**Solution methods and bounds for two-stage risk-neutral and multistage risk-averse stochastic mixed-integer programs with applications in energy and manufacturing**

by

**Ge Guo**

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering
Minor: Statistics

Program of Study Committee:
Sarah M. Ryan, Major Professor
Guiping Hu
Lizhi Wang
Farzad Sabzikar
James D. McCalley

Iowa State University

Ames, Iowa

2018

## DEDICATIONS

In dedication to my beloved parents, husband, and my son coming on the way.

# TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

## ACKNOWLEDGMENTS

I would like to take this opportunity to express my gratitude to those who have helped me throughout my research and the writing of this dissertation and made my five-year Ph.D. program an unforgettable experience.

I would like to thank my committee chair, Dr. Sarah Ryan, for her constant guidance throughout my Ph.D. study. Her conscientious scholarship and academic integrity has set a role model for me and has inspired me to pursue my career in academia. Her understanding and support in both academic and personal lives has made my years at ISU an enjoyable experience.

I would like to express my very sincere gratitude to my committee members, Dr. Guiping Hu, Dr. Lizhi Wang, Dr. Farzad Sabsikar, Dr. James D. McCalley, and Dr. Mingyi Hong for their great efforts and valuable feedbacks on my research.

I would like to offer my sincere appreciation to my co-authors, Dr. David Woodruff at University of California Davis, Dr. Jean-Paul Watson and Dr. Gabriel Hackebeil at Sandia National Laboratories for their support in the software implementation with Pyomo and their assistance on the computational experiments.

I would also like to thank my colleagues, my instructors, department faculty and staff for making my time at Iowa State University a wonderful experience.

## ABSTRACT

This dissertation presents an integrated method for solving stochastic mixed-integer programs, develops a lower bounding approach for multistage risk-averse stochastic mixed-integer programs, and proposes an optimization formulation for mixed-model assembly line sequencing (MMALS) problems.

It is well known that a stochastic mixed-integer program is difficult to solve due to its non-convexity and stochastic factors. The scenario decomposition algorithms display computational advantage when dealing with a large number of possible realizations of uncertainties, but each has its own advantages and disadvantages. This dissertation presents a solution method for solving large-scale stochastic mixed-integer programs that integrates two scenario-decomposition algorithms: Progressive Hedging (PH) and Dual Decomposition (DD). In this integrated method, fast progress in early iterations of PH speeds up the convergence of DD to an exact solution.

In many applications, the decision makers are risk-averse and are more concerned with large losses in the worst scenarios than with average performance. The PH algorithm can serve as a time-efficient heuristic for risk-averse stochastic mixed-integer programs with many scenarios, but the scenario reformulation for time consistent multistage risk-averse models does not exist. This dissertation develops a scenario-decomposed version of time consistent multistage risk-averse programs, and proposes a lower bounding approach that can assess the quality of PH solutions and thus identify whether the PH algorithm is able to find near-optimal solutions within a reasonable amount of time.

The existing optimization formulations for MMALS problems do not consider many real-world uncertainty factors such as timely part delivery and material quality. In addition, real-time

sequencing decisions are required to deal with inevitable disruptions. This dissertation formulates a multistage stochastic optimization problem with part availability uncertainty. A risk-averse model is further developed to guarantee customers' satisfaction regarding on-time performance.

Computational studies show that the integration of PH helps DD to reduce the run-time significantly, and the lower bounding approach can obtain convergent and tight lower bounds to help PH evaluate quality of solutions. The PH algorithm and the lower bounding approach also help the proposed MMALS formulation to make real-time sequencing decisions.

# CHAPTER 1. INTRODUCTION

In many applications where decisions are made without full information, stochastic programs are introduced to hedge against consequences of the possible realizations of random parameters so that the expected value of the objective is the best possible. Unlike deterministic programming, stochastic programming describes the uncertain parameters with probability distributions instead of specific values. In two-stage stochastic programs, two categories of decisions are considered. First-stage decisions have to be taken without full information on some random events while recourse actions are taken in response to a particular realization in the second stage. In multistage stochastic programs, the decisions made at each stage can only depend on information revealed before that stage but not after.

Stochastic integer programs are formulated when some of the decisions are required to be integers. For example, unit commitment problems make on and off decisions for thermal electric power generators to minimize the cost of serving unpredictable electricity load; server location problems decide on the optimal server locations to minimize the cost of serving uncertain potential clients; lot sizing problems determine a minimum cost setup, production and inventory schedule to satisfy a stochastic demand over time; and assembly line model sequencing problems yield an optimal production sequence of models subject to unreliable part delivery and quality.

The combination of stochastic parameters and discrete decisions leads to great difficulty in solving stochastic mixed-integer programs. Integer programming is NP-complete. Integer variables make an optimization problem non-convex which is far more difficult to solve than a convex optimization problem is. The solution time grows exponentially in the instance size. When many discrete realizations of the uncertain parameters are possible, a stochastic mixed-

integer program may be formulated as the extensive form of the deterministic equivalent, which is a very large mixed integer program, and therefore even more difficult to solve.

Due to the large scale of extensive forms of stochastic integer programs, decomposition algorithms are widely applied to speed up computation. Generally, the decomposition methods for stochastic integer programs fall into two groups: stage-based methods and scenario-based methods. The exemplary stage-based decomposition method, the L-shaped method, or Benders decomposition [1], is limited to instances with binary variables only in the first stage and easily computable recourse costs. However, the size of the master problem, which is solved for first-stage decisions, keeps increasing as Benders cuts pertaining to later stages are added in each iteration. In contrast, scenario-based decomposition methods display advantages in solving the large-scale instances when many realizations are present. Paradigms of scenario-based decomposition include the Dual Decomposition (DD) algorithm for two-stage stochastic integer programs [2] and the Progressive Hedging (PH) algorithm [3]. The PH algorithm has been proven to converge when all decision variables are continuous and can serve as a heuristic in the mixed-integer case. Computational studies have shown that it can find high-quality solutions within a reasonable number of iterations. To assess the quality of the solutions generated by PH relative to the optimal solution, Gade et al. [4] presented a lower bounding technique for the PH algorithm and showed that for convex problems, the lower bound obtained by PH is as tight as the lower bound from the Lagrangian dual. Furthermore, the PH algorithm can easily extend to multistage formulations with integers in any stage.

Traditional stochastic programming is risk-neutral and optimizes the expected performance across all scenarios. The optimal solutions to the risk-neutral models perform well in the long run over repeated instances. But for non-repetitive decision-making problems under

uncertainty, the risk-neutral solutions may perform poorly under certain realizations of the uncertain parameters. Therefore, risk-averse models have recently attracted attention in stochastic programming for decision makers who are more concerned with the large losses in the worst scenarios rather than average performance.

Among a selection of risk measures to include in risk-averse programs, some coherent measures possess suitable mathematical properties to construct efficient algorithms. Two broad categories of coherent risk terms are quantile and deviation based risk measures [5]. Quantile risk-measures are based on the quantiles of the probability distributions of the random objectives. Types of quantile based risk-measures include Excess Probability (EP), which measures the probability of exceeding a prescribed target level; and Conditional Value-at-Risk (CVaR), which measures the expectation of the $(1-\alpha)\%$ worst outcomes for a given probability $\alpha$. Deviation risk measures are based on the deviation of the expected value from a prescribed target, and they include Expected Excess (EE), which measures the expected value of the excess over a given target; and Semi-Deviation (SD), which measures the expected value of the excess over the mean.

When risk measures are extended to multiple stages, however, there is no natural way of measuring risk and one key issue of time consistency arises. Generally speaking, re-solving the problem at later stages yields the same optimal objective given the solutions from previous stages if an optimization problem is time consistent. In general, the property of time consistency does not hold for multistage risk-averse models and depends on how the risk measures are computed. Homem-de-Mello and Pagnoncelli [23] proposed a class of expected conditional risk measures (ECRMs) which prove to be time consistent and, for some risk measures, allow for risk-neutral reformulations. The scenario-decomposed reformulations with various risk measures

allow for the employment of scenario decomposition solution algorithms, such as the PH algorithm, to efficiently solve multistage risk-averse problems. However, approaches to assess PH solution quality for risk-averse models have not been explored. An approach to obtain lower bounds from the PH algorithm, if found, not only can evaluate the PH solution quality, but also can integrate the PH algorithm with exact algorithms that rely on lower bounds such as the DD algorithm for two-stage problems. Most importantly, the lower bounding approach can help the PH algorithm to find near-optimal solutions within a reasonable amount of computational time.

A novel application of multistage stochastic mixed-integer programs arises in shop floor control. Mixed-model assembly line manufacturing systems have become popular in recent years as an important part of the just-in-time production system, in which several models of the same basic product are manufactured on the same production line. The optimal design and operation of mixed-model assembly lines must address a long-term assembly line balancing problem to assign tasks to stations and a short-term model sequencing problem to determine the production sequence of a given set of models within the planning horizon.

A number of optimization models have been proposed for the model sequencing problem in mixed-model assembly line manufacturing system. Some formulations also considered uncertainty factors such as demand from customers and operation times. In the real world, however, mixed-model assembly lines are faced with more challenging uncertainties including timely part delivery, material quality, upstream sub-assembly completion and availability of other resources. In addition, assembly lines must meet deadlines imposed by customers or downstream stations. All those uncertainties play an important role in making sequencing decisions, but are not yet addressed in existing formulations.

In just-in-time manufacturing systems such as mixed-model assembly lines, the schedulers are most concerned with the on-time performance in worst scenarios rather than average performance when large losses are resulted from unreliable part availability. There are two open issues in this context. First, the objective of on-time performance has not been explored in existing optimization models. Second, risk-neutral models yield the optimal performance across all the scenarios while the optimization of performance in worst scenarios requires risk-averse models, which have not been explored either.

The formulation of optimization models to make sequencing decisions was motivated by a project aimed at developing a shop floor decision support system in collaboration with industrial partners. The optimization model could serve as a decision-support tool in the sequencing module of this system. Real-time resequencing decisions are required in this project to avoid wastage in time and costs of downtime caused by inevitable disruptions. Therefore, a time efficient solution algorithm is desired to find near-optimal solutions within a reasonable amount of time.

### 1.1 Problem Statement

This dissertation addresses the following interrelated questions.

First, the PH algorithm and the DD algorithm are two paradigms of scenario-based decomposition solution algorithms for two-stage stochastic programs and have both displayed computational advantages for solving stochastic mixed-integer programs with a large number of scenarios. However, both of these two solution algorithms have their own deficiencies. Although the PH algorithm can find high-quality solutions within a reasonable number of iterations, the solutions are not guaranteed to converge to global optimality in the case of mixed-integer problems. The DD algorithm, on the other hand, can achieve global convergence but the branch and bound process may be slow. Therefore, a question is brought up: can we combine the

advantages of the PH and DD algorithms by proposing an integrated approach? Given the PH lower bounding approach presented by Gade et a. [4] and the fact that for convex problems, the lower bound from PH is as tight as the lower bound from the Lagrangian dual, we are seeking a way to exchange the information of solutions and objectives between the PH algorithm and the DD algorithm through their lower bounds.

Second, the classical formulation of stochastic programs assumes that the decision maker is risk-neutral such that he or she will not mind large losses in some scenarios as long as those are offset by large gains in other scenarios. This formulation, however, does not reflect the situation where the decision maker is more concerned about large losses, that is, the decision maker is risk-averse. It is natural to consider risk-averse formulations of stochastic programs. Time consistency is an important issue in modeling multistage risk-averse models. In general, time consistency for multistage risk-averse stochastic programs is not guaranteed. Therefore, we are seeking a time consistent scenario-decomposed version of reformulations for multistage risk-averse models that allow for the application of scenario decomposition solution algorithms such as the PH algorithm. Essentially, we want to propose a lower bounding approach from the PH algorithm for multistage risk-averse programs to assess the PH solution quality and thus identify whether the PH algorithm is able to find near-optimal solutions within a reasonable amount of computational time.

Third, as the most important short-term problem in mixed-model assembly line manufacturing system, the model sequencing problem has been studied extensively as seen in a variety of optimization formulations to determine the optimal production sequence. However, the existing formulations are incomplete and do not model many real-world uncertainties such as timely part delivery and material quality. To incorporate those uncertainties in our decision-

making process, we would like to propose an optimization model with part availability modeled as a stochastic process. In addition, in just-in-time manufacturing systems such as mixed-model assembly lines, the schedulers tend to worry more about the on-time performance in worst scenarios. For such decision makers concerned with large losses in worst scenarios, a risk-averse model is preferred. The critical question is, which risk measure should be selected to represent the on-time performance in worst scenarios? Furthermore, can this risk-averse model provide us with near-optimal sequencing decisions in real time to decrease the downtime of assembly lines?

## 1.2 Literature Review

### 1.2.1 Solution Algorithms for Stochastic Integer Programs

The presence of discrete decision variables leads to great difficulty in solving stochastic integer programs due to the NP-hard nature of integer programming. Until now much progress has been made in developing algorithms, extending from special instances to more general stochastic mixed-integer programs. Stage-based decomposition algorithms were studied for various classes of two-stage stochastic integer programs. Laporte and Louveaux [1] proposed integer L-shaped methods for stochastic integer programs so long as the first-stage decisions are binary. Sen and Sherali [6] presented branch-and-cut approaches for two-stage stochastic integer programs. Ahmed [7] introduced a branch-and-bound algorithm for two-stage stochastic integer programs. As for multi-stage stochastic integer programs, several scenario decomposition methods were proposed. Lulli and Sen [8] presented a branch-and-price (B&P) algorithm for multi-stage stochastic integer programs. Carøe and Schultz [2] developed a dual decomposition (DD) algorithm based on scenario decomposition and Lagrangian relaxation. Lubin et al. [9] demonstrated the potential for parallel speedup by addressing the bottleneck of parallelizing dual decomposition. Originally proposed by Rockafellar and Wets [3] for stochastic programs with only continuous variables, progressive hedging (PH) has been successfully applied by Listes and

Dekker [10], Fan and Liu [11], Watson and Woodruff [12], and many others as a heuristic to solve stochastic mixed-integer programs. To assess the quality of the solutions generated by PH relative to the optimal solution, Gade et al. [4] presented a lower bounding technique for the PH algorithm and showed that for convex two-stage problems, the lower bound obtained by PH is as tight as the lower bound obtained from the Lagrangian dual.

### 1.2.2 Risk-averse Stochastic Programs

Extensive studies have been performed in the formulation and solution algorithms for two-stage risk-averse models. Schultz and Tiedemann [13] presented a mixed-integer linear programming formulation of a mean-risk model involving CVaR as risk measure in the framework of two-stage stochastic mixed-integer programming. Fábián [14] proposed decomposition frameworks for handing CVaR objectives and constraints in two-stage stochastic models. Miller and Ruszczynski [15] developed a nested formulation of a risk-averse two-stage program and presented a risk-averse multicut decomposition method. Noyan [16] developed decomposition algorithms for a risk-averse two-stage stochastic programming model with CVaR as the risk measure. Venkatachalam and Ntaimo [17] presented a stage-wise decomposition method for stochastic programs with binary variables in the second-stage with absolute semi-deviation risk measure.

For multi-stage stochastic programs, however, there is no obvious way of measuring risk. The difficulty in extending risk measures to the multistage setting has been discussed in several papers. Some papers discuss how to adapt existing algorithms from the risk-neutral case to the risk-averse case, often with the CVaR as the risk measure. Collado and Papp [18] introduced a partial bundle method for risk-averse multistage stochastic optimization. Eichhorn and Römisch [19] defined a class of polyhedral risk measures with favorable properties and proposed multiperiod extensions of the CVaR as polyhedral risk measures for multistage stochastic

programs. Guigues and Sagastizábal [20] proposed a risk-averse rolling-horizon time consistent approach and showed the risk-averse formulations of stochastic linear programs are numerically tractable. Kozmik and Morton [21] proposed a new approach of upper bound estimator for minimization problems in risk-averse multi-stage stochastic programs using CVaR as risk measure. Pflug and Pichler [22] proposed a time consistent formulation of multi-stage stochastic program with CVaR and presented a stage-wise dynamic decomposition.

It has been observed that one very important issue in modeling risk-averse multi-stage stochastic programs is that of time consistency. It is a desirable property for multi-stage stochastic programs. Shapiro [23] defines a problem to be time consistent if the solution at a node in the scenario tree does not depend on children of other nodes. Carpentier et al. [24] formulated the property of time consistency such that the optimal strategies obtained when solving the original problem remain optimal for all subsequent-stage problems. Pflug and Pichler [25] claim a multistage stochastic program to be time consistent if, when resolving the problem at later stages, the original solutions remain optimal for those stages. Homem-de-Mello and Pagnoncelli [26, p.189] define time consistency informally as “if you solve a multi-stage stochastic program and find solutions today, you should find the same solutions if you re-solve the problem tomorrow given what was observed and decided today”.

Given that the property of time consistency is not guaranteed for multistage risk-averse models, significant efforts are initiated to find time consistent risk measures for multistage stochastic programs. Ruszczynski and Shapiro [27] proposed a nested conditional risk measure for multistage optimization problems which proves to be time consistent. The nested conditional risk measure is formulated in a recursive function which is not given in explicit form. Homem-de-Mello and Pagnoncelli [26] addressed this drawback by proposing a class of expected

conditional risk measures which prove to be time consistent and can lead to a risk-neutral reformulation.

### 1.2.3 Mixed-Model Assembly Line Sequencing Problems

Modeling of sequencing problems has been studied in recent papers with the development of mixed-model assembly line manufacturing systems. Rahimi-Vahed [28] considered three objectives to be minimized in a mixed-model assembly line sequencing problem: total utility work, total production rate variation, and total setup cost. Rabbani et al. [29] developed a bi-objective optimization model to find the optimal sequence of products to minimize the total cost as well as to maximize levels of customer satisfaction. In addition, some studies are performed on the optimization modeling with a variety of uncertainties. Boysen [30] discussed three major sequencing approaches including mixed-model sequencing, car sequencing and level scheduling considering two major uncertainties of stochastic demand and task times. Zhao [31] formulated an optimization problem of daily scheduling to minimize the expected system cost including the inventory cost of holding products and the penalty cost of backorders with stochastic demand. Dong [32] proposed a stochastic programming formulation to minimize the expected work overload time for a mixed-model assembly U-lines with stochastic task times. As the problem is NP-hard, a simulated annealing algorithm is proposed to solve this problem.

## 1.3 Research Gap

Based on the current literature review, the existing gaps studied in this research are as follows:

(1) Given the advantages of two scenario decomposition solutions algorithms, there did not exist an approach to combine the computational efficiency of the PH algorithm and the global optimality of the DD algorithm. Now that the lower bound from PH is as tight as the lower bound from the Lagrangian dual for convex problems, it is yet unrevealed how the

information can be exchanged between the PH algorithm and the DD algorithm through their lower bounds. This gap has been filled in our research work [33].

(2) It is natural to consider risk-averse models for decision makers who are more concerned with large losses in worst scenarios. For multistage risk-averse models, the property of time consistency is desired. There did not exist scenario reformulations for time consistent multistage risk-averse stochastic programs to allow for the application of scenario decomposition solution algorithms such as the PH algorithm. Furthermore, a lower bounding approach is not available for multistage risk-averse models to help the PH algorithm to find near-optimal solutions within a reasonable amount of time. This gap has been filled in our research work [34].

(3) The existing formulations for model sequencing problems in mixed-model assembly line manufacturing systems are incomplete and do not model some important real-world uncertainties such as timely part delivery and material quality. In addition, the schedulers tend to be more concerned with on-time performance in worst scenarios. Thus, a risk-averse model with part availability uncertainty is to be proposed to find optimal sequencing decisions. Besides, a time efficient solution algorithm is to be identified to find near-optimal solutions in real time for a just-in-time manufacturing system. This gap has been filled in our research work [35].

### 1.4 Outline of Dissertation

Chapter 2 presents a method for integrating PH and DD algorithm for solving stochastic integer programs based on the correspondence between lower bounds obtained with PH and DD algorithm [33]. In chapter 3, we propose a scenario-decomposed version of risk-neutral reformulation for time consistent multistage risk-averse models, and present an approach to obtain convergent and tight lower bounds from the PH algorithm for time consistent multistage risk-averse models [34]. A multistage stochastic model for mixed-model assembly line sequencing problem is formulated in chapter 4 to increase on-time performance with part

availability uncertainty. The lower bounding approach from [34] is applied to its risk-averse version as the solution algorithm in the context of real-time resequencing [35]. Chapter 5 concludes this dissertation with its contributions, limitations, and future studies.

## REFERENCES

[1] G. Laporte, F.V. Louveaux. (1993) The integer L-shaped method for stochastic integer programs with complete recourse, Operations Research Letters, 13, 133-142

[2] C.C. Carøe, R. Schultz. (1999) Dual decomposition in stochastic integer programming, Operations Research Letters, 24 (1-2), 37-45

[3] R.T. Rockafellar, R.J.-B. Wets. (1991) Scenarios and policy aggregation in optimization under uncertainty, Mathematics of Operations Research, 16 (1), 119-147

[4] D. Gade, G. Hackebeil, S.M. Ryan, J.P. Watson, R.J. Wets, D.L. Woodruff. (2016) Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs, Mathematical Programming, 157 (1), 47-67

[5] R. Schultz. (2010) Risk aversion in two-stage stochastic integer programming, International Series in Operations Research & Management Science, 150, 165-187

[6] S. Sen, H.D. Sherali. (2002) Decomposition with branch-and-cut approaches for two-stage stochastic integer programming, Mathematical Programming, 106 (2), 203-223

[7] S. Ahmed, M. Tawarmalani, N.V. Sahinidis. (2004) A finite branch and bound algorithm for two-stage stochastic integer programs, Mathematical Programming, 100, 355-377

[8] G. Lulli, S. Sen. (2002) A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems, Management Science, 50 (6), 786-796

[9] M. Lubin, K. Martin, C. Petra, B. Sandikçi. (2013) On parallezing dual decomposition in stochastic integer programming, Operations Research Letters, 41 (3), 252-258

[10] O. Listes, R. Dekker. (2005) A scenario aggregation based approach for determining a robust airline fleet composition, Transportation Science, 39 (3), 367–382

[11] Y. Fan, C. Liu. (2010) Solving stochastic transportation network protection problem using the progressive hedging-based method, Networks and Spatial Economics, 10 (2), 193–208

[12] J.-P. Watson, D.L. Woodruff. (2011) Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems, Computational Management Science, 8 (4), 355-370

[13] R. Schultz, S. Tiedemann. (2006) Conditional value-at-risk in stochastic programs with mixed-integer recourse, Mathematical Programming, 105 (2), 365-386

[14] C.I. Fábián. (2008) Handling CVaR objectives and constraints in two-stage stochastic models. European Journal of Operational Research, 191 (3), 888–911

[15] N. Miller, A. Ruszczynski. (2011) Risk-averse two-stage stochastic linear programming: modeling and decomposition, Operations Research, 59 (1), 125-132

[16] N. Noyan. (2012) Risk-averse two-stage stochastic programming with an application to disaster management, Computer and Operations Research, 39 (3), 541-559

[17] S. Venkatachalam, L. Ntaimo. (2016) Absolute semi-deviation risk measure for ordering problem with transportation cost in supply chain, arXiv preprint arXiv:1605.08391

[18] R.A. Collado, D. Papp. (2011) Scenario decomposition of risk-averse multistage stochastic programming problems, Annals of Operations Research, 200 (1), 147-170

[19] A. Eichhorn, W. Römisch. (2005) Polyhedral risk measures in stochastic programming. SIAM Journal on Optimization, 16 (1), 69-95

[20] V. Guigues, C. Sagastizábal. (2013) Risk-averse feasible policies for large-scale multistage stochastic linear programs, Mathematical Programming, 138 (1-2), 167-198

[21] V. Kozmík, D.P. Morton. (2015) Evaluating policies in risk-averse multi-stage stochastic programming, Mathematical Programming, 152 (1-2), 275-300

[22] G.C. Pflug, A. Pichler. (2016) Time-inconsistent multistage stochastic programs: Martingale bounds, European Journal of Operations Research, 249 (1), 155-163

[23] A. Shapiro. (2009) On a time consistency concept in risk averse multistage stochastic programming, Operations Research Letters, 37 (3), 143-147

[24] P. Carpentier, J.P. Chancelier, G. Cohen, M.D. Lara, & P. Girardeau. (2012) Dynamic consistency for stochastic optimal control problems. Annals of Operations Research, 200 (1), 247-263

[25] G.C. Pflug, & A. Pichler. (2016) Time-inconsistent multistage stochastic programs: martingale bounds. European Journal of Operational Research, 249 (1), 155-163

[26] T. Homem-de-Mello, B.K. Pagnoncelli. (2016) Risk aversion in multistage stochastic programming: A modeling and algorithmic perspective, 249, 188-199

[27] A. Ruszczynski, & A. Shapiro. (2006) Conditional risk mappings. Mathematics of Operations Research, 31 (3), 544-561

[28] A. Rahimi-Vahed. (2007) A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem, Computers and Industrial Engineering, 53 (4), 642-666

[29] M. Rabbani, R. Heidari, and H. Farrokhi. (2018) A bi-objective mixed-model assembly line sequencing problem considering customer satisfaction and customer buying behavior, Engineering Optimization, 1-20

[30] N. Boysen, M. Fliedner, A. Scholl. (2009) Sequencing mixed-model assembly lines: Survey, classification and model critique, European Journal of Operations Research, 192, 349-373

[31] X. Zhao. (2006) Sequence-to-customer goal with stochastic demands for a mixed-model assembly line, International Journal of Production Research, 44 (24), 5279-5305

[32] J. Dong. (2014) Balancing and sequencing of stochastic mixed-model assembly U-lines to minimize the expectation of work overload time, International Journal of Production Research, 52 (24), 7529-7548

[33] G. Guo, G. Hackebeil, S.M. Ryan, J.-P. Watson, D.L. Woodruff. (2015) Integration of progressive hedging and dual decomposition in stochastic integer programs, Operations Research Letters, 43, 311-316

[34] G. Guo, S.M. Ryan. (2017) Progressive hedging lower bounds for time consistent risk-averse multistage stochastic mixed-integer programs, Available at: http://works.bepress.com/sarah_m_ryan/93/

[35] G. Guo, S.M. Ryan. (2017) Risk-averse stochastic integer programs for mixed-model assembly line sequencing problems. Industrial and Manufacturing Systems Engineering Annual Conference Proceedings, 98.

# CHAPTER 2. INTEGRATION OF PROGRESSIVE HEDGING AND DUAL DECOMPOSITION IN STOCHASTIC INTEGER PROGRAMS

A paper published in Operations Research Letters

**Abstract**

We present a method for integrating the Progressive Hedging (PH) algorithm and the Dual Decomposition (DD) algorithm of Carøe and Schultz for stochastic mixed-integer programs. Based on the correspondence between lower bounds obtained with PH and DD, a method to transform weights from PH to Lagrange multipliers in DD is found. Fast progress in early iterations of PH speeds up convergence of DD to an exact solution. We report computational results on server location and unit commitment instances.

*Keywords:* Stochastic programming; Mixed-integer programming; Progressive hedging; Dual decomposition; Lower bounding

## 2.1 Introduction

Stochastic mixed-integer programs find a broad application in energy, facility location, production scheduling and other areas where a set of decisions must be taken before full information is revealed on some random events and some of the decisions are required to be integer [1]. The combination of uncertainty and discrete decisions leads to the difficulty in solving stochastic mixed-integer programs.

Until now much progress has been made in developing algorithms to solve these problems, extending from special instances [12, 13, 23] to more general stochastic mixed-integer programs [2, 20]. Carøe and Schultz [3] developed a dual decomposition (DD) algorithm based on scenario decomposition and Lagrangian relaxation. Lubin et al. [14] demonstrated the potential for parallel speedup by addressing the bottleneck of parallelizing dual decomposition. Originally proposed by Rockafellar and Wets [19] for stochastic programs with only continuous

variables, progressive hedging (PH) has been successfully applied by Listes and Dekker [17], Fan and Liu [6], Watson and Woodruff [25], and many others as a heuristic to solve stochastic mixed-integer programs. To assess the quality of the solutions generated by PH relative to the optimal solution, Gade et al. [8] presented a lower bounding technique for the PH algorithm and showed that the best possible lower bound obtained from PH is as tight as the lower bound obtained using DD.

The PH algorithm can find high-quality solutions within a reasonable number of iterations, but is not guaranteed to converge to a globally optimal solution in the case of mixed-integer problems. The DD algorithm, on the other hand, will achieve convergence combined with branch and bound but may be slow. This paper combines advantages of both scenario decomposition methods. By transforming PH weights into Lagrangian multipliers as a starting point for DD, the convergence of DD can be sped up considerably.

The remainder of this paper is organized as follows. In Section 2 we describe the PH and DD algorithms, two scenario-based decomposition algorithms for stochastic mixed-integer programs. Our integration approach to transfer information from PH to DD is developed in Section 3. In Section 4, we document the implementation of our integration method and in Section 5, provide experimental results on a set of stochastic server location instances and two stochastic unit commitment instances.

### 2.2 Scenario Decomposition Algorithms for Stochastic Mixed Integer Programs

Decomposition methods for stochastic programs generally fall into two groups: stage-based methods and scenario-based methods [18]. The exemplary stage-based decomposition method is the L-shaped method, or Benders decomposition [21]. Paradigms of scenario-based decomposition include the PH algorithm [19] and the DD algorithm [3]. One advantage of scenario-based decomposition methods over the stage-based ones is their mitigation of the

computational difficulty associated with large problem instances by decomposing the problem by scenario and solving the subproblems in parallel. In practical applications, PH can easily be implemented as a "wrapper" for existing software for large-scale implementation of the deterministic scenario problems. In this section, we will discuss these two scenario-based decomposition methods for stochastic mixed-integer programs in detail.

### 2.2.1 Two-Stage Stochastic Mixed-Integer Program

We consider the following two-stage stochastic mixed-integer program:

$$z = \min\{cx + Q(x) : Ax = b, x \in X\}, \qquad (1)$$

where $Q(x) = \mathbb{E}_{\xi}\phi(x,\xi)$ and $\phi(x,\xi) = \min\{q(\xi)y : Wy = h(\xi) - T(\xi)x, y \in Y\}$. Here $c^T \in \mathbb{R}^{n_1}$ and $b \in \mathbb{R}^{m_1}$ are known vectors, while $A \in \mathbb{R}^{m_1 \times n_1}$ and $W \in \mathbb{R}^{m_2 \times n_2}$ are known matrices. The vector $\xi$ is a random variable defined on some probability space $(\Xi, \Im, P)$ and for each $\xi \in \Xi$, the vectors $q(\xi)^T \in \mathbb{R}^{n_2}$ and $h(\xi) \in \mathbb{R}^{m_2}$ and the matrix $T(\xi) \in \mathbb{R}^{m_2 \times n_1}$. The sets $X \subseteq \mathbb{R}^{n_1}_{+}$ and $Y \subseteq \mathbb{R}^{n_2}_{+}$ denote the mixed-integer requirements on the first-stage and second-stage variables. The decisions are two-stage in the sense that first-stage decisions $x$ have to be taken without full information on some random events while second-stage decisions $y$ are taken after full information is received on the realization of the random vector $\xi$. The notation $\mathbb{E}_{\xi}$ denotes expectation with respect to the distribution of $\xi$.

To avoid complications when computing the integral behind $\mathbb{E}_{\xi}$ we assume that we have only a finite number of realizations of $\xi$, known as scenarios $\xi^j$, $j = 1,...,r$, with corresponding probabilities $p^j$. Then problem (1) can be written as a large-scale deterministic mixed-integer linear program with a block-angular structure called the extensive form of the deterministic equivalent:

$$z = \min\left\{ cx + \sum_{j=1}^{r} p^j q^j y^j : (x, y^j) \in S^j,\ j = 1,...,r \right\}, \quad (2)$$

where $S^j = \left\{ (x, y^j) : Ax = b, x \in X, Wy^j = h^j - T^j x, y^j \in Y \right\}$.

The block-angular structure of Eq. (2) enables the decomposition methods to split it into scenario subproblems by introducing copies of the first-stage variables. This idea leads to the so-called scenario formulation of the stochastic program:

$$z = \min\left\{ \sum_{j=1}^{r} p^j (cx^j + q^j y^j) : (x^j, y^j) \in S^j,\ j = 1,...,r, x^1 = ... = x^r \right\}. \quad (3)$$

The subproblems are coupled by the non-anticipativity constraints, $x^1 = ... = x^r$, which force the first-stage decisions to be scenario-independent.

**2.2.2 Dual Decomposition**

The dual decomposition (DD) algorithm of Carøe and Schultz relaxes the non-anticipativity constraints and uses branch and bound to restore non-anticipativity. DD obtains lower bounds on the optimal value of problem (3) by solving the Lagrangian dual obtained by relaxing the non-anticipativity constraints.

The non-anticipativity requirement of problem (3) can be expressed by several equivalent representations. Lulli and Sen [15] as well as Lubin and Martin [14] introduce an additional variable $x.$ and model non-anticipativity as

$$x_j - x. = 0,\ j = 1,...r, \quad (4)$$

while Carøe and Schultz represent non-anticipativity by

$$\sum_{j=1}^{r} H^j x^j = 0, \quad (5)$$

where the matrix $H^j \in \mathbb{R}^{n_1(r-1)\times n_1}$.

Using non-anticipativity representation (4), the Lagrangian relaxation of non-anticipativity constraints may be written as

$$P(\lambda) = \min\left\{\sum_{j=1}^{r}[R_j(x^j, y^j, \mu^j) - \mu^j x.] : (x^j, y^j) \in S^j\right\}, \qquad (6)$$

where $R_j(x^j, y^j, \mu^j) = p^j(cx^j + q^j y^j) + \mu^j x^j$ for $j = 1,...,r$ and the parameter $(\mu^j)^T \in \mathbb{R}^{n_1}$. The Lagrangian (6) is separable into $P(\mu^1,...,\mu^r) = \sum_{j=1}^{r} P_j(\mu^j)$, where

$$P_j(\mu^j) = \min\left\{R_j(x^j, y^j, \mu^j) : (x^j, y^j) \in S^j\right\}, \qquad (7)$$

with the condition $\sum_{j=1}^{r} \mu^j = 0$ required for boundedness of the Lagrangian. The Lagrangian dual is expressed as

$$c_{LD} = \max_{\mu^1,...,\mu^r}\left\{P(\mu^1,...,\mu^r) : \sum_{j=1}^{r} \mu^j = 0\right\}. \qquad (8)$$

The non-anticipativity representation (5), on the other hand, leads to the Lagrangian relaxation in the form

$$D(\lambda) = \min\left\{\sum_{j=1}^{r} L_j(x^j, y^j, \lambda) : (x^j, y^j) \in S^j\right\}, \qquad (9)$$

where $L_j(x^j, y^j, \lambda) = p^j(cx^j + q^j y^j) + \lambda(H^j x^j)$ for $j = 1,...,r$, where the vector $\lambda = (\lambda^1,...,\lambda^{r-1})$ and the vector $(\lambda^j)^T \in \mathbb{R}^{n_1}$. The Lagrangian (9) is separable into $D(\lambda) = \sum_{j=1}^{r} D_j(\lambda)$, where

$$D_j(\lambda) = \min\left\{L_j(x^j, y^j, \lambda) : (x^j, y^j) \in S^j\right\}. \qquad (10)$$

The Lagrangian dual problem then becomes the problem

$$z_{LD} = \max_{\lambda} D(\lambda). \qquad (11)$$

The Lagrangian dual (11) is a convex non-smooth program and can be solved using subgradient methods.

Due to the integer requirements in Eq. (2), a duality gap may occur between the optimal value of the Lagrangian dual (11) and the optimal value of Eq. (2) as described in the proof of Proposition 2 in [3]. The Lagrangian dual (11) provides lower bounds on the optimal value of Eq. (2) and the optimal solutions of the Lagrangian relaxation. In general, these first-stage solutions will not coincide unless the duality gap vanishes. The DD algorithm employs a branch and bound procedure that uses Lagrangian relaxation of non-anticipativity constraints as lower bounds [3].

*STEP 1* Initialization: Set $z^* = \infty$ and let $\mathcal{P}$ consist of problem (2).

*STEP 2* Termination: If $\mathcal{P} = \varnothing$ and $z^* < \infty$, then $x^*$ with $z^* = cx^* + Q(x^*)$ is optimal.

*STEP 3* Node selection: Select and delete a problem $P$ from $\mathcal{P}$, solve its Lagrangian dual (11). If the associated optimal value $z_{LD}(P)$ equals infinity go to *STEP 2*.

*STEP 4* Bounding: If $z_{LD}(P)$ is greater than $z^*$ go to *STEP 2*. Otherwise proceed as follows; if the first-stage solutions $x^j$, $j = 1,...,r$, of the subproblems are

(1) identical, then set $z^* := \min\left\{z^*, cx^j + Q(x^j)\right\}$.

(2) not identical, then compute a suggestion $\hat{x} = Heu(x^1,...,x^r)$ using some heuristic. If $\hat{x}$ is feasible then let $z^* := \min\left\{z^*, c\hat{x} + Q(\hat{x})\right\}$. Go to Step 5.

*STEP 5* Branching: Select a component $x_{(k)}$ of $\hat{x}$ and add two new problems to $\mathcal{P}$ that differ from $P$ by the additional constraint $x_{(k)} \le \lfloor \hat{x}_{(k)} \rfloor$ and $x_{(k)} \ge \lfloor \hat{x}_{(k)} \rfloor + 1$, respectively, if $x_{(k)}$

is integer, or $x_{(k)} \le \hat{x}_{(k)} - \varepsilon$ and $x_{(k)} \ge \hat{x}_{(k)} + \varepsilon$, respectively, if $x_{(k)}$ is continuous. The value of $\varepsilon > 0$ must be chosen such that the two new problems have disjoint subdomains. Go to *STEP 3*.

**2.2.3 Progressive Hedging**

Proposed by Rockafellar and Wets [19], the progressive hedging (PH) algorithm is a scenario decomposition method for stochastic programs motivated by augmented Lagrangian theory. By decomposing the extensive form into scenario subproblems, the PH algorithm effectively reduces the computational burden of solving extensive forms directly, especially for large-scale problem. Solving scenario subproblems separately can also take advantage of any special structures that are present.

A scenario solution is said to be admissible if it is feasible in one scenario; a scenario solution is said to be implementable or non-anticipative if its first-stage decision is scenario-independent; a solution is feasible if it is both admissible and implementable. The idea of the PH algorithm is to aggregate the admissible solutions of modified scenario subproblems, which progressively causes the aggregated solution to be non-anticipative and optimal. The modified scenario subproblem comes from scenario decomposition of the augmented Lagrangian as a close approximation of problem (3). The modified cost function includes a penalty term relative to the non-anticipativity constraint and a proximal term that measures the deviation of the scenario solution from the aggregated solution for first-stage decisions. The weight vector $w \in \mathbb{R}^{n_1 \times S}$ is updated by the penalty parameter (vector) $\rho > 0$ in each iteration. This weight update rule is essential to the proofs of the convergence theorems [19].

The PH algorithm has been proven to converge when all decision variables are continuous and can serve as a heuristic in the mixed-integer case. The basic PH algorithm for two-stage stochastic mixed-integer programs proceeds as follows [8]:

*STEP 1* Initialization: Let $v := 0$ and $w_v^j := 0$, $j = 1,...,r$. For each $j = 1,...,r$, compute

$$(x_{v+1}^j, y_{v+1}^j) := \arg\min_{x^j, y^j} \left\{ cx^j + q^j y^j : (x^j, y^j) \in S^j \right\}$$

*STEP 2* Iteration update: $v \leftarrow v + 1$

*STEP 3* Non-anticipative policy: $x_v := \sum_{j=1}^{r} p^j x_v^j$

*STEP 4* Weight update: $w_v^j := w_{v-1}^j + \rho(x_v^j - x_v),\ j = 1,...,r$

*STEP 5* Decomposition: For each $j = 1,...,r$, compute

$$(x_{v+1}^j, y_{v+1}^j) := \arg\min_{x^j, y^j} \left\{ cx^j + q^j y^j + w_v^j x + \frac{\rho}{2} \left\| x - x_v \right\|^2 : (x^j, y^j) \in S^j \right\}$$

*STEP 6* Termination: If all the first-stage scenario solutions $x_{v+1}^j$ agree, then stop. Otherwise, return to Step 2.

While convergence is not guaranteed for mixed-integer problems, computational studies have shown that the PH algorithm can find high-quality solutions within a reasonable number of iterations [25]. The PH algorithm also applies to multi-stage stochastic programs with discrete variables in any stage.

### 2.3 Integration of PH and DD

In view of the fact that the PH algorithm can find high-quality solutions within a reasonable number of iterations but is not guaranteed to converge in the mixed-integer case and the DD algorithm is exact but may be slow, fast progress in early iterations of PH could speed up convergence of DD to an exact solution if the PH algorithm can be combined with the DD algorithm. We now demonstrate how PH and DD can be integrated through their lower bounds. We first review the lower bounding technique for the PH algorithm proposed by Gade et al. [8] and recall equivalence between the best lower bounds obtained by the PH algorithm and the

Lagrangian dual from the DD algorithm. Finally, we establish relationships between PH weights and DD multipliers.

### 2.3.1 Lower Bounds for PH

Although the PH algorithm has been successfully applied as a heuristic to solve multi-stage stochastic mixed-integer programs, it is limited by the lack of convergence guarantee as well as the lack of information to evaluate solution quality relative to the optimal objective. Gade et al. [8] corrected this deficiency of the PH algorithm by presenting a method to compute lower bounds in PH for two-stage and multi-stage stochastic mixed-integer programs. This not only allows us to assess the quality of the solutions in each iteration, but also can provide lower bounds for solution methods, such as branch-and-bound, that rely on them. We restate Proposition 1 of [8], which shows that the weights $w$ define implicit lower bounds, $D(w)$, on the optimal objective value of denoted by $z^*$.

Proposition 1 [8]. Let $w^j, j = 1,...r,$ satisfy $\sum_{j=1}^{r} p^j w^j = 0$. Let

$$D_j(w^j) := \min\left\{ p^j(cx^j + q^j y^j + w^j x^j) : (x^j, y^j) \in S^j \right\}. \qquad (12)$$

Then $D(w) := \sum_{j=1}^{r} D_j(w^j) \le z^*$.

It can be verified $\sum_{j=1}^{r} p^j w^j = 0$ is maintained in every iteration by the weight update rule.

Proposition 1 indicates that one can compute a lower bound on $z^*$ in any iteration of the PH algorithm using the current weights with approximately the same effort as one PH iteration.

### 2.3.2 Information Exchange between PH and DD

Theorem 5.1. of Rockafellar and Wets [19] states that, in the convex case, the sequence $\{(\hat{x}^v, \rho^{-1}w^v)\}_{v=1}^{\infty}$ from PH converges to a pair $(x^*, \rho^{-1}w^*)$ such that $x^*$ solves the primal problem and $w^*$ solves the dual problem. In the mixed-integer case, however, a duality gap may occur because of the introduced nonconvexity. We restate Proposition 2 in [3], which follows from Theorem II.3.6.2 in [27], to provide insight into why this duality gap arises.

Proposition 2. The optimal value $z_{LD}$ of the Lagrangian dual (11) equals the optimal value of the linear program

$$\min\left\{\sum_{j=1}^{r} p^j(cx^j + q^j y^j) : (x^j, y^j) \in conv\, S^j, j = 1,...,r, x^1 = ... = x^r\right\}, \quad (13)$$

where *conv* denotes convex hull.

Gade et al. [8] show that by applying the PH algorithm to the linear program (13), one can recover both primal and dual optimal solutions to (13) and (11), respectively. Furthermore, the best PH lower bound $D(w)$ obtained from (12) equals the Lagrangian dual $z_{LD}$ from (11) and $c_{LD}$ obtained from (8). Since both PH and DD can decompose by scenario, the equivalence between $D(w)$ and $z_{LD}$ can be realized by the equivalence for each scenario, that is, $D_j(w^j)$ from (12) equals $Q_j(\mu^j)$ from (7) and $D_j(\lambda)$ from (10). Based on this observation, the equivalence can be established by letting $p^j w^j = \lambda^j$ for the non-anticipativity representation of Lulli and Sen and Lubin et al. and $p^j w^j = \lambda H^j$ for that of Carøe and Schultz. More generally, this information exchange can be applied in any iteration of the PH algorithm to obtain a starting point for solving the Lagrangian relaxation in the DD algorithm. We will illustrate a software implementation of the weight exchange method in detail in the next section.

### 2.4 Implementation

### 2.4.1 DDSIP – Implementation of DD

DDSIP [16] is a C package for the Dual Decomposition algorithm of Carøe and Schultz for two-stage stochastic mixed-integer programs. Its main idea is the Lagrangian relaxation of the non-anticipativity constraints and it uses a branch-and-bound algorithm to reestablish non-anticipativity. The dual optimization employs ConicBundle [10] provided by C. Helmberg as an implementation of the proximal bundle method [11]. The mixed-integer scenario subproblems in the branch-and-bound tree are solved using CPLEX [28].

### 2.4.2 PySP – Implementation of PH

PySP [26] is an open-source software package for modeling and solving stochastic programs by leveraging the combination of a high-level programming language (Python) and the embedding of the base deterministic model in that language (Pyomo [9]). It provides an implementation of PH for stochastic programs. One must specify both the deterministic base model and the scenario tree model to formulate a stochastic program in PySP. The PySP library also provides a generic implementation of the lower bounding method for the PH algorithm in a plugin called phboundextension.py.

In the application of PH, a significant trade-off in terms of the speed of convergence and quality of the solution is observed as the PH parameter, $\rho$ , is varied, indicating that larger values of a scalar $\rho$ can accelerate the convergence of PH while lower values of $\rho$ can improve the quality of solutions and lower bounds [8]. Watson and Woodruff [25] developed a heuristic method for selecting per-element $\rho(i)$ called SEP that will allow the updates to proceed more quickly to a "good" value $w^*$ of the weight $w$. The value of the $\rho$ component for an integer variable with index $i$ is determined after PH iteration 0 by setting $\rho(i) = c(i)/(x^{\max} - x^{\min} + 1)$ , where $c(i)$ is the corresponding cost coefficient, $x^{\max}(i) = \max_j x_1^j(i)$ and $x^{\min}(i) = \min_j x_1^j(i)$.

The primary advantage of the SEP selection heuristic is its problem-independent nature. However, there is a high likelihood that more effective methods exist for any specific problem. For instance, Watson and Woodruff [25] have observed that the best performing alternative for a class of stochastic mixed-integer resource allocation programs is a straightforward yet effective "cost-proportional" method that sets $\rho(i)$ equal to a multiple $k > 0$ of the element unit cost $c(i)$. This method is denoted by $CP(k)$. As a control measure in our computational results, various fixed, global values of $\rho$ denoted by $FX(\cdot)$ are used. The $FX$ stands for fixed and the argument gives the scalar value of $\rho$.

### 2.4.3 Weight Exchange between PySP and DDSIP

DDSIP allows three ways to represent the non-anticipativity constraints in problem (3):

NONANT1: $x^1 = x^2, x^1 = x^3, ..., x^1 = x^r$ (14)

NONANT2: $x^1 = x^2, x^2 = x^3, ..., x^{r-1} = x^r$ (15)

NONANT3: $x^i = \sum_{j=1}^{r} p^j x^j, \forall i = 1, ..., r-1$ (16)

By writing the three sets of equalities in the form $\sum_{j=1}^{r} H^j x^j = 0$ as in Lagrangian relaxation (5), the matrices $H^j$ for representation (14) are $H^1 = \begin{bmatrix} I_{n_1} \\ \vdots \\ I_{n_1} \end{bmatrix}, H^2 = \begin{bmatrix} -I_{n_1} \\ 0_{n_1} \\ \vdots \\ 0_{n_1} \end{bmatrix}, \dots, H^r = \begin{bmatrix} 0_{n_1} \\ \vdots \\ 0_{n_1} \\ -I_{n_1} \end{bmatrix}$, the matrices $H^j$ for representation (15) are $H^1 = \begin{bmatrix} I_{n_1} \\ 0_{n_1} \\ \vdots \\ 0_{n_1} \end{bmatrix}, H^2 = \begin{bmatrix} -I_{n_1} \\ I_{n_1} \\ 0_{n_1} \\ \vdots \\ 0_{n_1} \end{bmatrix}, \dots, H^r = \begin{bmatrix} 0_{n_1} \\ \vdots \\ 0_{n_1} \\ -I_{n_1} \end{bmatrix}$

and the matrices $H^j$ for representation (16) are $H^1 = \begin{bmatrix} diag(p^1 - 1) \\ diag(p^1) \\ \vdots \\ diag(p^1) \end{bmatrix}, H^2 = \begin{bmatrix} diag(p^2) \\ diag(p^2 - 1) \\ diag(p^2) \\ \vdots \\ diag(p^2) \end{bmatrix}, \dots, H^{r-1} = \begin{bmatrix} diag(p^{r-1}) \\ \vdots \\ diag(p^{r-1}) \\ diag(p^{r-1} - 1) \end{bmatrix}$ and $H^r = \begin{bmatrix} diag(p^r) \\ \vdots \\ diag(p^r) \end{bmatrix}$ where $diag(x)$ is a $(n_1 \times n_1)$ matrix with $x$ on the main diagonal.

Motivated by the equivalence between the best PH lower bound and the Lagrangian dual of the linear programming relaxation, we equate the corresponding objective function coefficients in the bounding subproblems for each scenario; i.e., $p^j w^j = \lambda H^j$. This equation enables the information exchange between PH and the Lagrangian dual. Given a weight $w$ from PH, the corresponding Lagrangian multiplier vector $\lambda = [-p^2 w^2, \dots, -p^r w^r]$ for representation (14), $\lambda = [\sum_{j=1}^{1} p^j w^j, \sum_{j=1}^{2} p^j w^j, \dots, \sum_{j=1}^{r-1} p^j w^j]$ for representation (15) and $\lambda = [p^1(w^1 - w^r), \dots, p^{r-1}(w^{r-1} - w^r)]$ for representation (16).

A model-dependent user-defined PySP extension called ddextension.py is used to create input files for DDSIP from the PySP input files and the PH results. While DDSIP allows the specification of various types of starting information such as an initial feasible solution or cost bound, in this paper we focus on providing starting values of the multipliers for solving the Lagrangian dual.

### 2.5 Numerical Results

In this section, we study the impact of DDSIP starting multipliers on the run-time of DDSIP for stochastic mixed-integer instances. We consider summary results of the performance of DDSIP starting multipliers on a number of stochastic server location instances. We investigate

the interaction between the strategies for choosing the PH $\rho$ parameter and the quality of DDSIP starting multipliers on a stochastic unit commitment problem. We further examine various types of starting information such as multipliers combined with initial solutions for DDSIP on a stochastic modified WECC-240 instance. All the experiments are conducted on Linux Mint 13 running as a virtual machine (3.7 GB RAM with one core at 3.1 GHz).

### 2.5.1 Server Location

The stochastic server location problem (SSLP) is a two-stage stochastic mixed-integer program widely applied in a variety of domains such as network design of electric power, internet server and telecommunications systems. The goal is to find the optimal server locations to minimize the investment costs minus the revenue while satisfying the clients' demand and not exceeding the servers' capacities. First-stage variables decide whether to locate a server at each potential position and second-stage variables assign the clients to the servers. A "scenario" specifies a subset of potential clients that are present. As we examine the following empirical results, SSLP instances are named m.n.s, where m is the number of potential server locations, n is the number of potential clients and s is the number of scenarios. The data for each instance are available as three text files in SMPS format (http://www2.isye.gatech.edu/~sahmed/siplib/sslp/sslp.html).

We compare DDSIP run-times required to reduce the relative duality gap below 0.001 with and without starting multipliers on a set of SSLP instances. Several parameters can be set to tune the performance of DDSIP for a particular problem or instance, including the frequency with which the Lagrangian dual is solved in the branch-and-bound tree and the number of iterations for which ConicBundle is allowed to run. We first experimented with these DDSIP parameters. The DDSIP performs the best with regard to the running time without starting

multipliers when the Lagrangian dual is solved in every 10th node and the Lagrangian dual is allowed to run for 2 iterations for each SSLP instance. Therefore, this DDSIP parameter setting is used for each run of SSLP instances. The PH $\rho$ parameter selection methods are explored for each SSLP instance and the PH algorithm is allowed to converge. The DDSIP run-time results in Table 4 are obtained using the best PH $\rho$ parameter selection method for each instance, which is specified in the second column of Table 1. As demonstrated in Table 1, starting multipliers derived from PH weights can reduce DDSIP run-time by up to 50% in stochastic server location instances.

Table 2.1 *DDSIP run-time results on a set of SSLP instances.*

| DDSIP run-time (seconds) | | Non-anticipativity representation | | | | | |
|---|---|---|---|---|---|---|---|
| | | NONANT1 | | NONANT2 | | NONANT3 | |
| SSLP instance | ρ selection method | With or without starting multipliers from PH | | | | | |
| | | Without | With | Without | With | Without | With |
| 5.50.500 | FX(10) | 181 | 140 | 148 | 133 | 154 | 81 |
| 5.50.1000 | FX(10) | 651 | 534 | 700 | 500 | 567 | 342 |
| 5.50.1500 | FX(10) | 1088 | 974 | 1060 | 959 | 1071 | 963 |
| 10.50.50 | CP(1) | 112 | 74 | 98 | 77 | 99 | 78 |
| 10.50.100 | CP(1) | 238 | 175 | 238 | 200 | 240 | 193 |
| 10.50.500 | CP(1) | 1777 | 1221 | 1367 | 1033 | 1476 | 1122 |
| 15.45.10 | CP(1) | 96 | 46 | 95 | 46 | 95 | 45 |
| 15.45.15 | CP(1) | 259 | 123 | 246 | 169 | 281 | 235 |

### 2.5.2 Unit Commitment

The unit commitment problem to schedule electricity generating units over a given time horizon is extensively used in daily system operation. The uncertainty in net load associated with inaccurate demand forecasts and unpredictable power output from variable generation units has traditionally been managed by deterministically derived reserve margins [18]. Stochastic unit commitment explicitly accounts for the uncertainty via probabilistic scenarios. The objective is

to minimize the expected total operational cost such that load is satisfied in all scenarios, subject to operational constraints such as ramp rate limits, minimum startup and shutdown times, and power flow limits on transmission lines. The first-stage variables are on/off decisions for the generators which incur startup, no-load and shutdown costs. The second-stage variables include scenario-specific power output levels. We use the model of Carrión and Arroyo [4] as our core deterministic optimization model [7].

We first execute on a 5 bus test case of the AMES wholesale power market test bed system [22], augmented with additional unit commitment extensions [5]. The instance includes 5 generators, 5 buses and 6 transmission lines with a scheduling horizon of 24 hours in hourly increments. We consider 10 equally likely scenarios for the sequence of hourly loads. The extensive form of this instance has 16,194 variables (1,200 binary) and 24,092 constraints.

Table 2 shows the running times required for DDSIP to reduce the relative duality gap below 0.001 for different parameter values, both without any starting information and with starting multipliers obtained from the final weights obtained by fixing the PH penalty parameter $\rho = 1$ and allowing the PH algorithm to converge. In Table 2, CBFREQ specifies the frequency of solving the Lagrangian dual using ConicBundle, and CBITLI specifies the limit for the number of descent steps in solving the Lagrangian dual.

Table 2.2 *DDSIP run-time with different ConicBundle parameters for the 5-bus instance.*

| DDSIP running time (seconds) | Non-anticipativity representation | | | | | |
|---|---|---|---|---|---|---|
| | NONANT1 | | NONANT2 | | NONANT3 | |
| ConicBundle parameter (CBFREQ, CBITLI) | With or without starting multipliers from PH | | | | | |
| | Without | With | Without | With | Without | With |
| (1, 5) | 2164 | 237 | 2179 | 249 | 2624 | 278 |
| (1, 1000) | 714 | 203 | 2014 | 263 | 477 | 165 |
| (100, 20) | 527 | 156 | 603 | 139 | 149 | 141 |
| (50, 10) | 271 | 73 | 426 | 102 | 654 | 102 |

Table 2 displays only a selection of the DDSIP parameters we have explored. Among all the DDSIP parameters we have experimented with, the DDSIP parameters of (50, 10) perform the best with regard to DDSIP run-time without starting multipliers. Therefore, we adopt (50, 10) as the DDSIP parameter setting for further experiments on this 5 bus test case.

Next, we consider the interaction between the value of PH parameter $\rho$ and the quality of DDSIP starting multipliers derived from PH weights. We vary the strategy to compute PH $\rho$ values for DDSIP starting multipliers. The results are shown in Table 3, where data in the row labeled FX(1) are repeated from Table 2. Even though we chose the DDSIP parameters with the shortest DDSIP running time without starting multipliers, the starting multipliers transformed from PH weights can reduce the DDSIP running time by roughly an order of magnitude in this instance as demonstrated by Table 2.

Table 2.3 *DDSIP run-time with starting multipliers from PH using different $\rho$ computation strategies for the 5-bus instance.*

| DDSIP running time (seconds) | Non-anticipativity representation | | |
|---|---|---|---|
| PH ρ value selection method | NONANT1 | NONANT2 | NONANT3 |
| No starting multipliers | 271 | 426 | 654 |
| FX(1) | 73 | 102 | 102 |
| FX(10) | 94 | 85 | 48 |
| FX(30) | 77 | 90 | 69 |
| CP(10) | 32 | 163 | 40 |
| SEP(10) | 75 | 121 | 76 |

To assess the performance of DDSIP starting multipliers on utility-scale systems, we test on a stochastic WECC-240 instance with 5 scenarios. The WECC-240 instance is introduced in [24], which provides a simplified description of the western US interconnection. This instance consists of a single bus and 85 generators with a scheduling horizon of 48 hours in hourly increments. Because it was originally introduced to assess market design alternatives, we have

modified this instance to capture characteristics more relevant to reliability assessment, including startup, shutdown, and nominal ramping limits, startup cost curves, and minimum up and down times. The full set of modifications and the case itself can be obtained by contacting the authors. The instance has 31,674 variables (4,080 binaries) and 59,374 constraints for a single scenario problem.

Table 4 reports the DDSIP run-time required to reduce the optimality gap below 2% and the optimality gap of the resulting solution with or without DDSIP starting multipliers on the WECC-240 stochastic instance. The DDSIP parameter is set to be (50, 10) for each run. Moreover, we study various types of DDSIP starting information by providing both starting multipliers and initial solutions for solving the Lagrangian dual from the final iteration of PH. Based on extensive exploration of $\rho$-setting strategies, in the PH run we choose CP(0.1) to compute PH parameter $\rho$ value and limit the number of PH iterations to 100. Without starting information, DDSIP cannot reduce the optimality gap below 99% within 24 hours. Supplying starting multipliers derived from PH weights, however, allows DDSIP to converge to a near-optimal solution within minutes. By also supplying the primal solution from PH, the DDSIP run-time is further reduced by up to an order of magnitude.

Table 2.4 *DDSIP run-time and optimality gap on WECC-240 stochastic instance with various starting information. The symbol * denotes failure to converge within 24 hours.*

| DDSIP run-time (seconds) and optimality gap | Non-anticipativity representation | | | | | |
|---|---|---|---|---|---|---|
| | NONANT1 | | NONANT2 | | NONANT3 | |
| DDSIP Starting information | Run-time | Opt. Gap | Run-time | Opt. Gap | Run-time | Opt. Gap |
| None | * | - | * | - | * | - |
| Multipliers only | 877 | 1.98% | 1937 | 1.82% | 5056 | 1.82% |
| Both multipliers and solutions | 671 | 1.88% | 777 | 1.99% | 646 | 1.94% |

**Acknowledgments**

This work was funded by the US Department of Energy's Advanced Research Projects Agency – Energy. We are grateful to Ralf Gollmer for his consistent assistance with DDSIP software. Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

## REFERENCES

[1] J.R. Birge, F. Louveaux, Introduction to Stochastic Programming, second ed., Springer, New York, 2011

[2] C.C. Carøe, J. Tind, L-shaped decomposition of two-stage stochastic programs with integer recourse, Technical Report, Institute of Mathematics, University of Copenhagen, 1995

[3] C.C. Carøe, R. Schultz, Dual decomposition in stochastic integer programming, Operations Research Letters 24 (1-2) (1999) 37-45

[4] M. Carrión, J.M. Arroyo, A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem, IEEE Trans. Power Systems 21 (2006) 1371-1378

[5] E. Ela, M. Milligan, M. O'Malley, A flexible power system operations simulation model for assessing wind integration, Power and Energy Society General Meeting, 2011 IEEE

[6] Y. Fan, C. Liu, Solving stochastic transportation network protection problem using the progressive hedging-based method, Networks and Spatial Economics 10 (2) (2010) 193–208

[7] Y. Feng, S.M. Ryan, Solution sensitivity-based scenario reduction for stochastic unit commitment, Computational Management Science (2014) 1-34

[8] D. Gade, G. Hackebeil, S.M. Ryan, C. J.-P. Watson, R.J.B. Wets, D.L. Woodruff, Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs, under review

[9] W.E. Hart, J.-P. Watson, D.L. Woodruff, Pyomo: Modeling and solving mathematical programs in Python, Mathematical Programming Computation 3 (3) (2011) 219-260

[10] C. Helmberg, The ConicBundle library for convex optimization, Available at: http://www-user.tu-chemnitz.de/helmberg/ConicBundle/

[11] K.C. Kiwiel, Proximity control in bundle methods for convex nondifferentiable optimization, Mathematical Programming 46 (1990) 105-122

[12] G. Laporte, F.V. Louveaux, The integer L-shaped method for stochastic integer programs with complete recourse, Operations Research Letters 13 (1993) 133-142

[13] A. Løkketangen, D.L. Woodruff, Progressive hedging and tabu search applied to mixed integer (0,1) multi-stage stochastic programming, Journal of Heuristics, 2 (2) (1996) 111-128

[14] M. Lubin, K. Martin, C. Petra, B. Sandıkçı, On parallezing dual decomposition in stochastic integer programming, Operations Research Letters 41 (3) (2013) 252-258

[15] G. Lulli, S. Sen, A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems, Management Science 50 (6) (2004) 786-796

[16] A. Märkert, R. Gollmer, User's Guide to ddsip – A C Package for the Dual Decomposition of Two-Stage Stochastic Programs with Mixed-Integer Recourse, Available at: http://plato.asu.edu/ddsip-man.pdf

[17] O. Listes, R. Dekker, A scenario aggregation based approach for determining a robust airline fleet composition, Transportation Science 39 (3) (2005) 367–382

[18] S.M. Ryan, C. Silva-Monroy, J.P. Watson, R.J.B. Wets, D.L. Woodruff, Toward scalable, parallel progressive hedging for stochastic unit commitment, in Proceedings of the 2013 IEEE Power and Energy Society General Meeting

[19] R.T. Rockafellar, R.J.-B. Wets, Scenarios and policy aggregation in optimization under uncertainty, Mathematics of Operations Research 16 (1) (1991) 119-147

[20] S. Sen, Algorithms for stochastic mixed-integer programming models, in: K. Aardal, G.L. Nemhauser, R. Weismantel (Eds.), Discrete Optimization, Elsevier, 2005, pp. 515-558.

[21] R. Van Slyke, R.J.B. Wets, L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming, SIAM Journal of Applied Mathematics 17 (4) (1969) 638-663.

[22] J. Sun, L. Tesfatsion, Dynamic testing of wholesale power market designs: An open-source agent-based framework, Computational Economics 30 (3) (2007) 291-327

[23] S. Takriti, J.R. Birge, E. Long, A stochastic model for the unit commitment problem, IEEE Trans. Power Systems 11 (1996) 1497-1508

[24] J.E. Price, J. Goodin, Reduced network modeling of WECC as a market design prototype, Proceedings of the 2011 IEEE Power and Energy Society General Meeting

[25] J.-P. Watson, D.L. Woodruff, Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems, Computational Management Science 8 (4) (2011) 355-370

[26] J.-P. Watson, D.L. Woodruff, W.E. Hart, PySP: Modeling and solving stochastic programs in Python, Mathematical Programming Computation 4 (2) (2012) 109-149

[27] L.A. Wolsey, G.L. Nemhauser, Integer and Combinatorial Optimization, Wiley-Interscience, New York, 1988

[28] IBM ILOG CPLEX Optimization Studio V12.6, Available at: http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud

# CHAPTER 3. PROGRESSIVE HEDGING LOWER BOUNDS FOR TIME CONSISTENT RISK-AVERSE MULTISTAGE STOCHASTIC MIXED-INTEGER PROGRAMS

A paper submitted to Annals of Operations Research

## Abstract

Risk-averse models have attracted attention in stochastic programming in situations where the decision maker is more concerned about large losses than average performance. When the risk-averse stochastic program is multistage, however, one key issue of time consistency arises. While definitions of time consistency vary, overall this property multistage stochastic programs is not guaranteed and depends on how the risk measure is computed. Expected conditional risk measures, which extend single-period risk measures to multiple stages, have been proved to be time consistent by Homem-de-Mello and Pagnoncelli (2016) according to their definition and, for some risk measures, allow for risk-neutral reformulations. . We propose scenario-decomposed versions of these risk-neutral formulations for a variety of risk measures and present an approach to obtain convergent and tight lower bounds from the Progressive Hedging (PH) algorithm. For mixed-integer programs where convergence is not guaranteed, this method can assess the quality of PH solutions and also integrate with exact algorithms that rely on lower bounds. We report computational results on financial portfolio optimization, lot sizing and a realistic-scale generation expansion planning problem and show that convergent and tight lower bounds are found.

*Keywords:* Risk-averse stochastic optimization; Scenario decomposition; Progressive Hedging algorithm; Time consistency; Expected conditional risk measures; Lower bounding

## 3.1 Introduction:

Traditional stochastic programming is risk-neutral in the sense that it is concerned with the optimization of an expectation criterion. This may yield solutions that are good in the long

run over repeated instances. But for non-repetitive decision making problems under uncertainty, the classical stochastic programming approach may perform poorly under certain realizations of the uncertain parameters. Thus, risk-averse models have attracted attention in the stochastic programming literature.

The two-stage risk-averse stochastic program can extend from the risk-neutral model in a straightforward way (Schultz ,2006). In the multistage case, however, the picture is quite different and there is no natural or obvious way of measuring risk (Homem-de-Mello and Pagnoncelli 2016). Time consistency is an important issue in modeling multistage risk-averse models. Risk-neutral stochastic programs are time consistent, which means the solutions for later stages found originally  remain optimal if the probem is resolved in the later stages (Pflug and Pichler 2016). In general, time consistency for multistage risk-averse stochastic programs does not hold true. Significant efforts have been made to achieve time consistency for multistage risk-averse problems. One popular way is to adopt the nested conditional risk measure proposed by Shapiro and Ruszczynski (2006), which has a drawback that the problem must be solved according to the recursive Bellman equations. Homem-de-Mello and Pagnoncelli (2016) extend single-period coherent risk measures to a class of multi-period risk measures called expected conditional risk measures (ECRMs) and show they are time consistent according to their broader definition, which allows for multiple optimal solutions. One advantage of ECRMs is that its resulting risk-averse problem can be reformulated as risk-neutral model with some additional variables and constraints.

In this paper, we show that the risk-neutral reformulations of several ECRMs are scenario decomposable. The resulting scenario formulations enable the use of existing scenario

decomposition approaches such as the progressive hedging (PH) algorithm to efficiently solve the risk-averse problems.

The PH algorithm is a scenario decomposition method developed by Rockafellar and Wets (1991) for stochastic programs with continuous decision variables. It has been explored by Watson and Woodruff (2011) as an effective heuristic for solving stochastic mixed-integer programs. Gade et al. (2016) presented a lower bounding technique for the PH algorithm and showed that, in two-stage convex problems, the best lower bound obtained from PH algorithm is as tight as the lower bound obtained from using Dual Decomposition developed by Caroe and Schultz (1999). In many applications where computational efficiency is valued, near-optimal solutions are desired within a reasonable amount of computation time. This lower bounding approach can assess solution quality in any iteration of the PH algorithm and can also integrate with exact algorithms that rely on lower bounds (Guo et al. 2015). This lower bounding technique, however, is restricted to risk-neutral models.

In this paper, we show how to obtain PH lower bounds for time-consistent multistage risk-averse stochastic integer programs with scenario-decomposable ECRMs. In the case of expected conditional value-at-risk, the optimization problems solved to obtain the bounds may be unbounded. . To overcome this hindrance, we find bounds for the optimal values of the additional decision variables introduced to obtain the risk-neutral reformulation, which also help speed up the convergence of PH algorithm. Our numerical results show that convergent and tight lower bounds are found.

### 3.2 Multistage Stochastic Mixed-Integer Programs

Suppose $T$ is the number of stages. We denote the uncertain parameters by $\xi = (\xi_2, ..., \xi_T)$, whose probability distributions are known. The decision vectors are represented

as $x = (x_1, x_2, ..., x_T)$. The realization of $\xi_t$ at stage $t = 2, ..., T$ is known only when decisions $x_{t-1}$ have been made. The history of the data process up to stage $t$ is denoted as $\xi_{[t]} \equiv (\xi_2, ..., \xi_t)$. The decisions and realizations are sequenced as

$$x_1, \xi_2, x_2(x_1, \xi_2), \xi_3, x_3(x_1, x_2, \xi_{[3]}), \ldots, x_T(x_1, \ldots, x_{T-1}, \xi_{[T]}).$$

We write the risk-neutral multi-stage stochastic mixed-integer program as

$$z = \min_{x_1, \ldots, x_T} \left\{ c_1^T x_1 + \mathbb{E}_{\xi_2}[Q_2(x_1, \xi_2)] : Ax_1 = b, x_1 \in \mathbb{Z}_+^{p_1} \times \mathbb{R}_+^{n_1 - p_1} \right\} \tag{1}$$

For $t = 2, ...T$, $Q_t(x_{t-1}, \xi_{[t]})$ is defined recursively as

$$\begin{aligned} & Q_t(x_{t-1}, \xi_{[t]}) = \min_{x_t} \left\{ c_t^T(\xi_{[t]}) x_t + \mathbb{E}_{\xi_{t+1} | \xi_{[t]}}[Q_{t+1}(x_t, \xi_{[t+1]})] \right\} \\ & T_t(\xi_{[t]}) x_{t-1} + W_t(\xi_{[t-1]}) x_t = h_t(\xi_{[t]}) \\ & x_t \in \mathbb{Z}_+^{p_t} \times \mathbb{R}_+^{n_t - p_t} \end{aligned}$$

Here $c_1 \in \mathbb{R}^{n_1}, b \in \mathbb{R}^{m_1}$ and $c_t(\xi) \in \mathbb{R}^{n_t}, h_t(\xi) \in \mathbb{R}^{m_t}$ are given vectors, while $A \in \mathbb{R}^{m_1 \times n_1}$ and $T_t(\xi) \in \mathbb{R}^{m_t \times n_{t-1}}, W_t(\xi) \in \mathbb{R}^{m_t \times n_t}$ are given matrices. The sets $x_t \in \mathbb{Z}_+^{p_t} \times \mathbb{R}_+^{n_t - p_t}$ denote the integer requirements on the variables at each time stage. The decisions are non-anticipative in the sense that a decision can depend on information revealed before the stage but not after.

The notation $\mathbb{E}_{\boldsymbol{\xi}}$ denotes expectation with respect to the distribution of random variable $\boldsymbol{\xi}$. To avoid complications when computing the integral behind $\mathbb{E}_{\boldsymbol{\xi}}$ we assume that we have only a finite number of realizations $\xi$ with corresponding probabilities $p_\xi$. Let $n_t$ be a scenario node that belongs to the set of all scenario tree nodes $N_t$ at stage $t \in T$. Let $\xi(n_t)$ be a scenario that belongs to the set of scenarios $\Xi(n_t)$ that define the node $n_t \in N_t$. Let $n_t(\xi)$ be the corresponding tree node for scenario $\xi \in \Xi$ at stage $t \in T$. Let $\hat{x}(n_t(\xi))$ be the non-anticipative decision made at scenario tree node $n_t(\xi)$. Then problem (1) is decomposable by scenario and

can be written as its so-called scenario reformulation of the multistage stochastic mixed-integer program:

$$\min_{x_1,\ldots,x_T}\left\{\sum_{\xi\in\Xi} p_\xi\left[c_1^T x_1(\xi)+\sum_{t=2}^{T} q_t\left(n_t(\xi)\right)^T x_t(\xi)\right]: \right.$$
$$\left. x_t(\xi)\in X_\xi,\ p_\xi x_t(\xi)-p_\xi \hat{x}\left(n_t(\xi)\right)=0,\ \forall \xi\in\Xi,\ \forall t=1,\ldots,T\right\} \tag{2}$$

where

$$X_\xi=\left\{\begin{array}{l} x_t(\xi): Ax_1(\xi)=b,\ x_t(\xi),\ \hat{x}\left(n_t(\xi)\right)\in\mathbb{Z}_+^{p_1}\times\mathbb{R}_+^{n_1-p_1},\ \forall t=1,\ldots,T \\ T_t\left(n_t(\xi)\right)x_{t-1}(\xi)+W_t\left(n_{t-1}(\xi)\right)x_t(\xi)=h_t\left(n_t(\xi)\right),\ \forall \xi\in\Xi,\ \forall t=2,\ldots,T \end{array}\right\}$$

The above problem (2) can decompose into scenario sub-problems

$$\min_{x_1,\ldots,x_T}\left\{c_1^T x_1(\xi)+\sum_{t=2}^{T} q_t\left(n_t(\xi)\right)^T x_t(\xi): x_t(\xi)\in X_\xi,\ \forall t=1,\ldots,T\right\} \text{ for scenarios } \forall \xi\in\Xi$$

which are coupled by the non-anticipativity constraints $p_\xi x_t(\xi)-p_\xi \hat{x}\left(n_t(\xi)\right)=0$.

### 3.3 Risk Measures

We distinguish between two classes of risk measures according to whether they are defined via quantiles or via deviation measures. Quantile risk measures are based on the quantiles of the probability distributions of the costs. Types of quantile based risk-measures include conditional value-at-risk (CVaR), which measures the expectation of worst outcomes for a given probability; and excess probability (EP), which measures the probability of exceeding a prescribed target level. Deviation risk measures are given by expectations of deviations of the relevant random variable from its mean or from some prescribed target. Examples of deviation based risk-measures include expected excess (EE), which measures the expected value of the excess over a given target; and semi-deviation (SD), which measures the expected value of the

excess over the mean. We use the definitions and notations for two-stage problems from Schultz (2006) where $f(x(\xi),\xi)$ is the objective function for a two-stage stochastic program.

***Definition 3.1.*** *The* $\alpha-$conditional value-at-risk ($\alpha-CVaR$) *reflects the expectation of the* $(1-\alpha)\cdot 100\%$ *worst outcomes for a given probability level* $\alpha \in (0,1)$, *and can be expressed by the following minimization formula:*

$$Q_{\alpha-CVaR}(x) = \min_{\eta\in\mathbb{R}} g(\eta, x), \tag{3}$$

*where* $g(\eta,x) := \eta + \frac{1}{1-\alpha}\mathbb{E}\left[\max\left\{f(x(\xi),\xi)-\eta, 0\right\}\right]$.

***Definition 3.2.*** *Excess probability (EP) is the probability of exceeding a prescribed target level* $\beta \in \mathbb{R}$, *and is defined as:*

$$Q_{\mathbb{P}^\beta}(x) = \mathbb{P}\left[\xi \in \Xi : f(x(\xi),\xi) > \beta\right]. \tag{4}$$

***Definition 3.3.*** *Expected excess (EE) reflects the expected value of the excess over a given target* $\gamma \in \mathbb{R}$, *and is defined as:*

$$Q_{D^\gamma}(x) = \mathbb{E}\left[\max\left\{f(x(\xi),\xi)-\gamma, 0\right\}\right]. \tag{5}$$

***Definition 3.4.*** *Semi-deviation (SD) is similar in spirit to the expected excess, but with the prefixed target replaced by the mean, and is defined as:*

$$Q_{D^+}(x) = \mathbb{E}\left[\max\left\{f(x(\xi),\xi)-Q_{\mathbb{E}}(x), 0\right\}\right]. \tag{6}$$

## 3.4 Time Consistency of Risk-Averse Multistage Stochastic Programs

### 3.4.1 Risk-Averse Multistage Stochastic Programs

Risk-averse models have attracted considerable attention in stochastic programming in situations where the decision maker is more concerned about large losses than average performance. Risk aversion is addressed by replacing the expectation in traditional stochastic

programs with risk measures to identify the best decisions. Another way to handle risk is to include risk measures in the constraints with important applications, such as in portfolio optimization with CVaR constraints. Krokhmal, Palmquist, and Uryasev (2002) is the first paper to deal with optimization approach with CVaR constraints. Fabian (2008) studies and proposes solution schemes for two-stage CVaR-minimization and CVaR-constrained problems. Guigues and Sagastizábal (2013) propose a risk-averse rolling-horizon time consistent approach with CVaR constraints. In this paper, however, we focus on the stochastic programs to handle risk measures in the objective.

For two-stage stochastic programs, the risk-averse model extends immediately from the risk-neutral model by replacing the expectation of the second stage cost with some risk measure. Ahmed (2006) provides two classes of mean-risk two-stage stochastic linear programs involving the semi-deviation risk measure and the quantile deviation risk measure, and proposes a decomposition algorithm. Schultz and Tiedemann (2006) present a mixed-integer programming formulation of a two-stage stochastic mixed-integer model involving CVaR. Miller and Ruszczynski (2011) formulate a risk-averse two-stage stochastic linear programming problem with a composition of conditional risk measures and demonstrate improvement over the existing decomposition approaches. Noyan (2012) develops two Benders-based decomposition algorithms for a risk-averse two-stage stochastic linear programming model with CVaR risk measure.

When it comes to multistage models, however, there is no natural way of measuring risk, as risk measures can be applied at every stage additively or to the complete scenario path or be measured in a nested form (Homem-de-Mello and Pagnoncelli 2016). The challenges in extending risk measures to the multistage case have been discussed extensively. Collado and

Papp (2011) introduce a scenario decomposition method for risk-averse multistage stochastic linear programs by using the dual properties of dynamic measures of risk.

### 3.4.2 Time Consistency

The definitions of time consistency differ by their focus. Some focus on the sequences of random variables (Ruszczynski 2010; Kovacevic and Pflug 2014), some are defined for continuous time dynamic models (Detlefsen and Scandolo 2005; Cheridito et al. 2006; Bion-Nadal 2008), while others take the point of view of optimization and decision making at every stage (Shapiro 2009; Carpentier et al. 2012; Rudloff et al. 2014). Here, we are most interested in time consistency for multistage stochastic programs. Shapiro (2009) claims that for time consistency of a problem, the solution at a node in the scenario tree must not depend on children of other nodes. Carpentier et al. (2012) formulate the property of time-consistency such that the optimal strategies obtained when solving the original problem remain optimal for all subsequent-stage problems. Pflug and Pichler (2016) consider a multistage stochastic decision problem to be time consistent if, when resolving the problem at later stages, the original solutions remain optimal for those stages. . Homem-de-Mello and Pagnoncelli (2016) define time consistency in terms of an inherited optimality property. Here we use the same definition of time consistency as Homem-de-Mello and Pagnoncelli (2016) such that given the optimal solutions from previous stages, resolving the problem results in the same solutions for the later stages if the optimal solutions are unique. If the optimal solutions are not unique, resolving the problem at the later stages gives the same optimal objective as computed by the original optimal solutions.

### 3.4.3 Time Consistency for Risk-Averse Multistage Stochastic Programs

Risk-neutral and two-stage risk-averse stochastic programs are time consistent. For multistage risk-averse stochastic programs, however, time consistency is not guaranteed and depends on how the risk measure is computed. The risk-averse models with risks measured at

every stage separately or measured for the complete scenario path are shown to be time inconsistent (Pflug and Pichler 2016). To enforce time consistency for decision problems, significant efforts and investigations have been initiated to identify classes of time consistent multistage risk measures. Shapiro and Ruszczynski (2006) propose a nested conditional risk measure for multistage optimization problems which proves to be time consistent. The nested conditional risk measure is formulated in recursive function which is not given in explicit form. Homem-de-Mello and Pagnoncelli (2016) address this drawback by proposing a class of expected conditional risk measures (ECRMs) which prove to be time consistent. One important advantage of ECRMs is that their resulting risk-averse problem can be formulated by a risk-neutral model for a modified problem with some additional variables and constraints. We will show in the next section that the risk-averse multistage stochastic program of ECRMs, based on a variety of single-period risk measures, can be decomposed by scenario.

### 3.5 Scenario Reformulation for Expected Conditional Risk Measures

Here, we will use the notations from Homem-de-Mello and Pagnoncelli (2016). Consider a probability space $(\Omega, \mathcal{F}, P)$, and let $\mathcal{F}_1 \subset \mathcal{F}_2 \subset ... \subset \mathcal{F}_T$ be sub sigma-algebras of $\mathcal{F}$ such that each $\mathcal{F}_t$ corresponds to the information available up to stage $t$, with $\mathcal{F}_1 = \{\varnothing, \Omega\}$ and $\mathcal{F}_T = \mathcal{F}$. Let $\mathcal{Z}_t$ denote a space of $\mathcal{F}_t$-measurable functions from $\Omega$ to $\mathbb{R}$, and let $\mathcal{Z} := \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_T$. A multi-period risk function $\mathbb{F}$ is defined as a mapping from $\mathcal{Z}$ to $\mathbb{R}$.

Homem-de-Mello and Pagnoncelli (2016) define the following multi-period risk measures $\mathbb{F}$ as expected conditional risk measures (ECRMs):

$$\mathbb{F}(Z_1, ..., Z_T) = Z_1 + \rho_2(Z_2) + \mathbb{E}_{\xi_{[2]}}\left[\rho_3^{\xi_{[2]}}(Z_3)\right] + \cdots + \mathbb{E}_{\xi_{[T-1]}}\left[\rho_T^{\xi_{[T-1]}}(Z_T)\right] \quad (7)$$

where the subscript in $\mathbb{E}$ indicates that the expectation is taken with respect to the corresponding variables. Homem-de-Mello and Pagnoncelli (2016) prove that any $\mathbb{F}$ defined as in (7) is time consistent, provided that each $\rho_t^{\xi_{[t-1]}}$ satisfies some basic properties that automatically hold, for example, for coherent risk measures.

Homem-de-Mello and Pagnoncelli (2016) use a particular case of ECRMs with $\rho_t = CVaR_{\alpha_t}$ denoted as $\mathbb{E}$-CVaR and show that $\mathbb{E}$-CVaR has the appealing property that any risk-averse multistage stochastic program defined with $\mathbb{E}$ -CVaR can be written as a risk-neutral model with some additional variables. Thus, existing algorithms can be adopted to solve the $\mathbb{E}$-CVaR problems. The other risk measures, however, are not yet investigated. Besides, the scenario decomposition of risk-averse multistage stochastic programs are not explored. In the next section, therefore, we will discuss the scenario reformulation of multistage stochastic programs with ECRMs based on various risk measures.

Homem-de-Mello and Pagnoncelli (2016) write the optimization formulation for $\mathbb{E}$-CVaR as follows:

$$\begin{aligned}
&\min_{x_1,\ldots,x_T}\Big\{c_1^T x_1 + CVaR_{\alpha_2}\left(c_2^T(\xi_{[2]})x_2\right) + \mathbb{E}_{\xi_2}\Big[CVaR_{\alpha_3}^{\xi_{[2]}}\left(c_3^T(\xi_{[3]})x_3\right) \\
&+\mathbb{E}_{\xi_3}\Big[CVaR_{\alpha_4}^{\xi_{[3]}}\left(c_4^T(\xi_{[4]})x_4\right)+\cdots \\
&+\mathbb{E}_{\xi_{T-1}}\Big[CVaR_{\alpha_T}^{\xi_{[T-1]}}\left(c_T^T(\xi_{[T]})x_T\right)\,|\,\xi_{[T-2]}\Big]\cdots\,|\,\xi_{[2]}\Big]\Big], \\
&x_t(\xi)\in X_\xi,\ p_\xi x_t(\xi) - p_\xi \hat{x}\left(n_t(\xi)\right)=0, \forall \xi\in\Xi, \forall t=1,\ldots,T\Big\}
\end{aligned} \tag{8}$$

By substituting the CVaR representation of Rockafellar and Uryasev (2000), we write the optimization formulation of $\mathbb{E}$ -CVaR in the form of a dynamic program:

$$z_{E-CVaR_{\alpha_t}} = \min_{x_1,\eta_2}\begin{Bmatrix} c_1^T x_1 + \eta_2 + \mathbb{E}_{\xi_2}\left[Q_2\left(x_1,\eta_2,\xi_2\right)\right]: \\ Ax_1 = b \end{Bmatrix} \tag{9}$$

For $t = 2,\ldots T-1$, we have:

$$Q_t\left(x_{t-1},\eta_t,\xi_{[t]}\right) = \min_{x_t,\eta_{t+1},v_t}\left\{\begin{array}{l}\dfrac{1}{1-\alpha_t}v_t + \eta_{t+1} + \mathbb{E}_{\xi_{t+1}}\left[Q_{t+1}\left(x_t,\eta_{t+1},\xi_{t+1}\right)\right]: \\ T_t\left(\xi_t\right)x_{t-1} + W_t\left(\xi_t\right)x_t = h_t\left(\xi_t\right) \\ \eta_t + v_t \geq c_t\left(\xi_t\right)^T x_t \\ v_t \geq 0\end{array}\right\}$$

For the last period $t = T$ we have:

$$Q_T\left(x_{T-1},\eta_T,\xi_{[T]}\right) = \min_{x_T,v_T}\left\{\begin{array}{l}\dfrac{1}{1-\alpha_T}v_T: \\ T_T\left(\xi_T\right)x_{T-1} + W_T\left(\xi_T\right)x_T = h_T\left(\xi_T\right) \\ \eta_T + v_T \geq c_T\left(\xi_T\right)^T x_T \\ v_t \geq 0\end{array}\right\}$$

Given $\alpha_t$ for stage $t$, the auxiliary variable $\eta_t$ is a "$(t-1)$-stage variable" to represent the value-at-risk ($VaR_\alpha$); i.e. the minimum $\alpha_t$-quantile such that the probability that the $t$-stage cost exceeds it is at least $\alpha_t$. Another auxiliary variable $v_t$ is a "$t$-stage variable" to represent the excess of $t$-stage cost of above $\eta_t$. As can be seen from formulation (9), the $\mathbb{E}$-CVaR optimization problem can be formulated as a risk-neutral model with two new variables ($\eta_t$ and $v_t$) and two additional constraints in stages $t = 2,...,T$.

For this formulation, existing algorithms can be readily adapted to solve the $\mathbb{E}$-CVaR optimization problem such as stochastic dual dynamic programming (SDDP) algorithm developed by Pereira and Pinto (1991) for continuous variables and stochastic dual dynamic integer programming (SDDIP) algorithm developed by Zou, Ahmed and Sun (2016) for integer variables. When the number of possible realizations are significant, however, the scenario decomposition algorithms display advantages in computational efficiency via parallel computing.

In order to realize the scenario decomposition in $\mathbb{E}$ -CVaR problem, we propose the scenario reformulation of $\mathbb{E}$-CVaR in the following Proposition 5.1.

***Proposition 5.1.*** *Consider the case with finitely many realizations* $\xi$ *and corresponding probabilities* $p_\xi$. *Let* $\alpha \in (0,1)$. *Then the scenario reformulation of* $\mathbb{E}$*-CVaR optimization can be represented as:*

$$z_{\mathbb{E}-CVaR_\alpha} = \min_{\substack{x_1,\ldots,x_T,\\ \eta_2,\ldots,\eta_T,\\ v_2,\ldots,v_T}} \left\{ \begin{array}{l} \sum_{\xi\in\Xi} p_\xi \left[ c_1(\xi)^T x_1(\xi) + \eta_2(\xi) + \frac{1}{1-\alpha_2} v_2(\xi) \cdots + \eta_T(\xi) + \frac{1}{1-\alpha_T} v_T(\xi) \right]: \\ \eta_t(\xi) + v_t(\xi) \geq c_t(\xi_t)^T x_t(\xi), \forall t = 2,\ldots,T \\ x_t(\xi) \in X_\xi, p_\xi x_t(\xi) - p_\xi \hat{x}(n(\xi_t)) = 0, \forall \xi \in \Xi, \forall t = 1,...,T \end{array} \right\} \quad (10)$$

where

$$X_\xi = \left\{ \begin{array}{l} x_t(\xi) : Ax_1(\xi) = b, x_t(\xi), \hat{x}(n(\xi_t)) \in \mathbb{Z}_+^{p_1} \times \mathbb{R}_+^{n_1-p_1}, \forall t = 1,...,T \\ T_t(\xi) x_{t-1}(\xi) + W_t(\xi) x_t(\xi) = h_t(\xi), \forall \xi \in \Xi, \forall t = 2,\ldots,T \end{array} \right\}$$

Similarly, we write the scenario reformulation for the risk-averse multistage stochastic program for the ECRM based on $\rho_t = EP_{\beta_t}$, denoted as $\mathbb{E}$-EP, as follows.

***Proposition 5.2.*** *Consider the case with finitely many realizations* $\xi$ *and corresponding probabilities* $p_\xi$. *Given a prescribed target level* $\beta_t \in \mathbb{R}$ *for each stage* $t = 1,\ldots,T$. *Then there exists a constant* $M > 0$ *such that the scenario reformulation of* $\mathbb{E}$ *-EP optimization is equivalent to the following program:*

$$z_{\mathbb{E}-EP_{\beta_t}} = \min_{x_1,\ldots,x_T} \left\{ \begin{array}{l} \sum_{\xi\in\Xi} p_\xi \sum_{t=1,\ldots,T} \theta_t(\xi): \\ \beta_t + M_t \cdot \theta_t(\xi) \geq c_t(\xi_t)^T x(\xi)_t, \forall t = 1,\ldots,T \\ x_t(\xi) \in X_\xi, p_\xi x_t(\xi) - p_\xi \hat{x}(n(\xi_t)) = 0, \forall \xi \in \Xi, \forall t = 1,...,T \end{array} \right\}. \quad (11)$$

Note that the constant $M_t > 0$ can be selected as $\sup\left\{c_t(\xi)^T x_t(\xi) : x_t(\xi) \in X_\xi, \forall \xi \in \Xi\right\}$.

The scenario reformulation for ECRM based on $\rho_t = EE_{\gamma_t}$, denoted as $\mathbb{E}$-EE, is addressed in Proposition 5.3.

***Proposition 5.3.*** *Consider the case with finitely many realizations $\xi$ and corresponding probabilities $p_\xi$. Given a prescribed target level $\gamma_t \in \mathbb{R}$ for each stage $t = 1, \ldots, T$. Then the scenario reformulation of $\mathbb{E}$-EE optimization is equivalent to the following program:*

$$z_{\mathbb{E}-EE_{\gamma_t}} = \min_{x_1,\ldots,x_T} \left\{ \begin{array}{l} \sum_{\xi\in\Xi} p_\xi \sum_{t=1,\ldots,T} e_t(\xi): \\ \gamma_t + e_t(\xi) \geq c_t(\xi_t)^T x(\xi)_t, \forall t = 1,\ldots,T \\ x_t(\xi) \in X_\xi, p_\xi x_t(\xi) - p_\xi \hat{x}(n(\xi_t)) = 0, \forall \xi \in \Xi, \forall t = 1,...,T \end{array} \right\}. \quad (12)$$

For the ECRM with $\rho_t = SD$, denoted as $\mathbb{E}$-SD, we are able to formulate its risk-averse stochastic program as a risk-neutral problem as in Proposition 5.4. However, unlike the previous formulations of the $\mathbb{E}$-CVaR, $\mathbb{E}$-EP and $\mathbb{E}$-EE risk measures, the formulation of $\mathbb{E}$-SD optimization is not separable by scenario due to the presence of constraint $s_t(\xi) \geq \sum_{\xi\in\Xi} p_\xi \left[ c_t(\xi_t)^T x(\xi) \right], \forall t = 1,\ldots,T$ and, therefore, is not eligible for scenario decomposition.

***Proposition 5.4.*** *Consider the case with finitely many realizations $\xi$ and corresponding probabilities $p_\xi$. Then the $\mathbb{E}$-SD optimization problem is equivalent to the following program:*

$$z_{\mathbb{E}-SD} = \min_{x_1,\ldots,x_T} \left\{ \begin{array}{l} \sum_{\xi\in\Xi} p_\xi \sum_{t=1,\ldots,T} \left[ s_t(\xi) - c_t(\xi_t)^T x(\xi) \right]: \\ s_t(\xi) \geq c_t(\xi_t)^T x(\xi)_t, \forall t = 1,\ldots,T \\ s_t(\xi) \geq \sum_{\xi\in\Xi} p_\xi \left[ c_t(\xi_t)^T x(\xi) \right], \forall t = 1,\ldots,T \\ x_t(\xi) \in X_\xi, p_\xi x_t(\xi) - p_\xi \hat{x}(n(\xi_t)) = 0, \forall \xi \in \Xi, \forall t = 1,...,T \end{array} \right\}. \quad (13)$$

## 3.6 Lower Bounding Approach for Risk-Averse Problems

### 3.6.1 Progressive Hedging (PH) Algorithm

Proposed by Rockafellar and Wets (1991), the progressive hedging (PH) algorithm is a scenario decomposition method for stochastic programs motivated by augmented Lagrangian theory. By decomposing the extensive form into scenario sub-problems, the PH algorithm effectively reduces the computational burden by solving the scenario sub-problems in parallel instead of solving extensive forms directly, especially for large-scale instances. Solving scenario sub-problems separately can also take advantage of any special structures that are present.

For a two-stage stochastic mixed-integer program, a solution is said to be admissible in one scenario if it is feasible in this scenario; a solution is said to be implementable or non-anticipative if its first-stage decision is scenario-independent; and a solution is feasible if it is both admissible to all scenarios and implementable. The idea of the PH algorithm is to aggregate the admissible solutions of modified scenario subproblems which progressively causes the aggregated solution to be non-anticipative and optimal. The modified scenario subproblem comes from scenario decomposition of the augmented Lagrangian as a close approximation of problem (3). The modified cost function includes a penalty term relative to the non-anticipative constraint and a proximal term that measures the deviation of the scenario solution from the aggregated solution for first-stage decisions. The weight vector $w \in \mathbb{R}^{n_1 \times s}$ is updated by the

penalty parameter (vector) $\rho > 0$ in each iteration. This weight update rule is essential to the proofs of the convergence theorems (Rockafellar and Wets 1991).

The PH algorithm has been proven to converge when all decision variables are continuous. It can also serve as a heuristic in the mixed-integer case. While convergence is not guaranteed for mixed-integer problems, computational studies have shown that the PH algorithm can find high-quality solutions within a reasonable number of iterations (Watson and Woodruff 2011). The PH algorithm for multistage stochastic mixed-integer programs is restated as follows (Gade et al. 2016):

*STEP 1* Initialization: Let $v := 0$ and $w^v\left(n\left(\xi_t\right)\right) := 0, \forall \xi \in \Xi, t = 1,\ldots,T$. Compute for each $\xi \in \Xi$:

$$x_t^{v+1}\left(\xi\right) := \arg\min\left\{c^T x_1 + \sum_{t=2}^{T} q_t\left(n\left(\xi_t\right)\right)^T x_t : x_t \in X_\xi\right\}$$

*STEP 2* Iteration update: $v \leftarrow v+1$

*STEP 3* Non-anticipative policy: Compute for each $t = 1,\ldots,T-1$ and each $n\left(\xi_t\right) \in N_t$:

$$\hat{x}_t^v\left(n(\xi_t)\right) := \sum_{\xi(n_t)\in\Xi(n_t)} p_\xi x_t^v\left(\xi\right) \Big/ \sum_{\xi(n_t)\in\Xi(n_t)} p_\xi$$

*STEP 4* Weight update: Compute for each $t = 1,\ldots,T-1$ and for each $\xi \in \Xi$:

$$w_t^v\left(n\left(\xi_t\right)\right) := w_t^{v-1}\left(n\left(\xi_t\right)\right) + \rho(x_t^v\left(n\left(\xi_t\right)\right) - \hat{x}_t^v\left(n\left(\xi_t\right)\right))$$

*STEP 5* Decomposition: Compute for each $\xi \in \Xi$:

$$x_t^{v+1}\left(\xi\right) := \arg\min\left\{c^T x_1 + \sum_{t=2}^{T} q_t\left(n\left(\xi_t\right)\right)^T x_t + \sum_{t=1}^{T-1}\left[w_t^v\left(n\left(\xi_t\right)\right)^T x_t + \frac{\rho}{2}\left\|x_t - \hat{x}_t^v\left(n\left(\xi_t\right)\right)\right\|^2\right] : x_t \in X_\xi\right\}$$

*STEP 6* Termination: If at each tree node, all the scenario solutions agree to within some tolerance, then stop. Otherwise, return to Step 2.

The performance of PH using various fixed, global values of penalty parameter $\rho$ with a single scalar used for all variables has been extensively explored in the literature (Mulvey and Vladimirou 1991; Listes and Dekker 2005; Fan and Liu 2010). We denote the corresponding method by $FX(\cdot)$, where the $FX$ stands for fixed and the argument provides the single value of $\rho$. Watson and Woodruff (2011) observe that the objective cost per decision variable may range in magnitude and an effective $\rho$ values should be close in magnitude to the unit cost per decision variable in the objective. Specifically, they set $\rho(i)$ for each decision variable $i$ to be a multiple of the corresponding objective cost coefficient. The method is denoted by $CP(\cdot)$, where $CP$ stands for cost-proportional and the argument gives the cost multiplier $k > 0$. Previous experience (Watson and Woodruff 2011; Gade et al. 2016) indicates that larger values of $\rho$ can accelerate the convergence of PH while oscillation can occur when the weight vector is updated too aggressively by large values of $\rho$. While smaller values of $\rho$ lead to slow changes in weight vector as well as little movement in convergence of PH, the quality of resulting solutions and lower bounds is improved. In addition to the strategies for choosing the PH $\rho$ parameter, Watson and Woodruff (2011) introduced additional strategies such as variable fixing and slamming to break cycles and accelerate PH convergence.

### 3.6.2 Lower Bounds from PH on Multistage Stochastic Mixed-Integer Programs

Although the PH algorithm has been applied successfully as a heuristic to solve multistage stochastic mixed-integer programs, it is limited by the lack of a convergence guarantee as well as the lack of information to evaluate solution quality relative to the optimal objective value. Gade et al. (2016) addressed this deficiency by presenting a method to compute lower bounds in the PH algorithm for multistage stochastic mixed-integer programs. The lower

bounds not merely allow us to assess the quality of the solutions in each iteration, but also can provide lower bounds for solution methods like branch-and-bound that rely on lower bounds. We elaborate the lower bounding approach for two-stage stochastic mixed-integer programs proposed by Gade et al. (2016) to multistage cases and show that the weights $w$ define implicit lower bounds, $D(w)$, on the optimal objective value denoted by $z^*$.

***Proposition 6.1.*** *Let* $w = \left(w\left(n\left(\xi_t\right)\right)\right)_{\xi \in \Xi}$ *where* $w\left(n\left(\xi_t\right)\right) \in \mathbb{R}^{n_1}$ *satisfy*

$\sum_{\xi(n_t) \in \Xi(n_t)} p_\xi w\left(n\left(\xi_t\right)\right) = 0$ *for each* $n_t \in N_t$. *Let*

$$D_\xi\left(w\left(n\left(\xi_t\right)\right)\right) := \min\left\{c^T x_1 + \sum_{t=2}^{T} q_t\left(n\left(\xi_t\right)\right)^T x_t + \sum_{t=1}^{T-1} w_t\left(n\left(\xi_t\right)\right)^T x_t : x_t \in X_\xi\right\} \tag{14}$$

*Then* $D(w) := \sum_{\xi \in \Xi} p_\xi D_\xi\left(w\left(n\left(\xi_t\right)\right)\right) \le z^*$.

It can be verified $\sum_{\xi(n_t) \in \Xi(n_t)} p_\xi w\left(n\left(\xi_t\right)\right) = 0$ is maintained in every iteration by the weight update rule. Proposition 6.1 indicates that one can compute a lower bound on $z^*$ in any iteration of PH algorithm using the current weights with approximately the same effort as one PH iteration.

### 3.6.3 Scenario Bundling in Progressive Hedging

Motivated by Wets' strategy of aggregating scenarios in stochastic optimization (Wets 1989), Gade et al. (2016) formalized the bundle version of PH algorithm, which allows Steps 1 and 5 of the PH algorithm to solve smaller extensive forms of the original problem. We extend the bundle version of lower bounding approach for two-stage cases introduced by Gade et al. (2016) to multistage cases in Proposition 6.2. Suppose the set of all the scenario tree nodes $N_t$ at

stage $t = 2,\ldots,T-1$ is partitioned into bundles, $\beta_t$, of $K$ scenario tree nodes each. We denote the set of bundles by $B_t$, with $\beta_t \in B_t$. Let $P_{\beta_t} = \sum_{\xi \in \Xi(\beta_t)} p_\xi$.

***Proposition 6.2.*** *Let* $w = \left(w(\beta_t)\right)_{\beta_t \in B_t}$ *where* $w(\beta) \in \mathbb{R}^{n_1}$ *satisfy* $\sum_{\beta_t \in B_t} P_{\beta_t} w(\beta_t) = 0$. *Let*

$$D_{\beta_t}\left(w(\beta_t)\right) := \min_x \left\{ c^T x_1 + \sum_{t=2}^{T} \sum_{\xi \in \Xi(\beta_t)} \frac{p_\xi}{P_{\beta_t}} q_t(\xi)^T x_t + \sum_{t=1}^{T-1} w_t(\beta_t)^T x_t : x_t \in X_\beta \right\} \tag{15}$$

*Then* $D(w) := \sum_{\beta_t \in B_t} P_{\beta_t} D_{\beta_t}\left(w(\beta_t)\right) \le z^*$.

### 3.6.4 Lower Bounds on E-CVaR Stochastic Mixed-Integer Programs

By applying the lower bounding approach in Proposition 6.1 to the optimization formulation of the $\mathbb{E}$-CVaR problem, we have the following Proposition 6.3.

***Proposition 6.3.*** *Let* $w = \left(w(\xi)\right)_{\xi \in \Xi}$ *where* $w(\xi) \in \mathbb{R}^{n_1}$ *satisfy* $\sum_{\xi(n_t) \in \Xi(n_t)} p_\xi w(\xi) = 0$, $w' = \left(w'(\xi)\right)_{\xi \in \Xi}$ *where* $w'(\xi) \in \mathbb{R}$ *satisfy* $\sum_{\xi(n_t) \in \Xi(n_t)} p_\xi w'(\xi) = 0$, and $w'' = \left(w''(\xi)\right)_{\xi \in \Xi}$ *where* $w''(\xi) \in \mathbb{R}$ *satisfy* $\sum_{\xi(n_t) \in \Xi(n_t)} p_\xi w''(\xi) = 0$ *for each* $n_t \in N_t$. *Let*

$$D_\xi\left(w(\xi), w'(\xi), w''(\xi)\right) := \min \left\{ \begin{array}{l} c_1^T x_1 + \eta_2 + \frac{1}{1-\alpha_2} v_2 + \cdots + \eta_T + \frac{1}{1-\alpha_T} v_T \\ + w_1^T(\xi) x_1 + \sum_{t=2}^{T} w_t'^T(\xi) \eta_t + \sum_{t=2}^{T-1} w_t''^T(\xi) v_t : \\ \eta_t + v_t \ge c_t(\xi_t)^T x_t, \forall t = 2,\ldots,T \\ x_t(\xi) \in X_\xi,\ p_\xi x_t(\xi) - p_\xi \hat{x}\left(n(\xi_t)\right) = 0, \forall \xi \in \Xi, \forall t = 1,\ldots,T \end{array} \right\} \tag{16}$$

*Then* $D(w, w', w'') := \sum_{\xi \in \Xi} p_\xi D_\xi\left(w(\xi), w'(\xi), w''(\xi)\right) \le z^*_{\alpha-CVaR}$.

However, the lower bounding approach for solving the $\mathbb{E}$-CVaR problem is not as straightforward as the one for solving risk-neutral problems. Take a 3-stage stochastic mixed-

integer program for instance. The problem $D_\xi\left(w(\xi), w'(\xi), w''(\xi)\right)$ can be written as:

$$D_\xi\left(w(\xi), w'(\xi), w''(\xi)\right) := \min\left\{\begin{array}{l} c_1^T x_1 + w_1^T(\xi)x_1 + \left(w_2''^T(\xi) + \dfrac{1}{1-\alpha_2}\right)\left(\eta_2 + v_2\right) \\ + \left(w_2'^T(\xi) - w_2''^T(\xi) - \dfrac{\alpha_2}{1-\alpha_2}\right)\eta_2 + \left(w_3'^T(\xi)+1\right)\eta_3 + \dfrac{1}{1-\alpha_3}v_3 : x_t \in X_\xi \end{array}\right\}$$

AThis optimization problem is unbounded due to the unboundedness property of decision variables $\eta_2, \eta_3$ and $\eta_2 + v_2$. To solve the optimization problem $D_\xi\left(w(\xi), w'(\xi), w''(\xi)\right)$, we must find valid upper and lower bounds for the decision variables $\eta_2, \eta_3$ and a valid upper bound for $\eta_2 + v_2$. Such bounds are derived in Proposition 6.4. Those bounds lead to tighter lower bounds of $z^*_{\alpha-CVaR}$ and can be obtained with little computational effort. In addition, the introduction of the bounds also speeds up the convergence of the PH algorithm.

***Proposition 6.4.*** *Let $\eta_t^*, v_t^*$ be optimal values of $\eta_t, v_t, \forall t = 2,...,T$ for the $\mathbb{E}$-CVaR stochastic program. Then*

$$\begin{aligned} U_t &= \max_{\xi\in\Xi}\left\{\max_x\left\{c_t(\xi)^T x_t : x_t \in X_\xi\right\}\right\} \geq \eta_t^* + v_t^*, \\ L_t &= \min_{\xi\in\Xi}\left\{\min_x\left\{c_t(\xi)^T x_t : x_t \in X_\xi\right\}\right\} \leq \eta_t^*. \end{aligned} \tag{17}$$

**Proof**: Let $x_t^*(\xi), \eta_t^*(\xi), v_t^*(\xi)$ be optimal solutions for the $\mathbb{E}$-CVaR stochastic program.

(a) The definition of CVaR straightforwardly indicates that

$$U_t = \max_{\xi\in\Xi}\left\{\max_x\left\{c_t(\xi)^T x_t : x_t \in X_\xi\right\}\right\} \geq \eta_t^* + v_t^*.$$

(b) Based on the definition of CVaR (Schultz and Tiedemann 2006), we define the cumulative distribution function of $\eta_t$ to be $\Psi(x_t^*(\xi), \eta_t) := P(\{\xi \in \Xi : c_t\left(\xi\right)^T x_t^*(\xi) \leq \eta_t\})$ and define $\alpha - VaR$ as $\eta_{t,\alpha}(x_t^*(\xi)) := \min\{\eta_t : \Psi(x_t^*(\xi), \eta_t) \geq \alpha\}$. Since the cumulative distribution

function $\Psi(x_t^*(\xi),\eta_t)$ of $\eta_t$ is a monotonically increasing function over $\eta_t$ and a function defined as $\Psi'(x_t^*(\xi),\alpha) := \min\{\Psi(x_t^*(\xi),\eta_t) : \Psi(x_t^*(\xi),\eta_t) \geq \alpha\}$ is a monotonically increasing function over $\alpha$, then $\eta_{t,\alpha}(x_t^*(\xi))$ is monotonically increases over $\alpha$. Besides, since $\lim_{\alpha\to 0^+} \eta_{t,\alpha}\left(x_t^*(\xi)\right) = \min_{\xi\in\Xi}\left\{c_t(\xi)^T x_t^*(\xi)\right\}$, then we have $\min_{\xi\in\Xi}\left\{c_t(\xi)^T x_t^*(\xi)\right\} \leq \eta_{t,\alpha}\left(x_t^*(\xi)\right)$. Since $x_t^*(\xi), \forall \xi \in \Xi$ are feasible for the risk-neutral problem, then $c_t(\xi)^T x_t^*(\xi) \geq \min_{x_t}\left\{c_t(\xi)^T x_t : x_t \in X_\xi\right\}$ for each $\xi \in \Xi$. By taking the minimum for all $\xi \in \Xi$ on both sides, we have $\min_{\xi\in\Xi}\left\{c_t(\xi)^T x_t^*(\xi)\right\} \geq \min_{\xi\in\Xi}\left\{\min_{x_t}\left\{c_t(\xi)^T x_t : x_t \in X_\xi\right\}\right\}$. Thus, it is proved $L_t = \min_{\xi\in\Xi}\left\{\min_{x_t}\left\{c_t(\xi)^T x_t : x_t \in X_\xi\right\}\right\} \leq \eta_t^*$.

Therefore, to compute the lower bounds for the $\mathbb{E}$-CVaR problem, one must compute both $U_t$ and $L_t$ by solving the minimization problem and the maximization problem of its risk-neutral model for each stage $t, \forall t = 2,...,T$ beforehand and then solve the modified problem of (16) with two additional constraints (17) in each stage $t, \forall t = 2,...,T$.

### 3.6.5 Lower Bounds on E-EP Stochastic Mixed-Integer Programs

It is straightforward to apply the lower bounding approach in Proposition 6.1 to the multistage risk-averse stochastic mixed-integer problems with $\mathbb{E}$-EP, resulting in the lower bounding approach in Proposition 6.5.

***Proposition 6.5.*** *Let* $w = \left(w(\xi)\right)_{\xi\in\Xi}$ *where* $w(\xi) \in \mathbb{R}^{n_1}$ *satisfy* $\sum_{\xi(n_t)\in\Xi(n_t)} p_\xi w(\xi) = 0$ *for each* $n_t \in N_t$. *Let*

$$D_{\xi}\left(w(\xi)\right) := \min\left\{\begin{array}{l} \sum_{t=1,\ldots,T}\theta_t + \sum_{t=1}^{T-1} w_t^T(\xi)\theta_t : \\ \beta_t + M\cdot\theta_t \geq c_t(\xi_t)^T x_t, \forall t = 1,\ldots,T \\ x_t(\xi)\in X_{\xi},\ p_{\xi}x_t(\xi) - p_{\xi}\hat{x}\left(n(\xi_t)\right) = 0, \forall \xi\in\Xi, \forall t = 1,...,T \end{array}\right\} \tag{18}$$

*Then* $D(w) := \sum_{\xi\in\Xi} p_{\xi} D_{\xi}\left(w(\xi)\right) \leq z_{EP}^{*}$.

### 3.6.6 Lower Bounds on E-EE Stochastic Mixed-Integer Programs

Similar to Proposition 6.5, we can easily derive the lower bounding approach for multistage risk-averse stochastic mixed-integer programs with $\mathbb{E}$-EE in Proposition 6.6.

***Proposition 6.6.*** *Let* $w = \left(w(\xi)\right)_{\xi\in\Xi}$ *where* $w(\xi)\in\mathbb{R}^{n_1}$ *satisfy* $\sum_{\xi(n_t)\in\Xi(n_t)} p_{\xi} w(\xi) = 0$ *for each* $n_t \in N_t$. *Let*

$$D_{\xi}\left(w(\xi)\right) := \min\left\{\begin{array}{l} \sum_{t=1,\ldots,T} e_t + \sum_{t=1}^{T-1} w_t^T(\xi) e_t : \\ \gamma_t + M\cdot e_t \geq c_t(\xi_t)^T x_t, \forall t = 1,\ldots,T \\ x_t(\xi)\in X_{\xi},\ p_{\xi}x_t(\xi) - p_{\xi}\hat{x}\left(n(\xi_t)\right) = 0, \forall \xi\in\Xi, \forall t = 1,...,T \end{array}\right\} \tag{19}$$

*Then* $D(w) := \sum_{\xi\in\Xi} p_{\xi} D_{\xi}\left(w(\xi)\right) \leq z_{EE}^{*}$.

## 3.7 Numerical Results

In this section, we study the performance of the lower bounding approach for risk-averse stochastic mixed-integer test instances with $\mathbb{E}$ -CVaR. We investigate the interaction between the strategies for choosing the PH $\rho$ parameter and the quality of PH lower bounds as well as the scenario bundling strategies on a financial portfolio optimization instance. We consider summary results of the performance of the lower bounding approach on a number of lot sizing

instances. We further examine the lower bounding approach on a risk-averse large-scale power generation expansion planning instance whose extensive form is too large to solve.

We use PySP (Watson et al. 2012), an open-source software package for modeling and solving stochastic programs, to implement PH algorithm and a plugin called phboundextension to implement the lower bounding approach for PH algorithm. CPLEX is used to solve mixed-integer linear optimization programs. All the experiments are conducted on a Linux server with 31 GB and 8 processors with 4 cores per processor.

### 3.7.1 Portfolio Optimization Problem

The application of multistage stochastic programming has gained popularity in the financial industry to address the stochastic nature of financial problems. The multistage portfolio optimization (MPO) problem, or multistage financial asset allocation problem, finds the optimal decisions to rebalance the portfolio over time to maximize the expected value of the portfolio by the end of the planning horizon. We modify the portfolio optimization formulation from Dantzig and Infanger (1993). At the initial time period, a certain amount of wealth is available to a decision maker in asset $i = 1,\ldots n$ and in cash which we index as asset $n+1$ with $x_i^0, i = 1,\ldots,n+1$ to be the dollar value of initially available assets. At each time period $t = 1,\ldots T$, an investor can sell off an amount of asset $i$ worth $y_i^t$ for cash or buy an amount of asset $i$ worth $z_i^t$ from trades in previous periods, and his resulting amount of asset $i$ at period $t$ is denoted as $x_i^t$. Buying and selling causes transaction costs proportional to the dollar value of the asset traded. Buying one unit of asset $i$ requires $1+v_i$ units of cash and selling one unit of asset $i$ results in $1-\mu_i$ units of cash. At time period $t$, the return rate $r_i^t$ of asset $i$ from period $t$ to period $t+1$ is not known to the decision maker until after the decision is made on

rebalancing the portfolio for period $t$. Only the return rate on cash, $r_{n+1}^t$ and the return rate on asset $i$ from initial period, $r_i^0$ are assumed known. In addition to Dantzig and Infanger's formulation, it is required that the amount of assets sold or bought must be either zero or a positive value between its lower and upper bounds. A multistage stochastic mixed-integer programming formulation of MPO problem is:

$$\max \sum_{\xi \in \Xi} p_\xi \sum_{i=1}^{n+1} r_i^T(\xi) x_i^T(\xi) \tag{20a}$$

$$r_i^{t-1}(\xi) x_i^{t-1}(\xi) + z_i^t(\xi) - y_i^t(\xi) = x_i^t(\xi), \forall i = 1,\ldots n, t = 1,\ldots T, \xi \in \Xi \tag{20b}$$

$$r_{n+1}^{t-1}(\xi) x_{n+1}^{t-1}(\xi) - \sum_{i=1}^{n}(1+v_i) z_{n+1}^t(\xi) + \sum_{i=1}^{n}(1-\mu_i) y_{n+1}^t(\xi) = x_{n+1}^t(\xi), \forall t = 1,\ldots T, \xi \in \Xi \tag{20c}$$

$$m_i^t(\xi) l_i^y \le y_i^t(\xi) \le m_i^t(\xi) u_i^y, n_i^t(\xi) l_i^z \le z_i^t(\xi) \le n_i^t(\xi) u_i^z, \forall i = 1,\ldots n, t = 1,\ldots T, \xi \in \Xi \tag{20d}$$

$$x_i^t(\xi), y_i^t(\xi), z_i^t(\xi) \ge 0, m_i^t(\xi), n_i^t(\xi) \in \{0,1\}, \forall i = 1,\ldots n+1, t = 1,\ldots T, \xi \in \Xi \tag{20e}$$

We generate a test instance with 5 assets, 3 stages and 10 branches emanating from each scenario tree node. The 10 branches from each scenario tree node are sampled from normal distributions of stochastic parameters $r_i^t$ with identical probabilities for each asset for each period. The means of normal distributions are displayed in Table 1 and the standard deviations are 0.5. The fixed input parameters are displayed in Table 2.

Table 3.1 *Mean values of normal distributions of return rates of assets for MPO test instance*

| | Asset 1 | Asset 2 | Asset 3 | Asset 4 | Asset 5 |
|---|---|---|---|---|---|
| Period 1 | 1 | 1.1 | 1.2 | 1.3 | 1.4 |
| Period 2 | 1 | 0.9 | 0.8 | 0.7 | 0.6 |
| Period 3 | 0.8 | 0.9 | 1 | 1.1 | 1.2 |

Table 3.2 *Input parameters for MPO test instance*

| | |
|---|---|
| $x_i^0, i=1,\ldots,n+1$ | 100 |
| $v_i, i=1,\ldots,n$ | 0 |
| $\mu_i, i=1,\ldots,n$ | 0.5% |
| $r_{n+1}^t$ | 1.02 |
| $l_i^y, l_i^z$ | 30 |
| $u_i^y, u_i^z$ | 300 |

Here, we performed computational studies on the $\mathbb{E}$ -CVaR problem of this MPO instance given the upper and lower bounds for $\mathbb{E}$-CVaR variables. The preselected probability is set to be $\alpha_t = 0.8$ for each stage $t$ such that we are only concerned with the 20% worst scenarios at each stage. We perform multiple runs of the PH algorithm on this instance, varying the values of the penalty parameter $\rho$. Specifically, we consider fixed $\rho \in \left\{FX\left(10^{-2}\right), FX\left(10^{-3}\right), FX\left(10^{-4}\right)\right\}$ and record the time-series of the lower bound $D\left(w, w', w''\right)$ obtained at each PH iteration during each run. The lower bound results are shown in Figure 1 (a), which additionally displays the optimal solution value obtained from solving its extensive form. We also consider the PH lower bounds when bundling scenarios. Specifically, we vary the number of scenarios in each bundle considered by PH, while holding $\rho$ constant. Each scenario bundle is formed by some scenarios emanating from the same scenario tree node. An illustrative example is shown in Figure 1 (b), with $\rho = FX\left(10^{-3}\right)$.

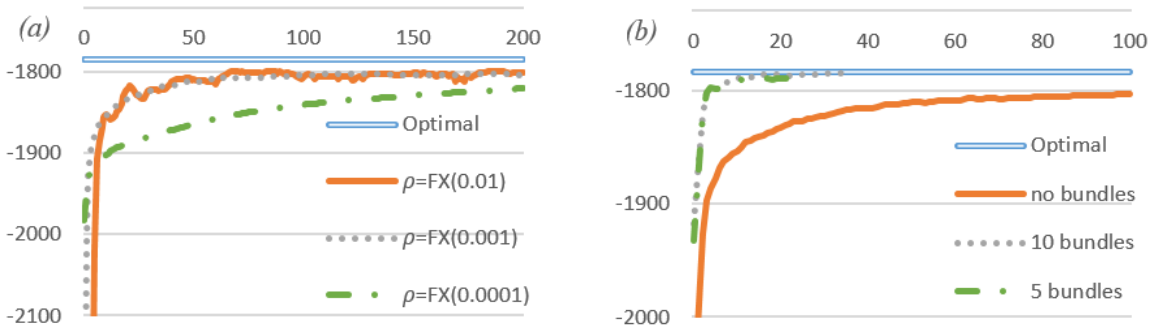

Figure 3.1 *Lower bounds from PH and optimal value from solving extensive from for MPO instance with (a) different penalty parameter values; (b) different scenario bundling strategies with* $\rho = FX\left(10^{-3}\right)$

As displayed in the PH lower bounding results in Figure 1 (a), larger $\rho$ values can lead to oscillations in the convergence of lower bounds. In contrast, lower $\rho$ values smoothen the convergence of lower bounds but can also slow down their convergence. Figure 1 (b) shows the advantage of scenario bundling for improving the quality of lower bound convergence but the disadvantage is that each PH iteration takes longer.

Table 3 further demonstrates that scenario bundling may reduce the number of PH iterations to converge as well as the total PH computational time. The computation time consumed per iteration, however, increases with the number of scenarios per bundle.

Table 3.3 *Computation time for MPO test instance with different scenario bundles*

| Number of bundles | 100 | 10 | 5 |
|---|---|---|---|
| Number of scenarios per bundle | 1 | 10 | 20 |
| Number of PH iterations to converge | 200 | 36 | 25 |
| Average PH execution time per iteration (seconds) | 4.5 | 11.8 | 16.9 |
| Total PH execution time (seconds) | 909 | 425 | 423 |

### 3.7.2 Lot Sizing Problem

The multistage lot-sizing problem (MLS) has been widely used as a test case for multistage stochastic integer programming algorithms (Burhaneddin and Özaltın 2014). It seeks to determine a minimum cost production and inventory holding schedule for a product to satisfy its stochastic demand over a finite discrete planning horizon. A multistage stochastic mixed-integer programming formulation of the MLS problem is:

$$\min \sum_{\xi \in \Xi} p_{\xi} \sum_{t \in T} \left( \alpha_t x_t(\xi) + \beta_t y_t(\xi) + h_t s_t(\xi) \right) \tag{21a}$$

$$s_{t-1}(\xi) + x_t(\xi) = d_t(\xi) + s_t(\xi), \forall t = 1,\ldots T, \xi \in \Xi \quad (21b)$$

$$x_t(\xi) \le My_t(\xi), \forall t = 1,\ldots T, \xi \in \Xi \quad (21c)$$

$$s_0(\xi) = 0, \forall \xi \in \Xi \quad (21d)$$

$$y_t(\xi) \in \{0,1\}; x_t(\xi), s_t(\xi) \ge 0, \forall \xi \in \Xi \quad (21e)$$

where the decision variables $x_t, s_t,$ and $y_t$ denote production level, inventory level, and setup indicator at period $t = 1,\ldots T$, the parameters $\alpha_t, \beta_t, h_t, d_t$ denote production cost, setup cost, inventory cost, and demand at period $t \in T$, the parameter $M$ denotes production capacity, and the parameter $p_\xi$ denotes the probability for each scenario $\xi \in \Xi$. Objective (21a) minimizes the total expected production, setup and inventory costs. Constraints (21b) enforce inventory balance conditions, (21c) enforce the production capacity limits, (21d) enforces no initial inventory, and (21e) enforce variable restrictions.

We populate data for MLS instances as in Guan et al. (2006). We generate a test instance with 4 stages and 5 branches emanating from each scenario tree node. The 5 branches from each scenario tree node are sampled from uniform distributions of stochastic parameters $d_t \sim U[0,100]$ with identical probabilities for each time period. The fixed input parameters are displayed in Table 4. The capacity is assigned to be 200.

Table 3.4 *Input parameters for MLS test instance*

| $t$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $h_t$ | 3 | 8 | 6 | 5 |
| $\alpha_t$ | 18 | 22 | 17 | 20 |
| $\beta_t$ | 99 | 91 | 102 | 108 |

Here, we performed computational studies on $\mathbb{E}$ -CVaR problems of (3-stage, 5-branch), (3-stage, 10-branch), (4-stage, 5-branch), and (4-stage, 10-branch) MLS instances with the upper and lower bounds for $\mathbb{E}$-CVaR variables and preselected probability set to be $\alpha_t = 0.2$ for each stage $t$. We perform multiple runs of the PH algorithm on this instance, varying the values of the penalty parameter $\rho$. Specifically, we consider fixed $\rho \in \left\{FX\left(10^{-3}\right), CP\left(10^{-3}\right), CP\left(10^{-4}\right)\right\}$ and record the time-series of the lower bound $D\left(w, w', w''\right)$ obtained at each PH iteration during each run. The lower bound results for various lot sizing instances are shown in Figure 2, which additionally displays the optimal solution value obtained from solving its extensive form.

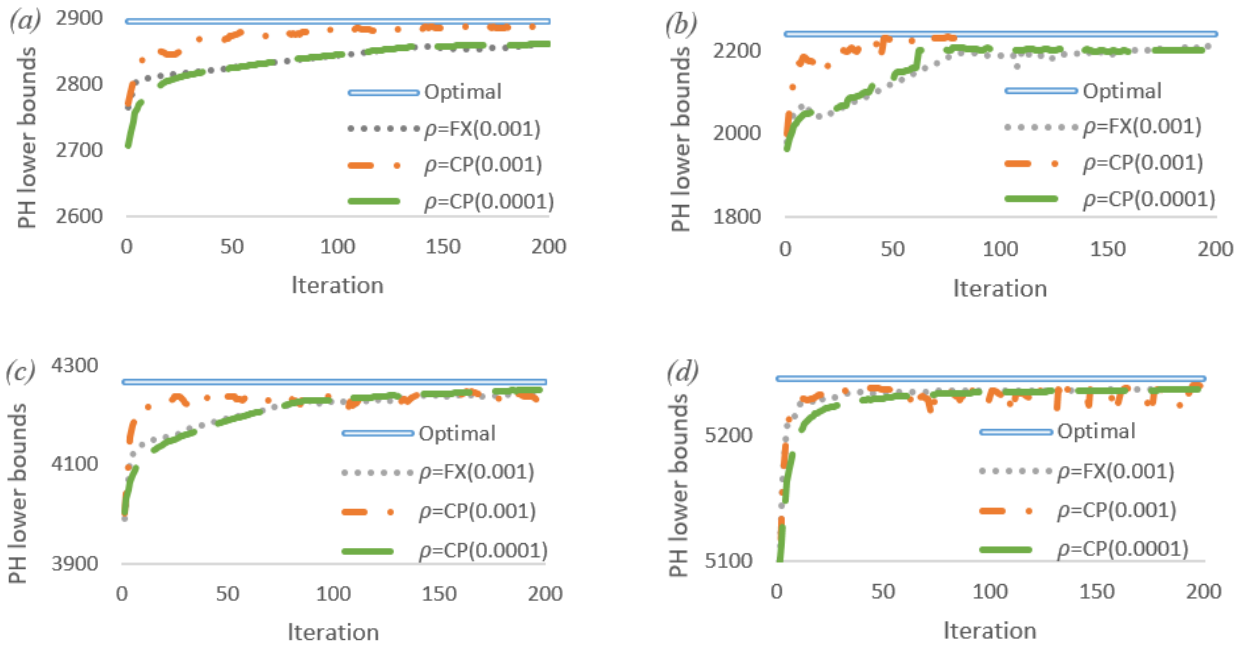


Figure 3.2 *Lower bounds from PH and optimal value from solving extensive form for MLS instances with (a) 3-stage, 5-branch; (b) 3-stage, 10-branch; (c) 4-stage, 5-branch; (d) 4-stage, 10-branch*

### 3.7.3 Generation Expansion Planning Problem

In a power generation expansion planning (GEP) problem, one seeks to determine a long-term construction and generation plan for different types of generators, taking into account the

uncertainties in future demand and fuel prices. Suppose there are $T$ time stages and $n$ types of expansion technologies available. Let $x_{it}$ represent the numbers of generators to be built for generator type $i$ in stage $t$, and $y_{it}$ represent the amount of electricity produced by generator type $i$ in stage $t$. The parameters $a_{it}, b_{it}$ denote investment and generation cost for generator type $i$ in stage $t$. The parameters $r_i, u_i, d_t$ denote the capacity rating of generator type $i$, the construction limits on generator type $i$, and the electricity demand at stage $t$. A multistage stochastic mixed-integer programming formulation of GEP problem is:

$$\min \sum_{\xi \in \Xi} p_{\xi} \sum_{t=1}^{T} \sum_{i=1}^{n} \left( a_{it} x_{it}(\xi) + b_{it}(\xi) y_{it}(\xi) \right) \tag{22a}$$

$$y_{it}(\xi) \le r_i \sum_{s=1}^{t} x_{is}(\xi), \forall i = 1, \ldots n, t = 1, \ldots T, \xi \in \Xi \tag{22b}$$

$$\sum_{t=1}^{T} x_{it}(\xi) \le u_i, \forall i = 1, \ldots n, \xi \in \Xi \tag{22c}$$

$$\sum_{i=1}^{n} y_{it}(\xi) \ge d_t(\xi), \forall t = 1, \ldots T, \xi \in \Xi \tag{22d}$$

$$x_{it} \in \mathbb{Z}_+, y_{it} \in \mathbb{R}_+, \forall i = 1, \ldots n, t = 1, \ldots T, \xi \in \Xi \tag{22e}$$

In the above formulation, objective (22a) minimizes the total expected investment cost and generation cost. Constraints (22b) enforce generation capacity, (22c) enforce the limitation on total number of generators, (22d) enforce demand satisfaction, and (22e) enforce variable restrictions.

We consider an instance of the GEP problem with a 10-year planning horizon where each year is considered as one period. There are 6 types of generators available for capacity expansion, namely Coal, Combined Cycle (CC), Combined Turbine (CT), Nuclear, Wind, and Integrated Gasification Combined Cycle (IGCC). Among these 6 types of generators, both CC

and CT power generators are fueled by natural gas. All the input parameters are deterministic except demand and natural gas price. We populate data for GEP problem as in Jin et al. (2011).

While Jin et al. (2011) consider a 10-period GEP instance as a two-stage problem, we consider a 10-period GEP instance as a 8-stage problem according to the division of the planning horizon and scenario tree generation in Feng et al. (2013). In our instance, each of the first six stages represent one period and each of the last two stages represent two periods. The stochastic parameters of demand and natural gas price are generated from two correlated geometric Brownian motions as in Jin et al. (2011). From each scenario tree node, 3 realizations of the pair of uncertain parameters and their probabilities are computed using moment matching method (Feng and Ryan 2013), thus leading to a large-scale mixed-integer problem with 2,187 scenarios. The data of fixed input parameters are obtained from GEP instance in Jin et al. (2011). We formulate its risk-averse $\mathbb{E}$-CVaR formulation with preselected probability set to be $\alpha_t = 0.2$ for each stage $t$, which has 244,944 variables and 205,578 constraints in total.

Due to the large number of scenarios and variables in this $\mathbb{E}$ -CVaR problem, its extensive form failed to compute an optimal objective with 48-hour time limit. To deal with this issue, the lower bounding approach from PH for risk-averse problems is employed here to compute a feasible solution with a reasonable optimality gap. In this instance, the objective cost coefficients of decision variables are in the unit of millions while the cost coefficients of the $\mathbb{E}$ E-CVaR related variables $\eta_t$ and $v_t$ are no greater than one. The unbalanced cost coefficients in objective prevent the PH algorithm from obtaining good variable-specific penalty parameters $\rho$, which significantly slows down its progress. Thus, additional variables $\lambda_t = 10^{-6}\eta_t$ and $u_t = 10^{-6}v_t$ are substituted for $\eta_t$ and $v_t$ in the E-CVaR optimization problem. In addition, the variable fixing and slamming strategies from Watson and Woodruff's PH extensions (2011) are

adopted here to accelerate PH convergence by forcing early agreement of variables at the expense of sub-optimal solutions. We fix decision variables once their value has stabilized to a fixed value over the past 3 iterations. After 10 iterations, we enforced slamming where a decision variable is fixed to its maximum solution across all scenarios for every 2 subsequent iterations.

Table 5 shows that while the extensive form failed to solve within 48 hours, the Progressive Hedging algorithm is able to provide a feasible solution with 3.2% optimality gap within 5 hours. The lower bounding approach allows us to assess the quality of feasible solutions generated by the algorithm by an upper bound on its optimality gap as the difference between the upper bound and lower bound.

Table 3.5 *PH run-time and optimality gap on 8-stage GEP instance with 48-hour time limit*

| | PH iterations | Run-time (hours) | Optimal objective (cost in thousand million dollars) | Lower bound | Upper bound | Optimality gap |
|---|---|---|---|---|---|---|
| Extensive Form | | 48 | N/A | N/A | N/A | N/A |
| Progressive Hedging | 12 | 6 | N/A | 3.41 | 3.45 | 1.2% |
| | 26 | 14 | N/A | 3.41 | 3.42 | 0.3% |

As the preselected probability $\alpha$ varies, the optimal solutions to the risk-averse programs change corresponding to optimize the expected values of costs in the $(1-\alpha)\cdot 100\%$ worst scenarios. Table 6 reports the best feasible solutions to first-stage decision variables for the E-CVaR problem of 8-stage GEP instance with different values of $\alpha$.

Table 3.6 *First-stage variable solutions for different values of* $\alpha$

| Number of generators to build by type | $\alpha$ values | | |
|---|---|---|---|
| | 0 | 0.2 | 0.8 |
| Baseload | 0 | 0 | 4 |
| CC | 0 | 0 | 0 |
| CT | 0 | 0 | 0 |
| Nuclear | 1 | 1 | 1 |
| Wind | 22 | 21 | 40 |
| IGCC | 0 | 0 | 4 |

Table 6 indicates that the optimal solutions may vary significantly according to the preselected probability and the optimal solutions to risk-neutral models do not necessarily guarantee best performance for risk-averse models.

### 3.8 Conclusions

We have proposed the scenario decomposition reformulations of multistage risk-averse stochastic programs with a variety of ECRMs. Based on the scenario reformulation, we presented a lower bounding approach from the PH algorithm. We discussed strategies for choosing the PH $\rho$ parameter and applied the scenario bundling strategy to help improve the quality of the PH lower bounds. Computing lower bounds for the PH algorithm allows us to assess the quality of the solutions generated by PH algorithm and also integrate with exact algorithms that rely on lower bounds. The integration of this lower bounding approach for risk-averse models with other exact algorithms remains as a promising area for potential future research. We also provided a remedy for the issue of unbounded optimization in the lower bounding problems introduced by $\mathbb{E}$-CVaR. Numerical results indicate that this lower bounding approach obtains convergent and tight lower bounds and displays its advantage in solving near-optimal solutions within reasonable run-time for large-scale stochastic problems whose extensive form fails to solve.

## REFERENCES

Ahmed, S. (2006). Convexity and decomposition of mean-risk stochastic programs. Mathematical Programming, 106 (3), 433-446.

Bion-Nadal, J. (2008). Dynamic risk measures: time consistency and risk measures from BMO martingales. Finance and Stochastics, 12(2), 219-244.

Burhaneddin, S., & Özaltın, O. Y. (2014) A scalable bounding method for multi-stage stochastic integer programs. Chicago Booth Research Paper, No. 14-21.

Carøe, C. C., & Schultz, R. (1999). Dual decomposition in stochastic integer programming. Operations Research Letters, 24(1-2), 37-45.

Carpentier, P., Chancelier, J.P., Cohen, G., Lara, M.D., & Girardeau, P. (2012). Dynamic consistency for stochastic optimal control problems. Annals of Operations Research, 200(1), 247-263.

Cheridito, P., Delbaen, F., & Kupper, M. (2006). Dynamic monetary risk measures for bounded discrete-time processes. Electronic Journal of Probability, 11(3), 57-106.

Collado, R. A., & Papp, D. (2011). Scenario decomposition of risk-averse multistage stochastic programming problems. Annals of Operations Research, 200(1), 147-170.

Dantzig, G. B., & Infanger, G. (1993). Multi-stage stochastic linear programs for portfolio optimization. Annals of Operations Research, 45, 59-76.

Detlefsen, K., & Scandolo, G. (2005). Conditional and dynamic convex risk measures. Finance and Stochastics, 9(4), 539-561.

Fabian, C. I. (2008). Handling CVaR objectives and constraints in two-stage stochastic models. European Journal of Operations Research, 191(3), 888-911.

Fan, Y., & Liu, C. (2010). Solving stochastic transportation network protection problem using the progressive hedging-based method. Networks and Spatial Economics, 10(2), 193-208.

Feng, Y., & Ryan, S. M. (2013). Scenario construction and reduction applied to stochastic power generation expansion planning. Computers & Operations Research, 40, 9-23.

Gade, D., Hackebeil, G., Ryan, S. M., Watson, J. P., Wets, R. J., & Woodruff, D. L. (2016). Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. Mathematical Programming, 157(1), 47-67.

Guan, Y., Ahmed, S., Nemhauser, G. L., & Miller, A. J. (2006). A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem. Mathematical Programming, 105(1), 55-84.

Guigues, V., & Sagastizábal, C. (2013). Risk-averse feasible policies for large-scale multistage stochastic linear programs. Mathematical Programming, 138(1-2), 167-198.

Guo, G., Hackebeil, G., Ryan, S. M., Watson, J. P., & Woodruff, D. L. (2015). Integration of progressive hedging and dual decomposition in stochastic integer programs. Operations Research Letters, 43(3), 311-316.

Homem-de-Mello, T., & Pagnoncelli, B. K. (2016). Risk aversion in multistage stochastic programming: a modeling and algorithmic perspective. European Journal of Operational Research, 249(1), 188-199.

Jin, S., Ryan, S. M., Watson, J. P., & Woodruff, D. L. (2011). Modeling and solving a large-scale generation expansion planning problem under uncertainty. Energy Systems, 2(3-4), 209-242.

Kovacevic, R., & Pflug, G. (2014). Are time consistent valuations information monotone? International Journal of Theoretical and Applied Finance, 17(1), 1443-1471.

Krokhmal, P., Palmquist, J., & Uryasev, S. (2002). Portfolio optimization with conditional value-at-risk objective and constraints. Journal of Risk, 4, 11-27.

Listes, O., & Dekker, R. (2005). A scenario aggregation based approach for determining a robust airline fleet composition. Transportation Science, 39, 367-382.

Miller, N., & Ruszczynski, A. (2011). Risk-averse two-stage stochastic linear programming: modeling and decomposition. Operations Research, 59(1), 125-132.

Mulvey, J. M., & Vladimirou, H. (1991). Applying the progressive hedging algorithm to stochastic generalized networks. Annals of Operations Research, 31, 399-424.

Noyan, N. (2012). Risk-averse two-stage stochastic programming with an application to disaster management. Computer and Operations Research, 39(3), 541-559.

Pereira, M., & Pinto, L. M. (1991). Multi-stage stochastic optimization applied to energy planning. Mathematical Programming, 52(1-3), 359-375.

Pflug, G. C., & Pichler, A. (2014). Multistage Stochastic Optimization. Springer Series in Operations Research and Financial Engineering.

Pflug, G. C., & Pichler, A. (2016). Time-inconsistent multistage stochastic programs: martingale bounds. European Journal of Operational Research, 249(1), 155-163.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. Journal of Risk, 2, 21-42.

Rockafellar, R. T., & Wets, R. J. B. (1991). Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research, 16(1), 119-147.

Rudloff, B., Street, A., & Valladão, D. (2014). On the economic interpretation of time consistent dynamic stochastic programming problems. European Journal of Operational Research, 234(3), 743-750.

Ruszczynski, A. (2010). Risk-averse dynamic programming for Markov decision processes. Mathematical Programming, 125(2), 235-261.

Ruszczynski, A., & Shapiro, A. (2006). Conditional risk mappings. Mathematics of Operations Research, 31(3), 544-561.

Schultz, R. (2011). Risk aversion in two-stage stochastic integer programming. In G. Infanger (Ed.), Stochastic Programming (pp. 165-187). New York: Springer.

Schultz, R., & Tiedemann, S. (2006). Conditional value-at-risk in stochastic programs with mixed-integer recourse. Mathematical Programming, 105(2), 365-386.

Shapiro, A. (2009). On a time consistency concept in risk averse multistage stochastic programming. Operations Research Letters, 37(37), 143-147.

Watson, J. P., & Woodruff, D. L. (2011). Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. Computational Management Science, 8(4), 355-370.

Watson, J. P., Woodruff, D. L., & Hart, W. E. (2012). PySP: modeling and solving stochastic programs in Python. Mathematical Programming Computation, 4(2), 109-149

Wets, R. J. B. (1989). The aggregation principle in scenario analysis and stochastic optimization. In Wallace, S.W. (Ed.), Algorithms and Model Formulations in Mathematical Programming (pp. 91-113). Berlin: Springer.

Zou, J., Ahmed, S., & Sun, X. A. (2016). Stochastic dual dynamic integer programming. Submitted for publication. http://www.optimization-online.org/DB_HTML/2016/05/5436.html.

IBM ILOG CPLEX Optimization Studio V12.6, Available at: https://www.ibm.com/products/ilog-cplex-optimization-studio

# CHAPTER 4. TIME CONSISTENT MULTISTAGE RISK-AVERSE STOCHASTIC MIXED-INTEGER PROGRAMMING APPLIED TO A MIXED-MODEL ASSEMBLY LINE SEQUENCING PROBLEM


## Abstract

The existing optimization formulations for mixed-model assembly line sequencing (MMALS) problems consider stochastic demand and task times, but neglect many real-world uncertainty factors such as timely part delivery, material quality, upstream sub-assembly completion and availability of other resources. Real-time resequencing decisions are required to deal with the inevitable disruptions. We present a multistage stochastic mixed-integer program for a sequencing problem to increase on-time performance considering part availability uncertainty. A risk-averse sequencing problem can be modeled to further improve on-time performance in the worst scenarios for decision makers more concerned about large losses than average performance. Computational studies are performed on a set of MMALS test cases with Progressive Hedging (PH) as the solution heuristic in the context of real-time resequencing. A lower bounding approach for time consistent risk-averse models is applied to evaluate the quality of the PH solutions.


*Keywords*: Mixed-model assembly line sequencing; Stochastic mixed-integer programming; Risk-averse optimization; Time consistency; Progressive Hedging; Lower bounds

## 4.1 Introduction

As an important part of the just-in-time production system, a mixed-model assembly line manufacturing system makes it possible to produce a variety of models of the same basic product on the same production line. The optimal design and operation of mixed-model lines must address a long-term line balancing problem to assign tasks to a sequence of stations and a short-term model sequencing problem to determine the production sequence of models within a planning horizon. In this paper, we are concerned with the model sequencing problem. A mixed-model assembly line

comprises a series of stations and a conveyor moving at a constant speed, which can assemble different models of the same basic product during a working shift. The model sequencing problem is to determine the sequence that specifies the feeding order of the models at the beginning of each working shift. This is a very complicated decision due to the large number of orders of models from customers, each with a specific due time and various requirements for parts and resources at different stations.

The development of mixed-model assembly lines has stimulated recent studies that address optimization formulations for model sequencing problems. Manavizadeh et al. (2015) proposed an optimization formulation and a heuristic solution method for both balancing and sequencing problems to minimize the cycle times, the wastages in stations and the work overload. Dörmer et al. (2015) presented an optimization model for master production scheduling problems in mixed-model assembly lines in the automotive industry to minimize the workload variability. Rabbani et al. (2018) developed a bi-objective optimization model to determine a sequence of products that minimizes the total cost and maximizes levels of customer satisfaction. In addition, some studies concern optimization models with a variety of uncertainties. Zhao (2006) proposed an optimization formulation for the mixed-model assembly line sequencing problem to minimize the expected cost including the inventory cost and the backorder cost in consideration of stochastic demand. Boysen et al. (2009) discussed three major sequencing approaches including mixed-model sequencing, car sequencing and level scheduling considering stochastic demand and task times. Dong (2014) presented a stochastic programming formulation to minimize the expected work overload time for mixed-model assembly U-lines with stochastic task times.

Despite the efforts in decision making for mixed-model sequencing with uncertainties, the existing formulations do not model many real-world uncertainty factors including timely part

delivery, material quality, upstream sub-assembly completion and availability of other resources. To incorporate the uncertainty of part delivery and material quality, we formulate a multi-stage stochastic mixed-integer program with a probabilistic model for part availability. To increase on-time performance, our objective function minimizes both lateness and earliness of the final products' finish times given their due time targets. The formulation of this optimization model to make real-time resequencing decisions is part of a project to develop a shop floor decision support system. This optimization model could serve as a decision-making tool in the sequencing module of this decision support system. Since this sequencing model accounts for all the possible outcomes of material availability, it effectively saves the time to solve the sequencing problem once again whenever a part availability issue arises, providing the shop floor with the optimal resequencing decisions in real time in all circumstances. Consequently, this model not only offers the sequencing decisions leading to optimal overall on-time performance, but could also help to prevent the assembly line from shutting down when required materials turn out to be unavailable.

For sequencing decision makers who are more concerned with on-time performance in worst scenarios rather than average performance, risk-averse models are preferred to risk-neutral models. Risk-averse stochastic programs have not been explored in the existing literature for mixed-model assembly line sequencing problems to our knowledge. In addition to the risk-neutral sequencing model, therefore, we present the risk-averse problem with Conditional Value-at-Risk (CVaR) adopted as our risk measure. With the CVaR risk measure, our risk-averse model is able to provide best resequencing decisions for the worst possible outcomes regarding the part availability, which helps decision makers to guarantee customers' satisfaction regarding on-time performance.

The Progressive Hedging (PH) algorithm is a scenario decomposition algorithm initially developed by Rockafellar and Wets (1991) for stochastic programs with continuous decision variables. It was further adapted as an effective heuristic solution method to solve stochastic mixed-integer programs by Watson and Woodruff (2011). The PH algorithm is adopted as our solution approach not only because of its capability to solve multi-stage models, but also due to its computational efficiency when dealing with many possible realizations of uncertainties. In many applications where real-time decisions are valued, such as model sequencing in manufacturing systems, near-optimal solutions are desired within a limited time frame. Gade et al. (2016) made it possible to evaluate the quality of solutions at each iteration of the PH algorithm by developing a lower bounding approach. This lower bounding method, however, is restricted to risk-neutral models. Guo and Ryan (2017) extended this lower bounding technique to obtain convergent and tight lower bounds for certain time consistent multistage risk-averse stochastic mixed-integer programs. In our computational study, we apply this lower bounding approach to a set of time consistent risk-averse MMALS instances. The numerical results show that convergent and tight lower bounds are found and that near-optimal feasible solutions can be found in real time especially for large-scale instances whose extensive form fails to solve directly. This PH algorithm together with the lower bounding method could be employed within our decision support system to make real-time resequencing decisions for the shop floor that can significantly reduce the downtime of assembly lines.

The remainder of this paper is organized as follows. In section 2, we formulate a deterministic optimization program for mixed-model assembly line sequencing problem to maximize on-time performance. Based on the deterministic model, in section 3, we develop its stochastic version with part availability uncertainty taken into consideration and further propose

its risk-averse stochastic version with CVaR risk measure to find optimal sequencing decisions in the worst scenarios. In section 4, we introduce a lower bounding approach to assess solution qualities for time consistent risk-averse stochastic programs. Computational studies are performed on a set of MMALS instances with their numerical results provided in section 5.

### 4.2 Formulation of Mixed-Model Assembly Line Sequencing Problem

We formulate a mixed-integer programming program for mixed-model assembly lines to make sequencing decisions. The objective of our model is to minimize both lateness and earliness of meeting customers' deadlines for finished assembly products. While it is obvious that the lateness must be avoided, completing the products too early also leads to undesirable inventory cost.

In our formulation, we schedule the sequencing of assembly units ordered by customers with given deadlines. We look for the optimal decisions on which unit to assemble at each assembly station at each period. It is assumed that the takt time, the rate at which a product needs to be completed to meet customer demand, keeps constant and we take one takt time as the length of one period. We also assume that the cycle time, the total time it takes to process one assembly unit at one station, is equal to the takt time. All the assembly units proceed throughout the assembly line from station to station in the same sequence. Each assembly station can either process one assembly unit at each period or stay idle. Each assembly unit can be processed at one assembly station at each period. If one assembly unit is ready to proceed to the next station while the next station is still busy or the required parts on the next station are not available yet, then this assembly unit will be pulled off the assembly line and wait beside the current station. The finish time for each assembly unit is defined as the time when this unit leaves the final assembly station. The length of earliness and lateness for each assembly unit is measured by the positive deviation and the negative deviation, respectively, between its finish time and its due time.

The deterministic model for our mixed-model assembly line sequencing problem is formulated as follows:

*Sets:*

$N$ : set of assembly units to be sequenced with $n = 1, \cdots, N$

$P$ : set of parts with $p = 1, \cdots, P$

$T$ : set of time periods with $t = 1, \cdots, T$ , where continuous time $t$ denotes the end of discrete period $t$

$S$ : set of assembly stations with $s = 1, \cdots, S$

*Decision variables:*

$v_{n,s,t}$ : binary variable equal to 1, if unit $n$ is being processed at station $s$ at time period $t$; and 0, otherwise

$u_{n,s,t}$ : binary variable equal to 1, if unit $n$ is pulled offline and waiting at station $s$ at time period $t$; 0, otherwise

$a_{p,s,t}$ : the number of good parts $p$ available at station $s$ at time period $t$

$f_n$ : finish time when unit $n$ leaves the last assembly station

$e_n$ : earliness of unit $n$

$l_n$ : lateness of unit $n$

*Input parameters:*

$D_n$ : due time when finished unit $n$ is required by downstream operations or customers

$L_{p,s,t}$ : number of good parts $p$ delivered at station $s$ at time period $t$

$R_{p,n,s}$ : number of parts $p$ consumed by unit $n$ at station $s$

$C_e$ : penalty for earliness ($/unit time)

$C_l$ : penalty for lateness ($/unit time)

*Objective:*

$$\min C_e \sum_{n=1}^{N} e_n + C_l \sum_{n=1}^{N} l_n \tag{1.1}$$

*Constraints:*

$$\sum_{n=1}^{N} v_{n,s,t} \le 1, \forall s = 1, \cdots, S, t = 1, \cdots, T \tag{1.2}$$

$$\sum_{s=1}^{S} \left( v_{n,s,t} + u_{n,s,t} \right) = 1, \forall n = 1, \cdots, N, t = 1, \cdots, T \tag{1.3}$$

$$\sum_{t=1}^{T} v_{n,s,t} = 1, \forall n = 1, \cdots, N, s = 1, \cdots, S \tag{1.4}$$

$$\begin{aligned} & v_{n,1,1} = 1, \forall n = 1, \cdots, N, v_{n,s,1} = 0, \forall n = 1, \cdots, N, s = 2, \cdots, S \\ & v_{n,s,t} + u_{n,s,t} = v_{n,s+1,t+1} + u_{n,s,t+1}, \forall n = 1, \cdots, N, s = 1, \cdots, S-1, t = 1, \cdots, T-1 \end{aligned} \tag{1.5}$$

$$\begin{aligned} & \sum_{n=1}^{N} \left( R_{p,n,s} v_{n,s,t} \right) \le a_{p,s,t}, \forall p = 1, \cdots, P, s = 1, \cdots, S, t = 1, \cdots, T \\ & a_{p,s,t} = a_{p,s,t-1} + L_{p,s,t} - \sum_{n=1}^{N} \left( R_{p,n,s} v_{n,s,t} \right), \forall p = 1, \cdots, P, s = 1, \cdots, S, t = 2, \cdots, T \end{aligned} \tag{1.6}$$

$$\sum_{t=1}^{T} \left( t v_{n,s,t} \right) = f_n, \forall n = 1, \cdots, N, s = S \tag{1.7}$$

$$f_n + e_n \ge D_n, \forall n = 1, \cdots, N \tag{1.8}$$

$$f_n - l_n \le D_n, \forall n = 1, \cdots, N \tag{1.9}$$

$$\begin{aligned} & v_{n,s,t}, u_{n,s,t} \in \{0,1\}, a_{p,s,t} \in \mathbb{N}, \forall n = 1, \cdots, N, s = 1, \cdots, S, t = 1, \cdots, T \\ & f_n, e_n, l_n \in \mathbb{R}^+, \forall n = 1, \cdots, N \end{aligned} \tag{1.10}$$

The objective (1.1) is to minimize the weighted total earliness and lateness over all the finished products. Constraints (1.2) enforce that each assembly station at each time period can either process one assembly unit or stay idle. Constraints (1.3) require that each assembly unit at each time period can stay at only one assembly station, either being processed or waiting offline.

Constraints (1.4) restrict that each assembly unit is processed for exactly one time period at each assembly station. Constraints (1.5) ensure that all assembly units proceed throughout all the stations on the assembly line in the same sequence. Constraints (1.6) state the availability balance condition and restrict the consumption of parts or resources at each assembly station to not exceed the availability at each time period. The variable of finish time for each assembly unit is defined in constraints (1.7) as the time when the unit leaves the last assembly station. The variables of earliness and lateness for each assembly unit are defined in constraints (1.8) and (1.9), respectively, as the positive deviation and the negative deviation between its finish time and its due time. Constraints (1.10) enforce restrictions on the decision variables.

In the above formulation, the input parameters for the numbers of delivered good parts $L_{p,s,t}$ are considered to be deterministic with assumptions of timely part delivery and flawless material quality. In the real world, however, the quantities of delivered good parts $\tilde{L}_{p,s,t}$ are random variables as a result of delay of deliveries and quality defects with stochastic values that are generally smaller than the deterministic parameters $L_{p,s,t}$ in formulation (1). Therefore, the optimal solutions yielded from solving the deterministic problem (1) might turn out to be infeasible given the actual number of available parts with delivery delays or quality issues. As a consequence, the problem (1) will need to be resolved with the updated information on the number of available parts, leading to the downtime of assembly lines while updating the sequencing decisions. Thus, it would be desirable to have initial solutions that are feasible with various extents of delays of deliveries and quality defects to optimize the overall on-time performance at the beginning of each working shift.

## 4.3 Time Consistent Multistage Risk-Averse Stochastic Mixed-Integer Formulation

### 4.3.1 Multistage Risk-Neutral Programs for Model Sequencing Problem

For a multistage stochastic program with $\tau$ time stages, we use $\xi = (\xi_2, \cdots, \xi_\tau)$ to denote the uncertain parameters with known probability distributions and use $x = (x_1, \cdots, x_\tau)$ to denote the decision vectors. The realization of uncertainty $\xi_t$ at stage $t = 2, \cdots, \tau$ is not revealed until the decisions $x_{t-1}$ are made. We use $\xi_{[t]} = (\xi_2, \cdots, \xi_t)$ to denote the history of the realizations up to stage $t$ and use $x_1, \xi_2, x_2(x_1, \xi_2), \xi_3, x_3(x_1, x_2, \xi_{[3]}), \ldots, \xi_\tau, x_\tau(x_1, \ldots, x_{\tau-1}, \xi_{[\tau]})$ to represent the sequence of decisions and realizations of uncertainties.

We represent the risk-neutral multistage stochastic mixed-integer program as follows:

$$z = \min_{x_1, \ldots, x_\tau} \left\{ c_1^T x_1 + \mathbb{E}_{\xi_2} \left[ Q_2 \left( x_1, \xi_{[2]} \right) \right] : A x_1 = b, x_1 \in \mathbb{Z}_+^{q_1} \times \mathbb{R}_+^{k_1 - q_1} \right\} \tag{2}$$

For $t = 2, \cdots, \tau$, $Q_t(x_{t-1}, \xi_{[t]})$ is defined recursively as:

$$\begin{aligned} & Q_t \left( x_{t-1}, \xi_{[t]} \right) = \min_{x_t} \left\{ c_t^T \left( \xi_{[t]} \right) x_t + \mathbb{E}_{\xi_{t+1} | \xi_{[t]}} \left[ Q_{t+1} \left( x_t, \xi_{[t+1]} \right) \right] \right\} \\ & T_t \left( \xi_{[t]} \right) x_{t-1} + W_t \left( \xi_{[t-1]} \right) x_t = h_t \left( \xi_{[t]} \right) \\ & x_t \in \mathbb{Z}_+^{q_t} \times \mathbb{R}_+^{k_t - q_t} \end{aligned}$$

Here $c_1 \in \mathbb{R}^{k_1}, b \in \mathbb{R}^{j_1}$ and $c_t(\xi_{[t]}) \in \mathbb{R}^{k_t}, h_t(\xi_{[t]}) \in \mathbb{R}^{j_t}$ are given vectors, while $A \in \mathbb{R}^{j_1 \times k_1}$ and $T_t(\xi_{[t]}) \in \mathbb{R}^{j_t \times k_{t-1}}, W_t(\xi_{[t-1]}) \in \mathbb{R}^{j_t \times k_t}$ are given matrices. The decisions are non-anticipative in the sense that a decision made in stage $t$ can depend on information revealed before that stage but not after.

The notation $\mathbb{E}_{\boldsymbol{\xi}}$ denotes expectation with respect to the distribution of random variable $\boldsymbol{\xi}$. To avoid complications when computing the integral behind $\mathbb{E}_{\boldsymbol{\xi}}$ we assume that we have only a finite number of realizations $\xi$ with corresponding probabilities $p_\xi$. In a scenario tree, we use $o_t$ to denote a scenario tree node that belongs to the set of all scenario tree nodes $O_t$ at stage

$t = 1, \cdots, \tau$. We use $\xi(o_t)$ to represent a scenario that belongs to the set of scenarios $\Xi(o_t)$ that define the node $o_t \in O_t$. We use $o_t(\xi)$ to denote the corresponding tree node for scenario $\xi \in \Xi$ at stage $t = 1, \cdots, \tau$. We use $\hat{x}(o_t(\xi))$ as the non-anticipative decision made at scenario tree node $o_t(\xi)$.

Then problem (2) can decompose by scenario and be represented as its scenario reformulation:

$$\begin{aligned} z = \min_{x_1,\ldots,x_\tau,\hat{x}} \Bigg\{ & \sum_{\xi \in \Xi} p_\xi \left[ c_1^T x_1 + \sum_{t=2}^{\tau} c_t^T \left( \xi_{[t]} \right) x_t \left( \xi \right) \right] : \\ & x_t \left( \xi \right) \in X_\xi, \, p_\xi x_t \left( \xi \right) - p_\xi \hat{x} \left( o_t(\xi) \right) = 0, \forall \xi \in \Xi, \forall t = 1, ..., \tau \Bigg\} \end{aligned} \tag{3}$$

where

$$X_\xi = \begin{Bmatrix} x_t \left( \xi \right) : A x_1 = b, x_t \left( \xi \right) \in \mathbb{Z}_+^{q_1} \times \mathbb{R}_+^{k_1 - q_1}, \forall t = 1, \cdots, \tau \\ T_t \left( \xi_{[t]} \right) x_{t-1} \left( \xi \right) + W_t \left( \xi_{[t-1]} \right) x_t \left( \xi \right) = h_t \left( \xi_{[t]} \right), \forall \xi \in \Xi, \forall t = 2, \cdots, \tau \end{Bmatrix} \tag{4}$$

Problem (3) can further decompose into scenario sub-problems

$\min_{x_1,\ldots,x_\tau} \left\{ c_1^T x_1 + \sum_{t=2}^{\tau} c_t^T \left( \xi_{[t]} \right) x_t : x_t \in X_\xi, \forall t = 1, \cdots, \tau \right\}$ for scenarios $\xi \in \Xi$ which are coupled by the non-anticipativity constraints $p_\xi x_t \left( \xi \right) - p_\xi \hat{x} \left( o_t(\xi) \right) = 0$.

Our deterministic formulation (1) assumes the part delivery is timely and no quality defects are present, that is, the information of part delivery $L_{p,s,t}$ is known with certainty. In the real world, however, the availability of parts at each period may not be revealed until after the sequencing decisions are made for that period. In order to make sequencing decisions in consideration of uncertainty from part delivery, this formulation is extended to a multistage stochastic program according to formulation (3) where each period defines one decision stage. In our multistage model, each scenario specifies a possible realization of part delivery over the schedule horizon with

marginal probability distribution $f_{p,s,t}(l) = \Pr(L_{p,s,t} = l)$ where scalar $l$ represents the number of good parts $p$ delivered at station $s$ at stage $t$. The sequencing decisions $x_t$ made at each stage are made with the part delivery information of the current and previous periods $\xi_{[t]}$ and previous decisions $(x_1, \ldots, x_{t-1})$. The delivery information of parts for the next periods is not revealed until after the sequencing decisions at the current stage are made. The scenario tree node $o_t(\xi)$ represent the realization of part delivery information for scenario $\xi \in \Xi$ at stage $t$. A set of non-anticipativity constraints $x_t(\xi) = \hat{x}(o_t(\xi))$ for all scenario tree nodes $o_t(\xi)$ is essential in our multistage formulation to enforce that all the decisions tied to the same scenario tree node are identical and cannot account for any information that has not been revealed yet. For a $\tau$-stage problem, note the decision variables $x_t$ in our multistage model include stage $t$ decision variables $v_{n,s,t}, u_{n,s,t}, a_{p,s,t}, \forall n = 1, \cdots, N, s = 1, \cdots, S$ when $t < \tau$ and include last stage decision variables $v_{n,s,\tau}, u_{n,s,\tau}, a_{p,s,t}, \forall n \in N, s = 1, \cdots, S$ and $f_n, e_n, l_n \in \mathbb{R}^+, \forall n \in N$ when $t \geq \tau$. The feasible region $X_\xi$ corresponds to constraints (1.2) - (1.10) given the realization $\xi$ for $\tilde{L}_{p,s,t}, \forall p = 1, \cdots, P, s = 1, \cdots, S, t = 1, \cdots, \tau$.

Compared to the deterministic model (1), the multistage stochastic version manages to account for all the possible realizations of part delivery and yields the sequencing decision for each realization resulting in the optimal expected on-time performance over all the scenarios. This makes the stochastic optimization model an effective real-time sequencing decision-making tool for a shop floor decision support system. While the deterministic model could yield infeasible solutions when required materials are unavailable or have quality issues, leading to the downtime

of assembly lines, the stochastic model considers all the possible scenarios and offers sequencing decisions in all circumstances.

### 4.3.2 Time Consistent Multistage Risk-Averse Programs with CVaR Risk Measure

The multistage formulation in equation (2) is risk-neutral such that the expected performance across all scenarios is minimized. On the other hand, there are many applications where decision makers are more interested in worst events that might happen. In manufacturing systems such as mixed-model assembly lines, the schedulers are more concerned with the on-time performance in the worst scenarios of part availability to guarantee the customers' satisfaction regarding the timely delivery of finished products, especially when the delivery of parts is very unpredictable. For such decision makers concerned about the worst scenarios, a risk-averse model is preferred. However, one key issue of time consistency arises when measuring risk in multistage models. Among various definitions of time consistency, we use the definition of time consistency from Homem-de-Mello and Pagnoncelli (2016) such that given the optimal solutions from previous stages, resolving the problem results in the same solutions for the later stages if the optimal solutions are unique or gives the same optimal objective otherwise. In general, this property of time consistency is not guaranteed for multistage stochastic programs and depends on how the risk measure is computed. Homem-de-Mello and Pagnoncelli (2016) proposed a class of expected conditional risk measures (ECRMs) which prove to be time consistent and, for some risk measures, allow for risk-neutral reformulations. Guo and Ryan (2017) presented scenario reformulations for multistage risk-averse models with ECRMs that allow for the application of scenario decomposition solution algorithms to efficiently solve the risk-averse problems. In the application of mixed-model assembly line sequencing problems where decision makers are concerned with costs in worst cases, we focus on Conditional Value-at-Risk (CVaR) as a coherent

risk measure to measure the expected value of earliness and lateness in the worst $100(1-\alpha)\%$ scenarios given a probability $\alpha$. We use $\mathbb{E}$-CVaR to denote the multistage risk-averse program with the expected conditional CVaR risk measure.

Here, we restate the scenario reformulation of $\mathbb{E}$-CVaR optimization from Guo and Ryan (2017) in Proposition 3 to demonstrate how $\mathbb{E}$-CVaR optimization can be computed in a deterministic equivalent formulation.

***Proposition 3.*** *Consider the case with finitely many realizations $\xi$ and corresponding probabilities $p_\xi$. Let $\alpha \in (0,1)$. Then the scenario reformulation of $\mathbb{E}$-CVaR optimization can be represented as:*

$$z_{\mathbb{E}-CVaR_\alpha} = \min_{\substack{x_1,\ldots,x_\tau,\hat{x} \\ \eta_2,\ldots,\eta_\tau, \\ v_2,\ldots,v_\tau}} \left\{ \begin{array}{l} \sum_{\xi\in\Xi} p_\xi \left[ c_1^T x_1 + \eta_2(\xi) + \frac{1}{1-\alpha_2} v_2(\xi) \cdots + \eta_\tau(\xi) + \frac{1}{1-\alpha_\tau} v_\tau(\xi) \right] : \\ \eta_t(\xi) + v_t(\xi) \geq c_t^T(\xi_t) x_t(\xi), \forall t = 2,\cdots,\tau \\ x_t(\xi) \in X_\xi,\ p_\xi x_t(\xi) - p_\xi \hat{x}(o(\xi_t)) = 0, \forall \xi \in \Xi, \forall t = 1,\cdots,\tau \end{array} \right\} \quad (5)$$

where $X_\xi$ is defined in (4).

### 4.4 Lower Bounding Approach for Time Consistent Risk-Averse Programs

#### 4.4.1 Progressive Hedging Algorithm

The PH algorithm is a scenario decomposition method developed by Rockafellar and Wets (1991) for stochastic linear programs. It demonstrates great advantage when dealing with a large number of scenarios by allowing decomposition of the large-scale extensive form into scenario sub-problems and optimization for each scenario in parallel. The computational burden can be further reduced by taking advantage of any special structures in the scenario sub-problems.

For a multistage stochastic mixed-integer program, a scenario solution is defined to be admissible to a specific scenario if it satisfies all the constraints in this scenario. A solution is

defined to be non-anticipative if its decisions tied to the same scenario tree node are identical. A solution is defined to be feasible if it is non-anticipative and all its scenario solutions are admissible to the corresponding scenarios. To find an optimal solution for a multistage stochastic program, the PH algorithm computes an aggregated solution at each scenario tree node from the admissible solutions of all the modified scenario sub-problems tied to the same scenario tree node and then updates the modified scenario sub-problems at each PH iteration. The aggregated solutions progressively converge to the optimal solution. The objective in each modified scenario sub-problem includes a penalty term for the non-anticipativity constraint and a proximal term to represent the squared deviation of its admissible solution for this scenario from the aggregated solution across all the scenarios. The weight vector $w \in \mathbb{R}^{k_1+k_2+\cdots+k_{\tau-1}}$ in modified scenario sub-problems is updated by the penalty vector $\rho > 0$ at each iteration.

The PH algorithm was further adapted by Watson and Woodruff (2011) to solve stochastic mixed-integer programs as an effective heuristic algorithm. Though its optimality is not guaranteed in the mixed-integer case, computational studies demonstrate that it finds high-quality solutions within a reasonable number of iterations (Watson and Woodruff 2011). We restate the PH algorithm for multistage stochastic mixed-integer programs from Gade et al. (2016):

*STEP 1* Initialization: Let $v := 0$ and $w^v(\xi) := 0, \forall \xi \in \Xi$. For each $\xi \in \Xi$, compute

$$x_t^{v+1}(\xi) := \arg\min \left\{ c_1^T x_1 + \sum_{t=2}^{\tau} c_t^T(\xi) x_t : x_t \in X_\xi \right\}$$

*STEP 2* Iteration counter update: $v \leftarrow v+1$

*STEP 3* Non-anticipative policy: $\hat{x}_t^v(o_t(\xi)) := \sum_{\xi(o_t)\in\Xi(o_t)} p_\xi x_t^v(\xi) \Big/ \sum_{\xi(o_t)\in\Xi(o_t)} p_\xi$

*STEP 4* Weight update: For each $\xi \in \Xi$ , compute

$$w_t^v(\xi) := w_t^{v-1}(\xi) + \rho\left(x_t^v(\xi) - \hat{x}_t^v(o_t(\xi))\right)$$

*STEP 5* Decomposition: For each $\xi \in \Xi$ , compute

$$x_t^{v+1}(\xi) := \arg\min\left\{c_1^T x_1 + \sum_{t=2}^{\tau} c_t^T(\xi) x_t + \sum_{t=1}^{\tau-1}\left[w_t^v(\xi)^T x_t + \frac{\rho}{2}\left\|x_t - \hat{x}_t^v(o_t(\xi))\right\|^2\right] : x_t \in X_\xi\right\}$$

*STEP 6* Termination: If all the scenario solutions are identical to within some tolerance at each tree node, then stop. Otherwise, return to Step 2.

The performance of the PH algorithm with various values of penalty parameter $\rho$ with a single scalar for all variables has been extensively studied in the literature (Mulvey and Vladimirou 1991; Listes and Dekker 2005; Fan and Liu 2010). Computational studies (Watson and Woodruff 2011; Gade et al. 2016) show that larger values of $\rho$ tend to speed up the convergence of PH but the upper bound may be bad and PH may converge to a non-optimal solution. The smaller values of $\rho$ generally slow down the convergence of PH but are more likely to result in high-quality solutions, and tight lower bounds.

### 4.4.2 Progressive Hedging Lower Bounds on Stochastic Mixed-Integer Programs

The PH algorithm has demonstrated its high-quality solutions and computational efficiency in a variety of applications of stochastic mixed-integer programs including power systems (Cheung et al. 2015, Ordoudis et al. 2015), network design (Crainic et al. 2011), lot-sizing (Haugen et al. 2001) and production scheduling (Lamghari et al. 2016). However, its optimality is not guaranteed and the algorithm itself lacks the capability to assess the quality of PH solutions. Gade et al. (2016) effectively addressed this deficiency by developing a lower bounding approach for the PH algorithm to evaluate its solution quality at each iteration. This

approach also allows the integration of PH with some exact solution algorithms for the two-stage problem such as dual decomposition that rely on lower bounds (Guo et al. 2015).

Here, we elaborate the extension of the lower bounding method proposed by Gade et al. (2016) for multistage problems in Proposition 4.1. Let $z^*$ denote the optimal objective function value of the problem (2) and (3). The weight vector $w$ defines an implicit lower bound $D(w)$ on $z^*$.

***Proposition 4.1.*** *Let* $w=\left(w(\xi)\right)_{\xi\in\Xi}$ *and* $w(\xi)=\left(w_t(\xi)\right)_{t=1,\cdots\tau}$ *where* $w_t(\xi)\in\mathbb{R}^{k_t}$ *satisfy* $\sum_{\xi(o_t)\in\Xi(o_t)} p_\xi w_t(\xi)=0$ *for each* $o_t\in O_t$. *Let*

$$D_\xi\left(w(\xi)\right):=\min\left\{c_1^T x_1+\sum_{t=2}^{\tau}c_t^T(\xi)x_t+\sum_{t=1}^{\tau-1}w_t^T(\xi)x_t : x_t\in X_\xi\right\} \tag{6}$$

*Then* $D(w):=\sum_{\xi\in\Xi}p_\xi D_\xi\left(w(\xi)\right)\le z^*$.

It is well known that convergence of PH can be accelerated by forming bundles of scenarios (Wets 1989). Instead of solving each scenario sub-problem, the bundle version of the PH algorithm solves smaller extensive forms of the original problem in steps 1 and 5 of the PH algorithm.

Here, we elaborate the bundle version of PH lower bounding method by Gade et al. (2016) for two-stage programs extended to multistage models in Proposition 4.2. Suppose the set of all the scenario tree nodes $O_t$ at stage $t=2,\cdots,\tau-1$ is partitioned into bundles $\beta_t$. We denote the set of bundles by $B_t$, with $\beta_t\in B_t$. Let $P_{\beta_t}=\sum_{\xi\in\Xi(\beta_t)}p_\xi$ and $X_{\beta_t}=\left\{x_t(\xi):\xi\in\beta_t\right\}$. Although the scenario tree nodes can be bundled at multiple stages, here we consider bundling only once at a single stage $t$.

***Proposition 4.2.*** *Let* $w=\left(w\left(\beta_t\right)\right)_{\beta_t\in B_t}$ *where* $w\left(\beta_t\right)\in\mathbb{R}^{k_t}$ *satisfies* $\sum_{\beta_t\in B_t} P_{\beta_t} w\left(\beta_t\right)=0$. *Let*

$$D_{\beta_t}\left(w\left(\beta_t\right)\right):=\min_x\left\{c_1^T x_1+\sum_{t=2}^{\tau}\sum_{\xi\in\Xi\left(\beta_t\right)}\frac{p_\xi}{P_{\beta_t}}c_t^T\left(\xi\right)x_t+\sum_{t=1}^{\tau-1}w^T\left(\beta_t\right)x_t : x_t\in X_{\beta_t}\right\} \tag{7}$$

*Then* $D(w):=\sum_{\beta_t\in B_t} P_{\beta_t} D_{\beta_t}\left(w\left(\beta_t\right)\right)\le z^*$ **.**

### 4.4.3 Lower Bounds on Time Consistent Risk-Averse Programs with CVaR Risk Measure

The above lower bounding approach has been successfully implemented in a variety of applications where numerous scenarios are present and time efficiency is valued (Gade et al. 2016, Cheung et al. 2015). The resulting solutions are expected to perform well across all the possible realizations and do not consider the risks in the worst scenarios. Guo and Ryan (2017) presented a set of PH lower bounding formulations for some commonly used risk measures including CVaR, excess probability and expected excess. Those risk measures are formulated based on the ECRMs proposed by Homem-de-Mello and Pagnoncelli (2016) which prove to be time consistent.

For some of the manufacturing systems such as mixed-model assembly lines operating on a just-in-time basis, the timely delivery of materials can be a large source of uncertainty for decision makers. Even the optimal sequencing decision solved from the multistage stochastic model may lead to significant costs in the worst scenarios of part availability. To increase the on-time performance for the worst cases, we adopt risk-averse stochastic models. We choose CVaR as our risk measure in our risk-averse model to measure the expectation of earliness and lateness in the worst scenarios whose costs fall above a given quantile of the cost distribution. Our time consistent risk-averse $\mathbb{E}$-CVaR optimization program for the mixed-model assembly line sequencing problem is formulated according to problem (5). Note that the objective function

(1.1) only includes $\tau$-stage decision variables. Therefore, the worst $100(1-\alpha)\%$ of scenarios in stage $\tau$ are considered in the $\mathbb{E}$-CVaR formulation of problem (1).

Here, we restate the lower bounding approach for $\mathbb{E}$-CVaR optimization programs from Guo and Ryan (2017) in Proposition 4.3. Let $U_t = \max_{\xi\in\Xi}\left\{\max_x \left\{c_t^T(\xi)x_t : x_t \in X_\xi\right\}\right\}$ and $L_t = \min_{\xi\in\Xi}\left\{\min_x \left\{c_t^T(\xi)x_t : x_t \in X_\xi\right\}\right\}$. Let $z^*_{\alpha-CVaR}$ denote the optimal objective function value of the problem (5). The weight vector $w$ defines an implicit lower bound on $z^*_{\alpha-CVaR}$.

**Proposition 4.3.** Let $w=\left(w(\xi)\right)_{\xi\in\Xi}$ and $w(\xi)=\left(w_t(\xi)\right)_{t=1,\cdots\tau}$ where $w_t(\xi)\in\mathbb{R}^{k_t}$ satisfy $\sum_{\xi(o_t)\in\Xi(o_t)} p_\xi w_t(\xi)=0$ for each $o_t\in O_t$. Let $w'=\left(w'(\xi)\right)_{\xi\in\Xi}$ and $w'(\xi)=\left(w'_t(\xi)\right)_{t=1,\cdots\tau}$ where $w'_t(\xi)\in\mathbb{R}^{k_t}$ satisfy $\sum_{\xi(o_t)\in\Xi(O_t)} p_\xi w'_t(\xi)=0$ for each $o_t\in O_t$. Let $w''=\left(w''(\xi)\right)_{\xi\in\Xi}$ and $w''(\xi)=\left(w''_t(\xi)\right)_{t=1,\cdots\tau}$ where $w''_t(\xi)\in\mathbb{R}^{k_t}$ satisfy $\sum_{\xi(o_t)\in\Xi(O_t)} p_\xi w''_t(\xi)=0$ for each $o_t\in O_t$. Let

$$D_\xi\left(w(\xi),w'(\xi),w''(\xi)\right) := \min\left\{\begin{array}{l} c_1^T x_1+\eta_2+\frac{1}{1-\alpha_2}v_2+\cdots+\eta_\tau+\frac{1}{1-\alpha_\tau}v_\tau \\ +w_1^T(\xi)x_1+\sum_{t=2}^{\tau} w_t'^T(\xi)\eta_t+\sum_{t=2}^{\tau-1} w_t''^T(\xi)v_t : \\ \eta_t+v_t\ge c_t^T(\xi)x_t,\forall t=2,\cdots,\tau \\ x_t(\xi)\in X_\xi,\ p_\xi x_t(\xi)-p_\xi\hat{x}\left(o(\xi_t)\right)=0,\forall\xi\in\Xi,\forall t=1,\cdots,\tau \\ \eta_t+v_t\le U_t,\forall t=1,\cdots,\tau \\ \eta_t\ge L_t,\forall t=1,\cdots,\tau \end{array}\right\} \quad (8)$$

*Then* $D(w,w',w'') := \sum_{\xi\in\Xi} p_\xi D_\xi\left(w(\xi),w'(\xi),w''(\xi)\right)\le z^*_{\alpha-CVaR}$.

### 4.5 Computational Study

In this section, we examine the performance of PH lower bounds for our time consistent risk-averse multistage stochastic mixed-integer formulation of the MMALS problem. We

generate a set of test instances and study the performance of the PH lower bounds. The interaction between the strategies for choosing the PH $\rho$ parameter and the quality of PH lower bounds is investigated as well as the scenario bundling strategies of PH. The PH algorithm is implemented in PySP (Hart et al. 2017), an open-source software package for modeling and solving stochastic programs with or without scenario bundling. The plugin called phboundextension is employed to implement the PH lower bounds for stochastic programs. CPLEX 12.7 is adopted to solve mixed-integer linear optimization problems (IBM ILOG CPLEX). All the experiments are conducted on a Linux server with 31 GB and 8 processors with 4 cores per processor.

We generate a set of MMALS instances denoted as MMALS-*P*.$\tau$ where *P* is the number of critical parts with uncertain availability and $\tau$ is the number of time stages. In each MMALS instance, the goal is to sequence a given set of various models to be assembled with various deadlines from customers. We consider the deliveries of the critical parts required for the assembly of each product as the uncertainty factor. According to the data collected from our industrial partners, the deliveries of parts are observed to be independent among different parts and across periods. The maximum lateness of delivery is two periods such that there are three scenarios for each part delivery, that is, timely delivery, delivery one period late, and delivery two periods late. Additionally, the delivery of each part generally follows a truncated geometric distribution where the probability of its being delivered in the next period, if the part has not arrived yet, is 0.7. When multiple copies of the same part are needed by the same unit, they arrive together, either all delivered on time or all delayed. When there exist more than one critical part delivered with uncertainty, the joint distribution of part deliveries is computed from the marginal distribution for each part with assumption of independence among parts. In a $\tau$ -

stage instance, we consider uncertain delivery of parts in the first $\tau$ time periods and assume the parts scheduled to deliver in later stages are delivered on time. For a MMALS instance with $\tau$ stages and *P* parts with uncertain availability, the input parameters of part deliveries are generated using probabilistic scenario trees with $\tau$ stages and $3^P$ branches emanating from each scenario tree node since there are 3 possible realizations for each part from each node of the previous stage, which leads to $3^{P(\tau-1)}$ scenarios. To represent the evolution of realizations of uncertainties for MMALS instances, we take MMALS-1.5 as an instance and display its scenario tree of part delivery uncertainties in Figure 1. With only one part with uncertain delivery, 3 possible realizations emanate from each scenario tree node for the first $\tau$ periods with $d^t$ representing the number of delayed periods at period $t = 1, \cdots, \tau$. After $\tau$ periods, only one realization emanates from each scenario node with $d^t = 0$ for periods $t > \tau$, meaning all the parts scheduled to deliver in stages later than $\tau$ periods are delivered in a timely manner. As seen from the scenario tree for MMALS-1.5, the number of nodes $|O_t|$ for each period $t \le \tau$ are computed as $|O_t| = 3^{t-1}$ while the number of nodes $|O_t|$ for each period $t > \tau$ stays constant as $|O_t| = 3^{\tau-1}$.

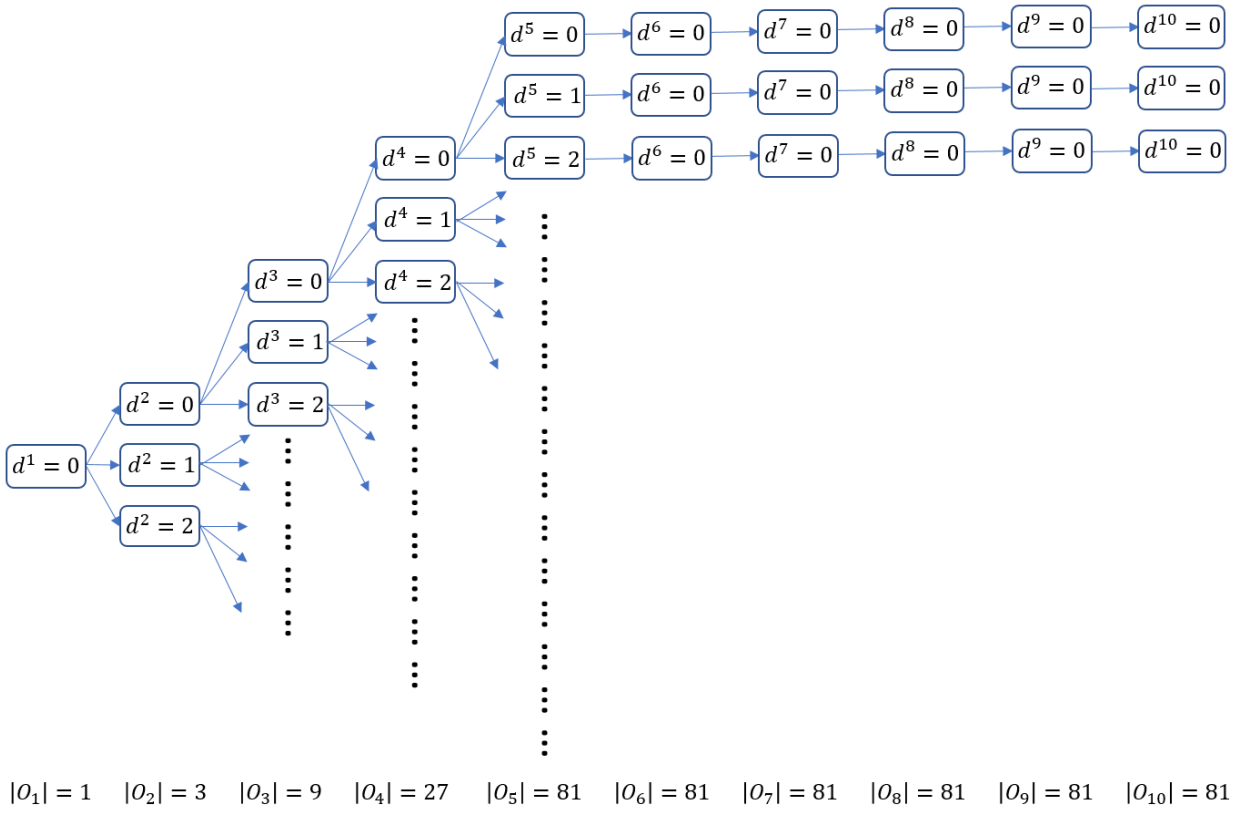


Figure 4.1 *Scenario tree to represent realizations of part delivery lateness for instance MMALS-1.5*

In our MMALS instances, we sequence ten various models at each working shift on a mixed-model assembly line of ten workstations. We consider 15 time periods with various time stages. The penalties for earliness and lateness are set to be $C_e = 1$ and $C_l = 3$. The input parameters of deadlines on the finished products are shown in Table 1. The input parameters of the part consumption are displayed in Table 2 and the input parameters of the number of delivered parts are shown in Table 3. The $P$ critical parts delivered with uncertainty are consumed in the first $P$ workstations in the assembly line, one such part type in each station.

Table 4.1 *Input parameters of due time for the finished products on MMALS instances*

| | $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *P=1* | $D_n$ | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| *P=2* | $D_n$ | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |

Table 4.2. *Input parameters of the number of part assumptions on MMALS instances*

| | $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *P=1* | $R_{1,n,1}$ | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 |
| *P=2* | $R_{1,n,1}$ | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 |
| | $R_{2,n,2}$ | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 |

Table 4.3. *Input parameters of the number of delivered good parts on MMALS instances*

| | $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *P=1* | $L_{1,1,t}$ | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 0 |
| *P=2* | $L_{1,1,t}$ | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 0 |
| | $L_{2,2,t}$ | 0 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 2 |

Computational studies on the risk-averse MMALS instances with $\mathbb{E}$-CVaR risk measure are as follows. Since the objective function (1.1) only includes $\tau$-stage decision variables, we need only to assign the confidence level in the final stage. The preselected probability is set to be $\alpha_\tau = 0.8$ such that we are concerned with the 20% worst scenarios in the final stage. Multiple runs of the PH algorithm are performed on this instance, varying the values of the penalty parameter $\rho$. Specifically, we consider fixed scalar values $\rho \in \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$ and record the time-series of the lower bound $D(w, w', w'')$ obtained at each PH iteration during each run. The lower bound results for a MMALS-2.3 instance are shown in Figure 2 (a), which additionally displays the optimal value obtained from solving its extensive form. As displayed in Figure 2 (a), larger values of $\rho$ tend to accelerate the convergence of lower bounds to the optimal objective. In contrast, smaller values of $\rho$ slow down as well as smooth the convergence and generally lead to higher quality of final solutions.

We also examine the PH lower bounds when bundling scenarios. Specifically, we vary the number of scenarios in each bundle considered by PH, while holding $\rho$ constant. We consider bundling at a single stage $t$ and each bundle is formed by scenarios emanating from one scenario tree node in stage $t-1$. An illustrative example for a MMALS-2.3 instance is shown in Figure 2 (b). The PH parameter is chosen to be $\rho = 10^{-3}$ because it takes a relatively large number of iterations for the PH lower bounds to converge to the optimal objective value. When scenarios are bundled when implementing the PH algorithm, the PH lower bounds start closer and converge faster to the optimal objective value.

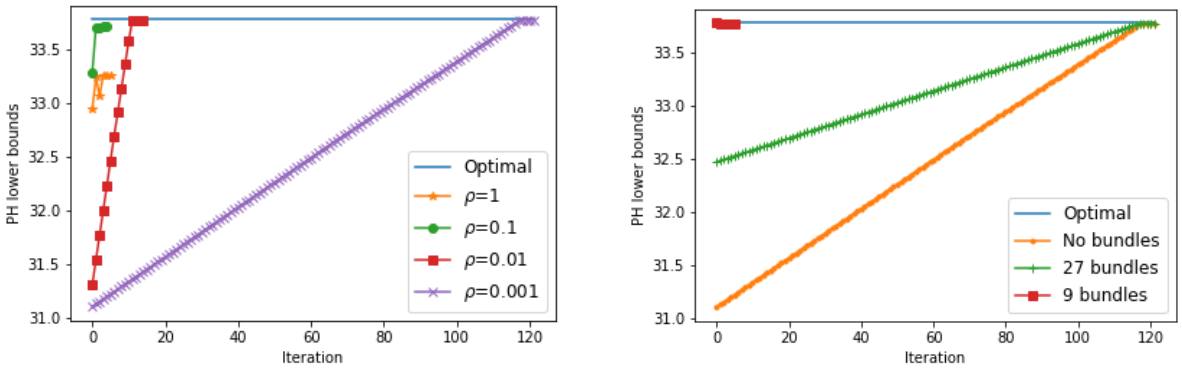


Figure 4.2 *Lower bounds from PH and optimal value by solving extensive form for MMALS-2.3 with (a) different penalty parameter values; (b) different scenario bundling strategies with* $\rho = 10^{-3}$

Finally, we consider summary results of a set of MMALS instances under the PH algorithm, shown in Table 4. In this experiment, we consider the PH algorithm behavior with and without the scenario bundling strategy under scalar PH parameter values $\rho \in \{10^{-1}, 10^{-2}\}$. In just-in-time production systems such as mixed-model assembly lines, the takt time and the cycle time tend to be very short and are generally measured in minutes, which requires real-time decisions within minutes to prevent the assembly lines from shutting down. In our computational experiments, the run-time limit is set to be 300 seconds for the PH algorithm and we record the

PH lower bounds as well as upper bounds on the optimality gaps. As we can see from Table 4, although the extensive form solves faster for relatively small-scale instances, the PH algorithm displays advantages when real-time decisions are desired for relatively large-scale instances where extensive forms fail to solve within a limited time frame. A parallel implementation of PH could reduce the run-times from those reported.

Table 4.4 *Lower bounds and run-time (in seconds) from PH and optimal objective and run-time (in seconds) from EF for a set of MMALS-P.$\tau$ instances*

| MMALS instance | Number of scenarios | PH algorithm with 300-second time limit | | | | | | | Extensive Form | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Value of $\rho$ | Stage of bundle | Number of bundles | Number of iterations | Lower bound | Gap upper bound | Run-time (s) | Optimal objective | Run-time (s) |
| 2.3 | 81 | 0.1 | NA | NA | 4 | 33.71 | 0.21% | 187 | 33.78 | 42 |
| | | | 3 | 9 | 3 | 33.78 | 0.00% | 120 | | |
| | | 0.01 | NA | NA | 9 | 33.36 | 1.26% | 300 | | |
| | | | 3 | 9 | 2 | 33.78 | 0.00% | 89 | | |
| 2.4* | 729 | 0.1 | NA | NA | 1 | 31.37 | 7.68% | 300 | 33.37 | 578 |
| | | | 4 | 81 | 1 | 32.01 | 5.53% | 300 | | |
| | | | 3 | 9 | 1 | 33.37 | 1.23% | 300 | | |
| | | 0.01 | NA | NA | 1 | 31.35 | 7.75% | 300 | | |
| | | | 4 | 81 | 1 | 32.01 | 5.53% | 300 | | |
| | | | 3 | 9 | 1 | 33.37 | 1.23% | 300 | | |
| 2.5* | 6561 | 0.1 | NA | NA | 1 | 32.09 | 14.02% | 300 | NA | NA |
| | | | 4 | 81 | 1 | 34.47 | 6.15% | 300 | | |
| | | | 3 | 9 | 1 | 34.48 | 6.12% | 300 | | |
| | | 0.01 | NA | NA | 1 | 32.09 | 14.02% | 300 | | |
| | | | 4 | 81 | 1 | 34.44 | 6.24% | 300 | | |
| | | | 3 | 9 | 1 | 34.44 | 6.24% | 300 | | |
| 1.5 | 81 | 0.1 | NA | NA | 11 | 32.11 | 5.45% | 300 | 33.86 | 18 |
| | | | 4 | 9 | 8 | 32.55 | 4.02% | 210 | | |
| | | 0.01 | NA | NA | 10 | 31.62 | 7.08% | 300 | | |
| | | | 4 | 9 | 13 | 32.39 | 4.54% | 300 | | |
| 1.6 | 243 | 0.1 | NA | NA | 3 | 31.10 | 8.87% | 300 | 33.86 | 71 |
| | | | 5 | 27 | 3 | 31.99 | 5.85% | 300 | | |
| | | | 4 | 9 | 4 | 32.46 | 4.31% | 300 | | |
| | | 0.01 | NA | NA | 2 | 31.04 | 9.09% | 300 | | |
| | | | 5 | 27 | 3 | 31.96 | 5.94% | 300 | | |
| | | | 4 | 9 | 4 | 32.34 | 4.70% | 300 | | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1.7* | 729 | 0.1 | NA | NA | 1 | 31.12 | 9.54% | 300 | 34.09 | 255 |
| | | | 5 | 27 | 1 | 32.09 | 6.23% | 300 | | |
| | | | 4 | 9 | 1 | 32.46 | 5.02% | 300 | | |
| | | 0.01 | NA | NA | 1 | 31.12 | 9.54% | 300 | | |
| | | | 5 | 27 | 1 | 32.04 | 6.40% | 300 | | |
| | | | 4 | 9 | 1 | 32.40 | 5.22% | 300 | | |
| 1.8* | 2187 | 0.1 | NA | NA | 1 | 31.29 | 9.68% | 300 | 34.32 | 670 |
| | | | 5 | 27 | 1 | 31.29 | 9.68% | 300 | | |
| | | 0.01 | NA | NA | 1 | 31.29 | 9.68% | 300 | | |
| | | | 5 | 27 | 1 | 31.29 | 9.68% | 300 | | |
| 1.9* | 6561 | 0.1 | NA | NA | 1 | 31.29 | 9.68% | 300 | NA | NA |
| | | | 6 | 81 | 1 | 31.29 | 9.68% | 300 | | |
| | | 0.01 | NA | NA | 1 | 31.29 | 9.68% | 300 | | |
| | | | 6 | 81 | 1 | 31.29 | 9.68% | 300 | | |

The PH algorithm is terminated either when the normalized average per-scenario deviation from the average solution is reduced below the convergence threshold of 0.01% or when the run-time exceeds 300 seconds. The bound on the optimality gap is computed as $\frac{UB-LB}{LB}$, where $LB$ is the value of the lower bound obtained from our PH lower bounding approach, and $UB$ is the objective value of the feasible solution obtained from PH in the last iteration. For the relatively large MMALS instances denoted by symbol *, however, only one iteration can be performed within the 300-second time limit. For those instances, $LB$ is equal to the 'wait-and-see' (WS) value; i.e., the expectation of the optimal objective values from all the scenario sub-problems, and $UB$ is the objective value of a feasible solution obtained by solving the extensive form of a pessimistic instance that is solvable within 300 seconds. Compared to the original $\tau$-stage instance, a pessimistic has a scenario tree modified to have only the worst realizations of part deliveries in periods $\tau$, $\tau-1$, …, $\tau-K+1$ for some $K$; i.e., all the parts scheduled to be delivered in the last $K$ periods are delivered two periods late. As observed from Table 4, we cannot tell the difference on the optimal objectives between MMALS-1.8 and

MMALS-1.9, which indicates that the uncertainties to be realized in the ninth period are too remote to affect the sequencing decisions and results from the initial stage. This observation suggests diminishing returns from including many stages of uncertainties in the stochastic sequencing model.

## 4.6 Conclusions

To develop a shop floor decision support system for just-in-time production systems such as mixed-model assembly lines, it is essential to develop a decision-making tool to make real-time resequencing decisions. Real-time decisions are necessary to avoid wastage in time and costs of downtime caused by inevitable disruptions. The real-time aspect of our decision-making tool is realized from two respects: the stochastic optimization model and the solution approach. On the modeling side, our multistage stochastic program formulation accounts for all the possibilities of part availability and thus provides a feasible sequencing decision no matter which scenario occurs. While a deterministic model might give a solution that turns out to be infeasible when some required parts are unavailable, this stochastic model effectively prevents the assembly line from shutting down when scheduled parts are not delivered on time. In addition, the optimal solution from a risk-neutral model may result in large costs in the worst scenarios when the uncertainty of timely delivery is significant. Therefore, a time consistent risk-averse model with an expected conditional CVaR risk measure is presented to optimize the on-time performance for the worst scenarios. For computational efficiency, we adopt the PH algorithm with a lower bounding approach to assess solution quality. Our numerical results demonstrate its computational advantage in finding near-optimal solutions in real time, especially for large-scale MMALS instances whose extensive forms cannot be solved within a limited time frame. In summary, our formulation of risk-neutral and risk-averse multistage stochastic optimization models, together with our adopted

solution approach, makes it possible for a shop floor decision support system to suggest real-time resequencing decisions to improve the on-time performance of mixed-model assembly lines.

## REFERENCES

Boysen N., Fliedner M., and Scholl A. (2009) Sequencing mixed-model assembly lines: survey, classification and model critique. European Journal of Operations Research, 192, 349-373

Cheung K., Gade D., Silva-Monroy C., Ryan S., Watson J.P. (2015) Toward scalable stochastic unit commitment. Part 2: solver configuration and performance assessment. Energy Systems, 6(3), 417-438

Crainic T.G., Fu X., Gendreau M., Rei W., and Wallace S.W. (2011) Progressive hedging-based metaheuristics for stochastic network design. Networks, 58(2), 114-124

Dong J. (2014) Balancing and sequencing of stochastic mixed-model assembly u-lines to minimize the expectation of work overload time. International Journal of Production Research, 52(24), 7529-7548

Dörmer J., Günther H. O., and Gujjula R. (2015) Master production scheduling and sequencing at mixed-model assembly lines in the automotive industry. Flexible Services and Manufacturing Journal, 27, 1-29

Fan, Y., & Liu, C. (2010). Solving stochastic transportation network protection problem using the progressive hedging-based method. Networks and Spatial Economics, 10(2), 193-208.

Gade D., Hackebeil G., Ryan S., Watson J., Wets R., and Woodruff D. (2016) Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. Mathematical Programming, 157(1), 47-67

Guo G., Hackebeil G., Ryan S., Watson J., and Woodruff D. (2015) Integration of progressive hedging and dual decomposition in stochastic integer programs. Operations Research Letters, 43, 311-316

Guo G., Ryan S. (2017). Progressive hedging lower bounds for time consistent risk-averse multistage stochastic mixed-integer programs, available at: http://works.bepress.com/sarah_m_ryan/93/

Hart W., Laird C., Watson J., Woodruff D., Hackebeil G., Nicholson B., and Siirola J. (2017) Stochastic Programming Extensions. In Pyomo - Optimization Modeling in Python. Second Edition (pp. 165-198) Springer Optimization and Its Applications, 67

Haugen K., Løkketangen A., and Woodruff D.L. (2001) Progressive hedging as a metaheuristic applied to stochastic lot-sizing. European Journal of Operational Research, 132, 116-122

Homem-de-Mello T., & Pagnoncelli B. K. (2016). Risk aversion in multistage stochastic programming: a modeling and algorithmic perspective. European Journal of Operational Research, 249(1), 188-199.

Lamghari A., and Dimitrakopoulos R. (2016) Progressive hedging applied as a metaheuristic to schedule production in open-pit mines accounting for reserve uncertainty. European Journal of Operational Research, 265(3), 843-855

Listes O., & Dekker R. (2005). A scenario aggregation based approach for determining a robust airline fleet composition. Transportation Science, 39, 367-382.

Manavizadeh N., Rabbani M., and Radmehr F. (2015) A new multi-objective approach in order to balancing and sequencing U-shaped mixed model assembly line problem: a proposed heuristic algorithm. The International Journal of Advanced Manufacturing Technology, 79, 415-425

Mulvey J. M., & Vladimirou, H. (1991). Applying the progressive hedging algorithm to stochastic generalized networks. Annals of Operations Research, 31, 399-424.

Ordoudis C., Pinson P., Zugno M., and Morales J.M. (2015) Stochastic unit commitment via Progressive Hedging – extensive analysis of solution methods. PowerTech 2015 IEEE Eindhoven

Rabbani M., Heidari R., and Farrokhi H. (2018) A bi-objective mixed-model assembly line sequencing problem considering customer satisfaction and customer buying behavior. Engineering Optimization, 1-20

Rahimi-Vahed A. (2007) A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem," Computers and Industrial Engineering, 53(4), 642-666

Rockafellar R., and Wets R. (1991) Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research, 16(1), 119–147

Schultz R. (2010) Risk aversion in two-stage stochastic integer programming. International Series in Operations Research and Management Science, 150, 165-187

Watson J., and Woodruff D. (2011) Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. Computational Management Science, 8(4), 355-370

Wets R. J. B. (1989). The aggregation principle in scenario analysis and stochastic optimization. In Wallace, S.W. (Ed.), Algorithms and Model Formulations in Mathematical Programming (pp. 91-113). Berlin: Springer.

Zhao X. (2006) Sequence-to-customer Goal with Stochastic Demands for a Mixed-model Assembly Line. International Journal of Production Research, 44(24), 5279-5305

# CHAPTER 5. SUMMARY AND DISCUSSION

This dissertation consists of three papers that, together, contribute to the solution approaches for multistage risk-neutral and risk-averse optimization problems with discrete decisions under uncertainty. The contributions, limitations, and future studies are discussed in this chapter.

The first paper addressed some complementary deficiencies of the PH and DD algorithms and presented an integrated method that takes advantage of PH's computational efficiency and DD's guarantee of global convergence. The computational studies indicate that the integration of the PH algorithm can help the DD algorithm to reduce the run-time by up to 50% in a set of stochastic server location instances, and help the DD algorithm to converge to a near-optimal solution within minutes in a stochastic unit commitment instance that takes the DD algorithm alone more than 24 hours to reduce the optimality gap below 99%. The success with this solution approach, however, relies on the prior experiment on the tuning of the PH parameters and the DD parameters which might consume some computational time. Further research can be performed to explore the strategies to compute effective and efficient PH and DD parameters to speed up the convergence of our integrated method. In addition, we shall compare the PH lower bound with other bounding approaches for multistage problems in our future study.

The second paper proposed a time consistent scenario-decomposed version of reformulations of multistage risk-averse stochastic programs with a variety of risk measures, which allows for the employment of scenario decomposition solution algorithms such as the PH algorithm in solving multistage risk-averse models. We further developed a lower bounding approach based on the scenario reformulations to help the PH algorithm to assess solution quality and to find near-optimal solutions within a reasonable amount of time for multistage risk-averse

stochastic mixed-integer problems. Our computational experiments show that this lower bounding approach provides convergent and tight lower bounds and that PH can obtain near-optimal solutions within reasonable run-time for large-scale stochastic mixed-integer instances whose extensive form cannot be solved. However, our numerical studies are limited to the risk-averse problems with CVaR as the risk measure. A complete computational study can be performed in our future study to include a variety of risk-averse models with different risk measures. It also remains as a promising area for potential future research to integrate this lower bounding approach with other exact algorithms for multistage risk-averse programs.

The third paper is part of a project to develop a shop floor decision support system for a just-in-time production system to provide real-time resequencing decisions. First, a stochastic optimization model was formulated to incorporate real-world uncertainties from timely part delivery and material quality such that a feasible sequencing decision is available in any scenario. A time consistent risk-averse model was further proposed to optimize on-time performance for the worst scenarios. Second, the lower bounding approach presented in the second paper was adopted as our solution method to find near-optimal sequencing decisions in real time. The computational results indicate the PH algorithm together with its lower bounding approach can yield real-time sequencing decisions with verifiable high quality in a just-in-time production system, especially for large-scale instances whose extensive form cannot be solved within a reasonable amount of time. This optimization model, however, is limited to the sequencing decision making in the main assembly lines. Future studies can be conducted to also include the sequencing decisions for the sub-assembly lines. Additionally, more real-world uncertainty factors such as stochastic demands and deadlines can be incorporated in our sequencing model in our future research.