which this strategy performs better than the other ones. When the three partitioning strategies are compared in the two-stage problems, no significant difference is observed as expected. However, for three- and four-stage problems, the *similar* partition performs better than *different* and *random* partitions. Since each group subproblems obtained by the *similar* partition includes scenarios with many common nodes of the scenario tree, these group subproblems have less complicated structure than the ones obtained by other partitions. Hence, in the lower bounding phase of the proposed algorithm, the *similar* partition yields group subproblems that are easier to solve. On the other hand, the lower bounds and candidate solutions provided by *different* partition are expected to be better than the ones given by *similar partition*. Overall, *similar* is the best scenario partition choice among all partitions in 47.06% of all instances. This value is 37.25% and 15.69% for *different* and *random* partitions, respectively.

When we analyze the test problem 4-SSLP-5-25-20 at DRA I, it is observed that CPLEX could only solve 2 out of 5 DEP instances optimally within the time limit and the average optimality gap is 10.36%. On the other hand, our algorithm could solve all of the five instances optimally and the average computation time of our proposed algorithm is 1830.9, 2539.3 and 2795.1 seconds for the *similar*, *different* and *random* partitions, respectively. We observe similar results for the test problem 4-SSLP-5-25-20 at DRA II as well. CPLEX could only solve 3 out of 5 DEP optimally within the time limit of 7200 seconds. The average gap is 3.37% for those instances. We again could solve all of the five instances optimally and the average computation time of our proposed algorithm is 1205.5, 1849.1 and 1957.0 seconds for the *similar*, *different* and *random* partitions, respectively.

Although the proposed algorithm with the simplest choice of the number of groups $J = 2$ enables us to solve the large scale instances efficiently, we also investigate the performance of the algorithm by changing the value of J. As the number of groups J increases, the number of scenarios per group subproblem decreases. In that case, it can be expected that the group subproblems are easier to solve and hence the running time of Algorithm 2 decreases with the number of groups J.

Figure 4.4: Average running time (in seconds) of the proposed algorithm with respect to different number of groups for five instances of 3-SSLP-5-25-16 problem with different degrees of risk-aversion and partitions.



In Figure 4.4, we present the average running time of the proposed algorithm in seconds with respect to different number of groups $J \in \{2, 4, 8, 16, 32, 64, 128, 256\}$ for five instances of 3-SSLP-5-25-16 problem with different degrees of risk-aversion and partitions.

Since the total number of scenarios in 3-SSLP-5-25-16 instances is 256, the number of scenarios in a group subproblem is 128, 64, 32, 16, 8, 4, 2 and 1 for the respective values 2, 4, 8, 16, 32, 64, 128 and 256 of J. Although it may be expected that the running time of Algorithm 2 decreases with increasing number of groups J, we observe the contrary in the results of our experiments in general. There can be two reasons of this. First, in the upper bounding phase, the algorithm requires evaluation of each candidate solution in $\mathcal{D}$. When the number of group subproblems is large, we can expect the cardinality of the set $\mathcal{D}$ to be large as well. Therefore, large values of J may yield evaluation of upper bound values for large number of candidate solutions. The second reason is that a larger number of groups J yields looser lower bounds and therefore the running time of the algorithm increases. In the extreme case of $J = 256$, the performance of the proposed algorithm is the worst compared to the other values of J. Note that this

result motivates to use scenario grouping instead of pure scenario decomposition. Nonetheless, increasing the J value can still be beneficial for some instances as shown in Figure 4.4. For example, for the instances with DRA I and *different* partition, the running time decreases as J increases from 2 through 8.

4.5.2 Generation Expansion Problem

The generation expansion problem (GEP) is an optimization problem that appears in power systems. The objective of GEP is to minimize the total cost due to construction and generation decisions for different types of generators over a fixed-length decision horizon. We use the mathematical model used in [27] where the problem data are adapted from [55].

In the deterministic version of the problem, the power demand at stage $t \in \{1, \ldots, T\}$ is given as d_t and it is assumed that I types of power generators are available.The fixed construction and unit production costs of a type $i \in \{1, \ldots, I\}$ generator are a_i and b_i, respectively. There is a limit u_i on the total number of type i generators constructed over the decision horizon. Moreover, the production amount of a type i generator cannot exceed U_i. The deterministic GEP is given as

$$\min \sum_{t=1}^{T} \sum_{i=1}^{I} \left(a_i z_{ti} + b_i y_{ti} \right), \tag{4.17}$$

$$\text{s.t. } y_{ti} \leq \left(\sum_{\tau=1}^{t} z_{\tau i} \right) U_i, \quad \forall i \in \{1, \ldots, I\}, t \in \{1, \ldots, T\}, \tag{4.18}$$

$$\sum_{t=1}^{T} z_{ti} \leq u_i, \quad \forall i \in \{1, \ldots, I\}, \tag{4.19}$$

$$\sum_{i=1}^{I} y_{ti} \geq d_t, \quad \forall i \in \{1, \ldots, I\}, \tag{4.20}$$

$$y_{ti} \geq 0, \quad z_{ti} \in \mathbb{Z}_+, \quad \forall i \in \{1, \ldots, I\}, t \in \{1, \ldots, T\}, \tag{4.21}$$

where z_{ti} is the number of generators of type $i \in \{1, \ldots, I\}$ constructed at stage

$t \in \{1, \ldots, T\}$ and y_{ti} is the total production amount of type i generators at stage t. The objective function (4.17) is the sum of construction and production costs over T stages. Constraints (4.18) and (4.19) ensure that the production schedule is feasible. The demand satisfaction is ensured in constraint (4.20). Domain restrictions are given in constraint (4.21).

We consider a risk-averse version of GEP where the demand at stages $t = 2, \ldots, T$ is random. The objective of risk-averse GEP is to determine the number of generators to be constructed and production amounts for each type of generators so as to minimize the dynamic coherent risk defined in (2.13) with conditional mean-CVaR at each stage. The DEP of risk-averse GEP can be obtained easily by defining additional variables and constraints similar to DEP (4.7) - (4.16) of SSLP.

A test problem of risk-averse GEP is represented as T-GEP-b where T is the number of stages and b is the number of different values that the random demand d_t can take at each stage $t \in \{2, \ldots, T\}$. Problem statistics for the risk-averse GEP instances used in the experiments are given in Table 4.4.

The deterministic problem parameters are given in Table 4.5. We assume that the first stage demand is $d_1 = D/2$ and the demand at stages $t \in \{2, \ldots, T\}$ is $d_t \sim U[D/4, 3D/4]$ where D is the total production capacity, that is, $D = \sum_{i=1}^{I} U_i u_i$.

Since both integer and continuous variables appear in all stages of GEP, we can use the extension of the proposed method presented in Section 4.4 to solve risk-averse GEP. Therefore, we binarize the integer variables in the first stage of GEP and perform the evaluate-and-cut procedure in Algorithm 2 for only these variables. In this case, the problem in the upper bounding phase of the algorithm does not enjoy decomposition.

In computational experiments on GEP instances, we consider *similar* partition with $J = 2$ and 4 where cardinalities of the groups of a partition are the same. In Table 4.6, we report the number of instances solved optimally (# opt out of five)

and solution time in seconds (Time) for DEP. Four hours of time limit is imposed when solving DEP of each instance. We also report the average number of iterations (# iter) and running time in seconds (Time) of the proposed algorithm for $J = 2$ and 4. As before, all reported values are averages of five instances of each test problem.

For small instances such as 3-GEP-10, 3-GEP-20, 3-GEP-40, 4-GEP-10 and 4-GEP-20 solution time of DEP is less than the running time of the algorithm for all degrees of risk aversion. For a moderate instance 3-GEP-100, the algorithm with $J = 2$ outperforms solving DEP for DRA I and II. For the largest instance, namely 4-GEP-50, CPLEX either does not report any feasible solution of DEP within the time limit or terminates with memory error before four hours. The algorithm with $J = 2$ also does not terminate for these instances in four hours. However, the algorithm with $J = 4$ solves all 4-GEP-50 instances in 11089.7, 11188.5 and 12850.2 seconds for DRA I, II and III, respectively. Although $J = 2$ is a better choice for small and moderate instances, for larger instances $J = 4$ yields better results.

The results of the computational experiments on risk-averse multi-stage SSLP and GEP demonstrate the effectiveness of our proposed algorithm. They also verify our initial claim that it is an easily implementable exact solution algorithm for risk-averse mixed-integer multi-stage stochastic problems with an objective of dynamic mean-CVaR and binary first stage decisions. Although, in GEP, we partially binarize the first stage variables and do not benefit decomposition in the upper bounding phase, the algorithm is able to solve large instances of this problem.

4.6 Conclusion

Risk-averse mixed-integer multi-stage stochastic programming problems form a class of challenging large scale and non-convex optimization problems. Moreover, no exact solution algorithm is available for these problems in the literature. In this

paper, we propose an exact solution algorithm for risk-averse mixed-integer multi-stage stochastic programming problems with an objective of dynamic mean-CVaR risk measure and binary first stage decisions. Later, we prove that the algorithm can be used to solve the general risk-averse mixed-integer multi-stage stochastic programming problems.

The proposed algorithm is based on an evaluate-and-cut procedure and a simple scenario tree decomposition method. The computational experiments on large instances of risk-averse multi-stage SSLP reveal that the proposed algorithm requires significantly less computational effort than solving the deterministic equivalent problem with CPLEX. Moreover, an extension of the proposed method is used to solve large instances of risk-averse GEP where mixed-integer decisions appear in all stages. We also discuss the effect of implementation details, such as partitioning strategy and number of groups in a partition, on the performance of the algorithm.

Since we did not make any structural assumptions on the problem structure such as complete or relative recourse, stage-wise independency, convexity, and linearity of feasibility constraints, our proposed algorithm could be applied to a wide range of problems.

Algorithm 2 An exact solution algorithm for problem (4.1).

1: **Input:** An instance of problem (4.1) and a partition $\mathcal{S}$ of scenarios.
2: **Initialize:**
3: Incumbent solution $x^* \leftarrow$ `null`,
4: Set of candidate first stage solutions $\mathcal{D} \leftarrow \emptyset$,
5: $LB \leftarrow -\infty$, $UB \leftarrow +\infty$.
6: **while** $UB > LB$ and $\mathcal{X}_1 \setminus \mathcal{D} \neq \emptyset$ **do**
7: `Lower Bounding`
8: **for all** $j \in \{1, 2, \ldots, J\}$ **do**
9: Solve the j^{th} group subproblem with first stage feasible set $\mathcal{X}_1 \setminus \mathcal{D}$ by relaxing integrality requirements at stages $\{2, 3, \ldots, T\}$.
10: **if** The group subproblem is infeasible **then**
11: Terminate and go to line 27
12: **else**
13: Let $\tilde{z}_j$ be the optimal value of the group subproblem.
14: Let x'_1 be an optimal first stage solution of the group subproblem.
15: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \{x'_1\}$
16: **end if**
17: **end for**
18: $LB \leftarrow \sum_{j=1}^{J} p_j \tilde{z}_j$
19: `Upper Bounding`
20: **for all** $x'_1 \in \mathcal{D}$ **do**
21: Solve problem (4.1) with first stage decision x'_1 i.e. $x_1 = x'_1$. Let $\overline{x}$ and $\overline{z}$ be an optimal solution and the optimal value, respectively. If the problem is infeasible $\overline{z} \leftarrow \infty$.
22: **if** $\overline{z} < UB$ **then**
23: $UB \leftarrow \overline{z}$ and $x^* \leftarrow \overline{x}$
24: **end if**
25: **end for**
26: **end while**
27: **Return:** x^* and UB.

Table 4.3: Computational study results for risk-averse multi-stage SSLP.

	Test Problem	DEP			Proposed Algorithm					
					similar		*different*		*random*	
		# opt	Gap	Time	# iter	Time	# iter	Time	# iter	Time
DRA I	2-SSLP-5-25-50	5	-	0.3	2	0.9	2	1.0	2	**0.8**
	2-SSLP-5-25-100	5	-	0.8	2	1.7	2	1.6	2	**1.5**
	2-SSLP-10-50-50	5	-	9	2.6	**5.8**	2	6.1	2.4	6.4
	2-SSLP-10-50-100	5	-	180.1	2.2	**11.7**	2	13.6	2.2	13.5
	2-SSLP-10-50-500	0	0.03%	7200	2	**201.3**	2	254.4	2	216.2
	2-SSLP-10-50-1000	0	0.89%	7200	2	**757.6**	2.4	793.9	2	761.1
	2-SSLP-10-50-2000	0	16.45%	7200	2	3511.9	2	**3487.2**	2	3654.0
	3-SSLP-5-25-10	5	-	3	2	**1.5**	2	2.0	2	1.8
	3-SSLP-5-25-20	5	-	15.1	2.2	10.4	2	**9.3**	2	9.5
	3-SSLP-5-25-50	5	-	1270	2	**193.8**	2	219.0	2	217.1
	3-SSLP-5-25-100	1	11.39%	7143	2	**4292.9**	2	4711.4	2	4867.9
	3-SSLP-10-50-10	5	-	82.1	2.4	15.9	2	**14.5**	2	15.1
	3-SSLP-10-50-20	5	-	2273.8	2.2	**201.2**	2	204.2	2	229.7
	3-SSLP-10-50-50	0	59.11%	7200	2	**7117.1**	2	7882.9	2	8597.9
	4-SSLP-5-25-10	5	-	112.6	2.2	**44.3**	2	48.6	2.2	55.1
	4-SSLP-5-25-20	2	10.36%	6914.2	2	**1830.9**	2	2539.3	2	2795.1
	4-SSLP-10-50-10	2	0.87%	6913.1	4.2	2758.6	2.4	**1775.0**	2.4	1822.9
DRA II	2-SSLP-5-25-50	5	-	0.4	2.2	1.0	2	0.9	2	**0.7**
	2-SSLP-5-25-100	5	-	0.8	2.2	1.6	2	**1.3**	2	1.5
	2-SSLP-10-50-50	1	0.07%	5940.6	2.4	7.1	2.6	**6.8**	2.6	6.9
	2-SSLP-10-50-100	0	0.10%	7200	2.4	15.2	2.4	17.9	2.2	**14.5**
	2-SSLP-10-50-500	0	1.21%	7200	2	216.2	2	**203.9**	2	205.5
	2-SSLP-10-50-1000	0	4.37%	7200	2	**760.9**	2.2	782.2	2.2	821.7
	2-SSLP-10-50-2000	0	14.94%	7200	2	**3390.2**	2	3506.1	2	3442.8
	3-SSLP-5-25-10	5	-	1.8	2.6	2.1	2	**1.3**	2.4	2.3
	3-SSLP-5-25-20	5	-	10.2	2	**7.3**	2	10.2	2	10.7
	3-SSLP-5-25-50	5	-	417.1	2	**163.1**	2	174.0	2.2	199.4
	3-SSLP-5-25-100	1	20.02%	7101.9	2	3519.2	2	3991.3	2	**3181.8**
	3-SSLP-10-50-10	5	-	72.4	2.8	18.0	2.2	**17.0**	2	25.4
	3-SSLP-10-50-20	5	-	1270.5	2.4	243.4	2.8	**202.1**	2.6	311.6
	3-SSLP-10-50-50	0	85.64%	7200	3	6231.4	2.2	**5818.9**	2.2	6152.9
	4-SSLP-5-25-10	5	-	93.8	2.6	50.1	2.4	**43.4**	2.4	61.7
	4-SSLP-5-25-20	3	3.37%	5699.2	2	**1205.5**	2	1849.1	2	1957.0
	4-SSLP-10-50-10	1	1.15%	6559	7.8	**6117.4**	2.2	7837.1	3	8766.2
DRA III	2-SSLP-5-25-50	5	-	0.3	2.2	**0.9**	2.2	1.0	2.4	0.8
	2-SSLP-5-25-100	5	-	0.9	2.2	1.6	2.2	**1.5**	2.2	1.7
	2-SSLP-10-50-50	0	0.02%	7200	2.6	6.7	2.6	7.2	2.2	**5.8**
	2-SSLP-10-50-100	0	0.04%	7200	2.2	**13.2**	2.4	14.4	2.4	15.7
	2-SSLP-10-50-500	0	3.22%	7200	2.2	212.3	2	216.0	2	**200.5**
	2-SSLP-10-50-1000	0	7.41%	7200	2	656.6	2	**649.2**	2	695.6
	2-SSLP-10-50-2000	0	15.59%	7200	2	**3077.3**	2.2	3245.8	2.2	3325.7
	3-SSLP-5-25-10	5	-	1.2	2.8	**2.4**	2.4	2.6	3.2	3.3
	3-SSLP-5-25-20	5	-	15.2	2.2	8.5	2.2	**8.4**	2.4	12.8
	3-SSLP-5-25-50	5	-	398.5	2	155.1	2	**134.7**	2	149.2
	3-SSLP-5-25-100	2	13.80%	6259.6	2	2483.6	2	**2446.5**	2	2643.6
	3-SSLP-10-50-10	5	-	37.9	15.8	**127.9**	12.4	249.7	9.8	212.2
	3-SSLP-10-50-20	5	-	1305.8	4.8	462.0	2	**247.2**	4	485.3
	3-SSLP-10-50-50	0	27.91%	7200	2.6	6286.8	2.8	5254.2	2	**4444.3**
	4-SSLP-5-25-10	5	-	67.9	4.4	**86.7**	3.4	91.6	3.2	124.9
	4-SSLP-5-25-20	4	2.84%	3779	2.2	1539.8	2	**939.5**	2.8	2267.4
	4-SSLP-10-50-10	2	1.34%	5547.4	6.4	**4288.5**	2.6	6760.2	5.2	6844.5
The percentage of test problems the partition performs best						**47.06%**		**37.25%**		**15.69%**

Table 4.4: Problem statistics for risk-averse GEP instances.

Number of stages	Test problem	Number of scenarios	Number of constraints	Number of integer variables
3	3-GEP-10	100	1,608	666
	3-GEP-20	400	6,208	2,526
	3-GEP-40	1,600	24,408	9,846
	3-GEP-100	10,000	151,008	60,606
	3-GEP-200	40,000	602,008	241,206
4	4-GEP-10	1,000	16,708	6,666
	4-GEP-20	8,000	126,608	50,526
	4-GEP-50	125,000	1,915,508	765,306

Table 4.5: Parameters of GEP instances.

i	1	2	3	4	5	6
a_i	1,446,000	795,000	575,000	1,613,000	1,650,000	1,671,000
b_i	16.62	38.82	43.17	0.51	5.00	16.91
U_i	412,450	142,350	138,700	430,700	63,875	204,400
u_i	1	2	2	1	3	1

Table 4.6: Computational study results for risk-averse GEP.

	Test Problem	DEP		Proposed Algorithm			
				$J = 2$		$J = 4$	
		#opt	Time	#iter	Time	#iter	Time
DRA I	3-GEP-10	5	0.1	6.8	1.7	8.6	3.6
	3-GEP-20	5	0.5	5.2	3.6	7	7.9
	3-GEP-40	5	8.4	2.8	15.2	3.2	19.6
	3-GEP-100	5	176.0	2	65.8	2.8	108.0
	3-GEP-200	5	6227.0	2.2	2115.4	2.4	718.9
	4-GEP-10	5	3.2	6.6	14.4	8.6	28.8
	4-GEP-20	5	81.1	5.2	161.9	8.6	336.5
	4-GEP-50	0	14400	***	***	11.6	11089.7
DRA II	3-GEP-10	5	0.1	6.6	1.6	12.8	4.6
	3-GEP-20	5	0.5	5.6	5.6	14	15.3
	3-GEP-40	5	8.5	4.4	25.0	5.8	43.9
	3-GEP-100	5	146.1	3	101.1	4	170.3
	3-GEP-200	5	5617.7	3	3106.1	3.2	909.9
	4-GEP-10	5	2.8	7.4	16.0	14.6	48.3
	4-GEP-20	5	70.5	8.8	277.3	15	592.7
	4-GEP-50	0	14400	***	***	16.6	11188.5
DRA III	3-GEP-10	5	1.4	7.6	1.9	21.8	8.1
	3-GEP-20	5	6.3	8.8	6.7	27.8	31.1
	3-GEP-40	5	8.1	7	40.3	11	88.5
	3-GEP-100	5	95.7	4.2	143.9	7	276.9
	3-GEP-200	5	5246.5	4.6	4712.6	5.8	1670.3
	4-GEP-10	5	7.9	14.2	31.3	24.6	83.7
	4-GEP-20	5	74.0	10.4	342.2	35.2	1361.4
	4-GEP-50	0	14400	***	***	36.4	12850.2

Chapter 5

The Value of Adaptive Commitment in Risk-averse Unit Commitment under Uncertainty

Unit commitment is an important optimization problem emerging from power systems. Uncertainty and variability in net load arising from the increasing penetration of renewable technologies have motivated study of various classes of stochastic unit commitment models in recent years. In the models with non-adaptive commitment, the generation schedule for the entire day is fixed whereas the dispatch is adapted to the uncertainty. On the other hand, in the models with adaptive commitment, the generation schedule is also allowed to dynamically adapt to the realization uncertainty. The latter ones provide more flexibility in the generation schedule, however, they require significantly higher computational effort than the former ones. In order to justify this additional computational effort, in this chapter, we provide theoretical and empirical analyses of the value of adaptive commitment for risk-averse multi-stage stochastic unit commitment models.

The rest of the chapter is organized as follows: In Section 5.1, we define

the risk-averse unit commitment problem and present stochastic models with or without adaptive commitment decisions. In Section 5.2, we define the value of adaptive commitment and provide analytical bounds for it. In Section 5.3, we present results of computational experiments. In Section 5.4, concluding remarks are presented.

The results of this chapter are available online at `arxiv.org/abs/1808.00999`.

5.1 Motivation and Problem Definition

Unit commitment (UC) is a challenging optimization problem used for day-ahead generation scheduling given net load forecasts and various operational constraints [56]. The output schedule includes on-off status of generators (commitment decisions) and the production amounts (economic dispatch decisions) for every time period.

There has been a great deal of research on deterministic UC models where the problem parameters are assumed to be known exactly (see, for example, [57] and references therein). However, these models cannot capture variability and uncertainty in the nature of the problem. Common sources of uncertainty are departures from forecasts and unreliable equipment. The departures from forecasts generally stem from the variability in net load and production amounts, whereas unreliable equipment may result in generator and transmission line outages a (see, [58] and [59], for a technical discussion). Moreover, the penetration of renewable energy has increased the volatility of power systems in recent years since the production amount of energy from wind and solar power are not controllable but can only be forecasted.

Robust optimization and stochastic programming are two common frameworks used to address the uncertainty in UC problems. In robust optimization models, it is assumed that the uncertain parameters take values in some uncertainty sets and the objective is to minimize the worst case cost ([60], [61], [62], [63] and [64]).

In stochastic programming models, the uncertainty is represented by a probability distribution ([65], [66], [67], [68] and [69]).

In UC models with non-adaptive commitment, the generation schedule is fixed for the entire day before the beginning of the day while dispatch is adapted to uncertainty as in [70] and [71]. On the other hand, the models with adaptive commitment, both the generation schedule and dispatch are allowed to dynamically adapt to uncertainty realization at each hour (see for example, [69], [72] and [73]). Therefore, they incorporate multistage forecasting information with varying accuracy and express relation between time periods appropriately. However, in general, the models with adaptive commitment are computationally difficult than the models with non-adaptive commitment. A detailed comparison can be found in [74] and [75].

The computational challenge of the models with adaptive commitment motivates the question on whether the effort to solve them is worthwhile. In [45], this question is addressed for a risk-neutral stochastic capacity planning problem. In this chapter, we address this question for risk-averse unit commitment problems where the objective is a dynamic measure of risk. We provide theoretical and empirical analysis on the value adaptive commitment which measures the relative advantage to solve the risk-averse unit commitment models with adaptive commitment over the models with non-adaptive commitment.

We first present an abstract deterministic formulation of the UC problem. Let I be the number of generators and T be the number of periods. Also, let $\mathcal{I} := \{1, \ldots, I\}$ and $\mathcal{T} := \{1, \ldots, T\}$ be the sets of generators and time periods, respectively. A canonical formulation of the UC problem is as follows:

$$\min \sum_{t=1}^{T} f_t(\boldsymbol{u}_t, \boldsymbol{v}_t, \boldsymbol{w}_t) \tag{5.1}$$

$$\text{s.t.} \sum_{i=1}^{I} v_{it} \geq d_t, \ \forall t \in \mathcal{T} \tag{5.2}$$

$$\underline{q}_i u_{it} \leq v_{it} \leq \overline{q}_i u_{it}, \ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{5.3}$$

$$(\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) \in \mathcal{X}_1, \tag{5.4}$$

$$(\boldsymbol{u}_t, \boldsymbol{v}_t, \boldsymbol{w}_t) \in \mathcal{X}_t(\boldsymbol{u}_{t-1}, \boldsymbol{v}_{t-1}, \boldsymbol{w}_{t-1}),\ \forall t \in \mathcal{T} \setminus \{1\} \tag{5.5}$$

$$\boldsymbol{u}_t \in \{0,1\}^I, \boldsymbol{v}_t \in \mathbb{R}^I_+, \boldsymbol{w}_t \in \mathbb{R}^k,\ \forall t \in \mathcal{T} \tag{5.6}$$

Decision variables u_{it} and v_{it} represent the binary on/off status and production of generator $i \in \mathcal{I}$ in period $t \in \mathcal{T}$, respectively. The bold symbols $\boldsymbol{u}_t := (u_{1t}, u_{2t}, \ldots, u_{It})$ and $\boldsymbol{v}_t := (v_{1t}, v_{2t}, \ldots, v_{It})$ are the vectors of status and production decisions in period $t \in \mathcal{T}$, respectively. The vector $\boldsymbol{w}_t$ denotes auxiliary variables associated with period $t \in \mathcal{T}$. These variables can be used for modeling various operational constraints. The objective (5.1) is the sum of production, start-up and shut-down costs in all periods where the function $f_t(\cdot)$ represents the total cost in a period $t \in \mathcal{T}$. Constraint (5.2) ensures satisfaction of the power demand where the net load d_t is equal to the total power demand minus the power generated from renewable sources. Constraint (5.3) enforces lower and upper production limits on the generators. Other operational restrictions such as transmission capacity constraints are represented by constraints (5.4) and (5.5). The temporal relationship between consecutive periods such as start-up, ramp-up, shut-down and ramp-down restrictions can also be included in the set constraint (5.5). Domain restrictions of the decision variables are given by constraint (5.6). An explicit deterministic model is given in Appendix A.

In the deterministic formulation (5.1)-(5.6), net load values are assumed to be known exactly. However, this is a restrictive assumption in practice. We assume that the net load is random and denoted by a random variable $\widetilde{d}_t$ in period $t \in \mathcal{T}$ from a probability space $(\Omega, \mathcal{F}, P)$. Here Ω is a sample space equipped with sigma algebra $\mathcal{F}$ and probability measure P. An element of the sample space Ω is called as a *scenario* (or a sample path) and represents a possible realization of the net load values in all periods. The sequence of sigma algebras $\{\emptyset, \Omega\} = \mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots \subset \mathcal{F}_T = \mathcal{F}$ is called as a *filtration* and it represents the gradually increasing information through the decision horizon $1, 2, \ldots, T$. The set of $\mathcal{F}_t-$measurable random variables is denoted by $\mathcal{Z}_t$ for $t \in \mathcal{T}$. The random net load $\widetilde{d}_t$ in period t is $\mathcal{F}_t-$measurable, that is $\widetilde{d}_t \in \mathcal{Z}_t$ for $t \in \mathcal{T}$. Note that since $\mathcal{F}_1 = \{\emptyset, \Omega\}$ by definition, $\mathcal{Z}_1 = \mathbb{R}$ and the demand in the first period is deterministic.

To extend the deterministic UC model to this uncertainty setting, we have that the decisions in period t to depend on realization of the history of net load process $\widetilde{d}_{[t]} := (\widetilde{d}_1, \ldots, \widetilde{d}_t)$ up to period t. Therefore, we use the $\mathcal{F}_t-$measurable vectors $\widetilde{\boldsymbol{u}}_t(\widetilde{d}_{[t]})$, $\widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]})$ and $\widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]})$ to represent status, production and auxiliary decisions in period $t \in \mathcal{T}$, respectively. The total cost at period t is also $\mathcal{F}_t-$measurable, i.e., $f_t(\widetilde{\boldsymbol{u}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]})) \in \mathcal{Z}_t$. We use conditional risk measures $\rho_{\mathcal{F}_{t+1}|\mathcal{F}_t} : \mathcal{Z}_{t+1} \to \mathcal{Z}_t$ in order to quantify the risk involved in a random cost at period $t+1$ based on the available informations at period t for $t \in \mathcal{T} \setminus \{T\}$. An example of a conditional risk measure is the *conditional mean-upper semi deviation*

$$\rho_{\mathcal{F}_{t+1}|\mathcal{F}_t}(Z_{t+1}) = \mathbb{E}[Z_{t+1}|\mathcal{F}_t] + \lambda \mathbb{E}[(Z_{t+1} - \mathbb{E}[Z_{t+1}|\mathcal{F}_t])_+|\mathcal{F}_t], \tag{5.7}$$

where $\mathbb{E}[Z_{t+1}|\mathcal{F}_t]$ is the conditional expectation with respect to the sigma algebra $\mathcal{F}_t$, $\lambda \in [0,1]$ is a parameter controlling the degree of risk aversion and $(Z_{t+1})_+$ is the point-wise positive part function for all $Z_{t+1} \in \mathcal{Z}_{t+1}$.

The objective of the risk averse UC (RA-UC) problem is to minimize the risk involved with the cost sequence $\{Z_t\}_{t=1}^T$ where $Z_t := f_t(\widetilde{\boldsymbol{u}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]}))$ is a shorthand notation for the total cost in period $t \in \mathcal{T}$. Thus, as in [3] and [37], we define the dynamic coherent risk measure $\varrho : \mathcal{Z}_1 \times \mathcal{Z}_2 \times \cdots \times \mathcal{Z}_T \to \mathbb{R}$ by using nested composition of the conditional risk measures $\rho_{\mathcal{F}_2|\mathcal{F}_1}(\cdot), \rho_{\mathcal{F}_3|\mathcal{F}_2}(\cdot), \ldots, \rho_{\mathcal{F}_T|\mathcal{F}_{T-1}}(\cdot)$, that is,

$$\varrho(Z_1, Z_2, \ldots, Z_T) := Z_1 + \rho_{\mathcal{F}_2|\mathcal{F}_1}(Z_2 + \cdots \rho_{\mathcal{F}_T|\mathcal{F}_{T-1}}(Z_T) \cdots)$$

is the risk associated with this cost sequence. Due to translational equivariance property of conditional risk measures, we have an alternative representation of the dynamic coherent measure of risk $\varrho(\cdot)$ as

$$\rho\left(\sum_{t=1}^{T} Z_t\right) := \varrho(Z_1, Z_2, \ldots, Z_T) \tag{5.8}$$

where $\rho = \rho_{\mathcal{F}_2|\mathcal{F}_1} \circ \rho_{\mathcal{F}_3|\mathcal{F}_2} \circ \cdots \circ \rho_{\mathcal{F}_T|\mathcal{F}_{T-1}} : \mathcal{Z} \to \mathbb{R}$ is called as a *composite risk measure* and $\mathcal{Z} := \mathcal{Z}_T$. The composite risk measure $\rho(\cdot)$ satisfies the coherence axioms (A1)-(A4). Therefore, $\rho(\cdot)$ is a coherent risk measure as shown in [3, Eqn.

Figure 5.1: Order of decisions in the model with non-adaptive commitment.

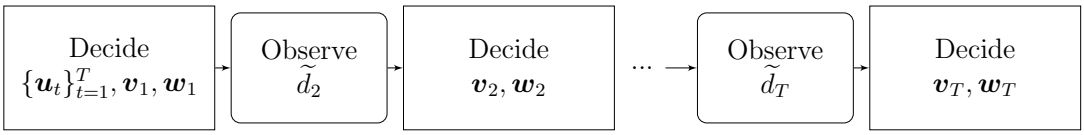
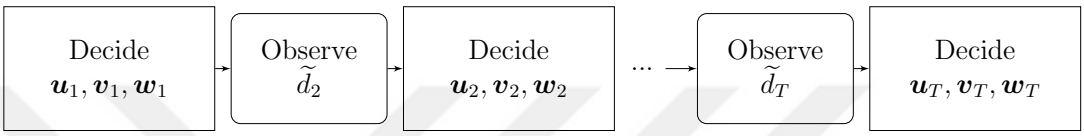


Figure 5.2: Order of decisions in the model with adaptive commitment.



6.234].

We consider two different models for the RA-UC problem. In the model with non-adaptive commitment, the on/off status decisions are fixed at the beginning of the day and production (or dispatch) decisions are adapted to uncertainty in the random demand. On the other hand, in the model with adapted commitment, both the status and production decisions are fully adapted to uncertainty in net load. In order to clarify the distinction between two models, the decision dynamics are depicted in 5.1 and 5.2, respectively.

The model with non-adaptive commitment (NC) for the RA-UC problem is given as

$$\min \rho \left[\sum_{t=1}^{T} f_t(\boldsymbol{u}_t, \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]})) \right], \tag{5.9}$$

$$\text{s.t.} \sum_{i \in \mathcal{I}} \widetilde{v}_{it}(\widetilde{d}_{[t]}) \geq \widetilde{d}_t, \ \forall t \in \mathcal{T}, \tag{5.10}$$

$$\underline{q}_i u_{it} \leq \widetilde{v}_{it}(\widetilde{d}_{[t]}) \leq \overline{q}_i u_{it}, \ \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{5.11}$$

$$(\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) \in \mathcal{X}_1, \tag{5.12}$$

$$(\boldsymbol{u}_t, \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]})) \in \mathcal{X}_t(\boldsymbol{u}_{t-1}, \widetilde{\boldsymbol{v}}_{t-1}(\widetilde{d}_{[t-1]}), \widetilde{\boldsymbol{w}}_{t-1}(\widetilde{d}_{[t-1]}), \widetilde{d}_{[t]}), \ \forall t \in \mathcal{T} \setminus \{1\}, \tag{5.13}$$

$$\boldsymbol{u}_t \in \{0,1\}^I, \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}) \in \mathbb{R}_+^I, \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]}) \in \mathbb{R}^k, \ \forall t \in \mathcal{T}. \tag{5.14}$$

The objective (5.9) of NC is the composite risk measure defined in (5.8) applied to the total cost sequence. The inequalities (5.10) and (5.11) are analogous to the constraints (5.2) and (5.3), respectively. The set constraint (5.12) is identical to (5.4) since the net load in the first period is deterministic. In constraint (5.13), $\mathcal{X}_t$ is an $\mathcal{F}_t-$measurable feasibility set. The domain constraint (5.14) states that only production and auxiliary decisions depend on the demand history and the status decisions are deterministic. However, in the model with adaptive commitment of the RA-UC problem, all decisions are made based on the history. Hence, the model with adaptive commitment (AC) can be written as

$$\min \rho\left[\sum_{t=1}^{T} f_t(\widetilde{\boldsymbol{u}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]}))\right], \tag{5.15}$$

$$\text{s.t.} \sum_{i\in\mathcal{I}} \widetilde{v}_{it}(\widetilde{d}_{[t]}) \geq \widetilde{d}_t, \ \forall t \in \mathcal{T}, \tag{5.16}$$

$$\underline{q}_i \widetilde{u}_{it}(\widetilde{d}_{[t]}) \leq \widetilde{v}_{it}(\widetilde{d}_{[t]}) \leq \overline{q}_i \widetilde{u}_{it}(\widetilde{d}_{[t]}), \ \forall i \in \mathcal{I}, t \in \mathcal{T}, \tag{5.17}$$

$$(\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) \in \mathcal{X}_1, \tag{5.18}$$

$$(\widetilde{\boldsymbol{u}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]})) \in$$
$$\mathcal{X}_t(\widetilde{\boldsymbol{u}}_{t-1}(\widetilde{d}_{[t-1]}), \widetilde{\boldsymbol{v}}_{t-1}(\widetilde{d}_{[t-1]}), \widetilde{\boldsymbol{w}}_{t-1}(\widetilde{d}_{[t-1]}), \widetilde{d}_{[t]}), \forall t \in \mathcal{T} \setminus \{1\}, \tag{5.19}$$

$$\widetilde{\boldsymbol{u}}_t(\widetilde{d}_{[t]}) \in \{0,1\}^I, \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}) \in \mathbb{R}^I_+, \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]}) \in \mathbb{R}^k, \forall t \in \mathcal{T}. \tag{5.20}$$

Note that the model AC is identical with NC except that the status decisions are fully adaptive to the random net load process.

An optimal solution of either NC and AC is a policy that minimizes the value of the dynamic coherent risk measure in the corresponding problem. Both in NC and AC, the optimality of a policy should only be with respect to possible future realizations given the available information at the time when the decision is made. This principle is called as time consistency. In [20, Example 2], it is shown that time consistency enables us to use the composite risk measure in minimization among all possible decisions instead of nested minimizations in a dynamic coherent measure of risk. We prefer conditional risk measures to represent the risk-averse behavior of decision makers since they yield time consistent formulation of the problem and their interpretation is clear.

5.2 Value of Adaptive Commitment

Although an optimal solution of AC provides a more flexible day-ahead schedule with respect to different realizations of parameters, the number of binary variables in AC is proportional to $\mathcal{N} \times I$ where $\mathcal{N}$ is the number of possible demand realizations in all periods if Ω is finite. However, the number of binary variables in NC is proportional to $T \times I$. Since $\mathcal{N} >> T$ for any non-trivial problem, computational difficulty of AC is significantly more than NC. Therefore, it is important to figure out if the additional effort to solve AC is worthwhile. We define the value of adaptive commitment in order to quantify the relative advantage of the optimal policy with adaptive commitment decisions over its counterpart with non-adaptive commitment decisions.

Definition 1. *The value of adaptive commitment (VAC) is the difference between the optimal values of NC and AC, that is, VAC* $= z^{NC} - z^{AC}$ *where* z^{NC} *and* z^{AC} *are the optimal values of NC and AC, respectively.*

Since an optimal solution of AC provides more flexibility in status decisions with respect to uncertain net load realizations, we have $z^{NC} \geq z^{AC}$ and therefore VAC ≥ 0. The complex structure of risk-averse UC problem prohibits exact calculation of VAC unless both NC and AC are solved optimally. Even calculation of bounds for VAC is not possible for UC problem. Thus, we provide theoretical bounds on the VAC under some assumptions.

Assumption 2. *There exists a generator* $j^* \in \mathcal{I}$ *such that* $\underline{q}_{j^*} \leq \widetilde{d}_t \leq \overline{q}_{j^*}$ *with probability 1 with no minimum start up and shut down time and no rumping limits for each* $t \in \mathcal{T}$.

Assumption 2 ensures that NC and AC always have at least one feasible solution and therefore both problems have *complete recourse*. Assumption 2 holds, for example, when it is possible to outsource the unmet power demand. In that case, decisions u_{j^*} and v_{j^*} represent outsourcing decision and amount of outsourced energy, respectively. Alternatively, u_{j^*} and v_{j^*} can be used to formulate the opportunity cost due to lost demand.

Assumption 3. *There exists an upper bound $d_t^{\max} \in \mathbb{R}_+$ on the net load values such that $0 \leq \widetilde{d}_t \leq d_t^{\max}$ with probability 1 for each $t \in \mathcal{T}$.*

Assumption 3 holds in practice and states that the net load in each period is bounded. We also define $\widetilde{D} := \sum_{t=1}^{T} \widetilde{d}_t$ as the total net load and $D^{\max} := \sum_{t=1}^{T} d_t^{\max}$ as an upper bound on $\widetilde{D}$.

Assumption 4. *The production cost at each stage is defined as*

$$f_t(\widetilde{\boldsymbol{u}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]})) = \sum_{i \in \mathcal{I}} g_i(\widetilde{u}_{it}(\widetilde{d}_{[t]}), \widetilde{v}_{it}(\widetilde{d}_{[t]}), \widetilde{w}_{it}(\widetilde{d}_{[t]}))$$

where $g_i(\cdot)$ is sum of a fixed commitment cost and a non-decreasing convex dispatch cost for all $i \in \mathcal{I}$.

If Assumption 4 holds, the function $g_i(\cdot)$ can be written as

$$g_i(\widetilde{u}_{it}(\widetilde{d}_{[t]}), \widetilde{v}_{it}(\widetilde{d}_{[t]}), \widetilde{w}_{it}(\widetilde{d}_{[t]})) = a_i \widetilde{u}_{it}(\widetilde{d}_{[t]}) + h_i(\widetilde{v}_{it}(\widetilde{d}_{[t]}))$$

for a coefficient $a_i \geq 0$ and a non-decreasing convex function $h_i(\cdot)$ with $h_i(0) = 0$ for all $i \in \mathcal{I}$. Assumption 4 is somewhat restrictive since it ignores start-up and shut-down costs. However this assumption is necessary for the analytical results. In Section 5.3, we will provide numerical results showing that the analytical results hold in instances with start-up and shut-down costs as well.

Theorem 2. *Under Assumptions 2, 3 and 4, we have that*

$$\alpha_* D^{\max} - \alpha^* \rho(\widetilde{D}) \leq \text{ VAC } \leq \alpha^* D^{\max} - \alpha_* \rho(\widetilde{D}),$$

where

$$\alpha_* := \min_{i \in \mathcal{I}} \left\{ a_i + h_i(\underline{q}_i) \right\} \Big/ \max_{i \in \mathcal{I}} \{\overline{q}_i\} \text{ and}$$
$$\alpha^* := \max_{i \in \mathcal{I}} \{a_i + h_i(\overline{q}_i)\} \Big/ \min_{i \in \mathcal{I}} \left\{ \underline{q}_i \right\}$$

are cost related problem parameters corresponding to under and over estimations on per unit production costs, respectively.

Proof. Assumption 2 implies that both NC and AC are feasible. Since the net loads are bounded due to Assumption 3, both models have at least one optimal solution.

Let $\{\widetilde{\boldsymbol{u}}_t^*, \widetilde{\boldsymbol{v}}_t^*, \widetilde{\boldsymbol{w}}_t^*\}_{t\in\mathcal{T}}$ be an optimal policy obtained by solving AC. By Assumption 4, we have

$$\sum_{t\in\mathcal{T}} f_t(\widetilde{\boldsymbol{u}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{v}}_t(\widetilde{d}_{[t]}), \widetilde{\boldsymbol{w}}_t(\widetilde{d}_{[t]})) = \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} g_i(\widetilde{u}_{it}(\widetilde{d}_{[t]}), \widetilde{v}_{it}(\widetilde{d}_{[t]}), \widetilde{w}_{it}(\widetilde{d}_{[t]})).$$

For a realization $d_1, d_2, \ldots, d_T$ of the random net load process $\widetilde{d}_1, \widetilde{d}_2, \ldots, \widetilde{d}_T$, let $[u_t^*, v_t^*, w_t^*] := [\widetilde{\boldsymbol{u}}_t^*, \widetilde{\boldsymbol{v}}_t^*, \widetilde{\boldsymbol{w}}_t^*](d_{[t]})$ be the optimal status and production decisions for $t \in \mathcal{T}$. Then, we have

$$\begin{aligned}
&\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} g_i(u_{it}^*, v_{it}^*, w_{it}^*) = \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} a_i u_{it}^* + h_i(v_{it}^*) \\
&\geq \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} a_i u_{it}^* + h_i(\underline{q}_i u_{it}^*) = \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} a_i u_{it}^* + h_i(\underline{q}_i) u_{it}^* \\
&= \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} [a_i + h_i(\underline{q}_i)] u_{it}^* \geq \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} [a_i + h_i(\underline{q}_i)] \frac{v_{it}^*}{\overline{q}_i} \\
&\geq \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} \frac{\min\limits_{i\in\mathcal{I}} \left\{a_i + h_i(\underline{q}_i)\right\}}{\max\limits_{i\in\mathcal{I}} \{\overline{q}_i\}} v_{it}^* \\
&= \alpha_* \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} v_{it}^* = \alpha_* \sum_{t\in\mathcal{T}} d_t,
\end{aligned}$$

where the first inequality holds due to feasibility and non-deceasing monotonicity of $h_i(\cdot)$. The second inequity also follows from feasibility. The second equality holds since $h(qu) = h(q)u$ for any function $h : \mathbb{R} \to \mathbb{R}$ with $h(0) = 0$ where $q \in \mathbb{R}_+$ and $u \in \{0, 1\}$.

Since $\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} g_i(u_{it}^*, v_{it}^*, w_{it}^*) \geq \alpha_* \sum_{t\in\mathcal{T}} d_t$ for any sample path $d_1, d_2, \ldots, d_T$, we have $\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} g_i(\widetilde{u}_{it}(\widetilde{d}_{[t]}), \widetilde{v}_{it}(\widetilde{d}_{[t]}), \widetilde{w}_{it}(\widetilde{d}_{[t]})) \succeq \alpha_* \sum_{t\in\mathcal{T}} \widetilde{d}_t = \alpha_* \widetilde{D}$. Due to the monotonicity axiom (A2) and positive homogeneity axiom

(A4), we get

$$\begin{aligned} z^{AC} &= \rho\left(\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} g_i(\widetilde{u}_{it}(\widetilde{d}_{[t]}), \widetilde{v}_{it}(\widetilde{d}_{[t]}), \widetilde{w}_{it}(\widetilde{d}_{[t]}))\right) \\ &\geq \rho(\alpha_*\widetilde{D}) = \alpha_*\rho(\widetilde{D}). \end{aligned}$$

Next, we consider a feasible policy $\{\widehat{\boldsymbol{u}}_t^*, \widehat{\boldsymbol{v}}_t^*, \widehat{\boldsymbol{w}}_t^*\}_{t\in\mathcal{T}}$ to the AC model where $\widehat{u}_{j^*t}(\widetilde{d}_{[t]}) = 1$, $\widehat{v}_{j^*t}(\widetilde{d}_{[t]}) = \widetilde{d}_t$ and all other status and generation variables are set to zero for a sample path $d_1, d_2, \ldots, d_t$. The feasibility of the solution is guaranteed by Assumption 2. Then,

$$\begin{aligned} z^{AC} &\leq \rho\left(\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} g_i(\widehat{u}_{it}(\widetilde{d}_{[t]}), \widehat{v}_{it}(\widetilde{d}_{[t]}), \widehat{w}_{it}(\widetilde{d}_{[t]}))\right) \\ &= \rho\left(\sum_{t\in\mathcal{T}} a_{j^*} + h_{j^*}(\widetilde{d}_t)\right) = \rho\left(\sum_{t\in\mathcal{T}} \frac{a_{j^*} + h_{j^*}(\widetilde{d}_t)}{\widetilde{d}_t}\widetilde{d}_t\right) \\ &\leq \rho\left(\sum_{t\in\mathcal{T}} \frac{a_{j^*} + h_{j^*}(\overline{q}_{j^*})}{\underline{q}_{j^*}}\widetilde{d}_t\right) \leq \frac{\max\limits_{i\in\mathcal{I}}\{a_i + h_i(\overline{q}_i)\}}{\min\limits_{i\in\mathcal{I}}\left\{\underline{q}_i\right\}}\rho\left(\sum_{t\in\mathcal{T}}\widetilde{d}_t\right) \\ &= \alpha^*\rho\left(\sum_{t\in\mathcal{T}}\widetilde{d}_t\right) \leq \alpha^*\rho(\widetilde{D}), \end{aligned}$$

where the first inequality follows from feasibility, the second inequality follows from Assumption 2 and the third equality follows from axiom (A4) and the definition of α^*. Thus, we get lower and upper bounds for the AC model, that is,

$$\alpha_*\rho(\widetilde{D}) \leq z^{AC} \leq \alpha^*\rho(\widetilde{D}). \tag{5.21}$$

Note that in the NC model, the status decisions in period $t \in \mathcal{T}$ are identical for all realizations of problem parameters in that period and satisfies $\max\{\widetilde{v}_{it}^*(\widetilde{d}_{[t]})\} \leq \overline{q}u_{it}^*$ and $D^{max} \leq \sum_{t\in\mathcal{T}}\max\{\widetilde{v}_{it}^*(\widetilde{d}_{[t]})\}$. Moreover, the policy $\{\widehat{\boldsymbol{u}}_t^*, \widehat{\boldsymbol{v}}_t^*, \widehat{\boldsymbol{w}}_t^*\}_{t\in\mathcal{T}}$ is also feasible for the NC model and $\rho(\widetilde{D}) \leq D^{max}$. Using these facts, a similar analysis can be used to obtain lower and upper bounds for NC model and we get

$$\alpha_* D^{\max} \leq z^{NC} \leq \alpha^* D^{\max}. \tag{5.22}$$

Hence, the claim of the theorem follows from (5.21) and (5.22). □

The inequalities given in (5.21) and (5.22) relate the optimal values of AC and NC, respectively, to the under and over estimations on per unit production costs.

If the generators are almost identical and lower and upper production limits are close enough, we have $\alpha_* \approx \alpha \approx \alpha^*$. Then, we have

$$\text{VAC} \approx \alpha(D^{\max} - \rho(\widetilde{D})). \tag{5.23}$$

Note that $0 \leq \rho(\widetilde{D}) \leq D^{\max}$ and the approximation (5.23) implies that the VAC increases with $D^{\max}$ and therefore variability in the net load. However, for fixed variability, the VAC decreases with $\rho(\widetilde{D})$ and therefore the degree of risk aversion.

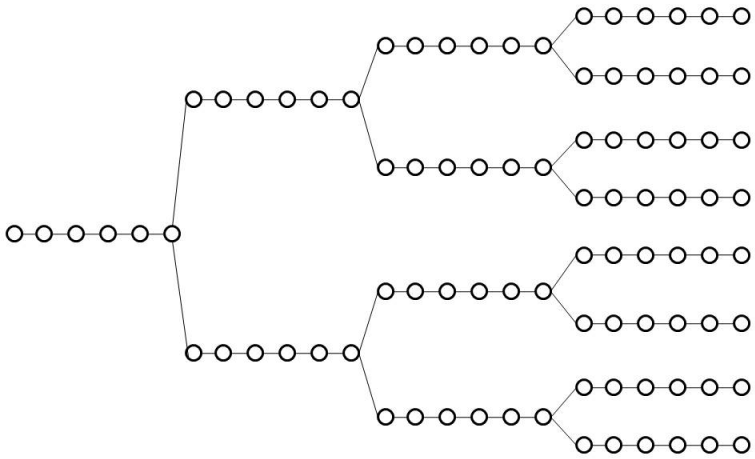
Assume that the net load in period $t \in \mathcal{T}$ is $\widetilde{d}_t = \overline{d}_t + \mathcal{U}[-\Delta, \Delta]$ where $\overline{d}_t$ is a deterministic value and $\mathcal{U}[-\Delta, \Delta]$ is an error term uniformly distributed between $-\Delta$ and Δ for some $\Delta \in \mathbb{R}_+$. Also, assume that the composite risk measure $\rho(\cdot)$ is obtained using conditional mean-upper semi deviation as given in (5.7) for simplicity. Then,

$$\begin{aligned} \text{VAC} &\approx \alpha(D^{\max} - \rho(\widetilde{D})) \\ &= \alpha\left(\sum_{t=1}^{T} d_t^{\max} - \rho\left(\sum_{t=1}^{T} \widetilde{d}_t\right)\right) \\ &= \alpha T\left(1 - \frac{\lambda}{4}\right)\Delta \end{aligned} \tag{5.24}$$

where the second equality follows from definitions of $d_t^{\max}$, $\widetilde{d}_t$ and evaluation of mean-upper semi deviation risk measure $\rho(\cdot)$. The approximation in (5.24) suggests that the VAC increases with the number of periods T and the variability in the net load Δ. However, VAC decreases with the degree of risk aversion λ.

The inverse relation between VAC and the degree of risk aversion may seem counter-intuitive at first glance. However, as the degree of risk aversion increases, $\rho(\widetilde{D})$ gets closer to $D^{\max}$, that is, the decision maker tries to minimize the cost in

Figure 5.3: Scenario tree for the system with 10 generators.



the most pessimistic scenarios. In that case, AC sacrifices its adaptivity in order to put emphasize to the most pessimistic scenarios. Thus, optimal values of NC and AC get closer.

5.3 Computational Experiments

The analytical results of the previous section rely on restrictive assumptions to simplify the structure of the RA-UC problem. In order to see how the VAC behave in the absence of these assumptions, we conduct two sets of computational experiments.

We first consider a power system with 10 generators in the computational experiments. We use the data set presented in [56] with some modifications. We also consider a random net load process with eight scenarios where the power demand at each hour is subject to uncertainty. The scenario tree depicting the random process is given in Figure 5.3. A similar scenario tree structure is used in [72].

The test data is presented in Appendix B. We use the base net load values presented in Table B.1 to generate random net load values. A variability parameter ϵ is used to control the dispersion of net load across all scenarios. Net load values for each scenario are presented in Table B.2. All other parameters except the production limits are set to the values given in [56]. The lower and upper production limits are increased by a half in order to avoid infeasibility in case of large variability in the net load amount. Start/shut-up/down limits are calculated as in [73]. A PC with two 2.2GHz processors and 6 GB of RAM is used in the computational experiments.

The quadratic production cost functions $\{h_i(\cdot)\}_{i\in\mathcal{I}}$ are approximated by a piecewise linear cost function with four pieces of equal lengths. This approximation of convex cost functions enables us to have a linear model for RA-UC problem and yields near-optimal solutions (see, for example, [71]). We also use a conditional mean-upper semi deviation risk measure (5.7) in each period.

We model and solve NC and AC for five different values of variability parameter ϵ and six different values of the penalty parameter λ. For each ϵ and λ pair, we calculate VAC in terms of difference of optimal values, that is,

$$\text{VAC } (\$) = z^{NC} - z^{AC},$$

and in terms of percentage

$$\text{VAC } (\%) = \frac{z^{NC} - z^{AC}}{z^{AC}},$$

The results on the VAC are presented in Figure 5.4.

Figure 5.4 verifies our analytical findings on VAC. We observe an increase in VAC with the uncertainty in net load values. The VAC and hence importance of adaptive commitment decisions increases as the dispersion among the scenarios increases. As expected, the day-ahead schedule obtained by solving AC is more adaptive and provides more flexibility in case of high variability of problem parameters.

Table 5.1: Solution times of NC for the system with 10 generators (in seconds)

$\epsilon\backslash\lambda$	0	0.1	0.2	0.3	0.4	0.5
0.1	7.5	10.4	9.6	7.7	7.2	7.2
0.2	4.2	3.8	3.5	4.0	3.7	3.2
0.3	12.2	10.9	9.5	8.1	7.8	6.0
0.4	7.9	3.8	4.1	4.0	3.3	2.7
0.5	8.8	5.4	6.3	4.8	4.8	4.6

We also observe decrease in the VAC with the level of risk aversion. In parallel with the analytical results in Theorem 2, higher risk aversion leads lower VAC. Hence, the importance of the multi-stage model decreases as risk aversion increases.

We also consider a rolling horizon policy obtained by solving NC in each period and fixing the decisions at that period with respect to the optimal solution of NC. In order to the measure the quality of the rolling horizon policy, we calculate the gap between the value of the rolling horizon policy and the optimal value of AC. The gap value GAP is calculated in terms of difference of objective values

$$\text{GAP } (\$) = z^{RH} - z^{AC},$$

and in terms of percentage

$$\text{GAP } (\%) = \frac{z^{RH} - z^{AC}}{z^{AC}}.$$

where z^{RH} is the value of the rolling horizon policy. Note that since rolling horizon provides a feasible policy to the multistage problem that is at least as good as that of NC, we have that $0 \leq \text{GAP} \leq \text{VAC}$. The results are presented in Figure 5.5.

We present the solution times for each NC and AC instance at Table 5.1 and Table 5.2, respectively. The required time to obtain the rolling horizon policy is also presented in Table 5.3.

Table 5.2: Solution times of AC for the system with 10 generators (in seconds)

$\epsilon\backslash\lambda$	0	0.1	0.2	0.3	0.4	0.5
0.1	1004.2	1280.0	1255.2	1489.7	1789.6	2009.1
0.2	328.3	381.6	400.4	444.7	324.6	393.8
0.3	480.0	1042.4	435.8	780.0	453.8	358.5
0.4	192.9	674.5	529.4	323.0	328.6	279.8
0.5	85.7	147.5	116.6	119.0	118.5	113.1

Table 5.3: Required time to obtain the rolling horizon policy for the system with 10 generators (in seconds)

$\epsilon\backslash\lambda$	0	0.1	0.2	0.3	0.4	0.5
0.1	16.6	15.1	14.7	13.6	14.9	12.8
0.2	8.0	9.0	9.0	8.7	8.0	8.5
0.3	15.1	17.3	15.1	15.2	14.6	11.4
0.4	9.0	10.4	8.3	9.1	7.7	7.8
0.5	10.2	9.6	9.0	12.3	9.7	9.5

In all instances, the rolling horizon policy performs much better than the policy obtained by solving NC problem with a small increase in computational effort. The GAP (%) of rolling horizon policy is 0.12% on average (with maximum 0.32%) whereas the VAC (%) is 1.42% on average (with maximum 3.20%). Thus, the rolling horizon policy can provide enough flexibility in generation schedule to obtain a near-optimal schedule in RA-UC problems with a reasonable computational effort.

The computational effort to solve the AC model is much larger than that of the NC model and the rolling horizon policy in all instances. The higher the net load variability leads higher VAC while decreasing the solution times as an additional benefit.

In the data given in [56], transmission capacity constraints are missing. Therefore, we conduct another set of experiments on the IEEE reliability test system [76] where transmission capacity constraints are also included in the model. This system has 24 buses, 34 transmission lines and 32 generators. In these experiments, we use the parameters presented in [77] and we consider $T = 6, 7, 8, 9$ and 10 stage problems with mean-upper semi deviation risk measure (5.7). As

in the previous set of experiments, the quadratic cost functions are replaced by their piece-wise linear approximations. We assume that the net load value at each stage can take values $(1-\epsilon)\overline{d}_t$ and $(1+\epsilon)\overline{d}_t$ with equal probabilities where $\overline{d}_t$ is the deterministic net load value at stage $t \in \mathcal{T}$ in the original data set. Thus, the resulting scenario tree is a binary tree where the number of scenarios is 2^{T-1} in a $T-$stage problem. Some instances of AC require long CPU times or cannot be solved optimally due to memory limitations. For these instances, the exact value of VAC cannot be calculated, however, we use the best objective value after two hours in calculation of an approximate VAC. The results of these experiments are presented in Figure 5.6 and Table 5.4.

Results in Table 5.4 reveal that with the existence of transmission capacity constraints, our findings on the relationship between VAC and degree of risk aversion, level of uncertainty of net load values and number of periods hold, in general. For the instances that cannot be solved within the time limit, the average optimality gap values are 0.03% and 0.07% for $T = 8$ and 9, respectively. Therefore, we obtain a good approximation of VAC and our findings are consistent in this approximation as well. However, for $T = 10$, the average optimality gap for the instances that cannot be solved within the time limit is 0.32%. Because of this poor approximation of VAC, we observe that approximate VAC fluctuates as λ increases for the instances $T = 10, \epsilon \in \{0.3, 0.4\}$. However, even $T = 10$, the results of the instances with $\epsilon \in \{0.1, 0.2, 0.5\}$ confirm our findings.

Especially for, $T = 9$ and 10, AC cannot be solved within two hours of time limit, on the other hand, the longest running time for NC is 1724.41 seconds. The average CPU times of NC and AC for the data set in [77] are given in Table 5.5.

The CPU time for AC, compared to NC, increases very rapidly as T increases even though the additional computational effort brings a benefit in the objective less than 1% in all instances. Therefore, implementing the policy obtained by solving NC can be a promising alternative under industry time constraints.

5.4 Conclusion

Recent improvements in the renewable power production technologies have motivated the stochastic unit commitment problems, since these models can explicitly address the variability in net load. The models with adaptive commitment provide completely flexible schedules where all decisions are adapted to the uncertainty. However, these models require high computational effort, and therefore, their non-adaptive counterparts are used to obtain approximate policies. In order to justify the additional effort to solve the model with adaptive commitment rather than its non-adaptive counterpart, we define the VAC and provide analytical and computational results on it. These results reveal that, for RA-UC problems, the VAC decreases with the degree of risk aversion, and increases with the level of uncertainty and number of time periods.

Figure 5.4: Results of the computational experiments on the VAC($) and VAC(%) for the system with 10 generators with respect to different variability (ϵ) and degree of risk aversion levels (λ).

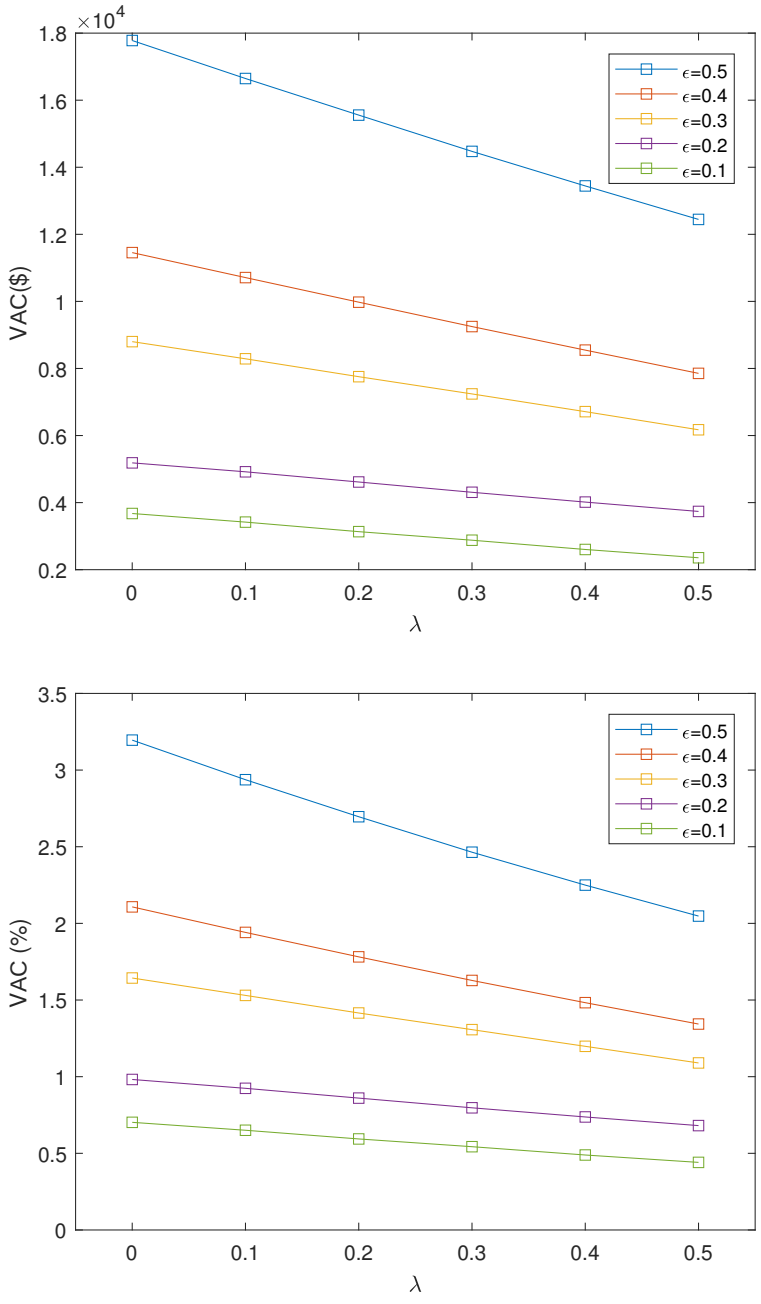


Figure 5.5: Results of the computational experiments on GAP($) and GAP(%) for the system with 10 generators with respect to different variability (ϵ) and degree of risk aversion levels (λ).

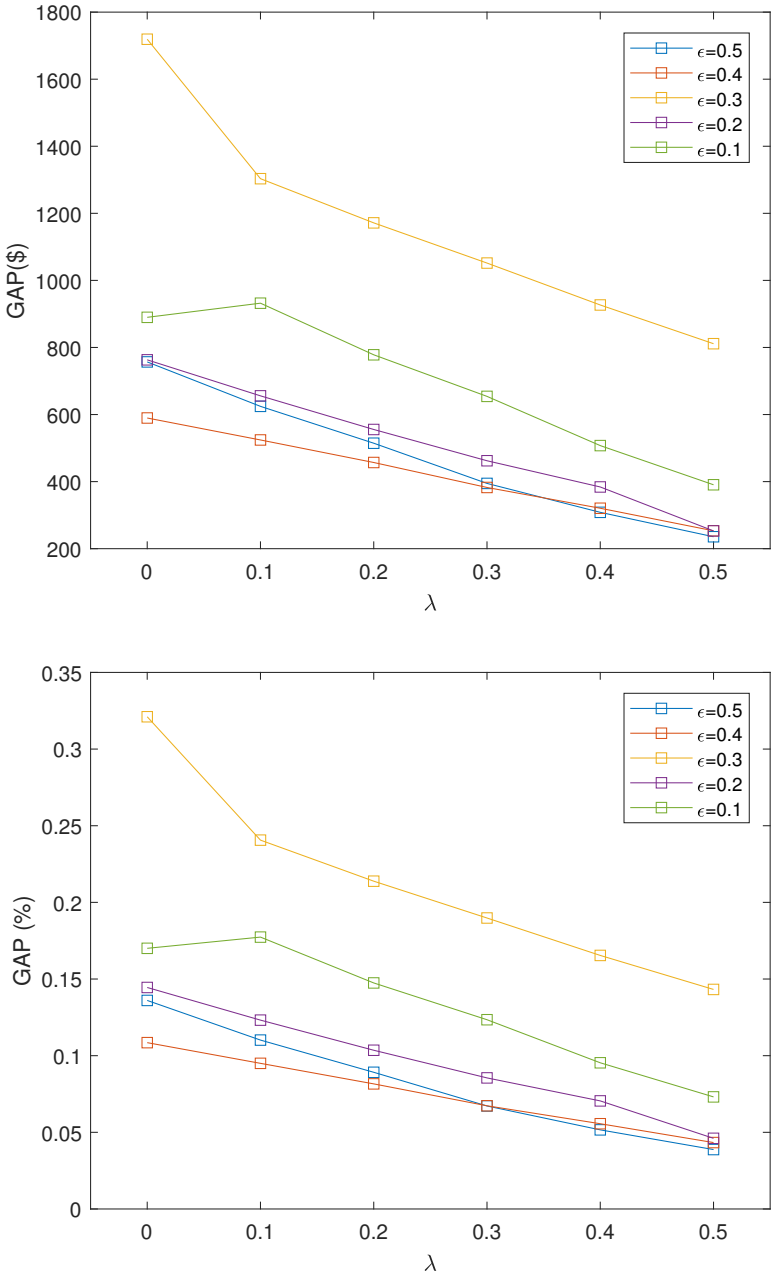


Figure 5.6: Results of the computational experiments on the VAC(%) for the system with 32 generators with respect to different variability (ϵ) and degree of risk aversion levels (λ).

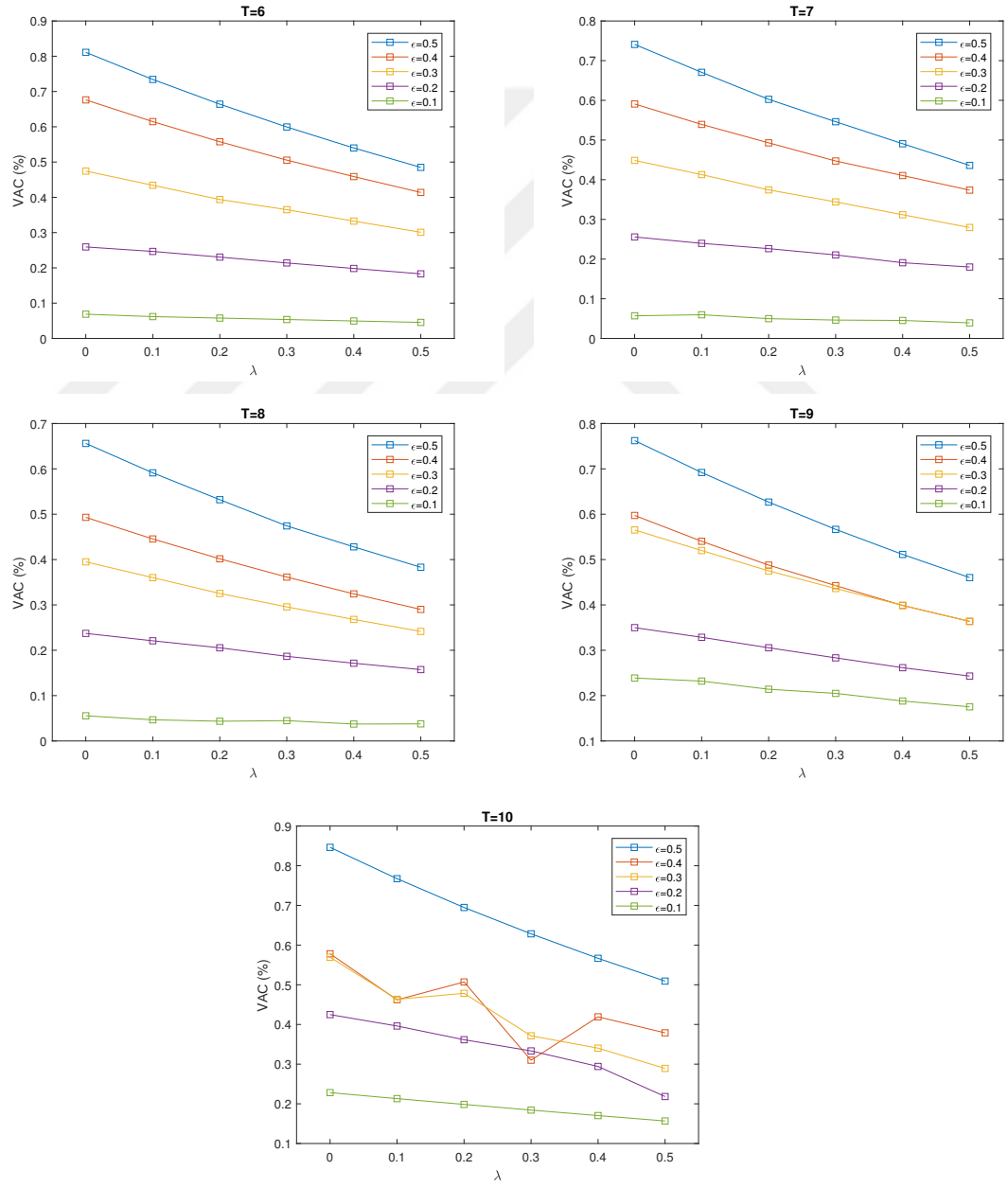


Table 5.4: VAC($) and VAC(%) for the system with 32 generators with respect to different variability (ϵ) and degree of risk aversion levels (λ).

T	$\epsilon\backslash\lambda$	0	0.1	0.2	0.3	0.4	0.5
6	0.1	107.86	97.51	91.31	85.11	78.91	72.73
		0.07%	0.06%	0.06%	0.05%	0.05%	0.05%
	0.2	410.26	393.11	370.31	346.26	323.32	300.70
		0.26%	0.25%	0.23%	0.21%	0.20%	0.18%
	0.3	755.90	699.71	641.85	601.45	554.11	506.83
		0.47%	0.43%	0.39%	0.36%	0.33%	0.30%
	0.4	1086.54	1002.56	923.08	848.99	781.85	715.17
		0.67%	0.61%	0.55%	0.50%	0.46%	0.41%
	0.5	1313.64	1211.58	1115.92	1025.51	939.97	858.95
		0.80%	0.73%	0.66%	0.60%	0.54%	0.48%
7	0.1	103.72	108.93	91.31	85.11	83.68	72.73
		0.06%	0.06%	0.05%	0.05%	0.05%	0.04%
	0.2	468.64	442.63	421.27	394.76	360.73	342.83
		0.26%	0.24%	0.23%	0.21%	0.19%	0.18%
	0.3	828.53	771.33	707.70	657.86	602.85	546.62
		0.45%	0.41%	0.37%	0.34%	0.31%	0.28%
	0.4	1100.65	1020.47	946.89	871.97	812.28	750.36
		0.59%	0.54%	0.49%	0.45%	0.41%	0.37%
	0.5	1391.73	1283.89	1175.38	1085.07	992.53	897.97
		0.74%	0.67%	0.60%	0.55%	0.49%	0.44%
8	0.1	115.55	97.51	91.31	94.47	78.91	80.03
		0.06%	0.05%	0.04%	0.04%	0.04%	0.04%
	0.2	497.70	467.20	438.09	401.12	371.40	343.64
		0.24%	0.22%	0.21%	0.19%	0.17%	0.16%
	0.3	835.65	771.17	704.73	648.16	594.22	541.80
		0.39%	0.36%	0.33%	0.30%	0.27%	0.24%
	0.4	1052.62*	966.11*	884.84*	808.41*	736.43	668.59*
		0.49%	0.45%	0.40%	0.36%	0.32%	0.29%
	0.5	1411.91	1297.33	1189.99	1080.89	993.60	905.40
		0.66%	0.59%	0.53%	0.47%	0.43%	0.38%
9	0.1	569.40	555.62	514.87	494.91	456.22	426.90
		0.24%	0.23%	0.21%	0.20%	0.19%	0.18%
	0.2	840.97	796.39	746.61	697.92	649.63	608.69
		0.35%	0.33%	0.31%	0.28%	0.26%	0.24%
	0.3	1370.20*	1275.94*	1180.00*	1096.45*	1015.09*	936.09*
		0.57%	0.52%	0.48%	0.44%	0.40%	0.36%
	0.4	1460.80*	1343.19*	1232.00*	1135.35*	1039.17*	961.80*
		0.60%	0.54%	0.49%	0.44%	0.40%	0.36%
	0.5	1881.41	1742.39	1609.45	1483.10	1363.85	1250.51
		0.76%	0.69%	0.63%	0.57%	0.51%	0.46%
10	0.1	620.71*	581.99*	544.72*	507.70*	471.40*	435.80*
		0.23%	0.21%	0.20%	0.18%	0.17%	0.16%
	0.2	1164.00*	1094.52*	1007.44*	936.85*	833.71*	624.08*
		0.42%	0.40%	0.36%	0.33%	0.29%	0.22%
	0.3	1574.31*	1298.25*	1356.30*	1066.33*	988.83*	850.78*
		0.57%	0.46%	0.48%	0.37%	0.34%	0.29%
	0.4	1615.47*	1313.53*	1463.80*	910.46*	1249.11*	1146.56*
		0.58%	0.46%	0.51%	0.31%	0.42%	0.38%
	0.5	2387.27	2209.31	2040.97	1882.50	1730.43	1584.75
		0.85%	0.77%	0.69%	0.63%	0.57%	0.51%

* VAC is calculated with the best objective value obtained after two hours.

Table 5.5: Average CPU times (in seconds) of NC and AC (for the instances that cannot be solved in two hours, the CPU times are taken as 7200 seconds.)

T	NC	AC
6	2.75	10.25
7	6.60	25.64
8	18.98	1551.31
9	116.83	4187.73
10	703.59	6445.46

Chapter 6

Approximations to Risk-averse Multi-stage Production Planning Problems

In this chapter, we consider risk-averse multi-stage production planning problems as a generalization of the unit commitment problem discussed in Chapter 5. In these problems, we make setup (or status) decisions of a set of production units and production amount decisions of each unit in order to satisfy random demand of a single product. For these problems, we consider two models with respect to their adaptivity to the uncertainty. In fully adaptive models, both setup and production decisions are given in on-line fashion, that is, they are adapted to demand uncertainty. However, in the models with non-adaptive setup decisions, the setup decisions are off-line, that is, they are fixed at the beginning of the decision horizon whereas the production decisions are on-line. We discuss the trade off between flexibility of the adaptive setup decisions and computational convenience of the model with non-adaptive setup decisions. As an intermediate case, we also consider a model with partially adaptive setup decisions. Moreover, we propose a rolling horizon approach for the fully adaptive model where the model with non-adaptive setup decisions is used as an approximation. In order to

reduce the computational difficulty of the rolling horizon algorithm, we consider restricting the production amounts to affine functions of demand realizations. We propose analytical results on the relation among the optimal values of the models with fully adaptive, partially adaptive, non-adaptive setup decisions and the objective value obtained from the rolling horizon algorithm. Finally, we conduct a set of computational experiments on a risk-averse multi-stage lot sizing problem to investigate the computational efficiency of the proposed models and verify the analytical results.

The rest of the chapter is organized as follows: In Sections 6.1 and 6.2, we present the deterministic and risk-averse models for multi-period production planning problems, respectively. Approximations to the fully adaptive risk-averse model are given in Section 6.3. In Section 6.4, we present a rolling horizon solution method for the risk-averse multi-stage production planning problems. In Section 6.5, we present results of our computational experiments. The concluding remarks are presented in Section 6.6.

6.1 Deterministic Problem

In a $T-$period planning horizon, we decide production schedule of I different units (or generators, machines etc.) in order to satisfy the demand of a certain product. The demand at period $t \in \{1, \ldots, T\}$ is given by d_t. Each unit has lower and upper production limits. Unit $i \in \{1, \ldots, I\}$ can produce at least $\underline{q}_i$ units and at most $\overline{q}_i$ units of product if it is working.

Let $\boldsymbol{u}_t := (u_{1t}, u_{2t}, \ldots, u_{It}) \in \{0,1\}^I$ and $\boldsymbol{v}_t := (v_{1t}, v_{2t}, \ldots, v_{It}) \in \mathbb{R}^I_+$ be the vectors of decision variables that represent status and production amounts at period t, respectively. The remaining auxiliary decisions at period t are represented by a $J-$dimensional real vector $\boldsymbol{w}_t \in \mathbb{R}^J_+$. The total cost at period t is given by a function $f_t(\boldsymbol{u}_t, \boldsymbol{v}_t, \boldsymbol{w}_t)$. Then, the deterministic multi-period production planning

problem is given in as:

$$\text{(DET)} \quad \min \quad \sum_{t=1}^{T} f_t(\boldsymbol{u}_t, \boldsymbol{v}_t, \boldsymbol{w}_t), \tag{6.1}$$

$$\text{s.t.} \quad \underline{q}_i u_{it} \leq v_{it} \leq \overline{q}_i u_{it}, \quad \forall i \in \{1, 2, \ldots, I\}, t \in \{1, 2, \ldots, T\}, \tag{6.2}$$

$$\sum_{i=1}^{I} v_{it} \geq d_t, \quad \forall t \in \{1, 2, \ldots, T\}, \tag{6.3}$$

$$(\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) \in \mathcal{X}_1, \tag{6.4}$$

$$(\boldsymbol{u}_t, \boldsymbol{v}_t, \boldsymbol{w}_t) \in \mathcal{X}_t(\boldsymbol{u}_{t-1}, \boldsymbol{v}_{t-1}, \boldsymbol{w}_{t-1}), \quad \forall t \in \{2, 3, \ldots, T\}, \tag{6.5}$$

$$\boldsymbol{u}_t \in \{0, 1\}_+^I, \boldsymbol{v}_t \in \mathbb{R}_+^I, \boldsymbol{w}_t \in \mathbb{R}_+^J, \quad \forall t \in \{1, 2, \ldots, T\}, \tag{6.6}$$

where the objective (6.1) is the total cost over T periods. Constraint (6.2) ensures that production amounts satisfy lower and upper production limits. Constraint (6.3) is demand satisfaction constraint. Constraints (6.4) and (6.5) reflect additional system dynamics and ensure that the production schedule is feasible.

The model DET provides a canonical formulation for a broad class of deterministic multi-period production planing problems including unit commitment and lot sizing problems. In unit commitment problems, for example, the objective is to minimize total energy production cost while satisfying load constraints through a multi-period decision horizon in a power system. The total cost is the sum of fixed commitment costs, convex dispatch cost, start up and shut down costs as well as other operational costs. The ramping and transmission requirements can easily be represented as a set constraint similar to (6.5). Similarly, in single-item lot sizing problems, the objective is the sum of fixed and variable costs of production and inventory holding costs through $T-$periods. Let $I = 2$ where $i = 1$ corresponds to the actual production unit and $i = 2$ is a dummy unit. Decision variable $u_{1t} \in \{0, 1\}$ takes value 1 if there is production at period t and 0 otherwise. Also let $u_{2t} \in \{0, 1\}$ represent the binary decision which indicates if there is a positive inventory at the beginning of period t. Also, let v_{1t} and v_{2t} be the production and inventory amounts in period t, respectively. Inventory balance requirement can also be represented by the set constraints. Thus, the lot sizing problem can also be written as the deterministic multi-period production

planning problem DET.

Although DET enables us to model multi-period decision problems, it assumes that demand values are deterministic and exactly known at the beginning of the entire decision horizon. This assumption can be quite restrictive in many real life applications. For example, net load values in unit commitment problems are subject to uncertainty due to unreliable equipment and power generated by renewable sources. Also, demand values in lot sizing problems are not exactly known but only be forecastable in many real-life applications. Therefore, we consider an extension of DET to a stochastic setting where the demand values at each period are random.

6.2 Risk-averse Multi-stage Production Planning Problem

Let $(\Omega, \mathcal{F}, P)$ be a *probability space* where Ω is a *sample space*, $\mathcal{F}$ be the *sigma algebra* representing the set of all events defined on Ω and P be a *probability measure*. The nested sequence of sigma algebras $\{\emptyset, \Omega\} = \mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots \subset \mathcal{F}_T = \mathcal{F}$ is called a *filtration* and it represents our gradually increasing information through a $T-$stage planning horizon. The set of all $\mathcal{F}_t-$measurable and $p-$integrable random variables are denoted by $\mathcal{Z}_t := \mathcal{L}_p(\Omega, \mathcal{F}_t, P)$ for $t \in \{1, 2, \ldots, T\}$ for some $p > 1$.

We assume that demand amount at period t is random and denoted by a random variable $\widetilde{d}_t$ in period $t \in \{1, 2, \ldots, T\}$. Then, $\widetilde{d}_t$ is $\mathcal{F}_t-$measurable, that is, $\widetilde{d}_t \in \mathcal{Z}_t$. The demand process $\widetilde{d}_1, \widetilde{d}_2, \ldots, \widetilde{d}_T$ evolves as a stochastic process with a specified probability distribution. An element $\omega \in \Omega$ is called as a *scenario* and corresponds to a realization $d_1, d_2, \ldots, d_T$ of the demand process $\widetilde{d}_1, \widetilde{d}_2, \ldots, \widetilde{d}_T$. Let p_ω be the probability of scenario ω. Note that since $\mathcal{F}_1 = \{\emptyset, \Omega\}$ by definition, $\mathcal{Z}_1 = \mathbb{R}$ and therefore the demand amount at the first period is deterministic.

To extend the deterministic model DET to this uncertainty setting, we

should ensure that the decisions at period t depend only on the history of demand realization $\widetilde{d}_{[t]} := (\widetilde{d}_1, \widetilde{d}_2, \ldots, \widetilde{d}_t)$ up to period t. This requirement is called as *non-anticipativity*. Therefore, we use $\mathcal{F}_t-$measurable mappings $\boldsymbol{u}_t(\widetilde{d}_{[t]}), \boldsymbol{v}_t(\widetilde{d}_{[t]}), \boldsymbol{w}_t(\widetilde{d}_{[t]})$ to represent our status, production and auxiliary decisions, respectively at period t. Note that, in that case, all decisions are fully adaptive to the underlying filtration. The total cost at period t is also $\mathcal{F}_t-$measurable, i.e., $f_t(\boldsymbol{u}_t(\widetilde{d}_{[t]}), \boldsymbol{v}_t(\widetilde{d}_{[t]}), \boldsymbol{w}_t(\widetilde{d}_{[t]})) \in \mathcal{Z}_t$. The feasibility sets $\mathcal{X}_t(\boldsymbol{u}_{t-1}(\widetilde{d}_{[t-1]}), \boldsymbol{v}_{t-1}(\widetilde{d}_{[t-1]}), \boldsymbol{w}_{t-1}(\widetilde{d}_{[t-1]}), \widetilde{d}_{[t]})$ are also given by the data process for $t \in \{2, \ldots, T\}$. The decisions at period t are made after observation of the demand realization at period t. We call a decision epoch as a stage and in our case, a stage corresponds to a period.

A sequence of measurable mappings $\{\boldsymbol{u}_t(\cdot), \boldsymbol{v}_t(\cdot), \boldsymbol{w}_t(\cdot)\}_{t=1}^T$ with respect to the underlying filtration is called as an *implementable* policy. Moreover, a policy is said to be *feasible* if it satisfies problem specific constraints analogous to (6.2)-(6.6) (see, for example, [37]).

As in the rest of the thesis, we use conditional risk measures to represent the attitude of a decision maker towards risk. The conditional risk measure $\rho_{\mathcal{F}_{t+1}|\mathcal{F}_t} : \mathcal{Z}_{t+1} \to \mathcal{Z}_t$ quantifies the risk involved in $\mathcal{F}_{t+1}-$measurable random variable based on the available information in period $t \in \{1, \ldots, T-1\}$.

At stage $t \in \{1, \ldots, T\}$, the objective of the decision maker is to minimize sum of the total cost at stage t and the risk-adjusted cost in the future stages over all implementable and feasible policies. Then, risk-averse (stochastic) multi-stage production planning problem can be written in a nested form:

$$\begin{aligned}
&\min_{(\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) \in \mathcal{X}_1} f_1(\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) \\
&+ \rho_{\mathcal{F}_2|\mathcal{F}_1}\left[\min_{(\boldsymbol{u}_2(\widetilde{d}_{[2]}), \boldsymbol{v}_2(\widetilde{d}_{[2]}), \boldsymbol{w}_2(\widetilde{d}_{[2]})) \in \mathcal{X}_2} f_2(\boldsymbol{u}_2(\widetilde{d}_{[2]}), \boldsymbol{v}_2(\widetilde{d}_{[2]}), \boldsymbol{w}_2(\widetilde{d}_{[2]})) \right. \\
&+ \rho_{\mathcal{F}_3|\mathcal{F}_2}\left\{ \min_{(\boldsymbol{u}_3(\widetilde{d}_{[3]}), \boldsymbol{v}_3(\widetilde{d}_{[3]}), \boldsymbol{w}_3(\widetilde{d}_{[3]})) \in \mathcal{X}_3} f_3(\boldsymbol{u}_3(\widetilde{d}_{[3]}), \boldsymbol{v}_3(\widetilde{d}_{[3]}), \boldsymbol{w}_3(\widetilde{d}_{[3]})) + \cdots \right.
\end{aligned}$$

$$+ \rho_{\mathcal{F}_T|\mathcal{F}_{T-1}}\left(\min_{(\boldsymbol{u}_3(\widetilde{d}_{[T]}),\boldsymbol{v}_3(\widetilde{d}_{[T]}),\boldsymbol{w}_3(\widetilde{d}_{[T]}))\in\mathcal{X}_T} f_T(\boldsymbol{u}_T(\widetilde{d}_{[T]}),\boldsymbol{v}_T(\widetilde{d}_{[T]}),\boldsymbol{w}_T(\widetilde{d}_{[T]}))\right)\cdots\Bigg\}\Bigg],$$

where dependence of feasibility sets to decision and demand processes is assumed but not explicitly written for notational brevity. The risk-averse multi-stage production planning problem can be written in several ways as also stated in [3]. As an example, the problem can be modeled as

(RA-A)

$$\begin{aligned}
\min \quad & f_1(\boldsymbol{u}_1,\boldsymbol{v}_1,\boldsymbol{w}_1) + \rho_{\mathcal{F}_2|\mathcal{F}_1}\Bigg[f_2(\boldsymbol{u}_2(\widetilde{d}_{[2]}),\boldsymbol{v}_2(\widetilde{d}_{[2]}),\boldsymbol{w}_2(\widetilde{d}_{[2]})) + \cdots \\
& + \rho_{\mathcal{F}_3|\mathcal{F}_2}\Bigg\{f_3(\boldsymbol{u}_3(\widetilde{d}_{[3]}),\boldsymbol{v}_3(\widetilde{d}_{[3]}),\boldsymbol{w}_3(\widetilde{d}_{[3]})) \\
& + \cdots + \rho_{\mathcal{F}_T|\mathcal{F}_{T-1}}\left(f_T(\boldsymbol{u}_T(\widetilde{d}_{[T]}),\boldsymbol{v}_T(\widetilde{d}_{[T]}),\boldsymbol{w}_T(\widetilde{d}_{[T]}))\right)\cdots\Bigg\}\Bigg], \qquad (6.7)\\
\text{s.t.} \quad & \underline{q}_i u_{it}(\widetilde{d}_{[t]}) \le v_{it}(\widetilde{d}_{[t]}) \le \overline{q}_i u_{it}(\widetilde{d}_{[t]}), \\
& \qquad \forall i \in \{1,2,\ldots,I\}, t \in \{1,2,\ldots,T\}, \qquad (6.8)\\
& \sum_{i=1}^{I} v_{it}(\widetilde{d}_{[t]}) \ge \widetilde{d}_t, \quad \forall t \in \{1,2,\ldots,T\}, \qquad (6.9)\\
& (\boldsymbol{u}_1,\boldsymbol{v}_1,\boldsymbol{w}_1) \in \mathcal{X}_1, \qquad (6.10)\\
& (\boldsymbol{u}_t(\widetilde{d}_{[t]}),\boldsymbol{v}_t(\widetilde{d}_{[t]}),\boldsymbol{w}_t(\widetilde{d}_{[t]})) \in \mathcal{X}_t(\boldsymbol{u}_t(\widetilde{d}_{[t-1]}),\boldsymbol{v}_t(\widetilde{d}_{[t-1]}),\boldsymbol{w}_t(\widetilde{d}_{[t-1]}),\widetilde{d}_{[t]}), \\
& \qquad \forall t \in \{2,3,\ldots,T\}, \qquad (6.11)\\
& \boldsymbol{u}_t(\widetilde{d}_{[t]}) \in \{0,1\}_+^I, \boldsymbol{v}_t(\widetilde{d}_{[t]}) \in \mathbb{R}_+^I, \boldsymbol{w}_t(\widetilde{d}_{[t]}) \in \mathbb{R}_+^J, \quad \forall t \in \{1,2,\ldots,T\}. \\
& \qquad (6.12)
\end{aligned}$$

The objective (6.7) is the risk value associate with the cost sequence $f_1(\cdot)$, $f_2(\cdot),\ldots,f_T(\cdot)$. Constraints (6.8), (6.9), (6.10), (6.11) and (6.12) are analogues to their deterministic counterparts (6.2), (6.3), (6.4), (6.5) and (6.6), respectively.

In order to get an alternative representation of this risk value, we define the dynamic coherent risk measure $\varrho : \mathcal{Z}_1 \times \mathcal{Z}_2 \times \cdots \times \mathcal{Z}_T \to \mathbb{R}$ by using nested composition of the conditional risk measures $\rho_{\mathscr{F}_2|\mathscr{F}_1}(\cdot), \rho_{\mathscr{F}_3|\mathscr{F}_2}(\cdot),\ldots,\rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}(\cdot)$,

that is,

$$\varrho(Z_1, Z_2, \ldots, Z_T) := Z_1 + \rho_{\mathscr{F}_2|\mathscr{F}_1}(Z_2 + \cdots \rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}(Z_T)\cdots),$$

for any cost sequence $\{Z_t\}_{t=1}^T$ where $Z_t \in \mathcal{Z}_t$ for any $t \in \{1, 2, \ldots, T\}$. Due to translational equivariance property of conditional risk measures, we have an alternative representation of the dynamic coherent measure of risk $\varrho(\cdot)$ as

$$\overline{\rho}\left(\sum_{t=1}^{T} Z_t\right) := \varrho(Z_1, Z_2, \ldots, Z_T), \tag{6.13}$$

where $\overline{\rho} = \rho_{\mathscr{F}_2|\mathscr{F}_1} \circ \rho_{\mathscr{F}_3|\mathscr{F}_2} \circ \cdots \circ \rho_{\mathscr{F}_T|\mathscr{F}_{T-1}} : \mathcal{Z} \to \mathbb{R}$ is called as a *composite risk measure* and $\mathcal{Z} := \mathcal{Z}_T$. Note that, in the risk-neural case $\overline{\rho}(\cdot)$ reduces to expectation operator $\mathbb{E}(\cdot)$. The composite risk measure $\overline{\rho}(\cdot)$ satisfies the coherence axioms (A1)-(A4) and therefore it is a coherent measure of risk as shown in [3, Eqn. 6.234].

$$\begin{aligned} \min. \quad & \overline{\rho}\left(\sum_{t=1}^{T} f_t(\boldsymbol{u}_t(\widetilde{d}_{[t]}), \boldsymbol{v}_t(\widetilde{d}_{[t]}), \boldsymbol{w}_t(\widetilde{d}_{[t]}))\right), \\ \text{s.t.} \quad & (6.8), (6.9), (6.10), (6.11) \text{ and } (6.12), \end{aligned} \tag{6.14}$$

where $f_1(\boldsymbol{u}_1(\widetilde{d}_{[1]}), \boldsymbol{v}_1(\widetilde{d}_{[1]}), \boldsymbol{w}_1(\widetilde{d}_{[1]})) := f_1(\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1)$ for notational consistency.

If the random demand values evolve as a discrete-time stochastic process with finite support, then the whole process can be represented by a $T-$stage scenario tree where Ω_t is the set of nodes at stage $t \in \{1, \ldots, T\}$. Each node n of the scenario tree at stage t corresponds to a possible realization of $d_{[t]}$. Note that in this case, each last stage node $n \in \Omega_T$ corresponds to a scenario $\omega \in \Omega$.

In finite realization case, the risk-averse multi-stage production planning problem can be written as a large deterministic problem called as *deterministic equivalent problem* (DEP) by defining a decision variable for each status, production and auxiliary decision at each node of the scenario tree. Therefore, the size of DEP grows exponentially with the number of stages T for any non-trivial scenario tree. Moreover, the problem quickly becomes computationally intractable

since the status decisions are binary in this DEP. This computational challenge motivates following questions:

- Is it possible to quantify the benefit of solving the model with a fully adaptive policy?
- Is it worthwhile to solve a large scale non-convex DEP to obtain a fully adaptive policy?
- In which instances, a fully adaptive policy is necessary?
- Can we sacrifice the adaptivity of our decisions to some extend for ease of computation?

In this chapter, we try to answer these questions by considering several approximations of the risk-averse multi-stage model of multi-period production planning problem. Some of these questions are answered for a risk-neutral multi-stage stochastic programming problem in [78]. Moreover, non-adaptive and partially adaptive polices are considered for specific applications of risk-neutral models such as lot-sizing [79] and inventory planning [80] problems.

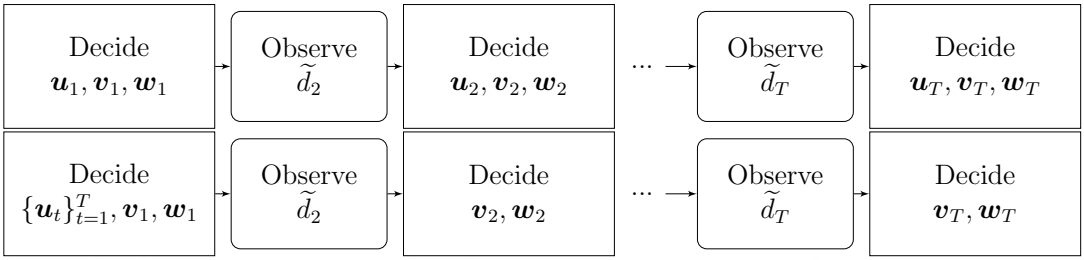
In the next sections, we obtain approximations to problem RA-A by varying adaptivity of setup decisions. We also consider a rolling horizon approach based on these approximations.

6.3 Approximations

6.3.1 Non-adaptive Setup Model

In this section, we consider an approximation of RA-A where the setup decisions are non-adaptive. The risk-averse multi-stage production planning problem with

Figure 6.1: Decision processes in RA-A (top) and RA-N (bottom) problems models .



non-adaptive setup decisions is given as

(RA-N)

$$\begin{aligned}
\text{min.} \quad & f_1(\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) + \rho_{\mathscr{F}_2|\mathscr{F}_1}\Bigg[f_2(\boldsymbol{u}_2, \boldsymbol{v}_2(\widetilde{d}_{[2]}), \boldsymbol{w}_2(\widetilde{d}_{[2]})) \\
& + \rho_{\mathscr{F}_3|\mathscr{F}_2}\Bigg\{ f_3(\boldsymbol{u}_3, \boldsymbol{v}_3(\widetilde{d}_{[3]}), \boldsymbol{w}_3(\widetilde{d}_{[3]})) + \cdots \\
& + \rho_{\mathscr{F}_T|\mathscr{F}_{T-1}}\left(f_T(\boldsymbol{u}_T, \boldsymbol{v}_T(\widetilde{d}_{[T]}), \boldsymbol{w}_T(\widetilde{d}_{[T]})) \right) \cdots \Bigg\}\Bigg], && (6.15) \\
\text{s.t.} \quad & \underline{q}_i u_{it} \leq v_{it}(d_{[t]}) \leq \overline{q}_i u_{it}, \quad \forall i \in \{1, 2, \ldots, I\}, t \in \{1, 2, \ldots, T\}, && (6.16) \\
& \sum_{i=1}^{I} v_{it}(\widetilde{d}_{[t]}) \geq \widetilde{d}_t, \quad \forall t \in \{1, 2, \ldots, T\} && (6.17) \\
& (\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) \in \mathscr{X}_1, && (6.18) \\
& (\boldsymbol{u}_t, \boldsymbol{v}_t(\widetilde{d}_{[t]}), \boldsymbol{w}_t(\widetilde{d}_{[t]})) \in \mathscr{X}_t(\boldsymbol{u}_{t-1}, \boldsymbol{v}_t(\widetilde{d}_{[t-1]}), \boldsymbol{w}_t(\widetilde{d}_{[t-1]}), \widetilde{d}_{[t]}), \\
& \qquad \forall t \in \{2, 3, \ldots, T\}, && (6.19) \\
& \boldsymbol{u}_t \in \{0, 1\}_+^I, \boldsymbol{v}_t(\widetilde{d}_{[t]}) \in \mathbb{R}_+^I, \boldsymbol{w}_t(\widetilde{d}_{[t]}) \in \mathbb{R}_+^J, \quad \forall t \in \{1, 2, \ldots, T\}. && (6.20)
\end{aligned}$$

Note that, in the model RA-N, the schedule decisions are deterministic and therefore, they can also be seen as first stage decisions. An interpretation of this fact is that the status decisions are made at the beginning of the decision horizon, however, the production and auxiliary decisions are made after observing the demand realization at each stage. Figure 6.1 highlights the difference of decision processes in problems with adaptive and non-adaptive schedules.

In the DEP of RA-A, the number of binary variables is $\mathcal{N} \times I$ where $\mathcal{N} := \sum_{t=1}^{T} |\Omega_t|$ is the number of nodes in the scenario tree. This number grows exponentially with the number of stages. On the other hand, in the DEP of RA-N, the number of binary variables is $T \times I$ whose value grows linearly with the number of stages. Since $\mathcal{N} >> T$ for any non-trivial problem, computational difficulty of RA-A is significantly more than RA-N.

The following example shows that the difference between the optimal values of RA-A and RA-N can be arbitrarily large even for a two-stage problem if the demand is unbounded.

Example 6. *Consider a risk-neutral instance of the problem with* $T = 2, I = \lceil \frac{1}{\epsilon} \rceil$ *for some* $0 < \epsilon < 1$, $\underline{q}_i = \overline{q}_i = 1$ *for all* $i \in \{1, \ldots, I\}$. *Let* $f_t(\boldsymbol{u}_t(\widetilde{d}_{[t]}), \boldsymbol{v}_t(\widetilde{d}_{[t]}), \boldsymbol{w}_t(\widetilde{d}_{[t]})) = \sum_{i=1}^{I} v_{it}(\widetilde{d}_t)$ *for* $t \in \{1, 2\}$. *The demand in the first period is 1 unit and demand in the second period takes values 1 and* $\frac{1}{\epsilon}$ *units equally likely.*

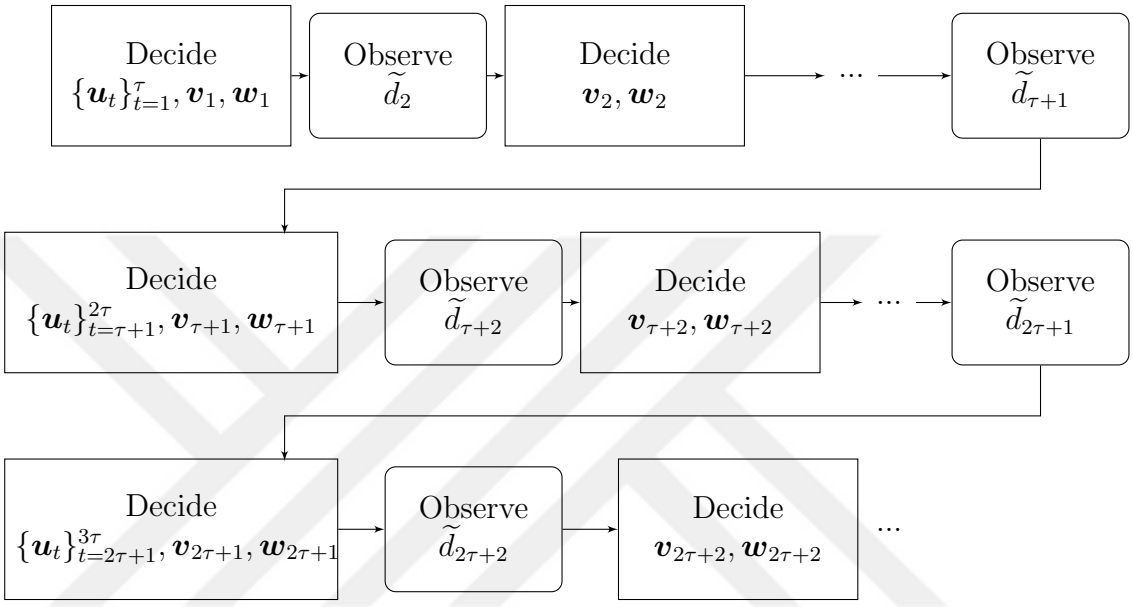
The optimal value of RA-A is $\frac{3}{2} + \frac{1}{2}\lceil \frac{1}{\epsilon} \rceil$ *and the optimal value of RA-N is* $1 + \lceil \frac{1}{\epsilon} \rceil$. *The difference between the optimal values is* $-\frac{1}{2} + \frac{1}{2}\lceil \frac{1}{\epsilon} \rceil$ *and it can be arbitrarily large since* ϵ *can be arbitrarily small.*

However, if Assumptions 2, 3 and 4 in Chapter 5 hold, then as a corollary of Theorem 2 in the same chapter, we have

$$\alpha_* \overline{\rho}(\widetilde{D}) \leq z^A \leq z^N \leq \alpha^* D^{\max} \tag{6.21}$$

where z^A and z^N are the optimal values of RA-A and RA-N, respectively and the parameters α_*, α^*, $\widetilde{D}$ and $D^{\max}$ are defined as in Chapter 5. Therefore, it is possible to calculate a bound on the approximation error due to non-adaptive setup decisions. Moreover, Assumptions 2, 3 and 4 are not too restrictive and already hold for the lot sizing problem with outsourcing option or lost demand.

Figure 6.2: Decision processes in problems with a partially adaptive setup decisions where the setup decisions are updated in every τ periods.



6.3.2 Partially Adaptive Setup Model

As an intermediate case between the models with adaptive and non-adaptive setup decisions, we consider a third model with partially adaptive setup decisions. In this model, the setup decisions are given in every τ periods such that $1 \leq \tau \leq T$ whereas the production and auxiliary decisions are given in every period. The decision process for the model with partially adaptive setup decisions is given in Figure 6.2.

Note that in a model with partially adaptive setup decisions, $\boldsymbol{u}_t$ depends only on the demand history up to period $\lceil \frac{t}{\tau} \rceil$, that is $\boldsymbol{u}_t$ is $\mathcal{F}_{\lceil \frac{t}{\tau} \rceil}$-measurable where $\lceil \cdot \rceil$ is the ceiling function. Thus, the risk-averse multi-stage production planning problem with a partially adaptive setup decisions is given as

(RA-P(τ))

$$\min \quad f_1(\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) + \rho_{\mathcal{F}_2|\mathcal{F}_1}\left[f_2(\boldsymbol{u}_2(\widetilde{d}_{\lceil \lceil \frac{2}{\tau} \rceil \rceil}), \boldsymbol{v}_2(\widetilde{d}_{[2]}), \boldsymbol{w}_2(\widetilde{d}_{[2]})) + \cdots \right.$$

$$+\rho_{\mathcal{F}_3|\mathcal{F}_2}\Bigg\{f_3(\boldsymbol{u}_3(\widetilde{d}_{[\lceil\frac{3}{\tau}\rceil]}),\boldsymbol{v}_3(\widetilde{d}_{[3]}),\boldsymbol{w}_3(\widetilde{d}_{[3]}))$$

$$+\cdots+\rho_{\mathcal{F}_T|\mathcal{F}_{T-1}}\left(f_T(\boldsymbol{u}_T(\widetilde{d}_{[\lceil\frac{T}{\tau}\rceil]}),\boldsymbol{v}_T(\widetilde{d}_{[T]}),\boldsymbol{w}_T(\widetilde{d}_{[T]}))\right)\cdots\Bigg\}\Bigg], \quad (6.22)$$

$$\text{s.t.}\quad \underline{q}_i u_{it}(\widetilde{d}_{[\lceil\frac{t}{\tau}\rceil]}) \le v_{it}(\widetilde{d}_{[t]}) \le \overline{q}_i u_{it}(\widetilde{d}_{[\lceil\frac{t}{\tau}\rceil]}),$$

$$\forall i\in\{1,2,\ldots,I\}, t\in\{1,2,\ldots,T\}, \quad (6.23)$$

$$\sum_{i=1}^{I} v_{it}(\widetilde{d}_{[t]}) \ge \widetilde{d}_t, \quad \forall t\in\{1,2,\ldots,T\} \quad (6.24)$$

$$(\boldsymbol{u}_1,\boldsymbol{v}_1,\boldsymbol{w}_1)\in\mathcal{X}_1, \quad (6.25)$$

$$(\boldsymbol{u}_t(\widetilde{d}_{[\lceil\frac{t}{\tau}\rceil]}),\boldsymbol{v}_t(\widetilde{d}_{[t]}),\boldsymbol{w}_t(\widetilde{d}_{[t]}))\in$$

$$\mathcal{X}_t(\boldsymbol{u}_t(\widetilde{d}_{[\lceil\frac{t-1}{\tau}\rceil]}),\boldsymbol{v}_t(\widetilde{d}_{[t-1]}),\boldsymbol{w}_t(\widetilde{d}_{[t-1]}),\widetilde{d}_{[t]}), \quad \forall t\in\{2,3,\ldots,T\}, \quad (6.26)$$

$$\boldsymbol{u}_t(\widetilde{d}_{[\lceil\frac{t}{\tau}\rceil]})\in\{0,1\}^I_+, \boldsymbol{v}_t(\widetilde{d}_{[t]})\in\mathbb{R}^I_+, \boldsymbol{w}_t(\widetilde{d}_{[t]})\in\mathbb{R}^J_+,$$

$$\forall t\in\{1,2,\ldots,T\}. \quad (6.27)$$

The model RA-P has $\sum_{k=0}^{\lceil T/\tau\rceil-1}|\Omega_{k\tau+1}|$ binary variables and provides an intermediate case between the models with adaptive and non-adaptive schedules in terms of adaptivity of setup decisions. Note that if $\tau=1$, the model RA-P coincides with the model RA-A where the setup decisions are adapted to the random process. Similarly, when $\tau=T$, it coincides with RA-N where the setup decisions are non-adaptive. Therefore, the RA-P enables the decision maker to control the desired level of adaptivity of the setup decisions and the number of binary variables in the model. In the next proposition, we prove the relationship between the optimal values of RA-A, RA-N and RA-P.

Proposition 9. *We have $z^A \le z^P(\tau) \le z^N$ where z^A is the optimal value of RA-A, $z^P(\tau)$ is the optimal value of RA-P(τ) and z^N is the optimal value of RA-N.*

Proof. An equivalent representation of RA-N can be obtained by adding additional constraints to RA-A to ensure that the setup decisions at stage t are identical for all possible realizations of $d_{[t]}$ for $t\in\{1,\ldots,T\}$. On the other

hand, an equivalent representation of RA-P(τ) can be obtained by adding additional constraints to RA-A such that the setup decisions at stage t are identical for the possible realizations of $d_{[t]}$ that share same history up to period $\lceil \frac{t}{\tau} \rceil$ for $t \in \{1, \ldots, T\}$. Thus, the claim in the proposition follows. □

Although smaller τ values yield more adaptive schedules, decreasing τ may increase the cost of the obtained schedules. The following examples shows an instance with $z^P(\tau_2) < z^P(\tau_1)$ where $1 < \tau_1 < \tau_2 < T$.

Example 7. *Consider a risk-neutral instance of the problem with* $T = 4, I = 2, \underline{q}_1 = \underline{q}_2 = \overline{q}_1 = \overline{q}_2 = 1$. *Let* $f_t(\boldsymbol{u}_t(\widetilde{d}_{[t]}), \boldsymbol{v}_t(\widetilde{d}_{[t]}), \boldsymbol{w}_t(\widetilde{d}_{[t]})) = \sum_{i=1}^{2} v_{it}(\widetilde{d}_{[t]})$ *for* $t \in \{1, 2, 3, 4\}$. *The demand in the first three periods is 1 unit and demand in the last period takes values 1 and 2 equally likely. Note that, in this case,* $z^A = z^P(3) = 4.5$, *however* $z^P(2) = z^N = 5$.

6.4 A Rolling Horizon Approach

In a classical rolling horizon (RH) approach, at each period, an approximate problem is solved for the rest of the decision horizon and immediate decisions are implemented for the current period. Iterating over all periods a feasible policy is obtained. RH approach is shown to be an effective tool in multi-stage scheduling problems (see, [81] and references therein for a more detailed discussion). A RH approach can be employed for RA-A by solving RA-N at every $\tau \in \{1, \ldots, T\}$ stages and implementing an optimal policy of this approximation for the next τ stages. We first make a complete recourse assumption in order to ensure that all problems can be solved optimally during execution of the RH approach.

Assumption 5. *Given any partial feasible policy* $\{\boldsymbol{u}_t(\cdot), \boldsymbol{v}_t(\cdot), \boldsymbol{w}_t(\cdot)\}_{t=1}^{t'}$, *the model RA-N starting from period* $t' + 1$ *is always feasible.*

Algorithm 3 outlines the RH approach proposed for RA-A.

Algorithm 3 Rolling Horizon Algorithm for RA-A

Require: An integer $\tau \in \{1, \ldots, T\}$
Initialize: Set update epoch $\bar{t} = 1$ and policy $\{\overline{\boldsymbol{u}}_t(\cdot), \overline{\boldsymbol{v}}_t(\cdot), \overline{\boldsymbol{w}}_t(\cdot)\}_{t=1}^{T} = null$
for $\bar{t} \leq T$ **do**
  for $n \in \Omega_{\bar{t}}$ **do**
    Given the demand history $d_1, d_2, \ldots, d_{\bar{t}-1}$ corresponding to node n and $\{\overline{\boldsymbol{u}}_t(\cdot), \overline{\boldsymbol{v}}_t(\cdot), \overline{\boldsymbol{w}}_t(\cdot)\}_{t=1}^{\bar{t}-1}$, solve RA-N. Let $\{\widehat{\boldsymbol{u}}_t(\cdot), \widehat{\boldsymbol{v}}_t(\cdot), \widehat{\boldsymbol{w}}_t(\cdot)\}_{t=\bar{t}}^{T}$ be an optimal policy.
    $(\overline{\boldsymbol{u}}_t(\cdot), \overline{\boldsymbol{v}}_t(\cdot), \overline{\boldsymbol{w}}_t(\cdot)) \leftarrow (\widehat{\boldsymbol{u}}_t(\cdot), \widehat{\boldsymbol{v}}_t(\cdot), \widehat{\boldsymbol{w}}_t(\cdot))$ for stages $t \in \{\bar{t}, \bar{t}+1, \ldots, \bar{t}+\tau-1\}$.
    Update: $\bar{t} \leftarrow \bar{t} + \tau$
  end for
end for
Return: Policy $\{\overline{\boldsymbol{u}}_t(\cdot), \overline{\boldsymbol{v}}_t(\cdot), \overline{\boldsymbol{w}}_t(\cdot)\}_{t=1}^{T}$

Let $z^{RH}(\tau)$ be the objective value corresponding to the policy obtained from Algorithm 3 for a given τ value. The following proposition shows the relation between $z^{RH}(\tau)$, z^A, $z^P(\tau)$ and z^N.

Proposition 10. *$z^A \leq z^P(\tau) \leq z^{RH}(\tau) \leq z^N$ for $\tau \in \{1, \ldots, T\}$.*

Proof. Due to Assumption 5, the policy $\{\overline{\boldsymbol{u}}_t(\cdot), \overline{\boldsymbol{v}}_t(\cdot), \overline{\boldsymbol{w}}_t(\cdot)\}_{t=1}^{T}$ obtained by Algorithm 3 is feasible for RA-P(τ) and the setup decision $\overline{\boldsymbol{u}}_t(\cdot)$ is $\mathcal{F}_{\lceil \frac{t}{\tau} \rceil}-$measurable. Therefore, $z^P(\tau) \leq z^{RH}(\tau)$. In the first epoch of Algorithm 3, that is for $\bar{t} = 1$, the initial policy is an optimal solution of RA-N with the objective value z^N. Throughout the update epochs, the objective value is non-increasing. Thus, $z^{RH}(\tau) \leq z^N$. □

Proposition 11. *Under Assumptions 2, 3 and 4 in Chapter 5, we have $\frac{z^{RH}(\tau)}{z^A} \leq \frac{\alpha^* D^{\max}}{\alpha_* \overline{\rho}(\widetilde{D})}$ for $\tau \in \{1, \ldots, T\}$.*

Proof. The proof follows from the inequalities given in (6.21) and Proposition 10. □

Note that Proposition 11 provides a worst-case bound for the performance of the rolling horizon algorithm. Obtaining worst case bounds of a RH approach for

an optimization problem is rare in the literature. For example, recently in [82], worst case bounds of a RH approach can only be shown for a multi-stage fixed charge transportation problem for some specific cases.

Algorithm 3 requires solving the RA-N model $\sum_{k=0}^{\lceil T/\tau \rceil - 1} |\Omega_{k\tau+1}|$ times. This number also grows exponentially with the number of stages, and therefore Algorithm 3 may require long computation times. In order to reduce the computational difficulty in Algorithm 3, we consider restricting the production decision in RA-N at stage t to an affine function of random demands up to t, that is,

$$\text{ADR:} \quad v_{it}(\widetilde{d}_{[t]}) = \sum_{t'=1}^{t} A_{it't}\widetilde{d}_{t'} + B_{it}, \quad \forall t \in \{1, 2, \ldots, T\}, i \in \{1, 2, \ldots, I\}, \tag{6.28}$$

where $A_{it't}$ and B_{it} for $t', t \in \{1, \ldots, T\}$ with $t' \leq t$ and $i \in \{1, \ldots, I\}$ are the real valued coefficients that define the affine policy. Another alternative is a simpler policy where the production decision at stage t is an affine function of random demand at stage t only, that is,

$$\text{SADR:} \quad v_{it}(\widetilde{d}_{[t]}) = A_{it}\widetilde{d}_t + B_{it}, \quad \forall t \in \{1, 2, \ldots, T\}, i \in \{1, 2, \ldots, I\} \tag{6.29}$$

with coefficients A_{it} and B_{it} for $t \in \{1, \ldots, T\}$, $i \in \{1, \ldots, I\}$ that define an affine policy. Note that, in finite support case, a realization of $\widetilde{d}_{[t]}$ can be represented by $t-$dimensional real-valued vector. Similarly, $\boldsymbol{v}_t(\widetilde{d}_{[t]})$ can also be represented by an $I-$dimensional real-valued vector. Thus, the decision rule in (6.28) can be represented

$$\text{ADR:} \quad \boldsymbol{v}_t(\widetilde{d}_{[t]}) = \sum_{t'=1}^{t} A_{t't}\widetilde{d}_{t'} + B_t, \quad \forall t \in \{1, 2, \ldots, T\}, \tag{6.30}$$

where $A_{t't} \in \mathbb{R}^{I \times t'}$ and $B_t \in \mathbb{R}^{I \times 1}$ for $t' \in \{1, \ldots, t\}$. Also, the decision rule in (6.29) can be written as

$$\text{SADR:} \quad \boldsymbol{v}_t(\widetilde{d}_{[t]}) = A_t\widetilde{d}_t + B_t, \quad \forall t \in \{1, 2, \ldots, T\}, \tag{6.31}$$

where $A_t \in \mathbb{R}^{I \times t}$ and $B_t \in \mathbb{R}^{I \times 1}$.

Affine decision rules have been considered in the context of stochastic programming since the pioneering work [83]. However, successful implementations such as [84] and [85] in multi-stage setting have appeared recently in the literature. In the risk-averse context, usage of affine decision rules have not attracted much attention. In [86], the decisions are restricted to affine functions of the problem parameters where the objective is a non-dynamic CVaR in a reservoir management problem. To the best of our knowledge, affine decision rules have not been used in a risk-averse multi-stage stochastic optimization problem with an objective of dynamic coherent risk measure.

The risk-averse multi-stage production planning problem with non-adaptive schedule and affine decision rules is given by

(RA-N-ADR)

$$\begin{aligned}
\min. \quad & f_1(\boldsymbol{u}_1, A_{11}d_1 + B_1, \boldsymbol{w}_1) + \rho_{\mathscr{F}_2|\mathscr{F}_1}\Bigg[f_2(\boldsymbol{u}_2, \sum_{t'=1}^{2} A_{t'2}\widetilde{d}_{t'} + B_2, \boldsymbol{w}_2(\widetilde{d}_{[2]})) \\
& + \rho_{\mathscr{F}_3|\mathscr{F}_2}\Bigg\{ f_3(\boldsymbol{u}_3, \sum_{t'=1}^{3} A_{t'3}\widetilde{d}_{t'} + B_3, \boldsymbol{w}_3(\widetilde{d}_{[3]})) + \cdots \\
& + \rho_{\mathcal{F}_T|\mathcal{F}_{T-1}}\left(f_T(\boldsymbol{u}_T, \sum_{t'=1}^{T} A_{t'T}\widetilde{d}_{t'} + B_T, \boldsymbol{w}_T(\widetilde{d}_{[T]})) \right) \cdots \Bigg\} \Bigg], \qquad (6.32) \\
\text{s.t.} \quad & \underline{q}\boldsymbol{u}_t \leq \sum_{t'=1}^{t} A_{t't}\widetilde{d}_{t'} + B_t \leq \overline{q}\boldsymbol{u}_t, \quad t \in \{1, 2, \ldots, T\}, \qquad (6.33) \\
& \mathbf{1}^\top \Big(\sum_{t'=1}^{t} A_{t't}\widetilde{d}_{t'} + B_t \Big) \geq \widetilde{d}_t, \quad \forall t \in \{1, 2, \ldots, T\} \qquad (6.34) \\
& (\boldsymbol{u}_1, A_{11}d_1 + B_1, \boldsymbol{w}_1) \in \mathcal{X}_1, \qquad (6.35) \\
& (\boldsymbol{u}_t, \sum_{t'=1}^{t} A_{t't}\widetilde{d}_{t'} + B_t, \boldsymbol{w}_t(\widetilde{d}_{[t]})) \in \\
& \mathcal{X}_t(\boldsymbol{u}_{t-1}, \sum_{t'=1}^{t-1} A_{t'(t-1)}\widetilde{d}_{t'} + B_{t-1}, \boldsymbol{w}_t(\widetilde{d}_{[t-1]}), \widetilde{d}_{[t]}), \quad \forall t \in \{2, 3, \ldots, T\}, \\
& \qquad (6.36) \\
& \boldsymbol{u}_t \in \{0,1\}^I_+, A_{t't} \in \mathbb{R}^{I \times t'}, B_t \in \mathbb{R}^{I \times 1}, \boldsymbol{w}_t(\widetilde{d}_{[t]}) \in \mathbb{R}^J_+, \\
& \qquad \forall t \in \{1, 2, \ldots, T\}, t' \in \{1, 2, \ldots, t\}, \qquad (6.37)
\end{aligned}$$

Algorithm 4 Rolling Horizon Algorithm with Affine Decision Rules for RA-A

Require: An integer $\tau \in \{1, \ldots, T\}$
Initialize: Set update epoch $\bar{t} = 1$ and policy $\{\overline{\boldsymbol{u}}_t(\cdot), \overline{\boldsymbol{v}}_t(\cdot), \overline{\boldsymbol{w}}_t(\cdot)\}_{t=1}^{T} = null$
for $\bar{t} \leq T$ **do**
  for $n \in \Omega_{\bar{t}}$ **do**
    Given the demand history $d_1, d_2, \ldots, d_{\bar{t}-1}$ corresponding to node n and $\{\overline{\boldsymbol{u}}_t(\cdot), \overline{\boldsymbol{v}}_t(\cdot), \overline{\boldsymbol{w}}_t(\cdot)\}_{t=1}^{\bar{t}-1}$, solve RA-N-ADR or RA-N-SADR. Let $\{\widehat{\boldsymbol{u}}_t(\cdot), \widehat{\boldsymbol{v}}_t(\cdot), \widehat{\boldsymbol{w}}_t(\cdot)\}_{t=\bar{t}}^{T}$ be an optimal policy.
    $(\overline{\boldsymbol{u}}_t(\cdot), \overline{\boldsymbol{v}}_t(\cdot), \overline{\boldsymbol{w}}_t(\cdot)) \leftarrow (\widehat{\boldsymbol{u}}_t(\cdot), \widehat{\boldsymbol{v}}_t(\cdot), \widehat{\boldsymbol{w}}_t(\cdot))$ for stages $t \in \{\bar{t}, \bar{t}+1, \ldots, \bar{t}+\tau-1\}$.
    Update: $\bar{t} \leftarrow \bar{t} + \tau$
  end for
end for
Return: Policy $\{\overline{\boldsymbol{u}}_t(\cdot), \overline{\boldsymbol{v}}_t(\cdot), \overline{\boldsymbol{w}}_t(\cdot)\}_{t=1}^{T}$

where $\mathbf{1} \in \mathbb{R}^I$ is the vector of all ones. $\underline{q} := (\underline{q}_1, \ldots, \underline{q}_I)$ and $\overline{q} := (\overline{q}_1, \ldots, \overline{q}_I)$. A similar model RA-N-SADR can be written for the simpler affine decision rule SADR.

Therefore, we can modify Algorithm 3 where we use affine decision rules in the approximating problem.

Let $z^{RH-ADR}(\tau)$ and $z^{RH-SADR}(\tau)$ be the objective values corresponding to the policy obtained from Algorithm 4 with ADR and SADR, respectively.

6.5 Computational Experiments

We conduct a set of computational experiments on a risk-averse lot sizing problem in order to observe the performance of the approximations to the risk-averse multi-stage production planning problem and verify the findings in the previous sections. All computational experiments are performed on an Intel(R) Core(TM) i7-4790 CPU@3.60 GHz computer with 8.00 GB of RAM. The algorithm is implemented on Java 1.8.0.31 where IBM ILOG CPLEX version 12.6 with default settings is used as the solver.

We generate problem instances similar to the ones in Section 3.3 except we define $\widetilde{d}_t \sim U[50-100\epsilon, 50+100\epsilon]$ where $\epsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ is a parameter that controls the variability of the demand values. The lower production limits are also set to one fourth of the upper production limits. We use conditional mean upper semi-deviation risk measure

$$\rho_{\mathcal{F}_{t+1}|\mathcal{F}_t}(Z_{t+1}) = \mathbb{E}[Z_{t+1}|\mathcal{F}_t] + \lambda\mathbb{E}[(Z_{t+1} - \mathbb{E}[Z_{t+1}|\mathcal{F}_t])_+|\mathcal{F}_t] \quad \forall Z_{t+1} \in \mathcal{Z}_{t+1}, \tag{6.38}$$

for all $t \in \{1, \ldots, T-1\}$ in our experiment. We also consider $\lambda \in \{0, 0.25, 0.5\}$ to control the degree of risk-aversion. Note that, if $\lambda = 0$, then the problem is risk-neutral.

We consider $T = 5$ and 6 stage instances where the demand can take K different values equally likely and independently at each stage. For $T = 5$, we let $K \in \{2, 3, 4\}$ and for $T = 6$, we let $K \in \{2, 3\}$. Note that for $K = 2, 3$ or 4, the corresponding scenario tree is a binary, ternary and quaternary tree, respectively. Moreover, the total number of scenarios in an instance is K^{T-1}. The detailed results are presented in Tables C.1-C.13 in Appendix C.

We first investigate how close the optimal value of RA-A to the optimal value of RA-N in order to measure the benefit of solving the model with adaptive setup decisions. Thus, we define a metric called as the *value of adaptive setup decisions* as:

$$\text{VAS } (\%) = \frac{z^N - z^A}{z^A},$$

similar to the definition of VAC in Chapter 5. VAS defines the percentage of increase in the objective value when RA-N is solved instead of RA-A. The exact value of VAS can only be calculated if both RA-A and RA-N are solved optimally. Since for some instances of RA-A cannot be solved optimally when $T = 5, K = 4$ and $T = 6, K = 3$, we present the analysis for the remaining instances. The results are presented in Figure 6.3.

As shown in Figure 6.3, VAS increases with the variability in demand values (ϵ) and the number of stages (T) and decreases with the degree of risk-aversion (λ). The largest VAS value 8.11% is attained when $T = 6, K = 2, \lambda = 0$ and

$\epsilon = 0.5$. Moreover, VAS is zero for all instances with low demand variability $\epsilon = 0.1$. The VAS and hence the importance of the model with adaptive setup decisions increase with variability and the number of stages and decrease with the degree of risk-aversion. Note that, this result is also consistent with the theoretical and computational results in Chapter 5. We consider a similar comparison for the models with adaptive and partially adaptive setup decisions. We define the *relative value of adaptive setup decisions* compared to partially adaptive setup decisions as:

$$\text{VAS}(\tau)\ (\%) = \frac{z^P(\tau) - z^A}{z^A},$$

where VAS(τ) (%) measures the percentage of increase in the objective value if RA-P with τ is solved instead of RA-A. The results are given in Figure 6.4.

As seen in Figure 6.4, VAS(τ)(%) values also increase with demand variability and the number of stages and decreases with the degree of risk-aversion, in general. Although we cannot always expect that VAS(2) $\leq$ VAS(3) as shown in Example 7, the average performance of RA-P with $\tau = 2$ (VAS(2)% = 0.45 on the average) is better than RA-P with $\tau = 3$ (VAS(3)% = 1.08 on the average).

We also define

$$\text{VAS-RH}(\tau)\ (\%) = \frac{z^{RH}(\tau) - z^A}{z^A},$$

in order to measure the percentage of increase in the objective value when the solution of the RH algorithm is used instead of the optimal fully adaptive setup decisions.

In Figure 6.5, we can observe that the relative importance of the adaptive setup decisions compared to the rolling horizon solutions also increases with variability and the number of stages and decreases with the degree of risk aversion. Moreover, as τ decreases, the RH algorithm performs better in terms of objective value with a cost of computation time. The average VAS-RH(τ) (%) values for $\tau = 1, 2$ and 3 are 1.40 %, 1.40 % and 1.47 %, respectively whereas the average computation times are 0.53, 0.32 and 0.16, respectively.

Finally, in order to measure the relative advantage of the optimal solution

with adaptive setup decisions compared to the solution obtained from the RH with affine decision rules, we define

$$\text{VAS-RH-ADR}(\tau)\ (\%) = \frac{z^{RH-ADR}(\tau) - z^A}{z^A},$$

and

$$\text{VAS-RH-SADR}(\tau)\ (\%) = \frac{z^{RH-SADR}(\tau) - z^A}{z^A},$$

as the percentage of increase in the objective value when the RH algorithm with ADR and SADR is used instead of RA-A, respectively. The results are given in Figures 6.6 and 6.7

It can be observed from Figures 6.6 and 6.7 that, the RH algorithm with affine decisions rules cannot perform better that the RH algorithm. For some instances, the approximation error is more than 18.24 %. Moreover, it is not easy to capture the effect on problem parameters such as ϵ, T and λ on the quality of the obtained solution. However, we present some larger instances where RH with affine decision rule saves computation time later in this section.

We investigate the relation among all approximations to RA-A for the instance $T = 6, K = 3, \lambda = 0.25$ and $\epsilon = 0.5$. The results of the experiments on this instances are summarized in Table 6.1.

As seen in Table 6.1, the model with adaptive setup decisions cannot be solved due to an "out of memory" error as in other large instances as shown in Table C.1. For this instance, even the model with partially adaptive setup decisions with $\tau = 2$ cannot be solved optimally after one hour of time limit and terminates with 1.75% optimality gap and the best incumbent value of 14212.81. On the other hand, the model with partially adaptive setup decisions with $\tau = 3$ can be solved only in 73.4 seconds with the optimal value of 14607.68. Although it requires only 0.8 seconds, the model with non-adaptive setup decisions performs the worst among these models. Moreover, we have $z^A \leq z^P(2) \leq z^N$ and $z^A \leq z^P(3) \leq z^N$, as we have expected.

Another result in Table 6.1 is the promising performance of the rolling horizon

Table 6.1: CPU times (in seconds) and the objective values for the instance $T = 6, K = 3, \lambda = 0.25$ and $\epsilon = 0.5$.

	Value	Time (in seconds)
z^A	***	***
$z^P(2)$	14212.81	3600 (1.75 %)
$z^P(3)$	14607.68	73.4
z^N	15383.73	0.8
$z^{RH}(1)$	14966.23	3.3
$z^{RH}(2)$	15026.68	1.6
$z^{RH}(3)$	15383.73	1.0
$z^{RH-ADR}(1)$	15402.47	6.9
$z^{RH-ADR}(2)$	15760.2	3.6
$z^{RH-ADR}(3)$	15705.36	2.0
$z^{RH-SADR}(1)$	15402.47	7.7
$z^{RH-SADR}(2)$	15760.2	3.7
$z^{RH-SADR}(3)$	15705.36	1.9

algorithm. For $\tau = 1, 2$ and 3, the running times of the rolling horizon algorithm are 3.3,1.6 and 1.0 seconds, respectively. Despite its short running time, the rolling horizon algorithm could improve the optimal solution of RA-N through iterations. Moreover, as τ increases, the number of problems to be solved through the execution of the rolling horizon algorithm decrease. The results also verify Proposition 10, that is, $z^A \leq z^{RH}(\tau) \leq z^N$ holds as expected for $\tau \in \{1, 2, 3\}$.

The results presented in Table 6.1, lead to the following chain of inequalities: $z^A \leq z^P(2) \leq z^P(3) \leq z^{RH}(1) \leq z^{RH}(2) \leq z^{RH}(3) = z^N \leq z^{RH-ADR}(1) = z^{RH-SADR}(1) \leq z^{RH-ADR}(3) = z^{RH-SADR}(3) \leq z^{RH-ADR}(2) = z^{RH-SADR}(2)$.

In Table 6.1, we can also observe that the rolling horizon policies with affine decision rules are neither time nor cost effective. Therefore, we conduct another set of experiments on larger problem instances. Table 6.2 presents the CPU times (in seconds) and the objective values of the rolling horizon algorithm without affine decision rules, with ADR and with SADR for $T = 10, K = 2, \lambda = \epsilon = 0.5$ instances with $\tau = 1$. For this large instance, RA-A and RA-P terminate due to

an "out-of-memory" error without reporting any feasible solution.

In Table 6.2, it can be seen that although the rolling horizon algorithm gives the best objective values, it takes longer times if no affine decision rule is used. The average CPU time for the rolling horizon algorithm is 51.5 seconds although ADR and SADR require 46.6 and 15.5 seconds, on the average. Another observation from Table 6.2 is that using SADR in the rolling horizon algorithm is computationally more efficient than ADR although both perform equally for all instances in terms of objective. Also, note that, the RH algorithm with or without affine decision rules can find a feasible solution to the problem despite RA-A and RA-P fail to do so.

6.6 Conclusion

In this chapter of the thesis, we consider a risk-averse model for multi-period production problem where the decisions are the setup and production decisions for a set of production units. Although the risk-averse model provides flexibility to adapt the dynamic environment, its solution requires great computational effort. In order to reduce this computational effort, we propose alternative models with non-adaptive and partially- adaptive setup decisions. Moreover, we propose a rolling horizon algorithm to the model with adaptive setup decisions. We also consider restricting the production decisions to affine functions of demand history to reduce to computation time of the rolling horizon algorithm. We present analytical and computational results which investigate the quality of solutions obtained by the approximations. The computational results reveal that the benefit of solving the fully adaptive model increases with variability of demand values and the number of stages and decreases with the degree of risk-aversion. Moreover, it is observed that the RH algorithm with affine decision rules can be employed to solve large problem instances for which RA-A and RA-P fail.

Figure 6.3: Results of the computational experiments on the VAS(%) for the risk-averse lot-sizing problem with respect to different variability levels (ϵ) and degrees of risk aversion (λ).

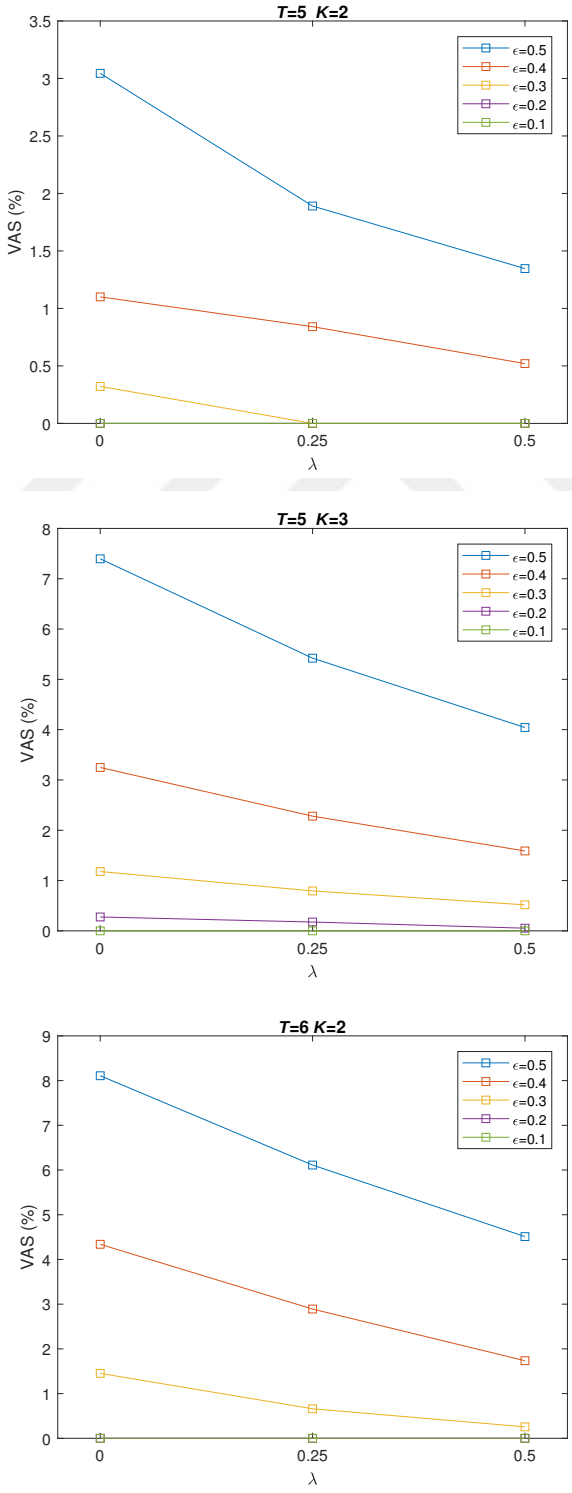


Figure 6.4: Results of the computational experiments on the VAS(τ)(%) for the risk-averse lot-sizing problem with respect to different variability levels (ϵ), degrees of risk aversion (λ) and $\tau \in \{2, 3\}$.

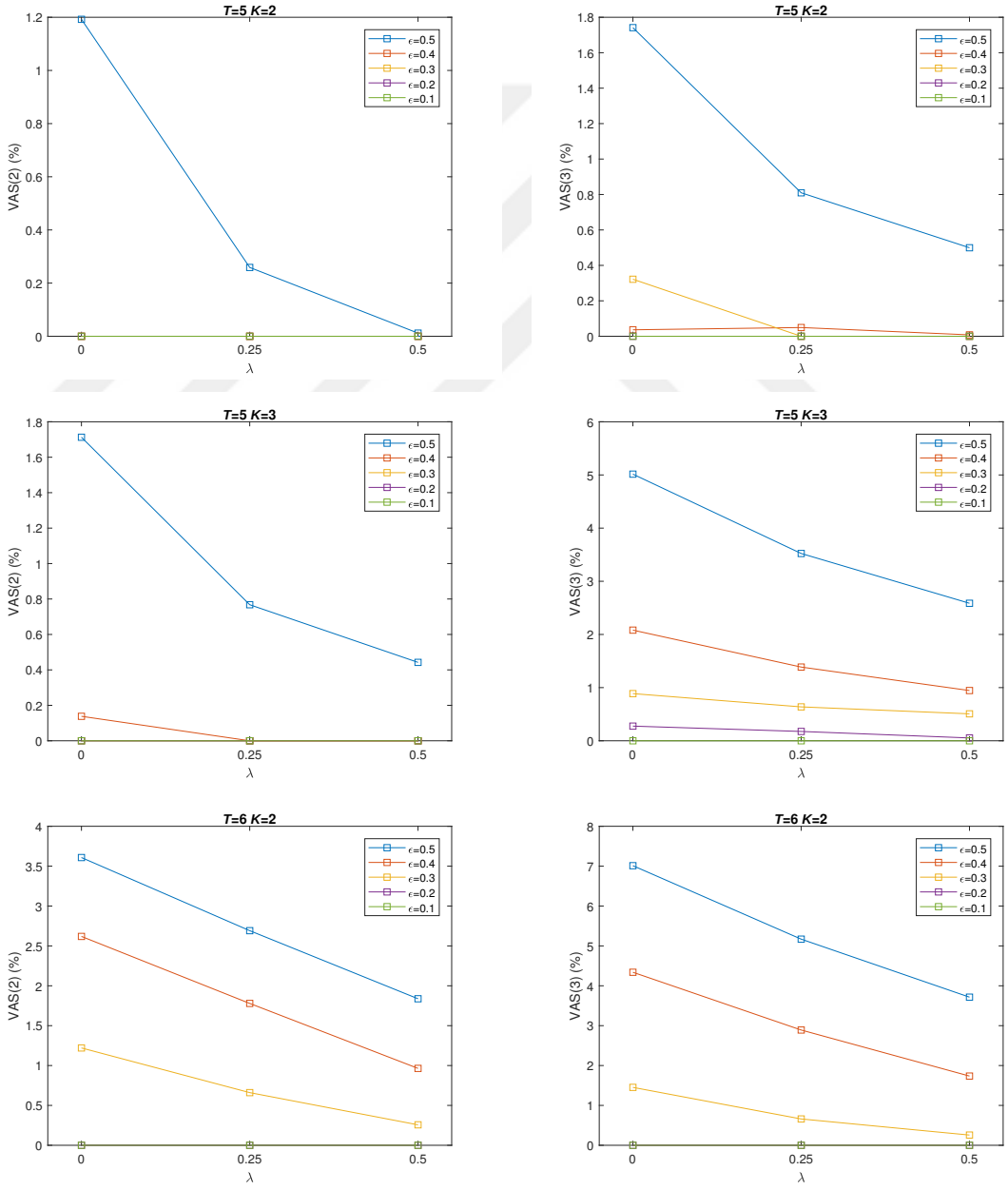


Figure 6.5: Results of the computational experiments on the VAS-RH(τ)(%) for the risk-averse lot-sizing problem with respect to different variability levels (ϵ), degrees of risk aversion levels (λ) and $\tau \in \{1, 2, 3\}$.

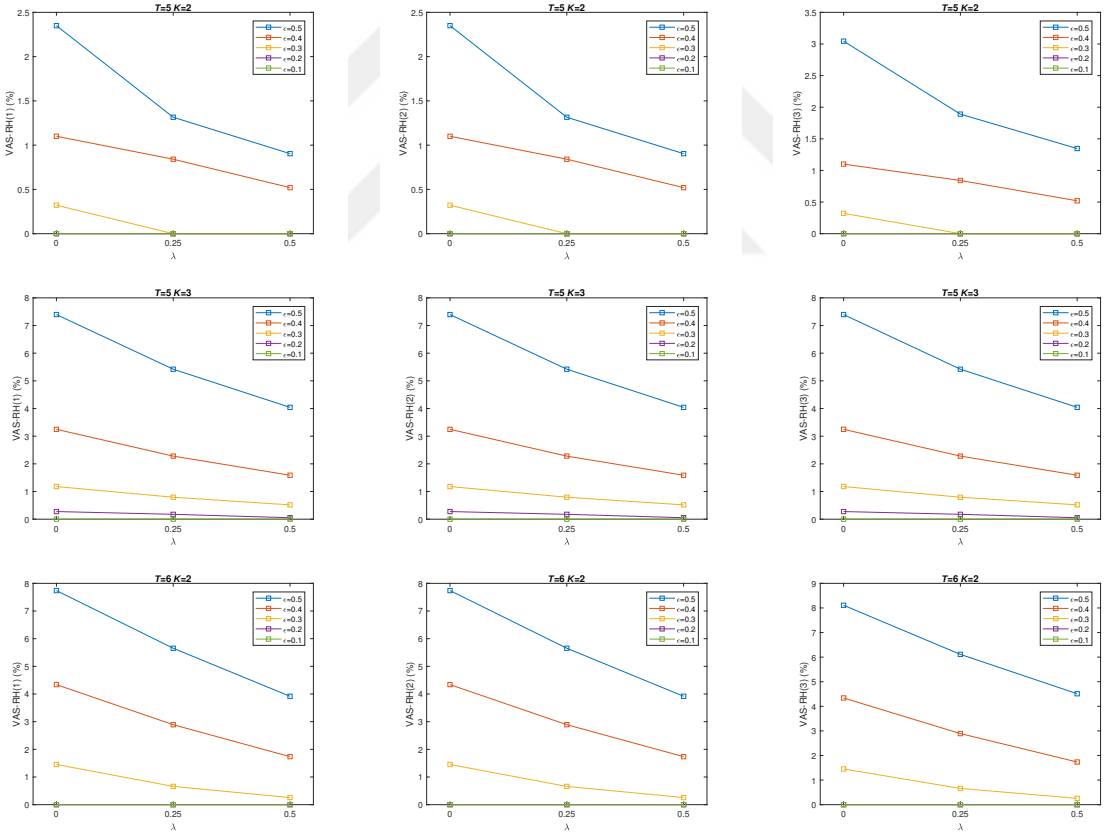


Figure 6.6: Results of the computational experiments on the VAS-RH-ADR(τ)(%) for the risk-averse lot-sizing with respect to different variability (ϵ), degree of risk aversion levels (λ) and $\tau \in \{1, 2, 3\}$.

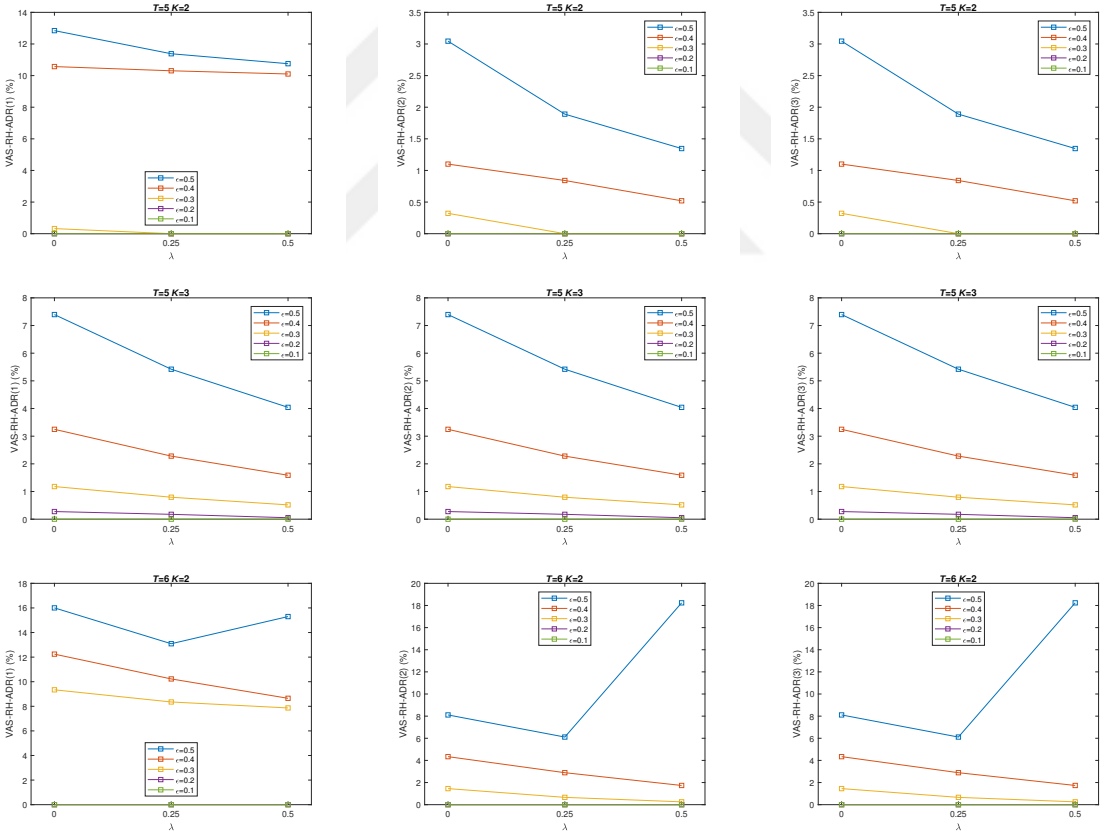


Figure 6.7: Results of the computational experiments on the VAS-RH-SADR(τ)(%) for the risk-averse lot-sizing with respect to different variability (ϵ), degree of risk aversion levels (λ) and $\tau \in \{1, 2, 3\}$.

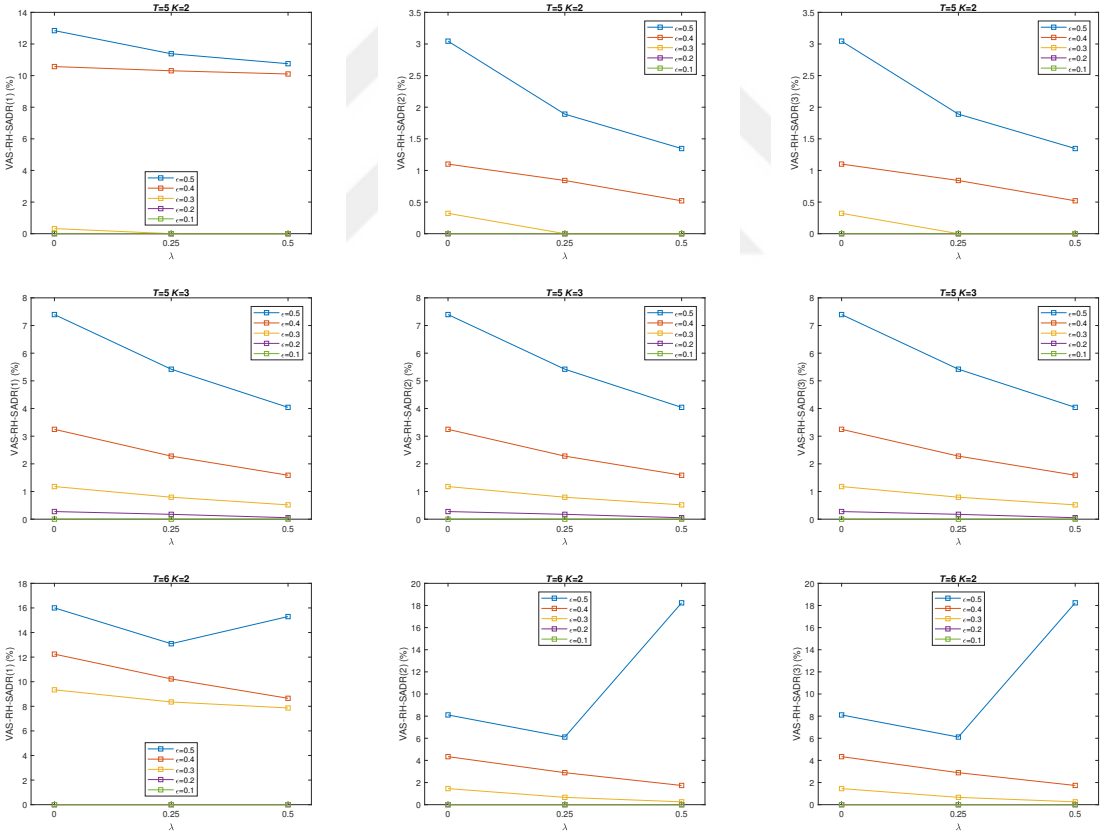


Table 6.2: the CPU times (in seconds) and the objective values of the rolling horizon algorithm without affine decision rules, with ADR and with SADR for a $T = 10, K = 3, \lambda = \epsilon = 0.5$ instances with $\tau = 1$.

λ	ϵ		RH	RH-ADR	RH-SADR
0	0.1	Time	51.3	44.0	13.0
		Obj.	18141.27	18141.27	18141.27
	0.2	Time	43.2	38.7	13.9
		Obj.	18877.02	19796.90	19796.90
	0.3	Time	43.4	37.9	15.4
		Obj.	19614.35	20571.58	20571.58
	0.4	Time	44.2	39.1	14.3
		Obj.	20349.43	21511.28	21511.28
	0.5	Time	45.6	41.1	14.6
		Obj.	21199.92	23636.98	23636.98
0.25	0.1	Time	56.2	50.8	14.5
		Obj.	18217.19	18217.19	18217.19
	0.2	Time	53.2	49.5	15.1
		Obj.	19030.81	19962.75	19962.75
	0.3	Time	52.6	49.0	16.6
		Obj.	19861.93	20766.52	20766.52
	0.4	Time	53.1	47.7	17.5
		Obj.	20674.30	21762.39	21762.39
	0.5	Time	57.2	52.3	17.1
		Obj.	21597.18	24102.10	24102.10
0.5	0.1	Time	55.1	49.8	14.5
		Obj.	18292.94	18292.94	18292.94
	0.2	Time	53.3	48.2	15.4
		Obj.	19184.90	20131.12	20131.12
	0.3	Time	53.1	48.5	16.3
		Obj.	20102.48	20967.80	20967.80
	0.4	Time	54.0	49.9	17.9
		Obj.	21001.19	22027.73	22027.73
	0.5	Time	56.9	51.7	17.0
		Obj.	22002.44	24565.23	24565.23

Chapter 7

Conclusion

In this dissertation, we consider risk-averse multi-stage mixed-integer stochastic programming which form a class of extremely challenging optimization problems. In Chapter 3, we propose a scenario tree decomposition algorithm, called as scenario grouping, for risk-averse multi-stage mixed-integer stochastic problems with a dynamic objective function defined via mean-CVaR. The suggested algorithm is used to find lower and upper bounds on the optimal value of the problem. We investigate the effect of scenario partitioning strategies on the quality of the different lower bounds by considering different partitioning strategies based on the structure of the scenario tree and disparateness of scenario realizations.

In Chapter 4, we extend the findings of Chapter 3 to an exact solution procedure. The proposed method is based on an evaluate-and-cut procedure where the lower bounds are obtained from group subproblems. Moreover, we show that, under the assumption that the first stage integer variables are bounded, the problem with mixed-integer variables in all stages can be solved with the proposed algorithm. As the computational experiments reveal, under modest computational settings, the proposed algorithm is able to solve large problem instances within a reasonable time.

In Chapter 5, we consider an application of risk-averse multi-stage mixed-integer stochastic programming problems to a power system optimization problem. Uncertainty and variability in net load arising from the increasing penetration of renewable technologies have motivated study of various classes of stochastic unit commitment models in recent years. In the models with non-adaptive commitment, the generation schedule for the entire day is fixed whereas the dispatch is adapted to the uncertainty. On the other hand, in the models with adaptive commitment, the generation schedule is also allowed to dynamically adapt to the uncertainty realization. The latter ones provide more flexibility in the generation schedule, however, they require significantly higher computational effort than the former ones. In order to justify this additional computational effort, in this chapter, we provide theoretical and empirical analyses of the value of adaptive commitment for risk-averse multi-stage stochastic unit commitment models.

In Chapter 6, we consider risk-averse multi-stage production planning problems as a generalization of the unit commitment problem discussed in Chapter 5. In these problems, we first consider two models with respect to their adaptivity to the uncertainty. In fully adaptive models, both setup and production decisions are given in on-line fashion, that is, they are adapted to demand uncertainty. However, in the models with non-adaptive setup decisions, the setup decisions are off-line, that is, they are fixed at the beginning of the decision horizon whereas the production decisions are on-line. As an intermediate case, we also consider a model with partially adaptive setup decisions. Moreover, we propose a rolling horizon approach for the problem. In order to reduce the computational effort of the rolling horizon algorithm, we consider restricting the production amounts to affine functions of demand realizations during the execution of the algorithm. We conduct a set of computational experiments on a risk-averse multi-stage lot sizing problem to investigate the computational efficiency of the proposed models and verify the analytical results.

In Chapters 3 and 4, the group subproblems can be assigned to different threads of a computer and solved in parallel during the execution of Algorithms 1 and 2. Similarly, parallel computing can be used to solve the problems in the upper

bounding phase of these algorithms. Parallel implementation of the proposed algorithms may decrease the running time significantly, especially when the number of groups is large. Therefore, it can be considered as a future research direction. Another possible extension of the study is to find better scenario partitioning strategies in Chapter 3. Since the scenario partitioning strategies have a direct impact on the quality of the bounds, it can be interesting to seek the partitioning strategy which yields the best lower and upper bounds.

We consider unit commitment and multi-period production planning problems in Chapters 5 and 6, respectively. However, the theoretical results and the proposed algorithms in these chapters can also be considered for other risk-averse multi-stage problems. Performance of the rolling horizon polices with affine decision rules are promising. As a future research direction, it would be interesting to consider these rolling horizon policies in different problems.

Bibliography

[1] G. B. Dantzig, "Linear programming under uncertainty," *Management Science*, vol. 1, no. 3-4, pp. 197–206, 1955.

[2] A. Prékopa, *Stochastic programming*. Kluwer Academic Publishers Group, Dordrecht, 1995.

[3] A. Shapiro, D. Dentcheva, *et al.*, *Lectures on stochastic programming: modeling and theory*, vol. 9. SIAM, 2009.

[4] J. R. Birge and F. Louveaux, *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[5] S. Ahmed, "Two-stage stochastic integer programming: A brief introduction," *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[6] N. Miller and A. Ruszczyński, "Risk-averse two-stage stochastic linear programming: Modeling and decomposition," *Operations Research*, vol. 59, no. 1, pp. 125–132, 2011.

[7] K. J. Arrow, *Aspects of the theory of risk-bearing*. Yrjö Jahnssonin Säätiö, 1965.

[8] J. W. Pratt, "Risk aversion in the small and in the large," *Econometrica: Journal of the Econometric Society*, vol. 32, no. 1, pp. 122–136, 1964.

[9] J. Hadar and W. R. Russell, "Rules for ordering uncertain prospects," *The American Economic Review*, vol. 59, no. 1, pp. 25–34, 1969.

[10] V. S. Bawa, "Optimal rules for ordering uncertain prospects," *Journal of Financial Economics*, vol. 2, no. 1, pp. 95–121, 1975.

[11] A. Charnes and W. W. Cooper, "Chance-constrained programming," *Management Science*, vol. 6, no. 1, pp. 73–79, 1959.

[12] B. K. Pagnoncelli, S. Ahmed, and A. Shapiro, "Sample average approximation method for chance constrained programming: theory and applications," *Journal of Optimization Theory and Applications*, vol. 142, no. 2, pp. 399–416, 2009.

[13] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[14] A. Shapiro, "Stochastic programming approach to optimization under uncertainty," *Mathematical Programming*, vol. 112, no. 1, pp. 183–220, 2008.

[15] S. Ahmed, "Convexity and decomposition of mean-risk stochastic programs," *Mathematical Programming*, vol. 106, no. 3, pp. 433–446, 2006.

[16] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[17] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking and Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[18] A. Ruszczyński and A. Shapiro, "Optimization of convex risk functions," *Mathematics of Operations Research*, vol. 31, no. 3, pp. 433–452, 2006.

[19] A. Ruszczyński and A. Shapiro, "Conditional risk mappings," *Mathematics of Operations Research*, vol. 31, no. 3, pp. 544–561, 2006.

[20] A. Shapiro, "On a time consistency concept in risk averse multistage stochastic programming," *Operations Research Letters*, vol. 37, no. 3, pp. 143–147, 2009.

[21] M. Pereira and L. M. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Mathematical Programming*, vol. 52, no. 1-3, pp. 359–375, 1991.

[22] A. Shapiro, "Analysis of stochastic dual dynamic programming method," *European Journal of Operational Research*, vol. 209, no. 1, pp. 63–72, 2011.

[23] A. Shapiro, W. Tekaya, J. P. da Costa, and M. P. Soares, "Risk neutral and risk averse stochastic dual dynamic programming method," *European Journal of Operational Research*, vol. 224, no. 2, pp. 375–391, 2013.

[24] A. Philpott, V. de Matos, and E. Finardi, "On solving multistage stochastic programs with coherent risk measures," *Operations Research*, vol. 61, no. 4, pp. 957–970, 2013.

[25] J. F. Bonnans, Z. Cen, and T. Christel, "Energy contracts management by stochastic programming techniques," *Annals of Operations Research*, vol. 200, no. 1, pp. 199–222, 2012.

[26] S. Bruno, S. Ahmed, A. Shapiro, and A. Street, "Risk neutral and risk averse approaches to multistage renewable investment planning under uncertainty," *European Journal of Operational Research*, vol. 250, no. 3, pp. 979–989, 2016.

[27] J. Zou, S. Ahmed, and X. A. Sun, "Stochastic dual dynamic integer programming," *Mathematical Programming*, pp. 1–42, 2017.

[28] C. C. CarøE and R. Schultz, "Dual decomposition in stochastic integer programming," *Operations Research Letters*, vol. 24, no. 1-2, pp. 37–45, 1999.

[29] R. Schultz, "Stochastic programming with integer variables," *Mathematical Programming*, vol. 97, no. 1-2, pp. 285–309, 2003.

[30] R. Schultz and S. Tiedemann, "Risk aversion via excess probabilities in stochastic programs with mixed-integer recourse," *SIAM Journal on Optimization*, vol. 14, no. 1, pp. 115–138, 2003.

[31] R. Schultz and S. Tiedemann, "Conditional value-at-risk in stochastic programs with mixed-integer recourse," *Mathematical Programming*, vol. 105, no. 2-3, pp. 365–386, 2006.

[32] R. T. Rockafellar and R. J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of Operations Research*, vol. 16, no. 1, pp. 119–147, 1991.

[33] J.-P. Watson and D. L. Woodruff, "Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems," *Computational Management Science*, vol. 8, no. 4, pp. 355–370, 2011.

[34] D. Gade, G. Hackebeil, S. M. Ryan, J.-P. Watson, R. J.-B. Wets, and D. L. Woodruff, "Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs," *Mathematical Programming*, vol. 157, no. 1, pp. 47–67, 2016.

[35] S. Atakan and S. Sen, "A progressive hedging based branch-and-bound algorithm for mixed-integer stochastic programs," *Computational Management Science*, pp. 1–40, 2018.

[36] A. Ruszczyński, "Decomposition methods," *Handbooks in operations research and management science*, vol. 10, pp. 141–211, 2003.

[37] R. A. Collado, D. Papp, and A. Ruszczyński, "Scenario decomposition of risk-averse multistage stochastic programming problems," *Annals of Operations Research*, vol. 200, no. 1, pp. 147–170, 2012.

[38] B. Sandıkçı, N. Kong, and A. J. Schaefer, "A hierarchy of bounds for stochastic mixed-integer programs," *Mathematical Programming*, vol. 138, no. 1-2, pp. 253–272, 2013.

[39] B. Sandıkçı and O. Y. Ozaltin, "A scalable bounding method for multistage stochastic programs," *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1772–1800, 2017.

[40] G. L. Zenarosa, O. A. Prokopyev, and A. J. Schaefer, "Scenario-tree decomposition: Bounds for multistage stochastic mixed-integer programs," 2014. Available on Optimization Online.

[41] F. Maggioni, E. Allevi, and M. Bertocchi, "Monotonic bounds in multistage mixed-integer stochastic programming," *Computational Management Science*, vol. 13, no. 3, pp. 423–457, 2016.

[42] F. Maggioni and G. C. Pflug, "Bounds and approximations for multistage stochastic programs," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 831–855, 2016.

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

[44] Y. Deng, S. Ahmed, and S. Shen, "Parallel scenario decomposition of risk-averse 0-1 stochastic programs," *INFORMS Journal on Computing*, vol. 30, no. 1, pp. 90–105, 2017.

[45] K. Huang and S. Ahmed, "The value of multistage stochastic programming in capacity planning under uncertainty," *Operations Research*, vol. 57, no. 4, pp. 893–904, 2009.

[46] A. İ. Mahmutoğulları, Ö. Çavuş, and M. S. Aktürk, "Bounds on risk-averse mixed-integer multi-stage stochastic programming problems with mean-cvar," *European Journal of Operational Research*, vol. 266, no. 2, pp. 595–608, 2018.

[47] D. A. Iancu, M. Petrik, and D. Subramanian, "Tight approximations of dynamic risk measures," *Mathematics of Operations Research*, vol. 40, no. 3, pp. 655–682, 2015.

[48] G. C. Pflug and W. Romisch, *Modeling, measuring and managing risk*. World Scientific, 2007.

[49] A. Eichhorn and W. Römisch, "Polyhedral risk measures in stochastic programming," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 69–95, 2005.

[50] Y. Guan, S. Ahmed, and G. L. Nemhauser, "Cutting planes for multistage stochastic integer programs," *Operations Research*, vol. 57, no. 2, pp. 287–298, 2009.

[51] C. D'Ambrosio, A. Frangioni, L. Liberti, and A. Lodi, "On interval-subgradient and no-good cuts," *Operations Research Letters*, vol. 38, no. 5, pp. 341–345, 2010.

[52] F. Glover, "Improved linear integer programming formulations of nonlinear integer problems," *Management Science*, vol. 22, no. 4, pp. 455–460, 1975.

[53] L. Ntaimo and S. Sen, "The million-variable "march" for stochastic combinatorial optimization," *Journal of Global Optimization*, vol. 32, no. 3, pp. 385–400, 2005.

[54] I. Bakir, N. Boland, B. Dandurand, and A. Erera, "Scenario set partition dual bounds for multistage stochastic programming: A hierarchy of bounds and a partition sampling approach," 2016. Available at Optimization Online.

[55] S. Jin, S. M. Ryan, J.-P. Watson, and D. L. Woodruff, "Modeling and solving a large-scale generation expansion planning problem under uncertainty," *Energy Systems*, vol. 2, no. 3-4, pp. 209–242, 2011.

[56] S. A. Kazarlis, A. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 83–92, 1996.

[57] N. P. Padhy, "Unit commitment-a bibliographical survey," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1196–1205, 2004.

[58] Y. Huang, P. M. Pardalos, and Q. P. Zheng, *Electrical power unit commitment: deterministic and two-stage stochastic programming models and algorithms*. Springer, 2017.

[59] P. A. Ruiz, C. R. Philbrick, E. Zak, K. W. Cheung, and P. W. Sauer, "Uncertainty management in the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 642–651, 2009.

[60] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng, "Adaptive robust optimization for the security constrained unit commitment problem," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 52–63, 2013.

[61] A. Lorca and X. A. Sun, "Multistage robust unit commitment with dynamic uncertainty sets and energy storage," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 1678–1688, 2017.

[62] L. Zhao and B. Zeng, "Robust unit commitment problem with demand response and wind energy," in *Power and Energy Society General Meeting, 2012 IEEE*, pp. 1–8, IEEE, 2012.

[63] R. Jiang, J. Wang, and Y. Guan, "Robust unit commitment with wind power and pumped storage hydro," *IEEE Transactions on Power Systems*, vol. 27, no. 2, p. 800, 2012.

[64] Q. Wang, J.-P. Watson, and Y. Guan, "Two-stage robust optimization for nk contingency-constrained unit commitment," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2366–2375, 2013.

[65] K. Cheung, D. Gade, C. Silva-Monroy, S. M. Ryan, J.-P. Watson, R. J.-B. Wets, and D. L. Woodruff, "Toward scalable stochastic unit commitment," *Energy Systems*, vol. 6, no. 3, pp. 417–438, 2015.

[66] M. Tahanan, W. Van Ackooij, A. Frangioni, and F. Lacalandra, "Large-scale unit commitment under uncertainty," *4OR*, vol. 13, no. 2, pp. 115–171, 2015.

[67] A. Papavasiliou, S. S. Oren, and B. Rountree, "Applying high performance computing to transmission-constrained stochastic unit commitment for renewable energy integration," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1109–1120, 2015.

[68] J. Wang, M. Shahidehpour, and Z. Li, "Security-constrained unit commitment with volatile wind power generation," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1319–1327, 2008.

[69] S. Takriti, J. R. Birge, and E. Long, "A stochastic model for the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 11, no. 3, pp. 1497–1508, 1996.

[70] Q. Wang, Y. Guan, and J. Wang, "A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 206–215, 2012.

[71] Q. P. Zheng, J. Wang, P. M. Pardalos, and Y. Guan, "A decomposition approach to the two-stage stochastic unit commitment problem," *Annals of Operations Research*, vol. 210, no. 1, pp. 387–410, 2013.

[72] T. Shiina and J. R. Birge, "Stochastic unit commitment problem," *International Transactions in Operational Research*, vol. 11, no. 1, pp. 19–32, 2004.

[73] R. Jiang, Y. Guan, and J.-P. Watson, "Cutting planes for the multistage stochastic unit commitment problem," *Mathematical Programming*, vol. 157, no. 1, pp. 121–151, 2016.

[74] Q. P. Zheng, J. Wang, and A. L. Liu, "Stochastic optimization for unit commitment—a review," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1913–1924, 2015.

[75] Á. Lorca, X. A. Sun, E. Litvinov, and T. Zheng, "Multistage adaptive robust optimization for the unit commitment problem," *Operations Research*, vol. 64, no. 1, pp. 32–51, 2016.

[76] P. M. Subcommittee, "Ieee reliability test system," *IEEE Transactions on Power Apparatus and Systems*, no. 6, pp. 2047–2054, 1979.

[77] C.-L. Tseng, S. S. Oren, C. S. Cheng, C.-a. Li, A. J. Svoboda, and R. B. Johnson, "A transmission-constrained unit commitment method in power system scheduling," *Decision Support Systems*, vol. 24, no. 3-4, pp. 297–310, 1999.

[78] F. Maggioni, E. Allevi, and M. Bertocchi, "Measures of information in multistage stochastic programming," in *STOPROG-2012-Stochastic Programming for Implementation and Advanced Applications*, pp. 78–82, The Association of Lithuanian Serials, 2012.

[79] J. H. Bookbinder and J.-Y. Tan, "Strategies for the probabilistic lot-sizing problem with service-level constraints," *Management Science*, vol. 34, no. 9, pp. 1096–1108, 1988.

[80] U. Özen, M. K. Doğru, and S. A. Tarim, "Static-dynamic uncertainty strategy for a single-item stochastic inventory control problem," *Omega*, vol. 40, no. 3, pp. 348–357, 2012.

[81] M. Zamarripa, P. A. Marchetti, I. E. Grossmann, T. Singh, I. Lotero, A. Gopalakrishnan, B. Besancon, and J. André, "Rolling horizon approach for production–distribution coordination of industrial gases supply chains," *Industrial and Engineering Chemistry Research*, vol. 55, no. 9, pp. 2646–2660, 2016.

[82] L. Bertazzi and F. Maggioni, "A stochastic multi-stage fixed charge transportation problem: Worst-case analysis of the rolling horizon approach," *European Journal of Operational Research*, vol. 267, no. 2, pp. 555–569, 2018.

[83] S. J. Garstka and R. J.-B. Wets, "On decision rules in stochastic programming," *Mathematical Programming*, vol. 7, no. 1, pp. 117–143, 1974.

[84] D. Kuhn, W. Wiesemann, and A. Georghiou, "Primal and dual linear decision rules in stochastic and robust optimization," *Mathematical Programming*, vol. 130, no. 1, pp. 177–209, 2011.

[85] M. Bodur and J. Luedtke, "Two-stage linear decision rules for multi-stage stochastic programming," *arXiv preprint arXiv:1701.04102*, 2017.

[86] C. Gauvin, E. Delage, and M. Gendreau, "Decision rule approximations for the risk averse reservoir management problem," *European Journal of Operational Research*, vol. 261, no. 1, pp. 317–336, 2017.

Appendix A

Deterministic Unit Commitment Formulation

Indexes and Sets

t : Period index,
i : Generator index,
T : Number of periods,
I : Number of generators,
$\mathcal{T}$: Set of periods,
$\mathcal{I}$: Set of generators,
l : Transmission line index,
$\mathcal{L}$: Set of transmission lines,

Parameters

a_i : Fixed cost of running generator $i \in \mathcal{I}$,
$h_i(\cdot)$: Production cost function of running generator $i \in \mathcal{I}$,
specifically, $h_i(v) = b_i v + c_i v^2$ for $v \geq 0$
with parameters $b_i, c_i \in \mathbb{R}_+$,
SU_i : Start-up cost of generator $i \in \mathcal{I}$,
SD_i : Shut-down cost of generator $i \in \mathcal{I}$,
$\underline{q}_i$: Minimum production amount of generator $i \in \mathcal{I}$,

$\bar{q}_i$: Maximum production amount of generator $i \in \mathcal{I}$,

d_t : Net load in period $t \in \mathcal{T}$,

M_i : Minimum up time of generator $i \in \mathcal{I}$,

L_i : Minimum down time of generator $i \in \mathcal{I}$,

V_i' : Start up rate of generator $i \in \mathcal{I}$,

V_i : Ramp up rate of generator $i \in \mathcal{I}$,

B_i' : Shut down rate of generator $i \in \mathcal{I}$,

B_i : Ramp down production limit of generator $i \in \mathcal{I}$,

C_l : Capacity of transmission line $l \in \mathcal{L}$,

K : Flow line distribution matrix.

Variables

u_{it} : Status of generator $i \in \mathcal{I}$ in period $t \in \mathcal{T}$,
(1 if generator i is ON in period t; 0 otherwise),

v_{it} : Production amount of generator $i \in \mathcal{I}$ in period $t \in \mathcal{T}$,

y_{it} : Start up decision of generator $i \in \mathcal{I}$ in period $t \in \mathcal{T}$,
(1 if $u_{i(t-1)} = 0$ and $u_{it} = 1$; 0 otherwise),

z_{it} : Shut down decision of generator $i \in \mathcal{I}$ in period $t \in \mathcal{T}$,
(1 if $u_{i(t-1)} = 1$ and $u_{it} = 0$; 0 otherwise).

Model

$$\min_{u,v,y,z} \sum_{t=1}^{T} \sum_{i=1}^{I} a_i u_{it} + h_i(v_{it}) + SU_i y_{it} + SD_i z_{it}, \tag{A.1}$$

s.t. (5.2), (5.3)

$$u_{it} - u_{i(t-1)} \leq u_{i\tau}, \ \forall t \in \mathcal{T}, \forall i \in \mathcal{I}, \\ \forall \tau \in \{t+1, \ldots, \min\{t + M_i, T\}\} \tag{A.2}$$

$$u_{i(t-1)} - u_{it} \leq 1 - u_{i\tau}, \ \forall t \in \mathcal{T}, \forall i \in \mathcal{I}, \\ \forall \tau \in \{t+1, \ldots, \min\{t + L_i, T\}\} \tag{A.3}$$

$$u_{it} - u_{i(t-1)} \leq y_{it},\ \forall t \in \mathcal{T}, \forall i \in \mathcal{I} \tag{A.4}$$

$$u_{i(t-1)} - u_{it} \leq z_{it},\ \forall t \in \mathcal{T}, \forall i \in \mathcal{I} \tag{A.5}$$

$$v_{it} - v_{i(t-1)} \leq V_i' y_{it} + V_i u_{i(t-1)},$$
$$\forall t \in \mathcal{T}, \forall i \in \mathcal{I} \tag{A.6}$$

$$v_{i(t-1)} - v_{it} \leq B_i' z_{it} + B_i u_{it},$$
$$\forall t \in \mathcal{T}, \forall i \in \mathcal{I} \tag{A.7}$$

$$-C_l \leq K v_t \leq C_l,$$
$$\forall t \in \mathcal{T}, \forall l \in \mathcal{L} \tag{A.8}$$

$$u_{it}, y_{it}, z_{it} \in \{0,1\}, v_{ti} \geq 0,\ \forall t \in \mathcal{T}, \forall i \in \mathcal{I}.$$

The objective (A.1) is total fixed, production, start up and shut down costs. Constraints (A.2), (A.3), (A.4) and (A.5) are minimum up time, minimum down time, start up and shut down constraints, respectively. The rump/start up rate constraint is given in (A.6). Similarly, (A.7) is the rump/shut down rate constraint. Constraints (A.8) are the flow balance constraints in linear form as given in [73].

Appendix B

Computational Experiment Data

Table B.1: Demand Data (MW = megawatt)

t	1	2	3	4	5	6
$\overline{d}_t$ (MW)	700	750	850	950	1000	1100
t	7	8	9	10	11	12
$\overline{d}_t$ (MW)	1150	1200	1300	1400	1450	1500
t	13	14	15	16	17	18
$\overline{d}_t$ (MW)	1400	1300	1200	1050	1000	1100
t	19	20	21	22	23	24
$\overline{d}_t$ (MW)	1200	1400	1300	1100	900	800

Table B.2: Scenario Data

		Period (or hour) t			
Scenario	Probability	1-6	7-12	13-18	19-24
1	0.125	$\overline{d}_t$	$(1-\epsilon)\overline{d}_t$	$(1-\epsilon)\overline{d}_t$	$(1-\epsilon)\overline{d}_t$
2	0.125	$\overline{d}_t$	$(1-\epsilon)\overline{d}_t$	$(1-\epsilon)\overline{d}_t$	$(1+\epsilon)\overline{d}_t$
3	0.125	$\overline{d}_t$	$(1-\epsilon)\overline{d}_t$	$(1+\epsilon)\overline{d}_t$	$(1-\epsilon)\overline{d}_t$
4	0.125	$\overline{d}_t$	$(1-\epsilon)\overline{d}_t$	$(1+\epsilon)\overline{d}_t$	$(1+\epsilon)\overline{d}_t$
5	0.125	$\overline{d}_t$	$(1+\epsilon)\overline{d}_t$	$(1-\epsilon)\overline{d}_t$	$(1-\epsilon)\overline{d}_t$
6	0.125	$\overline{d}_t$	$(1+\epsilon)\overline{d}_t$	$(1-\epsilon)\overline{d}_t$	$(1+\epsilon)\overline{d}_t$
7	0.125	$\overline{d}_t$	$(1+\epsilon)\overline{d}_t$	$(1+\epsilon)\overline{d}_t$	$(1-\epsilon)\overline{d}_t$
8	0.125	$\overline{d}_t$	$(1+\epsilon)\overline{d}_t$	$(1+\epsilon)\overline{d}_t$	$(1+\epsilon)\overline{d}_t$

Appendix C

Results of Computational Experiments in Chapter 6

Table C.1: CPU times (in seconds) and the objective values of RA-A instances.

			$T=5$			$T=6$	
λ	ϵ		$K=2$	$K=3$	$K=4$	$K=2$	$K=3$
0	0.1	Time	0.2	0.3	**1.24%**	0.3	64.6
		Obj.	9415.86	9916.17	9730.32	11343.05	11782.75
	0.2	Time	0.1	0.5	**3.10%**	0.1	**1.18%**
		Obj.	9853.30	10503.80	10154.41	12035.57	12389.17
	0.3	Time	0.1	1.7	**2.38%**	0.2	**1.95%**
		Obj.	10257.57	11019.13	10340.87	12545.89	12842.93
	0.4	Time	0.0	8.1	**1.45%**	0.3	**1.49%**
		Obj.	10611.26	11395.39	10548.54	12862.63	13145.20
	0.5	Time	0.0	1.9	1681.9	0.2	**1.28%**
		Obj.	10835.47	11529.25	10685.05	13092.75	13351.83
0.25	0.1	Time	0.0	0.5	27.2	0.1	68.8
		Obj.	9448.08	9943.67	9773.42	11388.43	11852.38
	0.2	Time	0.0	0.5	**0.78%**	0.3	**0.80%**
		Obj.	9917.74	10569.21	10247.60	12126.30	12519.57
	0.3	Time	0.0	0.8	**1.46%**	0.5	***
		Obj.	10387.21	11143.16	10538.67	12779.80	***
	0.4	Time	0.2	6.0	**1.37%**	0.4	***
		Obj.	10766.29	11610.79	10837.50	13220.07	***
	0.5	Time	0.1	5.7	***	0.5	***
		Obj.	11116.26	11875.63	***	13552.95	***
0.5	0.1	Time	0.0	0.3	***	0.1	15.9
		Obj.	9478.04	9971.71	***	11433.06	11916.47
	0.2	Time	0.0	0.6	50.8	0.3	**0.34%**
		Obj.	9977.66	10638.15	9815.76	12215.52	12649.76
	0.3	Time	0.03	0.58	**0.03%**	0.20	**1.43%**
		Obj.	10477.07	11257.47	10323.20	12964.76	13225.77
	0.4	Time	0.0	3.7	***	0.3	**1.36%**
		Obj.	10919.87	11800.19	***	13545.53	13574.87
	0.5	Time	0.0	4.4	***	0.5	**1.01%**
		Obj.	11323.72	12167.68	***	13973.84	13955.31

For the instances that cannot be solved within the time limit of one hour **the optimality gap** returned by CPLEX is presented instead of CPU time.
***: Terminated due to out of memory error.

Table C.2: CPU times (in seconds) and the objective values of RA-P instances with $\tau = 2$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.2	0.2	0.9	0.1	2.5
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.0	0.6	22.1	0.0	**0.24%**
		Obj.	9853.30	10503.80	10154.41	12035.57	12618.86
	0.3	Time	0.0	0.4	15.6	0.2	**1.54%**
		Obj.	10257.57	11019.13	10420.12	12699.01	13164.15
	0.4	Time	0.0	0.5	4.8	0.2	**1.59%**
		Obj.	10611.26	11411.13	10643.11	13199.50	13584.63
	0.5	Time	0.0	0.2	2.3	0.1	**1.23%**
		Obj.	10964.66	11726.72	10766.42	13565.14	13974.16
0.25	0.1	Time	0.3	0.3	2.4	0.1	22.7
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.1	0.5	20.2	0.2	1231.6
		Obj.	9917.74	10569.21	10252.41	12126.30	12745.46
	0.3	Time	0.0	0.5	22.8	0.1	**2.01%**
		Obj.	10387.21	11143.16	10621.29	12864.20	13339.73
	0.4	Time	0.0	1.1	16.3	0.3	**2.28%**
		Obj.	10766.29	11610.79	10936.06	13455.23	13801.65
	0.5	Time	0.0	0.7	8.7	0.3	**1.75%**
		Obj.	11145.04	11966.80	11126.52	13917.66	14212.81
0.5	0.1	Time	0.0	0.3	1.7	0.1	12.6
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.0	0.4	8.0	0.1	444.5
		Obj.	9977.66	10638.15	10326.57	12215.52	12871.53
	0.3	Time	0.0	0.5	4.6	0.3	**1.31%**
		Obj.	10477.07	11257.47	10775.18	12998.02	13530.38
	0.4	Time	0.0	0.7	15.0	0.4	**1.71%**
		Obj.	10919.87	11800.03	11214.69	13676.14	14017.41
	0.5	Time	0.0	0.7	9.3	0.3	**1.62%**
		Obj.	11325.07	12221.51	11478.86	14230.58	14451.53

For the instances that cannot be solved within the time limit of one hour **the optimality gap** returned by CPLEX is presented instead of CPU time.

Table C.3: CPU times (in seconds) and the objective values of RA-P instances with $\tau = 3$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.2	0.2	2.6	0.1	0.2
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.0	0.3	1.2	0.1	2.1
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.0	1.8	2.7	0.0	12.2
		Obj.	10290.58	11116.88	10463.23	12728.11	13413.88
	0.4	Time	0.0	0.3	1.9	0.1	39.8
		Obj.	10615.14	11632.54	10960.25	13420.75	13936.27
	0.5	Time	0.0	1.0	79.5	0.3	11.4
		Obj.	11024.17	12107.46	11492.06	14010.79	14372.72
0.25	0.1	Time	0.0	0.2	2.1	0.0	1.0
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.0	0.2	16.5	0.1	6.6
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.0	0.7	10.3	0.2	82.7
		Obj.	10387.21	11214.21	10623.76	12864.20	13551.12
	0.4	Time	0.0	0.8	35.3	0.2	139.1
		Obj.	10771.61	11771.79	11146.25	13602.20	14125.81
	0.5	Time	0.0	0.6	15.4	0.1	73.4
		Obj.	11206.26	12293.98	11705.29	14253.58	14607.68
0.5	0.1	Time	0.0	0.1	1.5	0.0	1.2
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.0	0.2	8.9	0.3	2.4
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.0	0.5	10.0	0.1	32.4
		Obj.	10477.07	11314.53	10770.83	12998.02	13697.08
	0.4	Time	0.0	0.3	27.4	0.5	52.2
		Obj.	10920.73	11911.71	11318.25	13780.64	14310.57
	0.5	Time	0.0	0.3	2.0	0.2	53.7
		Obj.	11380.28	12482.37	11902.68	14492.92	14840.45

Table C.4: CPU times (in seconds) and the objective values of RA-N instances.

			$T=5$			$T=6$	
λ	ϵ		$K=2$	$K=3$	$K=4$	$K=2$	$K=3$
0	0.1	Time	0.0	0.1	1.7	0.1	0.8
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.0	0.5	2.7	0.1	0.6
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.0	0.3	0.8	0.0	0.6
		Obj.	10290.58	11149.09	10739.95	12728.11	13568.30
	0.4	Time	0.2	0.1	0.7	0.0	0.7
		Obj.	10728.04	11765.52	11383.94	13420.75	14426.11
	0.5	Time	0.3	0.1	0.7	0.0	0.7
		Obj.	11165.37	12381.91	12064.79	14154.40	15243.87
0.25	0.1	Time	0.0	0.1	0.8	0.1	0.7
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.0	0.3	0.8	0.2	0.8
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.0	0.2	0.8	0.3	0.8
		Obj.	10387.21	11231.59	10824.89	12864.20	13694.89
	0.4	Time	0.0	0.1	0.8	0.9	0.7
		Obj.	10856.87	11875.51	11477.91	13602.20	14594.89
	0.5	Time	0.0	0.1	0.8	0.0	0.8
		Obj.	11326.43	12519.40	12166.62	14381.19	15383.73
0.5	0.1	Time	0.0	1.0	0.8	0.1	0.7
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.0	0.1	0.8	0.3	0.7
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.0	0.5	0.7	0.0	0.8
		Obj.	10477.07	11315.67	10914.17	12998.02	13824.30
	0.4	Time	0.0	0.1	0.8	0.1	0.8
		Obj.	10976.67	11987.61	11579.24	13780.64	14767.45
	0.5	Time	0.1	0.1	0.8	0.1	0.8
		Obj.	11476.20	12659.52	12277.52	14604.21	15536.19

Table C.5: CPU times (in seconds) and the objective values of the rolling horizon algorithm with $\tau = 1$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.5	0.8	1.6	0.6	2.1
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.0	0.5	2.3	0.2	2.7
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.1	0.9	2.1	0.3	2.1
		Obj.	10290.58	11149.09	10739.37	12728.11	13568.30
	0.4	Time	0.1	0.9	2.0	0.2	2.2
		Obj.	10728.04	11765.52	11378.28	13420.75	14359.41
	0.5	Time	0.1	0.9	2.0	0.5	2.4
		Obj.	11090.09	12381.91	12057.26	14105.94	14761.46
0.25	0.1	Time	0.1	1.0	2.0	0.7	2.1
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.3	0.5	2.1	0.8	2.2
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.1	0.7	2.0	0.7	2.5
		Obj.	10387.21	11231.59	10824.34	12864.20	13694.89
	0.4	Time	0.1	0.7	2.3	0.6	3.4
		Obj.	10856.87	11875.51	11473.08	13602.20	14520.07
	0.5	Time	0.4	0.6	2.3	1.3	3.3
		Obj.	11262.48	12519.40	12158.59	14319.22	14966.23
0.5	0.1	Time	0.1	0.5	2.1	2.2	2.2
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.1	0.5	2.2	0.7	2.3
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.3	0.6	2.3	0.6	2.3
		Obj.	10477.07	11315.67	10913.68	12998.02	13824.30
	0.4	Time	0.1	0.5	2.3	0.5	3.4
		Obj.	10976.67	11987.61	11575.31	13780.64	14688.32
	0.5	Time	0.1	0.5	2.4	1.0	3.1
		Obj.	11426.08	12659.52	12269.08	14521.23	15183.45

Table C.6: CPU times (in seconds) and the objective values of the rolling horizon algorithm with $\tau = 2$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.0	0.4	1.2	0.3	1.1
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.0	0.3	1.3	0.2	1.1
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.1	0.7	1.4	0.1	2.9
		Obj.	10290.58	11149.09	10739.95	12728.11	13568.30
	0.4	Time	0.2	0.4	1.3	0.1	1.2
		Obj.	10728.04	11765.52	11383.94	13420.75	14359.41
	0.5	Time	0.1	0.5	1.3	0.1	1.1
		Obj.	11090.09	12381.91	12064.79	14105.94	14841.07
0.25	0.1	Time	0.0	1.9	1.2	0.3	1.0
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.1	0.6	1.4	0.2	1.1
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.2	0.7	1.3	0.4	1.1
		Obj.	10387.21	11231.59	10824.89	12864.20	13694.89
	0.4	Time	0.1	0.4	1.2	0.2	1.6
		Obj.	10856.87	11875.51	11477.91	13602.20	14520.07
	0.5	Time	0.0	0.5	1.2	0.4	1.6
		Obj.	11262.48	12519.40	12166.62	14319.22	15026.68
0.5	0.1	Time	0.1	0.5	1.3	0.2	0.8
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.1	0.4	1.3	0.5	1.0
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.3	0.6	1.2	0.3	1.1
		Obj.	10477.07	11315.67	10914.17	12998.02	13824.30
	0.4	Time	0.0	0.5	1.4	0.4	1.6
		Obj.	10976.67	11987.61	11579.24	13780.64	14688.32
	0.5	Time	0.0	0.5	1.3	0.7	1.5
		Obj.	11426.08	12659.52	12277.52	14521.23	15226.41

Table C.7: CPU times (in seconds) and the objective values of the rolling horizon algorithm with $\tau = 3$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.0	0.3	0.7	0.3	0.4
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.0	0.5	1.2	0.0	0.5
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.0	0.3	1.6	0.1	0.5
		Obj.	10290.58	11149.09	10739.37	12728.11	13568.30
	0.4	Time	0.1	0.2	1.3	0.1	0.5
		Obj.	10728.04	11765.52	11378.28	13420.75	14426.11
	0.5	Time	0.2	0.3	1.2	0.2	1.1
		Obj.	11165.37	12381.91	12057.26	14154.40	15243.87
0.25	0.1	Time	0.0	0.2	0.7	0.1	0.6
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.1	0.3	0.9	0.1	0.5
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.1	0.4	1.2	0.1	0.6
		Obj.	10387.21	11231.59	10824.34	12864.20	13694.89
	0.4	Time	0.1	0.5	1.1	0.1	0.7
		Obj.	10856.87	11875.51	11473.08	13602.20	14594.89
	0.5	Time	0.1	0.4	1.1	0.2	1.0
		Obj.	11326.43	12519.40	12158.59	14381.19	15383.73
0.5	0.1	Time	0.0	0.2	0.9	0.1	0.5
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.1	0.4	0.9	0.1	0.5
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.1	0.2	1.0	0.1	0.4
		Obj.	10477.07	11315.67	10913.68	12998.02	13824.30
	0.4	Time	0.0	0.2	1.2	0.1	0.6
		Obj.	10976.67	11987.61	11575.31	13780.64	14767.45
	0.5	Time	0.1	0.3	1.0	0.1	0.9
		Obj.	11476.20	12659.52	12269.08	14604.21	15536.19

Table C.8: CPU times (in seconds) and the objective values of the rolling horizon algorithm with ADR and $\tau = 1$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.3	2.8	6.8	0.9	4.8
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.4	0.9	5.0	0.9	4.9
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.4	1.3	5.3	0.8	4.2
		Obj.	10290.58	11149.09	10739.95	13717.79	13568.30
	0.4	Time	0.2	1.4	5.5	0.6	4.0
		Obj.	11732.89	11765.52	11386.30	14437.31	14426.11
	0.5	Time	0.5	0.9	5.2	0.4	6.8
		Obj.	12227.26	12381.91	12458.11	15188.14	15108.42
0.25	0.1	Time	1.9	0.8	4.9	0.5	4.6
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.3	0.9	5.6	0.6	4.6
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.3	1.0	5.1	0.5	4.2
		Obj.	10387.21	11231.59	10824.89	13847.22	13694.89
	0.4	Time	0.7	0.8	5.6	1.1	4.4
		Obj.	11875.76	11875.51	11478.89	14572.01	14594.89
	0.5	Time	0.4	0.9	5.0	0.7	6.9
		Obj.	12381.55	12519.40	12650.93	15326.88	15402.47
0.5	0.1	Time	0.5	0.8	5.2	0.4	4.4
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.3	1.0	4.9	0.5	4.5
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.3	0.9	3.3	0.5	4.5
		Obj.	10477.07	11315.67	12038.14	13984.65	13824.30
	0.4	Time	0.1	0.9	5.5	0.5	4.7
		Obj.	12023.04	11987.61	11579.25	14717.37	14767.45
	0.5	Time	0.4	0.9	5.2	0.5	6.8
		Obj.	12541.36	12659.52	12942.87	16111.47	15683.67

Table C.9: CPU times (in seconds) and the objective values of the rolling horizon algorithm with ADR and $\tau = 2$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.2	0.8	3.7	0.4	2.0
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.1	0.6	3.6	0.2	1.7
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.1	0.9	2.9	0.6	4.6
		Obj.	10290.58	11149.09	10739.95	12728.11	13568.30
	0.4	Time	0.2	0.6	3.1	0.5	2.0
		Obj.	10728.04	11765.52	11383.94	13420.75	14426.11
	0.5	Time	0.1	0.7	3.0	0.2	3.5
		Obj.	11165.37	12381.91	12462.40	14154.40	15528.24
0.25	0.1	Time	0.1	1.2	3.0	0.2	2.1
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.1	0.7	3.3	0.3	1.9
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.4	1.0	3.3	0.2	2.0
		Obj.	10387.21	11231.59	10824.89	12864.20	13694.89
	0.4	Time	0.1	0.6	3.1	0.3	1.9
		Obj.	10856.87	11875.51	11477.91	13602.20	14594.89
	0.5	Time	0.1	0.6	3.1	0.2	3.6
		Obj.	11326.43	12519.40	12654.86	14381.19	15760.20
0.5	0.1	Time	0.4	0.5	3.0	0.2	1.9
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.2	0.7	2.8	0.3	2.0
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.4	0.7	2.2	0.2	2.0
		Obj.	10477.07	11315.67	12038.14	12998.02	13824.30
	0.4	Time	0.3	0.7	2.8	0.3	2.0
		Obj.	10976.67	11987.61	11579.24	13780.64	14767.45
	0.5	Time	0.2	0.8	2.9	0.2	3.7
		Obj.	11476.20	12659.52	12942.87	16522.34	16001.52

Table C.10: CPU times (in seconds) and the objective values of the rolling horizon algorithm with ADR and $\tau = 3$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.2	0.4	1.6	0.2	0.6
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.1	0.6	2.0	0.3	0.7
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.1	0.4	2.3	0.1	0.6
		Obj.	10290.58	11149.09	10739.95	12728.11	13568.30
	0.4	Time	0.0	0.2	2.6	0.2	0.5
		Obj.	10728.04	11765.52	11386.30	13420.75	14426.11
	0.5	Time	0.1	0.3	1.7	0.1	2.9
		Obj.	11165.37	12381.91	12458.11	14154.40	15571.46
0.25	0.1	Time	0.1	0.3	1.6	0.2	0.8
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.1	0.3	1.9	0.2	0.8
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.0	0.3	1.9	0.2	0.7
		Obj.	10387.21	11231.59	10824.89	12864.20	13694.89
	0.4	Time	0.0	0.3	2.4	0.2	0.7
		Obj.	10856.87	11875.51	11478.89	13602.20	14594.89
	0.5	Time	0.0	0.4	2.2	0.1	2.0
		Obj.	11326.43	12519.40	12650.93	14381.19	15705.36
0.5	0.1	Time	0.3	0.5	1.6	0.2	0.7
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.1	0.7	1.7	0.2	0.7
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.1	0.4	0.9	0.2	0.7
		Obj.	10477.07	11315.67	12038.14	12998.02	13824.30
	0.4	Time	0.2	0.4	2.5	0.4	0.7
		Obj.	10976.67	11987.61	11579.25	13780.64	14767.45
	0.5	Time	0.2	0.2	2.1	0.3	2.1
		Obj.	11476.20	12659.52	12849.40	16522.34	15845.34

Table C.11: CPU times (in seconds) and the objective values of the rolling horizon algorithm with SADR and $\tau = 1$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.3	1.3	5.6	0.6	5.2
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.2	2.0	5.5	0.5	5.4
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.4	1.4	6.3	0.4	5.6
		Obj.	10290.58	11149.09	10739.95	13717.79	13568.30
	0.4	Time	0.1	1.3	7.4	0.5	4.9
		Obj.	11732.89	11765.52	11386.30	14437.31	14426.11
	0.5	Time	0.3	1.5	5.9	4.5	7.4
		Obj.	12227.26	12381.91	12458.11	15188.14	15108.42
0.25	0.1	Time	0.3	1.7	6.1	0.5	5.2
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.2	3.1	6.3	1.9	5.3
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.3	1.3	6.0	1.2	5.1
		Obj.	10387.21	11231.59	10824.89	13847.22	13694.89
	0.4	Time	0.1	1.2	6.9	1.7	5.5
		Obj.	11875.76	11875.51	11478.89	14572.01	14594.89
	0.5	Time	0.3	1.1	6.3	0.6	7.7
		Obj.	12381.55	12519.40	12650.93	15326.88	15402.47
0.5	0.1	Time	0.1	1.0	5.6	0.5	5.4
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.3	1.0	6.2	0.6	5.4
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.2	1.1	4.2	0.5	5.3
		Obj.	10477.07	11315.67	12038.14	13984.65	13824.30
	0.4	Time	0.3	1.0	6.7	0.5	5.2
		Obj.	12023.04	11987.61	11579.25	14717.37	14767.45
	0.5	Time	0.2	1.1	6.2	0.6	8.4
		Obj.	12541.36	12659.52	12942.87	16111.47	15683.67

Table C.12: CPU times (in seconds) and the objective values of the rolling horizon algorithm with SADR and $\tau = 2$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.5	0.9	4.1	0.9	1.9
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.1	0.7	3.8	0.1	2.5
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.1	0.9	3.7	0.1	2.1
		Obj.	10290.58	11149.09	10739.95	12728.11	13568.30
	0.4	Time	0.3	0.8	3.6	0.4	1.7
		Obj.	10728.04	11765.52	11383.94	13420.75	14426.11
	0.5	Time	0.1	1.0	3.9	0.1	3.6
		Obj.	11165.37	12381.91	12462.40	14154.40	15528.24
0.25	0.1	Time	0.0	0.8	3.6	0.4	2.2
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.3	1.0	3.4	0.2	2.1
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.2	0.9	3.6	0.4	2.0
		Obj.	10387.21	11231.59	10824.89	12864.20	13694.89
	0.4	Time	0.1	1.3	3.7	0.1	2.0
		Obj.	10856.87	11875.51	11477.91	13602.20	14594.89
	0.5	Time	0.1	0.8	3.7	0.6	3.7
		Obj.	11326.43	12519.40	12654.86	14381.19	15760.20
0.5	0.1	Time	0.2	0.7	4.0	0.3	2.0
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.1	0.7	3.5	0.3	2.2
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.1	0.8	2.8	0.3	2.1
		Obj.	10477.07	11315.67	12038.14	12998.02	13824.30
	0.4	Time	0.1	0.7	3.5	0.3	1.9
		Obj.	10976.67	11987.61	11579.24	13780.64	14767.45
	0.5	Time	0.3	0.7	3.6	0.6	3.8
		Obj.	11476.20	12659.52	12942.87	16522.34	16001.52

Table C.13: CPU times (in seconds) and the objective values of the rolling horizon algorithm with SADR and $\tau = 3$.

			$T = 5$			$T = 6$	
λ	ϵ		$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$
0	0.1	Time	0.2	0.4	2.1	0.2	0.6
		Obj.	9415.86	9916.17	9730.32	11343.05	11852.98
	0.2	Time	0.1	0.2	2.3	0.1	0.7
		Obj.	9853.30	10532.76	10188.14	12035.57	12710.73
	0.3	Time	0.0	0.3	2.6	0.1	0.6
		Obj.	10290.58	11149.09	10739.95	12728.11	13568.30
	0.4	Time	0.1	0.2	2.8	0.1	0.7
		Obj.	10728.04	11765.52	11386.30	13420.75	14426.11
	0.5	Time	0.1	0.3	2.0	0.2	2.7
		Obj.	11165.37	12381.91	12458.11	14154.40	15571.46
0.25	0.1	Time	0.1	0.4	1.7	0.1	0.9
		Obj.	9448.08	9943.67	9773.42	11388.43	11895.18
	0.2	Time	0.1	0.2	2.2	0.1	0.9
		Obj.	9917.74	10587.75	10255.84	12126.30	12795.12
	0.3	Time	0.1	0.3	2.1	0.3	0.8
		Obj.	10387.21	11231.59	10824.89	12864.20	13694.89
	0.4	Time	0.0	0.4	2.5	0.1	0.8
		Obj.	10856.87	11875.51	11478.89	13602.20	14594.89
	0.5	Time	0.1	0.3	2.0	0.0	1.9
		Obj.	11326.43	12519.40	12650.93	14381.19	15705.36
0.5	0.1	Time	0.4	0.5	1.6	0.1	0.7
		Obj.	9478.04	9971.71	9815.76	11433.06	11938.33
	0.2	Time	0.2	0.5	2.0	0.2	0.8
		Obj.	9977.66	10643.80	10326.57	12215.52	12881.40
	0.3	Time	0.0	0.3	1.0	0.1	0.8
		Obj.	10477.07	11315.67	12038.14	12998.02	13824.30
	0.4	Time	0.1	0.2	2.6	0.1	0.7
		Obj.	10976.67	11987.61	11579.25	13780.64	14767.45
	0.5	Time	0.1	0.4	2.1	0.1	2.1
		Obj.	11476.20	12659.52	12849.40	16522.34	15845.34