Essays on Logic-Based Benders Decomposition, Portfolio Optimization, and Fair Allocation of Resources

by

Özgün Elçi

Submitted to the Tepper School of Business
in partial fulfillment of the requirements for the degree of

PhD in Operations Research

at the

CARNEGIE MELLON UNIVERSITY

August 2022

Dissertation Committee:
Gérard Cornuéjols (co-chair)
John Hooker (co-chair)
Pınar Keskinocak
Matthias Köppe
Peter Zhang

Essays on Logic-Based Benders Decomposition, Portfolio Optimization, and Fair Allocation of Resources

by

Özgün Elçi

Submitted to the Tepper School of Business
on August 30, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis offers methodological and computational contributions to several fields of operations research including stochastic programming, decomposition-based methods, robust optimization, and fairness in resource allocation. In the first chapter, we introduce the stochastic planning and scheduling problem and formulate it as a two-stage stochastic program. We devise a logic-based Benders decomposition algorithm that can solve this problem exactly. We present an extensive numerical analysis on the effectiveness of the proposed solution algorithm. In the second chapter, we extend our analysis on the planning and scheduling problem. We introduce new Benders cuts and improve some of the cuts proposed in the literature. We then focus on a class of sequence-dependent scheduling problems. We provide an exact solution method for this problem by deriving novel logic-based Benders cuts. The cuts we propose generalize some of the other well-known cuts in the literature. The numerical experiments show that the proposed method outperforms the benchmark. In the third chapter, we study the classical Markowitz model for portfolio optimization. We focus on a robust portfolio optimization model that attempts to address the uncertainty in the expected returns. We provide a theoretical analysis on the selection of the error covariance matrix that is used to define the uncertainty set.

Our results show that the class of diagonal estimation-error matrices can achieve an arbitrarily small loss in the expected portfolio return as compared to the optimum. The computational experiments we perform show that even using the identity matrix as the error covariance matrix outperforms the classical Markowitz model. In the fourth and the last chapter, we focus on the use of optimization models for fair allocation of scarce resources. We study several social welfare functions that are used in optimization models to balance efficiency and equity in resource distribution. We analyze the structural properties of the socially optimal distributions based on each social welfare function. We discuss the implications of selecting a social welfare function with respect to the incentives it creates for both the players and the social planner. We then extend our analysis to hierarchical networks. Our analysis offers a novel approach to evaluating the adequacy of well-known social welfare functions for distribution of scarce resources.

To my parents Nazan and Cumhur Elçi
and my brother Ezgi Elçi

Acknowledgments

A PhD is not a one-person job. I have had the great fortune of receiving support from many people in the last ten years of my academic studies. Here is an incomplete list of people to whom I would like to express my gratitude.

First and foremost, I would like to thank my advisors John Hooker and Gérard Cornuéjols. I started working with John early in my PhD studies and wrote my first summer paper with him. Ever since, his guidance, patience, and formidably broad expertise have shaped my research and the course of my PhD. I started working with Gérard right before the Covid pandemic hit; throughout everything, Gérard remained kind and understanding. Seeing his keen intellectual curiosity and problem-solving acumen was indispensable to my growth as a researcher. Working with John and Gérard was the pinnacle of my academic career, an experience that is unlikely to be beaten.

Aside from my advisors, I was very lucky to work with Matthias Köppe and Peter Zhang. They were both very generous with their time, and their ideas and suggestions were instrumental to this dissertation. Matthias and Peter also served on my PhD committee, along with Pınar Keskinocak. I am thankful to all my committee members for taking time out to be on my thesis committee and providing me with valuable feedback. I would like to thank all the Tepper faculty members in the Operations Research group, including Javier Peña and Ben Moseley for helping me in difficult times. I also would like to thank Laila Lee and Lawrence Rapp for their kindness and administrative support.

I would like to thank Semra Ağralı and Ethem Çanakoğlu for advising me during

my undergraduate degree and encouraging me to continue in graduate studies in operations research. I would like to thank my master's degree advisors Kerem Bülbül and Nilay Noyan. Their endless support and tutelage has brought me to places that I had hardly imagined when I was a young master's student.

During my graduate studies, I have had the pleasure of building relationships with so many great colleagues and friends. I miss Murat Mustafa Tunç, Bahar Imanlou, Gizem Ermiş, Can Küçükgül, and Yağmur Özdemir, and all the songs we sang together. Umut Koray Tuncay was the roommate of a lifetime. I learned a great deal of coding from Semih Atakan and Halil Şen. Many thanks to all my friends in my PhD cohort including Violet Chen, Yuyan Wang, Sagnik Das, Melda Korkut, Musa Çeldir, Franco Berbeglia, Arash Haddadan, Mehmet Aydemir, and Neda Mirzaeian. I had wonderful times with Jackson Singer, TC Eley, Lorenzo and Martina Tomaselli, Victoria Glavin, Tarık Bilgiç, Gabbi Guedes, and Nathan Nikolic. I will miss Meriç İşgenç, Barış Ötüş and Volkan Cirik, their kindness and support, and all the walks we have taken in Squirrel Hill.

I would like to thank my sweet partner Marina Lucia Clementi for supporting me more than anyone during the last three years of my PhD. Bob Clementi and Paula Martino have been incredibly generous to me in the time I have known them. Their kindness and hospitality made me feel at home in Pittsburgh.

Lastly, I would like to thank my parents Nazan and Cumhur Elçi, and my brother Ezgi Elçi, for their unconditional love. My family has shaped who I am as a scholar and as a person, and their support has meant everything to me. Mom, Dad, and Ezgi, I dedicate this dissertation to you.

Contents

List of Figures

List of Tables

Introduction

> *"Cum enim mundi universi fabrica sit perfectissima atque a Creatore sapientissimo absoluta, nihil omnino in mundo contingit, in quo non maximi minimive ratio quaepiam eluceat; quamobrem dubium prorsus est nullum, quin omnes mundi effectus ex causis finalibus ope methodi maximorum et minimorum aeque feliciter determinari queant, atque ex ipsis causis efficientibus."* [1]
>
> *Leonhard Euler (De Curvis Elasticis, 1774)*

Mathematical optimization plays an essential role in the planning, design, and operation of engineering systems. It provides a modeling framework for formulating real-world problems in detailed mathematical terms and techniques for solving those mathematical models. Today, the field of operations research offers significant tools for decision making through interdisciplinary collaboration between mathematicians, engineers, and practitioners. In this dissertation, we make contributions to several frontiers of operations research, from logic-based Benders decomposition to portfolio optimization and fair allocation of resources.

[1] "Briefly and very freely translated: Nothing in the world takes place without optimization, and there is no doubt that all aspects of the world that have a rational basis can be explained by optimization methods." (Grötschel, 2012)

Logic-based Benders decomposition. The idea of decomposition in mathematical optimization relies on breaking down an intractable problem into smaller problems that are easier to solve. One such approach for solving large-scale linear/integer programs is proposed by Jacques F. Benders in his seminal work in 1962. An important restriction of the classical Benders decomposition is that it can only be applied to the problems where subproblems are linear programs. Logic-based Benders decomposition, introduced by John Hooker in 2000, extends the classical Benders decomposition to cases in which subproblems can be arbitrary optimization problems.

In chapter 1, we apply logic-based Benders decomposition (LBBD) to two-stage stochastic planning and scheduling problems in which the second stage is a scheduling task. We solve the master problem with mixed-integer/linear programming and the subproblem with constraint programming. As Benders cuts, we use simple nogood cuts as well as analytical logic-based cuts we develop for this application. We find that LBBD is computationally superior to the integer L-shaped method. In particular, a branch-and-check variant of LBBD can be faster by several orders of magnitude, allowing significantly larger instances to be solved. This is due primarily to computational overhead incurred by the integer L-shaped method while generating classical Benders cuts from a continuous relaxation of an integer programming subproblem. To our knowledge, this is the first application of LBBD to two-stage stochastic optimization with a scheduling second-stage problem, and the first comparison of LBBD with the integer L-shaped method. The results suggest that LBBD could be a promising approach to other stochastic and robust optimization problems with integer or combinatorial recourse.

In chapter 2, we study a class of sequence-dependent parallel machine scheduling

problems in which the objective is makespan minimization. In contrast to the traditional literature, we develop an exact algorithm that finds the optimal solution in finitely many iterations. We achieve this by devising an algorithm that utilizes the LBBD framework. We derive novel logic-based Benders cuts by analyzing the combinatorial structure of the scheduling problem. The proposed cuts generalize other well known cuts in the literature. We demonstrate the efficacy of the cuts via a computational study.

Portfolio optimization. The publication of Harry Markowitz's theory of portfolio selection has been instrumental in the understanding of financial markets and the development of financial decision making. His famous work in 1952 suggested that financial decision making is a quantitative trade-off between risk and return. Ever since, the concepts of diversification and portfolio optimization have been studied extensively and quantitative techniques have become widespread in the investment industry.

In chapter 3, we study an extension of the classical Markowitz model. It is well known that the classical Markowitz model for portfolio optimization is extremely sensitive to estimation errors on the expected asset returns. Robust optimization mitigates this issue. We focus on ellipsoidal uncertainty sets around a point estimate of the expected asset returns. An important issue is the choice of the matrix that specifies this ellipsoid. In this paper we investigate the performance of diagonal estimation-error matrices. We show that diagonal estimation-error matrices can achieve an arbitrarily small loss in the expected portfolio return as compared to the optimum. We then formulate the problem of finding the best estimation error matrix as a bilevel program. Finally we analyze the use of an identity matrix as the estimation-error matrix. The results of our simulation show that robust portfolio models featuring

an identity matrix as an estimation-error matrix outperform the classical Markowitz model when the size of the uncertainty set is chosen properly.

Fair allocation of resources. Social welfare optimization is a paradigm that is used to incorporate ethical norms into decision making processes. When used to improve societal outcomes, optimization models that simply focus on utilitarian goals may produce extreme and undesirable solutions. To balance equity and efficiency, researchers have proposed various ways to incorporate both goals in the decision-making process. From the perspective of optimization, this can be done by using a social welfare function (SWF) that combines equity and efficiency in the objective function.

In chapter 4, we focus on the use of optimization models for fair allocation of scarce resources. We show that the plethora of SWFs proposed in the literature can also produce extreme solutions. For example, the maximin (Rawlsian) criterion ignores less fortunate individuals except for the very worst one; alpha fairness SWF may equate an egalitarian solution with an extremely imbalanced solution; the Kalai-Smorodinsky bargaining solution may favor individuals that are already privileged by proportionally allocating resources to everyone based on their utility upper bounds; and a threshold SWFs with leximax criterion may produce more moderate outcomes at the price of increasing complexity in model formulation and analysis. We analyze these SWFs roughly in increasing order of their complexity. Such complexity arises from the goal of preventing extreme outcomes and thus may attenuate the extremity of solutions. But it also can at the same time lead to more complex and undesirable properties. We illustrate the latter point with a representative example throughout, focusing on a simple hierarchical resource allocation model. In conclusion, one needs to be careful in choosing a SWF to avoid unacceptable outcomes.

Chapter 1

Stochastic Planning and Scheduling with Logic-Based Benders Decomposition

This chapter is a joint work with John Hooker.

1.1 Introduction

Benders decomposition has seen many successful applications to two-stage stochastic optimization, where it typically takes the form of the *L-shaped method* (Benders, 1962; Van Slyke and Wets, 1969). It offers the advantage that the second-stage problem decouples into a separate problem for each possible scenario, allowing much faster computation of the recourse decision.

A limitation of classical Benders decomposition, however, is that the subproblem must be a linear programming problem, or a continuous nonlinear programming problem in the case of "generalized" Benders decomposition (Geoffrion, 1972). This is necessary because the Benders cuts are derived from dual multipliers (or Lagrange multipliers) in the subproblem. Yet in many problems, the recourse decision is a combinatorial optimization problem that does not yield dual multipliers. This issue has been addressed by the *integer L-shaped method* (Laporte and Louveaux, 1993), which formulates the subproblem as a mixed-integer/linear programming (MILP) problem and obtains dual multipliers from its linear programming (LP) relaxation. To ensure finite convergence, classical Benders cuts from the LP relaxation are augmented with "integer cuts" that simply exclude the master problem solutions enumerated so far.

Unfortunately, a combinatorial subproblem may be difficult to model as an MILP, in the sense that many variables are required, or the LP relaxation is weak. This is particularly the case when the recourse decision poses a scheduling problem. We therefore investigate the option of applying *logic-based Benders decomposition* (LBBD) to problems with a second-stage scheduling decision (Hooker, 2000b; Hooker and Ottosson, 2003), because it does not require dual multipliers to obtain Benders cuts. Rather, the cuts are obtained from an "inference dual" that is based on a structural analysis of the subproblem. This allows the subproblem to be solved by a method that is best suited to compute optimal schedules, without having to reformulate it as an MILP.

We investigate the LBBD option by observing its behavior on a generic planning and scheduling problem in which scheduling takes place after the random events have been observed. The planning element is an assignment of jobs to facilities that

occurs in the first stage. Jobs assigned to each facility are then scheduled in the second stage subject to time windows. We assume that the job processing time is a random variable, but the LBBD approach is easily modified to accommodate other random elements, such as the release time. The subproblem decouples into a separate scheduling problem for each facility and each scenario. For greater generality, we suppose the recourse decision is a *cumulative* scheduling problem in which multiple jobs can run simultaneously on a single facility, subject to a limit on total resource consumption at any one time.

We solve the first-stage problem by MILP, which is well suited for assignment problems. More relevant to the present study is our choice to solve the scheduling subproblem by constraint programming (CP), which has proved to be effective on a variety of scheduling problems, perhaps the state of the art in many cases. We therefore formulate the subproblem in a CP modeling language rather than as an MILP. In view of the past success of LBBD on a number of deterministic planning and scheduling problems, we test the hypothesis that it can obtain similar success on stochastic problems with many scenarios. We perform computational experiments while minimizing makespan, total tardiness, and total assignment cost. We also derive new logic-based Benders cuts for the minimum makespan problem that have not been used in previous work.

In addition to standard LBBD, we experiment with *branch and check*, a variation of LBBD that solves the master problem only once and generates Benders cuts on the fly during the MILP branching process (Hooker, 2000b; Thorsteinsson, 2001). We find that both versions of LBBD are superior to the integer L-shaped method. In particular, branch and check is faster by several orders of magnitude, allowing significantly larger instances to be solved. We also conduct a variety of tests

to identify factors that explain the superior performance of LBBD, the relative effectiveness of various Benders cuts, and the impact of modifying the integer L-shaped method in various ways. To our knowledge, this is the first computational comparison between LBBD and the integer L-shaped method on any kind of stochastic optimization problem. It also appears to be the first application of LBBD to two-stage stochastic optimization with a scheduling second-stage problem.

The remainder of this paper is organized as follows. We introduce the stochastic planning and scheduling problem in Section 1.3. This is followed by Section 1.4 where we propose the logic-based Benders decomposition based solution methods for solving three variants of the stochastic planning and scheduling problem. We present the computational results in Section 1.7 and give our concluding remarks in Section 1.8. Additional computational experiments and details on the data set can be found in the electronic companion to the paper.

1.2 Previous Work

A wide range of problems can be formulated as two-stage stochastic programs. For theory and various applications, we refer the reader to Birge and Louveaux (2011), Shapiro et al. (2009), Prékopa (2013), and the references therein. Allowing discrete decisions in the second-stage problem significantly expands the applicability of the two-stage stochastic framework, as for example to last-mile relief network design (Noyan et al., 2015) and vehicle routing with stochastic travel times (Laporte et al., 1992).

Benders decomposition (Benders, 1962) has long been applied to large-scale optimization problems (Geoffrion and Graves, 1974; Cordeau et al., 2001; Binato et al.,

2001; Contreras et al., 2011). Rahmaniani et al. (2017) provide an excellent survey of enhancements to the classical method. In particular, it has been applied to two-stage stochastic programs with linear recourse by means of the L-shaped method (Van Slyke and Wets, 1969). Its applicability was extended to integer recourse by the integer L-shaped method of Laporte and Louveaux (1993), which was recently revisited and improved by Angulo et al. (2016) and Li and Grossmann (2018). Other Benders-type algorithms that have been proposed for integer recourse include disjunctive decomposition (Sen and Higle, 2005) and decomposition with parametric Gomory cuts (Gade et al., 2014). The essence of these two methods is to convexify the integer second-stage problem using disjunctive cuts and Gomory cuts, respectively. Still other decomposition-based methods in the literature include progressive hedging for multi-stage stochastic convex programs (Rockafellar and Wets, 1991) and a dual decomposition method for multi-stage stochastic programs with mixed-integer variables (Carøe and Schultz, 1999). We refer the reader to Küçükyavuz and Sen (2017) for a review of two-stage stochastic mixed-integer programming.

Logic-based Benders decomposition was introduced by Hooker (2000b) and further developed in Hooker and Ottosson (2003). Branch and check, a variant of LBBD, was also introduced by Hooker (2000b) and first tested computationally by Thorsteinsson (2001), who coined the term "branch and check." A general exposition of both standard LBBD and branch and check, with an extensive survey of applications, can be found in Hooker (2019a). A number of these applications have basically the same mathematical structure as the planning and scheduling problem studied here, albeit generally without a stochastic element.

In more recent work, Atakan et al. (2017) focus on a one-stage stochastic model for single-machine scheduling in which they minimize the value-at-risk of several

random performance measures. Bülbül et al. (2016) consider a two-stage chance-constrained mean-risk stochastic programming model for single-machine scheduling problem, but the scheduling decisions do not occur in the second stage. Rather, the second-stage problem is a simple optimal timing problem that can be solved very rapidly. The deterministic version of the planning and scheduling problem we consider here is solved by LBBD in Hooker (2007) and Ciré et al. (2016). We rely on some techniques from these studies.

We are aware of three prior applications of LBBD to stochastic optimization. Lombardi et al. (2010) use LBBD to assign computational tasks to chips and to schedule the tasks assigned to each chip. In this application, the scheduling problem parameters are not random, and the expected recourse has an analytic solution. The authors reformulate the problem as a single-stage stochastic program using the analytical solution. Fazel-Zarandi et al. (2013) solve a stochastic location-routing problem with LBBD, but there is no actual recourse decision in the second stage, which only penalizes vehicles if the route determined by first-stage decisions exceeds their threshold capacity. Guo et al. (2019) use LBBD to schedule patients in operating rooms, where the random element is the surgery duration. Here the scheduling takes place in the master problem, where patients are assigned to operating rooms and surgery dates. The subproblem checks whether there is time during the day to perform all the surgeries assigned to a given operating room, and if not, finds a cost-minimizing selection of surgeries to cancel on that day. Unstrengthened nogood cuts are used as LBBD cuts, along with classical Benders cuts derived from a network flow model of the subproblem that is obtained from a binary decision diagram.

The present study therefore appears to be the first application of LBBD to two-stage

stochastic optimization with scheduling in the second stage. It is also the first to compare any application of stochastic LBBD with the integer L-shaped method.

1.3 The Problem

We study a two-stage stochastic programming problem that, in general, has the following form:

$$\min_{\mathbf{x}\in X} \left\{ f(\mathbf{x}) + \mathbb{E}_\omega[Q(\mathbf{x},\omega)] \right\} \tag{1.1}$$

where $Q(\mathbf{x},\omega)$ is the optimal value of the second-stage problem:

$$Q(\mathbf{x},\omega) = \min_{\mathbf{y}\in Y(\omega)} \left\{ g(\mathbf{y}) \right\} \tag{1.2}$$

Variable $\mathbf{x}$ represents the first-stage decisions, while $\mathbf{y}$ represents second-stage decisions that are made after the random variable ω is realized. We suppose that ω ranges over a finite set Ω of possible scenarios, where each scenario ω has probability π_ω. The first-stage problem (1.1) may therefore be written as

$$\min_{\mathbf{x}\in X} \left\{ f(\mathbf{x}) + \sum_{\omega\in\Omega} \pi_\omega Q(\mathbf{x},\omega) \right\}$$

We consider a generic planning and scheduling problem in which the first stage assigns jobs to facilities, and the second stage schedules the jobs assigned to each facility. The objective is to minimize expected makespan or expected total tardiness. We assume that only the processing times are random in the second stage, but a slight modification of the model allows for random release times and/or deadlines as well.

We therefore suppose that each job j has a processing time p_{ij}^{ω} on facility i in scenario ω and must be processed during the interval $[r_j, d_j]$. For greater generality, we allow for cumulative scheduling (Aggoun and Beldiceanu 1993, Baptiste et al. 2001), where each job j consumes resources c_{ij} on facility i, and the total resource consumption must not exceed K_i.

To formulate the problem we let variable x_j be the facility to which job $j \in J$ is assigned. The first-stage problem is

$$\min_{\mathbf{x}} \left\{ g(\mathbf{x}) + \sum_{\omega \in \Omega} \pi_\omega Q(\mathbf{x}, \omega) \;\middle|\; x_j \in I, \text{ all } j \in J \right\} \tag{1.3}$$

where I indexes the facilities. In the second-stage problem, we let s_j be the time at which job j starts processing. We also let $J_i(\mathbf{x})$ be the set of jobs assigned to facility i, so that $J_i(\mathbf{x}) = \{j \in J \mid x_j = i\}$. Thus

$$Q(\mathbf{x}, \omega) = \min_{\boldsymbol{s}} \left\{ h(\boldsymbol{s}, \mathbf{x}, \omega) \;\middle|\; s_j \in [r_j, d_j - p_{x_j j}^{\omega}], \text{ all } j \in J; \sum_{\substack{j \in J_i(\mathbf{x}) \\ 0 \le t \le s_j + p_{ij}^{\omega}}} c_{ij} \le K_i, \text{ all } i \in I, \text{ all } t \right\}$$

where $h(\boldsymbol{s}, \mathbf{x}, \omega)$ denotes the second-stage objective function given the first-stage decision $\boldsymbol{x}$ and scenario ω.

The two-stage problem (1.1) is risk-neutral in the sense that it is concerned with minimizing expectation. However, the LBBD approach presented here can be adapted to a more general class of problems that incorporate a dispersion statistic $\mathbb{D}_\omega$ that measures risk, such as variance, as in the classical Markowitz (1952) model.

Then the problem (1.1) becomes

$$\min_{\mathbf{x}\in X} \left\{ f(\mathbf{x}) + (1-\lambda)\mathbb{E}_\omega[Q(\mathbf{x},\omega)] + \lambda\mathbb{D}_\omega[Q(\mathbf{x},\omega)] \right\} \tag{1.4}$$

and the first-stage planning and scheduling problem (1.3) becomes

$$\min_{\mathbf{x}} \left\{ g(\mathbf{x}) + (1-\lambda)\sum_{\omega\in\Omega} \pi_\omega Q(\mathbf{x},\omega) + \lambda\mathbb{D}_\omega\big[Q(\mathbf{x},\omega)\big] \;\Big|\; x_j \in I, \text{ all } j \in J \right\} \tag{1.5}$$

Formulations (1.4) and (1.5) also accommodate robust optimization, as for example when $\lambda = 1$ and

$$\mathbb{D}_\omega(Q(\mathbf{x},\omega)) = \max_{\omega\in\Omega}\{Q(\mathbf{x},\omega)\}$$

and Ω is an uncertainty set. See Ahmed (2006) for a discussion of various tractable and intractable risk measures.

1.4 Logic-Based Benders Decomposition

Logic-based Benders decomposition (LBBD) is designed for problems of the form

$$\min_{\mathbf{x},\mathbf{y}} \left\{ f(\mathbf{x},\mathbf{y}) \;\big|\; C(\mathbf{x},\mathbf{y}),\ \mathbf{x}\in D_x,\ \mathbf{y}\in D_y \right\} \tag{1.6}$$

where $C(\mathbf{x},\mathbf{y})$ denotes a set of constraints that contain variables $\mathbf{x}$ and $\mathbf{y}$, and D_y and D_x represent variable domains. The rationale behind dividing the variables into two groups is that once some of the decisions are fixed by setting $\mathbf{x} = \bar{\mathbf{x}}$, the remaining *subproblem* becomes much easier to solve, perhaps by decoupling into smaller problems. In our study, the smaller problems will correspond to scenarios

and facilities. The subproblem has the form

$$\mathrm{SP}(\bar{\mathbf{x}}) = \min_{\mathbf{y}} \left\{ f(\bar{\mathbf{x}}, \mathbf{y}) \;\middle|\; C(\bar{\mathbf{x}}, \mathbf{y}),\ \mathbf{y} \in D_y \right\} \tag{1.7}$$

The key to LBBD is analyzing the subproblem solution so as to find a function $B_{\bar{\mathbf{x}}}(\mathbf{x})$ that provides a lower bound on $f(\mathbf{x}, \mathbf{y})$ for any given $\mathbf{x} \in D_x$. The bound must be sharp for $\mathbf{x} = \bar{\mathbf{x}}$; that is, $B_{\bar{\mathbf{x}}}(\bar{\mathbf{x}}) = \mathrm{SP}(\bar{\mathbf{x}})$. The bounding function is derived from the *inference dual* of the subproblem in a manner discussed below. In classical Benders decomposition, the subproblem is an LP problem, and the inference dual is the LP dual.

Each iteration of the LBBD algorithm begins by solving a *master problem*:

$$\mathrm{MP}(\bar{\mathbf{X}}) = \min_{\mathbf{x},\beta} \left\{ \beta \;\middle|\; \beta \geq B_{\bar{\mathbf{x}}}(\mathbf{x}),\ \text{all } \bar{\mathbf{x}} \in \bar{\mathbf{X}};\ \ \mathbf{x} \in D_x \right\} \tag{1.8}$$

where the inequalities $\beta \geq B_{\bar{\mathbf{x}}}(\mathbf{x})$ are *Benders cuts* obtained from previous solutions $\bar{\mathbf{x}}$ of the subproblem. There may be several cuts for a given $\bar{\mathbf{x}}$, but for simplicity we assume in this section there is only one. Initially, the set $\bar{\mathbf{X}}$ can be empty, or it can contain a few solutions obtained heuristically to implement a "warm start." The optimal value $\mathrm{MP}(\bar{\mathbf{X}})$ of the master problem is a lower bound on the optimal value of the original problem (1.6). If $\bar{\mathbf{x}}$ is an optimal solution of the master problem, the corresponding subproblem is then solved to obtain $\mathrm{SP}(\bar{\mathbf{x}})$, which is an upper bound on the optimal value of (1.6). A new Benders cut $\beta \geq B_{\bar{\mathbf{x}}}(\mathbf{x})$ is generated for the master problem and $\bar{\mathbf{x}}$ added to $\bar{\mathbf{X}}$ in (1.8). The process repeats until the lower and upper bounds provided by the master problem and subproblem converge; that is, until $\mathrm{MP}(\bar{\mathbf{X}}) = \min_{\bar{\mathbf{x}} \in \bar{\mathbf{X}}} \{\mathrm{SP}(\bar{\mathbf{x}})\}$. The following is proved in Hooker (2000b):

Theorem 1. *If D_x is finite, the LBBD algorithm converges to an optimal solution of* (1.6) *after a finite number of iterations.*

The inference dual of the subproblem seeks the tightest bound on the objective function that can be inferred from the constraints. Thus the inference dual is

$$\mathrm{DSP}(\bar{\mathbf{x}}) = \max_{P \in \mathcal{P}} \left\{ \gamma \;\middle|\; \big(C(\bar{\mathbf{x}}, \mathbf{y}),\ \mathbf{y} \in D_y\big) \overset{P}{\Rightarrow} \big(f(\bar{\mathbf{x}}, \mathbf{y}) \geq \gamma\big) \right\} \tag{1.9}$$

where $A \overset{P}{\Rightarrow} B$ indicates that proof P deduces B from A. The inference dual is always defined with respect to set $\mathcal{P}$ of valid proofs. In classical linear programming duality, valid proofs consist of nonnegative linear combinations of the inequality constraints in the problem. We assume a strong dual, meaning that $\mathrm{SP}(\bar{\mathbf{x}}) = \mathrm{DSP}(\bar{\mathbf{x}})$. The dual is strong when the inference method is complete. For example, the classical Farkas Lemma implies that nonnegative linear combination is a complete inference method for linear inequalities. Indeed, any exact optimization method is associated with a complete inference method that it uses to prove optimality, perhaps one that involves branching, cutting planes, constraint propagation, and so forth.

In the context of LBBD, the proof P that solves the dual (1.9) is the proof of optimality the solver obtains for the subproblem (1.7). The bounding function $B_{\bar{\mathbf{x}}}(\mathbf{x})$ is derived by observing what bound on the optimal value *this same proof* P can logically deduce for a given $\mathbf{x}$, whence the description "logic-based." In practice, the solver may not reveal how it proved optimality, or the proof may be too complicated to build a useful cut. One option in such cases is to tease out the structure of the proof by re-solving the subproblem for several values of $\mathbf{x}$ and observing the optimal value that results. This information can be used to design *strengthened nogood cuts* that provide useful bounds for many values of $\mathbf{x}$ other than $\bar{\mathbf{x}}$. Another approach is

to use *analytical Benders cuts*, which deduce bounds on the optimal value when $\bar{\mathbf{x}}$ is changed in certain ways, based on structural characteristics of the subproblem and its current solution. We will employ both of these options.

Branch and check is a variation of LBBD that solves the master problem only once and generates Benders cuts on the fly. It is most naturally applied when the master problem is solved by branching. Whenever the branching process discovers a solution $\bar{\mathbf{x}}$ that is feasible in the current master problem, the corresponding subproblem is solved to obtain one or more Benders cuts, which are added to the master problem. Branching then continues and terminates in the normal fashion, all the while satisfying Benders cuts as they accumulate. Branch and check can be superior to standard LBBD when the master problem is much harder to solve than the subproblems.

A common enhancement of LBBD and other Benders methods is a *warm start*, which includes initial Benders cuts in the master problem. Recent studies that benefit from this technique include Angulo et al. (2016), Elçi and Noyan (2018), and Heching et al. (2019). Benders cuts can also be aggregated before being added to the master problem, a technique first explored in Birge and Louveaux (1988). A particularly useful enhancement for LBBD is to include a relaxation of the subproblem in the master problem, where the relaxation is written in terms of the master problem variables (Hooker, 2007; Fazel-Zarandi and Beck, 2012). We employ this technique in the present study.

1.5 Benders Formulation of Planning and Scheduling

We apply LBBD to the generic planning and scheduling problem by placing the assignment decision in the master problem and the scheduling decision in the subproblem. The master problem is therefore

$$\min_{\mathbf{x}} \left\{ g(\mathbf{x}) + \sum_{\omega \in \Omega} \pi_\omega \beta_\omega \;\middle|\; \text{Benders cuts; subproblem relaxation; } x_j \in I, \text{ all } j \in J \right\}$$

where β_ω is an auxiliary variable that captures second-stage objective function value under scenario ω. The Benders cuts provide lower bounds on each β_ω. The cuts and subproblem relaxation are somewhat different for each variant of the problem we consider below. The scheduling subproblem decouples into a separate problem for each facility and scenario. If $\bar{\mathbf{x}}$ is an optimal solution of the master problem, the scheduling problem for facility i and scenario ω is

$$\mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) = \min_{\boldsymbol{s}} \left\{ h_i(\boldsymbol{s}, \bar{\mathbf{x}}, \omega) \;\middle|\; s_j \in [r_j, d_j - p_{ij}^\omega], \text{ all } j \in J_i(\bar{\mathbf{x}}); \sum_{\substack{j \in J_i(\bar{\mathbf{x}}) \\ 0 \le t \le s_j + p_{ij}^\omega}} c_{ij} \le K_i, \text{ all } t \right\}$$

We solve the master problem and subproblem by formulating the former as an MILP problem and the latter as a CP problem. In the master problem, we let variable

$x_{ij} = 1$ when job j is assigned to facility i. The master problem becomes

$$\begin{array}{ll} \text{minimize} & \hat{g}(\mathbf{x}) + \sum_{\omega \in \Omega} \pi_\omega \beta_\omega \\ \text{subject to} & \sum_{i \in I} x_{ij} = 1, \;\; j \in J \\ & \text{Benders cuts} \\ & \text{subproblem relaxation} \\ & x_{ij} \in \{0,1\}, \; i \in I, \; j \in J \end{array} \tag{1.10}$$

where $\mathbf{x}$ now denotes the matrix of variables x_{ij}. If $\bar{\mathbf{x}}$ is an optimal solution of the master problem, the subproblem for each facility i and scenario ω becomes

$$\begin{array}{ll} \text{minimize} & \hat{h}_i(\boldsymbol{s}, \bar{\mathbf{x}}, \omega) \\ \text{subject to} & \text{cumulative}\Big(\big(s_j \,\big|\, j \in J_i(\bar{\mathbf{x}})\big), \; \big(p_{ij}^\omega \,\big|\, j \in J_i(\bar{\mathbf{x}})\big), \; \big(c_{ij} \,\big|\, j \in J_i(\bar{\mathbf{x}})\big), \; K_i\Big) \\ & s_j \in [r_j, d_i - p_{ij}^\omega], \; j \in J_i(\bar{\mathbf{x}}) \end{array} \tag{1.11}$$

The optimal value of (1.11) is again $\mathrm{SP}_{i\omega}(\bar{\mathbf{x}})$. The cumulative global constraint in (1.11) is a standard feature of CP models and requires that the total resource consumption at any time on facility i be at most K_i.

To solve a problem (1.5) that incorporates risk, one need only replace the objective function of (1.10) with

$$\hat{g}(\mathbf{x}) + (1 - \lambda) \sum_{\omega \in \Omega} \pi_\omega \beta_\omega + \lambda \mathbb{D}_\omega[\beta_\omega]$$

and otherwise proceed as in the risk-neutral case.

1.5.1 Minimum Makespan Problem

We begin by considering a minimum makespan problem in which the jobs have release times and no deadlines. The first-stage objective function is $g(\mathbf{x}) = 0$, and so we have $\hat{g}(\mathbf{x}) = 0$ in the MILP model (1.10). The second-stage objective function is the finish time of the last job to finish:

$$h(\boldsymbol{s}, \mathbf{x}, \omega) = \max_{j \in J(\mathbf{x})} \left\{ s_j + p^{\omega}_{x_j j} \right\}$$

This objective function is incorporated into the CP problem (1.11) by setting $\hat{h}_i(\boldsymbol{s}, \bar{\mathbf{x}}, \omega) = M$ and adding to (1.11) the constraints $M \geq s_j + p^{\omega}_{ij}$ for all $j \in J_i(\bar{\mathbf{x}})$. Since there are no deadlines, we assume $d_j = \infty$ for all $j \in J$.

Both strengthened nogood cuts and analytic Benders cuts can be developed for this problem. A simple nogood cut for scenario ω can take the form of a set of inequalities

$$\beta_\omega \geq \beta_{i\omega}, \; i \in I \tag{1.12}$$

where each $\beta_{i\omega}$ is bounded by

$$\beta_{i\omega} \geq \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) \Big(\sum_{j \in J_i(\bar{\mathbf{x}})} x_{ij} - |J_i(\bar{\mathbf{x}})| + 1 \Big) \tag{1.13}$$

and where $\bar{\mathbf{x}}$ is the solution of the current master problem and $|J_i(\bar{\mathbf{x}})|$ denotes the cardinality of set $J_i(\bar{\mathbf{x}})$. The cut says that if all the jobs in $J_i(\bar{\mathbf{x}})$ are assigned to facility i, possibly among other jobs, then the makespan of facility i in scenario ω is at least the current makespan $\mathrm{SP}_{i\omega}(\bar{\mathbf{x}})$. The cut is weak, however, because if even one job in $J_i(\bar{\mathbf{x}})$ is not assigned to i, the bound in (1.13) becomes useless. The cut can be

strengthened by heuristically assigning proper subsets of the jobs in $J_i(\bar{\mathbf{x}})$ to facility i, and re-computing the minimum makespan for each subset, to discover a smaller set of jobs that yields the same makespan. This partially reveals which job assignments serve as premises of the optimality proof. Then $J_i(\bar{\mathbf{x}})$ in (1.13) is replaced with this smaller set to strengthen the cut. This simple scheme, and variations of it, can be effective when the makespan problem solves quickly (Hooker, 2007).

A stronger cut can be obtained without re-solving the makespan problem by using an analytical Benders cut. We introduce a cut based on the following lemma:

Lemma 2. *Consider a minimum makespan problem P in which each job $j \in J$ has release time r_j and processing time p_j, with no deadlines. Let M^* denote the minimum makespan for P, and $\hat{M}$ the minimum makespan for the problem $\hat{P}$ that is identical to P except that the jobs in a nonempty set $\hat{J} \subset J$ are removed. Then*

$$M^* - \hat{M} \leq \Delta + r^+ - r^- \tag{1.14}$$

where $\Delta = \sum_{j \in \hat{J}} p_j$, and $r^+ = \max_{j \in J}$ and $r^- = \min_{j \in J}\{r_j\}$ are the latest and earliest release times of the jobs in set J.

Proof. Consider any solution of $\hat{P}$ with makespan $\hat{M}$. We will construct a feasible solution for P by extending this solution. If $\hat{M} > r^+$, we schedule all the jobs in $\hat{J}$ sequentially starting from time $\hat{M}$, resulting in makespan $\hat{M} + \Delta$. This is a feasible solution for P, and we have $M^* \leq \hat{M} + \Delta$. The lemma follows because $r^+ - r^-$ is nonnegative. If $\hat{M} < r^+$, we schedule all the jobs in $\hat{J}$ sequentially starting from time r^+ to obtain a solution with makespan of $r^+ + \Delta$. Again this is a feasible solution

for P, and we have $M^* \leq r^+ + \Delta$. This implies

$$M^* - \hat{M} \leq r^+ - \hat{M} + \Delta \tag{1.15}$$

Because $\hat{M}$ is at least r^-, (1.15) implies (1.14), and the lemma follows. □

We can now derive a valid analytical cut:

Theorem 3. *A valid Benders cut for scenario ω can be obtained by adding inequalities* (1.12) *and the following to the master problem:*

$$\beta_{i\omega} \geq \left\{ \begin{array}{ll} \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \Big(\sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij}) p^{\omega}_{ij} + r^+ - r^- \Big), & \textit{if } x_{ij} = 0 \textit{ for some } j \in J_i(\bar{\mathbf{x}}) \\ \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}), & \textit{otherwise} \end{array} \right\}, \; i \in I \tag{1.16}$$

where $r^+ = \max_{j \in J_i(\bar{\mathbf{x}})}\{r_j\}$ *and* $r^- = \min_{j \in J_i(\bar{\mathbf{x}})}\{r_j\}$.

Proof. The cut clearly provides a sharp bound $\max_{i \in I}\{\mathrm{SP}_{i\omega}(\bar{\mathbf{x}})\}$ when $\mathbf{x} = \bar{\mathbf{x}}$, because the second line of (1.16) applies in this case. The validity of the cut follows immediately from Lemma 2. □

We linearize the cut (1.16) as follows:

$$\begin{array}{ll} \beta_{i\omega} \geq \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij})(p^{\omega}_{ij} + r^+ - r^-) & (a) \\ \beta_{i\omega} \geq \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij}) p^{\omega}_{ij} - (r^+ - r^-) & (b) \end{array} \tag{1.17}$$

The Benders cut (1.16) is inserted into the master problem by including inequalities

(1.17) for each $i \in I$ and $\omega \in \Omega$, along with the inequalities (1.12).

Corollary 4. *The inequalities* (1.17) *yield a valid Benders cut equivalent to* (1.16).

Proof. Let $k = \sum_{j \in J_i(\bar{\mathbf{x}})}(1 - x_{ij})$. If $k = 0$, (1.17a) is identical to the second line of (1.16), while (1.17b) is implied by (1.17a) and therefore valid. If $k = 1$, both (1.17a) and (1.17b) are identical to the first line of (1.16). If $k \geq 2$, (1.17b) is identical to the first line of (1.16), while (1.17a) is implied by (1.17b) and therefore valid. □

A Benders cut can also be derived for the case in which all release times are equal and the jobs have deadlines. The following is an immediate consequence of a theorem proved in Hooker (2007):

Theorem 5. *Suppose all release times are $r_j = 0$, and each job j has a deadline d_j. A valid Benders cut for scenario ω can be obtained by adding inequalities* (1.12) *and the following to the master problem:*

$$\beta_{i\omega} \geq \left\{ \begin{array}{ll} \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \Big(\displaystyle\sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij}) p_{ij}^{\omega} + d^{+} - d^{-} \Big), & \text{if } x_{ij} = 0 \text{ for some } j \in J_i(\bar{\mathbf{x}}) \\ \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}), & \text{otherwise} \end{array} \right\}, \; i \in I \tag{1.18}$$

where $d^{+} = \max_{j \in J_i(\bar{\mathbf{x}})}\{d_j\}$ and $d^{-} = \min_{j \in J_i(\bar{\mathbf{x}})}\{d_j\}$.

The linearization provided in Hooker (2007) for this cut introduces a continuous variable for each job j. This adds a considerable computational burden for the stochastic problem, since it requires a new continuous variable for each scenario ω and each facility i. However, we can avoid additional variables by formulating a

linearization parallel to (1.17) that uses the following inequalities:

$$\begin{aligned} \beta_{i\omega} &\geq \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij})(p_{ij}^{\omega} + d^{+} - d^{-}) \qquad (a) \\ \beta_{i\omega} &\geq \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij}) p_{ij}^{\omega} - (d^{+} - d^{-}) \qquad (b) \end{aligned} \tag{1.19}$$

The Benders cut is inserted into the master problem by including inequalities (1.19) for each $i \in I$ and $\omega \in \Omega$, along with the inequalities (1.12). The proof of validity is similar to the proof of Corollary 4.

Corollary 6. *The inequalities* (1.19) *yield a valid Benders cut equivalent to* (1.18)*.*

Finally, we add a subproblem relaxation to the master problem. We use a relaxation from Hooker (2007), modified to be scenario-specific:

$$\beta_{i\omega} \geq \frac{1}{K_i} \sum_{j \in J} c_{ij} p_{ij}^{\omega} x_{ij}, \quad i \in I,\ \omega \in \Omega \tag{1.20}$$

This relaxation is valid for arbitrary release times and deadlines.

1.5.2 Minimum Cost Problem

In the minimum cost problem, there is only a fixed cost ϕ_{ij} associated with assigning job j to facility i. So we have

$$\hat{g}(\mathbf{x}) = \sum_{i \in I} \sum_{j \in J} \phi_{ij} x_{ij}$$

in the MILP master problem (1.10), and we set $\beta_{\omega} = 0$ for $\omega \in \Omega$. The subproblem decouples into a feasibility problem for each i and ω, because $\hat{h}_i(\boldsymbol{s}, \bar{\mathbf{x}}, \omega) = 0$.

A Benders cut is generated for each i and ω when the corresponding scheduling problem (1.11) is infeasible. A simple nogood cut is

$$\sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij}) \geq 1 \tag{1.21}$$

We strengthen the cut heuristically by re-solving the scheduling problem $|J_i(\bar{\mathbf{x}})|$ times, each time removing a different job j from $J_i(\bar{\mathbf{x}})$. We add the nogood cut (1.21), with j removed from $J_i(\bar{\mathbf{x}})$, whenever the scheduling subproblem is infeasible.

To create a subproblem relaxation for the master problem, one can exploit the fact that we now have two-sided time windows $[r_j, d_j]$. Let $J(t_1, t_2)$ be the set of jobs j for which $[r_j, d_j] \subseteq [t_1, t_2]$. Adapting an approach from Hooker (2007), one can add the following inequalities to the master problem for each $i \in I$:

$$\frac{1}{K_i} \sum_{j \in J(t_1,t_2)} p_{ij}^{\min} c_{ij} x_{ij} \leq t_2 - t_1, \;\; t_1 \in \{\bar{r}_1, \ldots, \bar{r}_{n'}\}, \; t_2 \in \{\bar{d}_1, \ldots, \bar{d}_{n''}\} \tag{1.22}$$

where $\bar{r}_1, \ldots, \bar{r}_{n'}$ are the distinct release times among $r_1, \ldots, r_n$, and $\bar{d}_1, \ldots, \bar{d}_{n''}$ the distinct deadlines among $d_1, \ldots, d_n$. Some of these inequalities may be redundant, and a method for detecting them is presented in (Hooker, 2007). Because the relaxation must be valid across all scenarios, the processing time is set to $p_{ij}^{\min} = \min_{\omega \in \Omega}\{p_{ij}^{\omega}\}$.

1.5.3 Minimum Tardiness Problem

In this section, we consider a minimum tardiness problem in which jobs are all released at time zero but have different due dates $\bar{d}_j$. There are no hard deadlines, and so we let $d_j = \infty$ for all $j \in J$. As in the minimum makespan problem, there

is no first-stage cost, so that $\hat{g}(\mathbf{x}) = 0$ in the MILP model (1.10). The second-stage objective function is expected total tardiness, and we have

$$\hat{h}_i(\boldsymbol{s}, \mathbf{x}, \omega) = \sum_{j \in J_i(\mathbf{x})} \left(s_j + p_{ij}^{\omega} - \bar{d}_j\right)^+$$

in the CP scheduling problem (1.11). Here $\alpha^+ = \max\{0, \alpha\}$.

The following analytic Benders cut can be adapted from Hooker (2012):

$$\beta_\omega \geq \sum_{i \in I} \left(\mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} \Big(\sum_{j' \in J_i(\bar{\mathbf{x}})} p_{ij'}^{\omega} - \bar{d}_j \Big)^+ (1 - x_{ij}) \right) \tag{1.23}$$

The cut is added to (1.10) for each $\omega \in \Omega$. Strengthened nogood cuts similar to those developed for the makespan problem can also be used.

Two subproblem relaxations can be adapted from Hooker (2007). The simpler one is analogous to (1.22) and adds the following inequalities to (1.10) for each i and ω

$$\beta_{i\omega} \geq \frac{1}{K_i} \sum_{j' \in J(0, \bar{d}_j)} p_{ij'}^{\omega} c_{ij'} x_{ij'} - \bar{d}_j, \quad j \in J \tag{1.24}$$

along with the bounds $\beta_{i\omega} \geq 0$. A second relaxation more deeply exploits the structure of the subproblem. For each facility i and scenario ω, let τ_i^{ω} be a permutation of $\{1, \ldots, n\}$ such that $p_{i\tau_i^{\omega}(1)}^{\omega} c_{i\tau_i^{\omega}(1)} \leq \cdots \leq p_{i\tau_i^{\omega}(n)}^{\omega} c_{i\tau_i^{\omega}(n)}$. We also assume that jobs are indexed so that $\bar{d}_1 \leq \cdots \leq \bar{d}_n$. Then we add the following inequalities to the master problem (1.10) for each i and ω:

$$\beta_{i\omega} \geq \frac{1}{K_i} \sum_{j' \in J} p_{i\tau_i^{\omega}(j')}^{\omega} c_{i\tau_i^{\omega}(j')} x_{i\tau_i^{\omega}(j')} - \bar{d}_j - (1 - x_{ij}) U_{ij\omega}, \quad j \in J$$

where

$$U_{ij\omega} = \frac{1}{K_i} \sum_{j' \in J} p^{\omega}_{i\tau_i^{\omega}(j')} c_{i\tau_i^{\omega}(j')} - \bar{d}_j$$

1.6 The Integer L-Shaped Method

The integer L-Shaped method is a Benders-based algorithm proposed by Laporte and Louveaux (1993) to solve two-stage stochastic integer programs. It terminates in finitely many iterations when the problem has complete recourse and binary first-stage variables. It is similar to branch and check in that Benders cuts are generated while solving the first-stage problem by branching. It differs in that it uses subgradient cuts derived from a linear programming relaxation of the subproblem rather than combinatorial cuts derived from the original subproblem. It also uses a simple integer nogood cut to ensure convergence, but the cut is quite weak and does not exploit the structure of the subproblem as does branch and check. We describe the integer L-shaped method as it applies to minimizing makespan in the planning and scheduling problem.

We first state an MILP model of the deterministic equivalent problem, as it will play a benchmarking role in computational testing. We index discrete times by $t \in T$ and introduce a 0–1 variable z^{ω}_{ijt} that is 1 if job j starts at time t on facility i in scenario ω. The model is

$$
\begin{aligned}
\text{minimize} \quad & \sum_{\omega \in \Omega} \pi_\omega \beta_\omega && (a) \\
\text{subject to} \quad & \sum_{i \in I} x_{ij} = 1, \;\; j \in J && (b) \\
& \beta_\omega \geq \beta_{i\omega}, \;\; i \in I, \; \omega \in \Omega && (c) \\
& x_{ij} \in \{0,1\}, \;\; i \in I, \; j \in J && (d) \\
& \beta_{i\omega} \geq \sum_{t \in T} (t + p_{ij}^\omega) z_{ijt}^\omega, \;\; i \in I, \; j \in J, \; \omega \in \Omega && (e) \\
& z_{ijt}^\omega \leq x_{ij}, \;\; i \in I, \; j \in J, \; t \in T, \; \omega \in \Omega && (f) \\
& \sum_{i \in I} \sum_{t \in T} z_{ijt}^\omega = 1, \;\; j \in J, \; \omega \in \Omega && (g) \\
& \sum_{j \in J} \sum_{t' \in T_{tij}^\omega} c_{ij} z_{ijt'}^\omega \leq K_i, \;\; i \in I, \; t \in T, \; \omega \in \Omega && (h) \\
& z_{ijt}^\omega = 0, \;\; i \in I, \; \omega \in \Omega, \; j \in J, \; \text{all } t \in T \text{ with } t < r_j && (i) \\
& z_{ijt}^\omega \in \{0,1\}, \;\; i \in i, \; j \in J, \; t \in T, \; \omega \in \Omega && (j)
\end{aligned}
\tag{1.25}
$$

where $T_{tij}^\omega = \{t' \mid 0 \leq t' \leq t - p_{ij}^\omega\}$. In the integer L-shaped method, the first stage minimizes (1.25a) subject to (1.25b)–(1.25d) and Benders cuts that provide bounds on β_ω. The Benders cuts consist of classical Benders cuts derived from the linear relaxation of the second-stage scheduling problem for each i and ω, as well as integer cuts. If $\bar{\mathbf{x}}$ is an optimal solution of the first-stage problem, the second-stage problem

for facility i and scenario ω is

$$\begin{aligned}
\text{minimize} \quad & M \\
\text{subject to} \quad & M \geq \sum_{t \in T} (t + p_{ij}^{\omega}) z_{ijt}^{\omega}, \;\; j \in J_i(\bar{\mathbf{x}}) \\
& \sum_{t \in T} z_{ijt}^{\omega} = 1, \;\; j \in J_i(\bar{\mathbf{x}}) \\
& \sum_{j \in J} \sum_{t' \in T_{tij}^{\omega}} c_{ij} z_{ijt'}^{\omega} \leq K_i, \;\; t \in T \\
& z_{ijt}^{\omega} \in \{0,1\}, \;\; j \in J_i(\bar{\mathbf{x}}), \; t \in T \\
& z_{ijt}^{\omega} = 0, \;\; j \in J_i(\bar{\mathbf{x}}), \text{ all } t \in T \text{ with } t < r_j \\
& z_{ijt}^{\omega} \in \{0,1\}, \;\; j \in J_i(\bar{\mathbf{x}}), \; t \in T
\end{aligned} \tag{1.26}$$

The following integer L-shaped cut is used for each ω to ensure convergence:

$$\beta_{\omega} \geq \big(\mathrm{SP}_{\omega}(\bar{\mathbf{x}}) - \mathrm{LB}_{\omega}\big)\Big(\sum_{j \in S(\bar{\mathbf{x}})} x_{ij} - \sum_{j \notin S(\bar{\mathbf{x}})} x_{ij} - |S(\bar{\mathbf{x}})| + 1\Big) + \mathrm{LB}_{\omega} \tag{1.27}$$

where $S(\bar{\mathbf{x}}) \coloneqq \{i : x_i = 1\}$ and LB_{ω} is a global lower bound on makespan under scenario ω. We obtain LB_{ω} by solving the LP relaxation of (1.25) for fixed scenario ω. The same lower bound is used to strengthen the initial master problem in LBBD and branch-and-check methods by adding bounds of the form

$$\beta_{\omega} \geq \mathrm{LB}_{\omega}, \;\; \omega \in \Omega. \tag{1.28}$$

1.7 Computational Study

In this section, we describe computational experiments we conducted for all three objective functions described in Section 1.5. In addition, we use the minimum makespan problem to test the effect of several modifications to the LBBD and integer L-shaped methods.

One effect that has been observed in previous work (Hooker 2007, Ciré et al. 2016) is that the relative advantage of LBBD for planning and scheduling tends to increase with the number of facilities, for a given fixed number of jobs. In particular, the advantage of LBBD is much less pronounced when there are only two facilities. This is because a larger number of facilities results in more decoupling of the subproblem and a smaller number of jobs assigned to each facility (the complexity of the scheduling problem is highly sensitive to the number of assigned jobs). Stochastic planning and scheduling is similar in that the subproblem is smaller when there are more facilities, but there is a difference as well: the scheduling problem size remains constant as the number of *scenarios* increases. We might therefore expect that computational tests will show the relative advantage of stochastic LBBD to be less with two facilities than with a greater number, while it is an open question how its advantage will vary with the number of scenarios. To test the former hypothesis, we design experiments with two and four facilities. To investigate the latter question, we run tests with a wide range of scenario counts (1 to 500).

All experiments are conducted on a personal computer with a 2.80 GHz Intel® Core™ i7-7600 processor and 24 GB memory running on a Microsoft Windows 10 Pro. All MILP and CP formulations are solved in `C++` using the `CPLEX` and `CP Optimizer` engines of `IBM® ILOG® CPLEX® 12.7 Optimization Studio`, respectively. We

use a single thread in all computational experiments. We modify `CP Optimizer` parameters to execute an extended filtering and DFS search. The rest of the parameters are set to their default values for both `CPLEX` and `CP Optimizer` engines. Lastly, we use the `Lazy Constraint Callback` function of `CPLEX` to implement branch and check.

1.7.1 Minimum Makespan Problem

For the minimum makepsan problem, we generate problem instances by combining ideas from Hooker (2007) and Atakan et al. (2017). We first generate the deterministic problem as in Hooker (2007). Let $|I| = m$ and $|J| = n$. The capacity limits of the facilities is set to $K_i = 10$ for all $i \in I$, and integer capacity requirements of jobs are drawn from a uniform distribution on $[1, 10]$. Integer release times are drawn from a uniform distribution on $[0,\ 2.5n(m+1)/m]$. For each facility $i \in I$, integer mean processing times $\bar{p}_{ij}$ are drawn from a uniform distribution on $[2,\ 25 - 10(i-1)/(m-1)]$. This causes facilities with a higher index to process jobs more rapidly.

We then follow Atakan et al. (2017) by perturbing the mean processing times to obtain a set of scenarios. In particular, we first divide the jobs into two groups, one group containing jobs i for which $0 < \bar{p}_{ij} \leq 16$, and the other group containing the remainder of the jobs. We then generate a perturbation parameter ϵ^ω for each scenario $\omega \in \Omega$ from a mixture of uniform distributions. Specifically, for jobs in the first group, ϵ^ω is distributed uniformly on the interval $[-0.1,\ 0.5]$ with probability 0.9 and on the interval $[2.0,\ 3.0]$ with probability 0.1. For jobs in the second group, ϵ^ω is distributed uniformly on the interval $[-0.1,\ 0.5]$ with probability 0.99 and on the interval $[1.0,\ 1.5]$ with probability 0.01. Finally, we generate the processing times

Table 1.1: Computation times in seconds (averaged over 3 instances) of the integer L-shaped and branch-and-check methods for 2 and 4 facilities.

		2 facilities			4 facilities		
Tasks	Scenarios	Integer L-shaped method	B&Ch nogood cuts	B&Ch analytic cuts	Integer L-shaped method	B&Ch nogood cuts	B&Ch analytic cuts
10	1	127.3	1.5	0.9	2405.0††	0.5	0.4
	5	839.2	4.6	2.1	*	2.5	2.4
	10	2316.9†	6.1	3.1	*	3.9	3.7
	50	*	28.4	17.3	*	24.4	17.4
	100	*	56.1	36.7	*	41.9	28.6
	500	*	375.6	279.1	*	268.3	164.8
14	1	1831.6†	4.3	3.8	3016.6††	0.9	1.7
	5	*	16.3	16.9	*	5.6	4.8
	10	*	37.7	26.1	*	12.4	7.3
	50	*	186.0	134.4	*	88.0	34.2
	100	*	411.0	357.9	*	189.5	83.2
	500	*	2431.9†	2424.3†	*	854.7	443.9
18	1	2416.2††	208.3	155.5	2404.7††	1.5	1.0
	5	*	1102.4	988.4	*	57.0	20.7
	10	*	2184.0	1888.3†	*	116.4	39.5
	50	*	*	*	*	458.2	187.8
	100	*	*	*	*	943.7	332.7
	500	*	*	*	*	2804.5††	2825.9††

†Average excludes one instance that exceeded an hour in computation time.

††Average excludes two instances that exceeded an hour.

*All three instances exceeded an hour.

under scenario $\omega \in \Omega$ by letting $p_{ij}^{\omega} = \lceil \bar{p}_{ij}(1 + \epsilon^{\omega}) \rceil$. All problem instances used throughout this paper are publicly available (see electronic companion A.1.3).

Table 1.1 summarizes the relative performance of LBBD and the integer L-shaped method on instances of various sizes. The table focuses on the branch-and-check variant of LBBD because we found it to be superior to standard LBBD. Statistics

for standard LBBD and other variants are reported in subsequent tables. The specific methods compared are as follows:

- *Integer L-shaped method.* We decouple the second-stage problem by facility and scenario, and we solve the resulting problems and their LP relaxations using the MILP engine of `CPLEX` whenever a candidate incumbent solution is identified. We then add the integer cut (1.27), as well as the classical Benders cut from the LP relaxation for each scenario. The initial bounds (1.28) are included in the master problem, even though they are not standard, because previous experience indicates that they significantly enhance performance. The subproblem relaxation (1.20) is likewise included in the master problem for fair comparison with LBBD and branch and check, where it is standard.

- *Branch and check with nogood cuts.* We use (1.12) and unstrengthened nogood cuts (1.13). We solve the decoupled subproblems by `CP Optimizer`. The initial bounds (1.28) are included in the master problem.

- *Branch and check with analytical cuts.* We use (1.12) and analytical cuts (1.17) rather than nogood cuts. The decoupled subproblems are solved by `CP Optimizer`. The initial bounds (1.28) are again included in the master problem.

The results indicate that branch and check is clearly superior to the L-shaped method. It is already orders of magnitude faster in those few smaller instances where the L-shaped method could solve the problem within an hour. Perhaps not surprisingly, the analytic Benders cuts are almost always more effective than the nogood cuts. These data also confirm the hypothesis that the advantage of branch and check is greater when there are 4 facilities rather than 2, indeed dramatically greater as instance size

increases.

Table 1.2 probes algorithmic performance more deeply by comparing computation times and optimality gaps for seven algorithms. Three of the methods are described above, and the remaining four are as follows:

- *Deterministic equivalent MILP.* We solve the deterministic equivalent model (1.25) using the MILP engine in `CPLEX`, which we also use to solve the first stage of the other six models.

- *Integer L-shaped method with CP.* We modify the standard method by solving the second-stage subproblems with CP rather than MILP. Integer cuts are as before, and classical Benders cuts are derived from the LP relaxation of the MILP model as before. The initial bounds (1.28) and subproblem relaxation (1.20) are again included in the master problem.

- *Standard LBBD with nogood cuts.* We use (1.12) and unstrengthened nogood cuts (1.13). We solve the decoupled subproblems by `CP Optimizer`. The initial bounds (1.28) are included in the master problem for comparability with the integer L-shaped method.

- *Standard LBBD with analytical cuts.* We use (1.12) and analytical cuts (1.17) rather than nogood cuts. The decoupled subproblems are solved by `CP Optimizer`. The initial bounds (1.28) are again included in the master problem.

In addition to average computation time (in seconds), Table 1.2 reports the optimality gap obtained for each solution method, defined as (UB − LB)/UB. For the deterministic equivalent and branch-and-check methods, UB and LB are,

Table 1.2: Average computation time in seconds over 3 instances (upper half of table) and average relative optimality gap (lower half) for various solution methods, based on 10 jobs and 2 facilities.

Scenarios	Determ. equiv. MILP	Integer L-shaped method	Integer L-shaped with CP	LBBD Nogood cuts	LBBD Analytic cuts	B&Ch Nogood cuts	B&Ch Analytic cuts
1	2.4	127.3	27.9	2.0	0.6	1.5	0.9
5	475.8††	839.2	149.3	12.1	3.0	4.6	2.1
10	*	2316.9†	437.8	27.4	7.3	6.1	3.1
50	*	*	2517.8††	243.1	42.8	28.4	17.3
100	*	*	*	952.8	118.8	56.1	36.7
500	*	*	*	*	900.9	375.6	279.1
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	7.8	0.0	0.0	0.0	0.0	0.0	0.0
10	12.4	3.8	0.0	0.0	0.0	0.0	0.0
50	17.4	21.7	13.9	0.0	0.0	0.0	0.0
100	25.4	25.4	21.7	0.0	0.0	0.0	0.0
500	44.5	25.8	25.4	13.5	0.0	0.0	0.0

†Average excludes one instance that exceeded an hour in computation time.

††Average excludes two instances that exceeded an hour.

*All three instances exceeded an hour.

respectively, the upper and lower bounds obtained from CPLEX upon solution of the master problem. For standard LBBD, UB and LB are, respectively, the smallest subproblem optimal value and the largest master problem optimal value obtained during the Benders algorithhm.

As one might expect, the integer L-shaped implementations are faster than solving the deterministic equivalent MILP, because they exploit the scenario-based block structure of two-stage stochastic programs. We also see that the integer L-shaped method can be significantly accelerated by solving the exact subproblem with CP rather than MILP (to obtain upper bounds and generate the integer cut), since CP is more effective for this type of scheduling problem.

It is clear from Table 1.2 that all four implementations of LBBD substantially outperform the integer L-shaped method, even when the latter uses CP. Furthermore, the two branch-and-check implementations scale much better than standard LBBD, due mainly to time spent in solving the master problem in standard LBBD. This confirms the rule of thumb that branch and check is superior when solving the master problem takes significantly longer than solving the subproblems. The results also indicate that analytical Benders cuts are more effective than unstrengthened nogood cuts in both standard LBBD and branch and check.

Table 1.3 provides a more detailed comparison of the integer L-shaped method with the branch-and-check implementations. The L-shaped method with CP is shown, as we have seen that it is faster than solving the subproblem with MILP. Interestingly, solving a CP formulation of the subproblem is much faster than solving the LP relaxation of an MILP formulation. This illustrates the computational cost of using the larger MILP formulation. We also see that the stronger analytical cuts reduce the number of times the subproblem must be solved, and therefore the number of cuts generated and the resulting size of the master problem. Furthermore, the number of subproblem calls is roughly constant as the number of scenarios increases. Finally, the subproblem solutions consume about half of the total computation time in the branch-and-cut algorithms. Previous experience suggests that for best results, the computation time should, in fact, be about equally split between the master problem and subproblem (Ciré et al. 2016).

Given the computational burden of solving the LP relaxation of the MILP subproblem, we experimented with running the integer L-shaped method with only integer cuts. This obviates the necessity of solving the LP relaxation of an MILP model. The results appear in Table 1.4. The three implementations shown in the table are

Table 1.3: Analysis of the integer L-shaped method with CP subproblems and two branch-and-check algorithms. Each number is an average over 3 problem instances.

	Integer L-shaped with CP					B&C with nogood cuts				B&C with analytical cuts			
	Time (sec)			Statistics		Time (sec)		Statistics		Time (sec)		Statistics	
Scenarios	Total	CPsub	LPsub	Cuts	Calls	Total	CPsub	Cuts	Calls	Total	CPsub	Cuts	Calls
1	27.9	0.9	2.1	450	452	1.5	0.7	282	150	0.9	0.4	180	95
5	149.3	5.6	16.4	2692	541	4.6	2.1	1289	144	2.1	0.8	658	79
10	437.8	15.8	73.4	5114	515	6.1	3.1	2390	134	3.1	1.3	1243	75
50	2517.8†	97.3†	500.2†	20002†	401†	28.4	14.8	12616	148	17.3	7.6	7684	94
100	*	*	*	*	*	56.1	29.7	25880	152	36.7	16.0	15800	99
500	*	*	*	*	*	375.6	169.0	127404	150	279.1	76.9	76029	96

†Average excludes two instances that exceeded an hour.
*Computation terminated for all 3 instances after one hour.

exactly the same except for the cuts used and therefore permit a direct comparison of the effectiveness of the cuts. The integer L-shaped method actually runs faster using only integer cuts, without any classical Benders cuts obtained from the LP relaxation. We also see that the analytical cuts are much more effective than integer cuts, which are quite weak.

Finally, these data allow us to address the question, posed earlier, as to whether the advantage of branch and check relative to the L-shaped method increases with the number of scenarios. The advantage appears to be roughly constant for 2 facilities and perhaps increasing for 4 facilities, although the latter is uncertain because the L-shaped method (even with no LP relaxation) quickly times out.

We also experimented with a different distribution of processing times. We simulated a situation in which processing proceeds normally except when there is a delay due to mechanical breakdown or other causes. Accordingly, we defined processing time to be a random variable that is equal to the mean quantity specified above with 80%

Table 1.4: Performance of the integer L-shaped method with integer cuts only (no cuts from the LP relaxation).

		2 facilities			4 facilities		
Tasks	Scenarios	Integer L-shaped method	L-shaped integer cuts only	B&Ch analytic cuts	Integer L-shaped method	L-shaped integer cuts only	B&Ch analytic cuts
10	1	127.3	2.8	0.9	2405.0††	78.1	0.4
	5	839.2	9.0	2.1	*	906.5	2.4
	10	2316.9†	16.0	3.1	*	2213.0†	3.7
	50	*	87.2	17.3	*	*	17.4
	100	*	209.5	36.7	*	*	28.6
	500	*	1166.6	279.1	*	*	164.8
14	1	1831.6†	48.9	3.8	3016.6††	2403.9††	1.7
	5	*	229.5	16.9	*	2402.8††	4.8
	10	*	284.7	26.1	*	*	7.3
	50	*	1850.6	134.4	*	*	34.2
	100	*	2810.4††	357.9	*	*	83.2
	500	*	*	2424.3†	*	*	443.9
18	1	2416.2††	1358.6†	155.5	2404.7††	1346.8†	1.0
	5	*	3048.4††	988.4	*	2405.5††	20.7
	10	*	3477.2††	1888.3†	*	*	39.5
	50	*	*	*	*	*	187.8
	100	*	*	*	*	*	332.7
	500	*	*	*	*	*	2825.9††

†Average excludes one instance that exceeded an hour in computation time.

††Average excludes two instances that exceeded an hour.

*All three instances exceeded an hour.

probability, but 1.5 times as large with 15% probability, and 4 times as large as 5% probability. The results appear in Table 1.5. Comparison with Table 1.4 reveals that the relative advantage of branch and check is even greater with this processing time distribution than with the original one.

Table 1.5: Average computation time for the makespan problem with alternate processing times.

		2 facilities		4 facilities	
Jobs	Scenarios	L-shaped integer cuts only	B&C analytic cuts	L-shaped integer cuts only	B&C analytic cuts
10	1	18.5	0.3	3246.3	0.9
	5	141.3	2.6	*	1.9
	10	292.3	2.7	*	2.8
	50	2425.0	11.6	*	15.1
	100	*	22.3	*	21.5
	500	*	129.3	*	149.1
14	1	1971.2	5.2	1368.6	0.8
	5	*	18.1	*	3.7
	10	*	43.3	*	5.6
	50	*	241.8	*	26.2
	100	*	704.4	*	60.1
	500	*	3503.4	*	407.4
18	1	*	213.3	1770.1	2.7
	5	*	2190.4	*	10.1
	10	*	2932.3	*	31.5
	50	*	*	*	316.6
	100	*	*	*	634.3
	500	*	*	*	3162.8

†Average excludes one instance that exceeded an hour in computation time.
††Average excludes two instances that exceeded an hour.
*All three instances exceeded an hour.

1.7.2 Minimum Cost Problem

In this section, we present the results of the computational experiments on the minimum cost problem. We use the same instances we used for the minimum makespan problem, with the addition of costs and deadlines. The fixed cost associated with assigning jobs to facility i is drawn from a uniform distribution on the interval $[400/\alpha, 800/\alpha]$, where $\alpha = 25 - (i-1)(10/(m-1))$, so that the faster

facilities tend to be more expensive. The deadline d_j of job j is obtained as follows. Let $L = 20 \times n/m$. We set $d_j = r_j + \beta$ where r_j is the release time of job j and β is drawn from a uniform distribution on $[0.75 \times \alpha L,\ 1.25 \times \alpha L]$ with $\alpha = 2/3$.

Table 1.6: Average computation time in seconds over 3 instances for the minimum cost problem.

		2 facilities		4 facilities	
Jobs	Scenarios	Determ. equiv. MILP	B&Ch analytic cuts	Determ. equiv. MILP	B&Ch analytic cuts
10	1	0.3	0.1	0.2	0.2
	5	4.9	0.8	1.2	1.2
	10	3.7	1.7	2.7	1.7
	50	6.5	4.9	15.1	9.0
	100	19.2	6.1	18.5	15.5
	500	353.1	31.9	201.5	82.5
14	1	4.1	0.3	0.6	0.3
	5	19.0	1.5	2.8	1.5
	10	46.6	3.3	4.9	2.6
	50	85.5	9.1	24.7	10.5
	100	445.4	19.9	75.4	20.6
	500	*	156.9	844.0	130.0
18	1	23.2	32.8	2.8	1.8
	5	56.5	97.7	17.6	4.8
	10	416.8	128.6	38.4	11.3
	50	2527.9††	587.1	107.2	64.7
	100	3348.2††	1425.1†	335.1	133.9
	500	*	2526.0††	*	716.9

†Average excludes one instance that exceeded an hour in computation time.
††Average excludes two instances that exceeded an hour.
*All three instances exceeded an hour.

We compare LBBD performance solely with the deterministic equivalent MILP formulation, since the type of integer cut used in the L-shaped method is an optimality cut and is not defined for infeasible subproblems. The MILP model is the

same as (1.25) except that the objective function is replaced by $\sum_{i\in I}\sum_{j\in J}\phi_{ij}x_{ij}$, β variables are eliminated, and constraint (i) is modified to reflect two-sided time windows.

As is evident in Table 1.6, the deterministic equivalent MILP performs better on this problem than on the minimum makespan problem. However, the branch-and-check method scales better than the MILP formulation and is superior for solving the larger instances.

1.7.3 Minimum Tardiness Problem

In this section, we present the results of the computational experiments on the minimum tardiness problem. We use the same instances as for the minimum makespan problem, with due dates added. The due dates are obtained in the same fashion as the deadlines for the minimum cost problem, except that we set α equal to 1/3 rather than 2/3.

The MILP model we used for the integer L-shaped method is the same as (1.25) except that constraint (i) is modified to reflect two-sided time windows, constraints (c) and (e) are replaced by the following:

$$\beta_\omega \geq \sum_{i\in I}\sum_{j\in J}\beta_{ij\omega}, \;\; \omega \in \Omega \qquad (c)$$

$$\beta_{ij\omega} \geq \sum_{t\in T}(t + p^\omega_{ij})z^\omega_{ijt} - d_j, \;\; \beta_{ij\omega} \geq 0, \;\; i \in I,\; j \in J,\; \omega \in \Omega \qquad (e)$$

The implementation of the integer L-shaped method is otherwise identical to the one applied to the minimum makespan problem. We use the logic-based cuts and the simpler subproblem relaxation (1.24) presented in Section 1.5.3.

Table 1.7: Average computation time in seconds over 3 instances for the minimum tardiness problem.

		2 facilities			4 facilities		
Jobs	Scenarios	Determ. equiv. MILP	L-shaped integer cuts only	B&Ch analytic cuts	Determ. equiv. MILP	L-shaped integer cuts only	B&Ch analytic cuts
10	1	3.1	4.0	2.7	2.1	*	7.5
	5	36.7	8.5	4.8	7.9	*	31.6
	10	868.4	19.7	10.8	29.1	*	60.2
	50	2614.5††	100.3	55.1	248.4	*	280.0
	100	2787.5††	215.8	110.1	1312.3	*	1108.8
	500	*	1262.3	641.9	*	*	*
14	1	4.7	7.9	2.6	2.9	*	9.9
	5	1307.9	71.5	14.6	21.4	*	25.5
	10	2431.8††	94.1	16.3	45.4	*	36.9
	50	*	387.6	92.0	1574.1†	*	374.3
	100	*	969.1	152.4	2945.2††	*	782.7
	500	*	2765.1††	762.7	*	*	2605.2†
18	1	5.8	18.1	5.3	3.6	*	7.8
	5	250.0	12.8	6.3	162.6	*	24.3
	10	599.5	94.3	25.4	1092.6	*	37.3
	50	*	2415.1††	1075.9	*	*	158.2
	100	*	2408.1††	1573.3†	*	*	382.8
	500	*	2429.6††	2532.0††	*	*	2390.4†

†Average excludes one instance that exceeded an hour in computation time.

††Average excludes two instances that exceeded an hour.

*All three instances exceeded an hour.

The results in Table 1.7 shows that the branch-and-check method clearly outperforms both of the benchmark methods. Interestingly, the integer L-shaped method is unable to solve any of the instances even without the overhead created by an LP relaxation.

1.8 Conclusion

In this study, we applied logic-based Benders decomposition (LBBD) to two-stage stochastic optimization with a scheduling task in the second stage. While Benders decomposition is often applied to such problems, notably in the integer L-shaped method, the necessity of generating classical Benders cuts requires that the subproblem be formulated as a mixed-integer/linear programming problem and cuts generated from its continuous relaxation. We observed that this process incurs substantial computational overhead that LBBD avoids by generating logic-based cuts directly from a constraint programming model of the scheduling subproblem. Although the integer cuts used with the L-shaped method can be regarded as a special case of logic-based Benders cuts, they are extremely weak, even weaker than simple nogood cuts often used in an LBBD context. Furthermore, the type of subproblem analysis that has been used for past applications of LBBD permits much stronger logic-based cuts to be derived, again without the overhead of obtaining a continuous relaxation.

Computational experiments found that, due to these factors, LBBD solves a generic stochastic planning and scheduling problem much more rapidly than the integer L-shaped method. The speedup is several orders of magnitude for the minimum makespan problem when a branch-and-check variant of LBBD is used. Branch and check is also superior when minimizing assignment cost or total tardiness, although its

advantage for the minimum cost problem is less pronounced. These outcomes suggest that LBBD could be a promising approach to other two-stage stochastic and robust optimization problems with integer or combinatorial recourse, particularly when the subproblem is relatively difficult to model as an integer programming problem.

Acknowledgement. This chapter is published in INFORMS Journal of Computing, see Elçi and Hooker (2022).

Chapter 2

On Logic-Based Benders Decomposition and Sequence-Dependent Scheduling

This chapter is a joint work with John Hooker.

2.1 Introduction

Sequence-dependent scheduling has played a fundamental role in designing manufacturing and business systems (Allahverdi, 2015). In many practical applications, there are several facilities (machines) available to process tasks and a setup time is needed to prepare the facilities between each task. When the setup time is independent of the sequence of the tasks, it can easily be captured as part of the processing time. However, the setup time in many real-world applications is sequence-dependent

(Allahverdi and Soroush, 2008) and it is crucial to develop models and algorithms that can account for the setup time correctly.

In this paper, we focus on a parallel (unrelated) machine scheduling problem (PMSP) with sequence-dependent setup times in which the objective is makespan minimization. Using the three-field notation introduced by Graham et al. (1979), the problem that we study belongs to the class of $(R, ST_{\text{sd}}, C_{\max})$. For greater generality, we assume that each task must be processed within a time window, i.e, the release times and deadlines of the tasks can be different for each task.

Parallel machine scheduling problem with sequence- dependent setup times and hard time windows (PMSP-TW) is notoriously a very difficult problem to solve. It is NP-Hard (Lenstra et al., 1977) and exact solution methods are scarce in the literature. Most studies focus on developing a heuristic method (see, e.g., Ying et al., 2012; Lin and Ying, 2014).

Among the exact methods, Tran et al. (2016) develop a logic-Based Benders decomposition (LBBD) algorithm to solve PMSP without time windows. In our study, we extend their work and derive novel logic-based Benders cuts for the case with hard time-windows. We summarize our contributions as follows:

- We analyze the LBBD cuts proposed in Hooker (2007) and Elçi and Hooker (2022).
  - We improve the LBBD cuts in Elçi and Hooker (2022) that are proposed for the planning and scheduling problem with no deadlines but with non-zero release times.
  - We show that the LBBD cuts proposed in this study and in Hooker (2007)

are tight, i.e., cannot be dominated by another set of LBBD cuts.

  - We introduce another set of LBBD cuts for the planning and scheduling problem considered in Elçi and Hooker (2022).

- We develop LBBD cuts for PMSP-TW.

  - This cut generalizes the some of the LBBD cuts proposed earlier in the literature. First, it generalizes the cuts proposed in Hooker (2007) and Elçi and Hooker (2022) so that the scheduling problem can feature both non-zero release times and hard deadlines. Furthermore, it generalizes the LBBD cuts proposed in Tran et al. (2016) to a setup with hard time windows.

- We develop a branch-and-check algorithm for PMSP-TW. Our method is one of the very few exact methods to solve the PMSP with both release times and due dates.

- We present an extensive computational study that shows the effectiveness of the proposed solution method in various settings.

  - We show that our new LBBD cuts for the planning and scheduling problem considered in Elçi and Hooker (2022) together with the improved version of the existing cuts perform better.

  - We demonstrate that the proposed branch-and-check algorithm for PMSP-TW performs better than the benchmark method.

The rest of the paper is organized as follows. In the next section, we review the relevant literature. In Section 2.3 we describe PMSP-TW in detail. The next two

chapters are devoted to the analysis of the planinng and scheduling problem and PMSP-TW, respectively. We present the computational experiments in Section 2.6 and conclude our paper in Section 2.7.

2.2 Previous Work

Our work is closely related to two bodies of literature: (i) sequence-dependent machine scheduling and (ii) logic-based Benders decomposition for scheduling and routing. In this section, we review the relevant papers and summarize our contributions to each body of literature.

2.2.1 Sequence-Dependent Machine Scheduling

The setup cost/time is a very crucial in designing scheduling systems. Allahverdi and Soroush (2008) discusses the role the setup cost/time play in today's modern manufacturing systems. Accordingly, there is a growing body of literature that focuses on modeling and algorithm design for scheduling problems with sequence-dependent setup cost/time. We refer the readers to Allahverdi (2015) for an excellent review on scheduling problems with setup time/cost.

In this study, we focus on PMSP-TW that incorporates setup time as part of the scheduling problem. PMSP-TW generalizes PMSP so as to feature hard time windows for each task. The objective function that we consider is makespan minimization. Within the PMSP literature, there are several studies that focus on makespan minimization. Most of the studies develop a heuristic/metaheuristic method. Among these, we name hybrid artificial bee colony algorithm (Lin and Ying, 2014), and a restricted simulated annealing algorithm (Ying et al., 2012).

Both of these studies assume that the release times are equal to zero and there are no deadlines.

The literature on sequence-dependent machine scheduling with time windows is scarce. Ying and Lin (2012) study PMSP with sequence-dependent setup times and deadline constraints. They propose a heuristic artificial-bee-colony-based algorithm to solve this problem. Jula and Rafiey (2012) consider a single machine with sequence-dependent setup times and strictly enforced time-window constraints on the start times of each task. They develop a heuristic algorithm based on network flow methods.

The literature on exact solution methods for PMSP is scarce too. Rocha et al. (2008) focus on a scheduling problem with unrelated parallel machines, sequence and machine-dependent setup times, due dates and weighted tasks. The objective function is to minimize the sum of the makespan and weighted total tardiness. They develop an exact branch-and-bound algorithm. Tran et al. (2016) focus on PMSP where the objective is to minimize makespan. They develop an exact branch-and-check method.

Our main contributions to the sequence-dependent machine scheduling literature is to introduce a PMSP model that features hard time windows and propose an exact solution method based on the logic-based Benders decomposition framework.

We finish this section by noting that PMSP-TW is very closely related to vehicle routing problem with time windows (VRP-TW). There is a very rich literature on VRP-TW, we refer the reader to Toth and Vigo (2002) for an excellent overview. Our work is a step forward to devise an exact decomposition-based method for VRP-TW.

2.2.2 Logic-Based Benders Decomposition

Logic-based Benders decomposition (LBBD) is a well celebrated framework that offers a method to solve large-scale combinatorial optimization problems exactly. It is introduced by Hooker (2000a) and further studied in Hooker and Ottosson (2003). Branch-and-check is a variant of LBBD which is also introduced in Hooker (2000a) and the first computational experiments are presented in Thorsteinsson (2001).

LBBD framework has found use for exact solution methods in applications including planning and scheduling problem (Hooker, 2007), plant location problem (Fazel-Zarandi and Beck, 2012), and network design (Solak et al., 2014). We refer the readers to Hooker (2019b) for an overview on LBBD for large-scale optimization.

Within the LBBD literature, Tran et al. (2016) is the closest study to our work. They develop a branch-and-check method to solve PMSP. Furthermore, Hooker (2007) and Elçi and Hooker (2022) are related to our study in that we improve and generalize the logic-based Benders cuts proposed in those studies.

2.3 The Problem

We consider a generic sequence-dependent parallel machine scheduling problem. The objective is to minimize makespan. We suppose that each task $j \in J$ has a processing time p_{ij} on facility $i \in I$. In addition, each task $j \in J$ is available within a time window denoted by $[r_j, d_j]$. In our notation, r_j and d_j denote the release time and deadline of each task $j \in J$, respectively. We assume that there are no precedence constraints among tasks. Furthermore, each facility (machine) is available all the time and can handle one task at a time with no preemption.

The setup time needed to prepare facilities is sequence and facility dependent. We have that s_{ijk} denotes the setup time if task k is processed right after task j on facility i. We assume that the setup times follow the triangle inequality: $s_{ijk} \leq s_{ijl} + s_{ilk}$. This assumption is common in the literature, see Kohl et al. (1999) for a discussion on time-related triangular inequality.

In the remainder of this section, we present two exact solution methods to solve PMSP-TW. The first one is a mixed-integer programming model (MIP) that is similar to the MIP formulation given in Toth and Vigo (2002) for VRP-TW. The second method is a branch-and-check algorithm that relies on our analysis of the sequence-dependent scheduling problem in Section 2.5.

2.3.1 MIP Formulation of PMSP-TW

In this section, we present an MIP model for PMSP-TW. We first define the decision variables of our problem as follows:

$$y_{ijk} = \begin{cases} 1, & \text{if task } k \in \bar{J} \text{ is processed directly after task } j \in \bar{J} \text{ on facility } i \in I \\ 0, & \text{otherwise} \end{cases}$$

$$w_{ij} = \quad \text{the start time of processing of task } j \in \bar{J} \text{ on facility } i \in I.$$

We suppose that $J = \{1, \ldots, n\}$ denote the set of tasks. Let 0 and $n+1$ be dummy tasks and $\bar{J} \coloneqq \{0, \ldots, n+1\}$. Let $J^+(j)$ and $J^-(j)$ denote the set of tasks that can succeed and precede a given task $j \in \bar{J}$, respectively. By definition, the dummy task 0 cannot succeed any task, and the dummy task $n+1$ cannot precede any task.

Using the above notation, we next present the MIP model.

$$\text{minimize} \quad C_{\max} \tag{2.1a}$$

$$\text{subject to} \quad C_{\max} \geq w_{i,n+1}, \quad i \in I, \tag{2.1b}$$

$$\sum_{i \in I} \sum_{k \in J^+(j)} y_{ijk} = 1, \quad j \in J, \tag{2.1c}$$

$$\sum_{j \in J^+(0)} y_{i0j} = 1, \quad i \in I, \tag{2.1d}$$

$$\sum_{k \in J^+(j)} y_{ijk} - \sum_{k \in J^-(j)} y_{ikj} = 0, \quad i \in I,\ j \in J, \tag{2.1e}$$

$$\sum_{j \in J^-(n+1)} y_{ij,n+1} = 1, \quad i \in I, \tag{2.1f}$$

$$w_{ij} + p_{ij} + s_{ijk} - w_{ik} \leq \big(1 - y_{ijk}\big) M_{ijk}, \quad i \in I,\ j \in J,\ k \in J^+(j), \tag{2.1g}$$

$$r_j \sum_{k \in J^+(j)} y_{ijk} \leq w_{ij} \leq d_j \sum_{k \in J^+(j)} y_{ijk}, \quad i \in I,\ j \in \bar{J}, \tag{2.1h}$$

$$w_{ij} \geq 0, \quad i \in I,\ j \in \bar{J}, \tag{2.1i}$$

$$y_{ijk} \in \{0, 1\}, \quad i \in I,\ j \in J,\ k \in J^+(j). \tag{2.1j}$$

The objective function (2.1a) minimizes the makespan. We make sure that each task is assigned to a facility via constraint set (2.1c). Constraint sets (2.1d) – (2.1f) make sure that the tasks are correctly scheduled. Constraint sets (2.1g) and (2.1h) guarantee that time-window considerations are respected. The non-negativity and binary restrictions are given in constraint sets (2.1i) and (2.1j).

The time-window constraints given in (2.1g) features non-linear terms. We linearize

such terms using the fact that variables $\mathbf{y}$ are binary as follows:

$$w_{ij} + p_{ij} + s_{ijk} - w_{ik} \leq \big(1 - y_{ijk}\big) M_{ijk}, \quad i \in I,\ j \in J,\ k \in J^{+}(j),$$

Here M_{ijk} are large constants (Toth and Vigo, 2002). We let $M_{ijk} = \max\{d_j + s_{ijk} + p_{jk} - r_k, 0\}$.

2.3.2 Logic-Based Benders Decomposition for PMSP-TW

In this section, we show how we use logic-based Benders decomposition framework to solve PMSP-TW. We define the following decision variables to be used in the master problem:

$$x_{ij} = \begin{cases} 1, & \text{if task } j \in J \text{ is served by facility } i \in I \\ 0, & \text{otherwise} \end{cases}$$

To this end, we have the following master problem.

$$\begin{aligned}
\text{minimize} \quad & \beta && \text{(2.2a)}\\
\text{subject to} \quad & \beta \geq \beta_i, \quad i \in I, && \text{(2.2b)}\\
& \sum_{i \in I} x_{ij} = 1, \quad j \in J, && \text{(2.2c)}\\
& \text{subproblem relaxation}, && \text{(2.2d)}\\
& 0 \geq B^{\text{fea}}_{i,\bar{\mathbf{x}}}(\mathbf{x}) \quad \text{for all } \bar{\mathbf{x}} \text{ that yields infeasibility}, && \text{(2.2e)}\\
& \beta_k \geq B^{\text{opt}}_{i,\bar{\mathbf{x}}}(\mathbf{x}), \quad i \in I, && \text{(2.2f)}\\
& x_{ij} \in \{0,1\}, \quad j \in J,\ i \in I. && \text{(2.2g)}
\end{aligned}$$

Let $J_i(\bar{\mathbf{x}})$ denote the set of tasks assigned to facility i at a given iteration. The subproblem decouples into separate facility-specific problems. We model the subproblem as a constraint programming model (CP) model. The subproblem for facility $i \in I$ has the following form.

$$\text{minimize} \quad M \tag{2.3a}$$

$$\text{subject to} \quad M \geq b_j + p_{ij}, \quad j \in J_i(\bar{\mathbf{x}}, \tag{2.3b}$$

$$\texttt{noOverlap}\Big(\big(b_j \,\big|\, j \in J_i(\bar{\mathbf{x}})\big),\, \big(p_{ij} \,\big|\, j \in J_i(\bar{\mathbf{x}})\big),\, \big(s_{ijk} \,\big|\, j,k \in J_i(\bar{\mathbf{x}})\big)\Big), \tag{2.3c}$$

$$b_j \in [r_j, d_j - p_{ij}],\ j \in J_i(\bar{\mathbf{x}}) \tag{2.3d}$$

In this formulation, $b_j \in J$ denote the start time of task j. `noOverlap` is a constraint available in many commercial CP solvers. Furthermore, some solvers allow the modelling of sequence-dependent scheduling via a `noOverlap` constraint. We refer the readers to Laborie et al. (2018) for a detailed explanation of the modeling of sequence-dependent single-facility scheduling problem via `IBM CP Optimizer`.

The above master problem is a reformulation of the original problem (2.1). It relies on the existence of valid Benders feasibility and optimality cuts shown in equations (2.2e) and (2.2f), respectively. In the next section, we describe the details of the branch-and-check algorithm that makes use of the LBBD-based formulation (2.2).

2.3.3 Outline of the Branch-and-Check Algorithm

We propose the following branch-and-check algorithm to solve PMSP-TW. The above pseudo-code summarizes the major steps of our branch-and-check algorithm. In the

```
Initialization;
Invoke Solver to solve the relaxed master problem;
while Solver determines that optimality gap is greater than the threshold do
    Identify a new candidate incumbent solution x̄ and β̄;
    for i ∈ I do
        Update the subproblem formulation (2.3) based on J_i(x̄);
        Solve the subproblem;
        if If the subproblem is infeasible then
            Add a feasibility cut;
            Break the for-loop and continue with a new candidate incumbent
             solution;
        else
            Let SP_i(x̄) denote the optimal value of the subproblem;
            if β̄_i < SP_i(x̄) then
                Add an optimality cut;
            end
        end
    end
end
```

Algorithm 1: Branch-and-check algorithm for PMSP-TW.

remainder of this section, we provide some implementation details of our proposed algorithm. We defer the discussion on optimality cuts to Section 2.5 as the derivation of the optimality cuts requires an analysis of the combinatorial structure of the sequence-dependent scheduling problem.

We use the following feasibility cuts whenever an infeasible subproblem is idendified.

$$B^{\text{fea}}_{i,\bar{\mathbf{x}}}(\mathbf{x}) = 1 - \sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij}). \tag{2.4}$$

Furthermore, we stop solving the subproblems corresponding to the other facilities, once an infeasible subproblem is identified.

Lastly, the following inequalities are valid for the master problem (Hooker, 2007).

$$\beta_i \geq \sum_{j \in J} p_{ij} x_{ij} \tag{2.5}$$

We use the above inequalities as subproblem relaxation as part of the relaxed master problem during the initialization stage of our branch-and-check algorithm

2.4 Revisiting the Planning and Scheduling Problem

In this section, we revisit the planning and scheduling problem studied in Hooker (2007) and Elçi and Hooker (2022). We begin by presenting two lemmas that establish valid lower bounds for the makespan problem.

Lemma 7. *(Elçi and Hooker, 2022) Consider a minimum makespan problem P in which each task $j \in J$ has release time r_j and processing time p_j, with no deadlines. Let M^* denote the minimum makespan for P, and $\hat{M}$ the minimum makespan for the problem $\hat{P}$ that is identical to P except that the tasks in a nonempty set $\bar{J} \subset J$ are removed. Then*

$$M^* - \hat{M} \leq \Delta + r^+ - r^- \tag{2.6}$$

where $\Delta = \sum_{j \in \bar{J}} p_j$, and $r^+ = \max_{j \in J}\{r_j\}$ and $r^- = \min_{j \in J}\{r_j\}$ are the latest and earliest release times of the tasks in set J.

Lemma 8. *(Hooker, 2007) Consider a minimum makespan problem P in which each task $j \in J$ has deadline d_j and processing time p_j. Assume that all tasks are released at time 0. Let M^* denote the minimum makespan for P, and $\hat{M}$ the minimum makespan for the problem $\hat{P}$ that is identical to P except that the tasks in a nonempty set $\bar{J} \subset J$ are removed. Then*

$$M^* - \hat{M} \leq \Delta + d^+ - d^- \tag{2.7}$$

where $\Delta = \sum_{j \in \bar{J}} p_j$, and $d^+ = \max_{j \in J}\{d_j\}$ and $d^- = \min_{j \in J}\{d_j\}$ are the latest and earliest deadline of the tasks in set J.

We next show how to improve Lemma 7.

Lemma 3 - improved. Consider the same problem in Lemma 7. We have that

$$M^* - \hat{M} \leq \Delta + C \tag{2.8}$$

where $C = \max\{0,\ r^+ - r^- - p^-\}$ with $\Delta = \sum_{j \in \bar{J}} p_j$, $r^+ = \max_{j \in J}\{r_j\}$, $r^- =$

$\min_{j\in J}\{r_j\}$ and $p^- = \min_{j\in J}\{p_j\}$.

Proof. Consider any solution of $\hat{P}$ with makespan $\hat{M}$. We will construct a feasible solution for P by extending this solution.

There are two cases to consider. Suppose that $r^+ - r^- - p^- > 0$. If $\hat{M} > r^+$, we schedule all the tasks in $\bar{J}$ sequentially starting from time $\hat{M}$. This is a feasible solution for P, and we have $M^* \leq \hat{M} + \Delta$. The lemma follows because $r^+ - r^- - p^- > 0$. If $\hat{M} < r^+$, we schedule all the tasks in $\bar{J}$ sequentially starting from time r^+. Again this is a feasible solution for P and we have that $M^* \leq r^+ + \Delta$. This implies

$$M^* - \hat{M} \leq r^+ - \hat{M} + \Delta$$

The lemma follows because $\hat{M}$ is at least $r^- + p^-$.

Now consider the case $r^+ - r^- - p^- < 0$. Since $r^- + p^- > r^+$, we have that $\hat{M} > r^+$. We schedule all the tasks in $\bar{J}$ sequentially after $\hat{M}$. This is a feasible solution for P, and we have that $M^* \leq \hat{M} + \Delta$. The lemma follows since $C = 0$. □

This bound yields a logic-based Benders cut for the stochastic planning and scheduling problem studied in Elçi and Hooker (2022).

Theorem 9. *The following is a valid Benders cut for the minimum makespan problem with no deadlines.*

$$\beta_{i\omega} \geq \left\{ \begin{array}{ll} \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \Big(\sum_{j\in J_i(\bar{\mathbf{x}})} (1 - x_{ij}) p^{\omega}_{ij} + C \Big), & \text{if } x_{ij} = 0 \text{ for some } j \in J_i(\bar{\mathbf{x}}) \\ \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}), & \text{otherwise} \end{array} \right\}, \; i \in I \tag{2.9}$$

where $J_i(\bar{\mathbf{x}}) = \{j \in J : x_{ij} = 1\}$, $C = \max\{0,\ r^+ - r^- - p^-\}$ *with* $r^+ = \max_{j \in J_i(\bar{\mathbf{x}})}\{r_j\}$, $r^- = \min_{j \in J_i(\bar{\mathbf{x}})}\{r_j\}$ *and* $p^- = \min_{j \in J_i(\bar{\mathbf{x}})}\{p_j\}$.

In order to linearize the cut (2.9), we first establish the following lemma.

Lemma 10. *Consider the following non-linear Benders cut.*

$$\beta \geq \left\{ \begin{array}{ll} \mathrm{SP}(\bar{\mathbf{z}}) - \Big(\sum_{j \in J(\bar{\mathbf{z}})} (1 - z_j) c_j + C \Big), & \text{if } z_j = 0 \text{ for some } j \in J(\bar{\mathbf{z}}) \\ \mathrm{SP}(\bar{\mathbf{z}}), & \text{otherwise} \end{array} \right\} \tag{2.10}$$

where $\bar{\mathbf{z}}$ *is a given binary solution of the master problem,* $\mathrm{SP}(\bar{\mathbf{z}})$ *is the optimal value of the subproblem and* $J(\bar{\mathbf{z}}) = \{j \in J : z_j = 1\}$. *Suppose that the constants* c_j *and* C *are non-negative constants. Then the cut can be linearized as follows:*

$$\begin{aligned} &\beta \geq \mathrm{SP}(\bar{\mathbf{z}}) - \sum_{j \in J(\bar{\mathbf{z}})} (1 - z_j)(c_j + C) && (a) \\ &\beta \geq \mathrm{SP}(\bar{\mathbf{z}}) - \sum_{j \in J(\bar{\mathbf{z}})} (1 - z_j) c_j - C && (b) \end{aligned} \tag{2.11}$$

Proof. Let $k = \sum_{j \in J(\bar{\mathbf{z}})} (1 - z_j)$. If $k = 0$, (2.11a) is identical to the second line of (2.10). Furthermore, (2.11b) is dominated by (2.11a). Therefore (2.11) is valid. If $k = 1$, both (2.11a) and (2.11b) are identical to the first line of (2.10) and therefore valid. If $k \geq 2$, (2.11b) is identical to the first line of (2.10), while (2.11a) is dominated by (2.11b). Therefore (2.11) is valid. □

The lemma above is powerful to linearize any logic-based Benders cut that has the same non-linear structure of (2.10), see, for example, Corollary 1 and 2 in Elçi and

Hooker (2022). We also linearize the cut (2.9) using Lemma 10.

$$\begin{aligned} \beta_{i\omega} &\geq \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij})(p_{ij}^{\omega} + C) \qquad (a) \\ \beta_{i\omega} &\geq \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij}) p_{ij}^{\omega} - C \qquad (b) \end{aligned} \tag{2.12}$$

Corollary 11. *The inequalities* (2.12) *yield a valid Benders cut equivalent to* (2.9)*.*

In the remainder of this section, we will show that the lower bounds presented above cannot be improved.

We say that a lower bound for a logic-based cut is *tight* if there is an instance for which the lower bound holds as an equality. If the lower bound is tight for a logic-based cut, it means that it is the best cut one can find (more correctly, it cannot be dominated by another cut).

Theorem 12. *The lower bound given in* (2.8) *is tight.*

Proof. Let $J = \{1, 2\}$. Suppose that $r_1 < r_2$ and $p_1 < p_2$. We have that $r^+ = r_2$, $r^- = r_1$, and $p^- = p_1$. Let $\bar{J} = \{2\}$ so that we remove task 2 from the set J. We have that $M^* = r_2 + p_2$ and $\hat{M} = r_1 + p_1$. Suppose that $r_1 + p_1 < r_2$. This implies that $C = r^+ - r^- - p^- > 0$.

The lower bound from in (2.8) states that

$$M^* - \hat{M} \leq \Delta + C$$

Plugging in the values for this instance show that the left hand side is equal to the right hand side. □

We next establish a similar result for Lemma 8.

Theorem 13. *The lower bound given in Lemma 8 is asymptotically tight when* $d^+ = \min\left\{ \max_{j\in J}\{d_j\},\ \sum_{j\in J} p_j\right\}$.

Proof. First, it is easy to see that the modified d^+ given above does not affect the makespan problem since $\sum_{j\in J} p_j$ is a trivial upper bound on the makespan problem when release times are all equal to 0.

Let $J = \{1, 2, 3\}$. Suppose that $\mathbf{d} = (2, 1, \infty)$, $\mathbf{p} = (1, 1, 2)$, $\mathbf{c} = (2, 1, 1)$, and $C = 2$. We have that $M^* = p_1 + p_2 + p_3$. Let $\bar{J} = \{1\}$. Then, $\hat{M} = p_3$, $\Delta = p_1$, $d^+ = p_1 + p_2 + p_3$, and $d^- = d_2$. This example, presented in Hooker (2007), shows that $M^* - \hat{M} \leq \Delta$ is not a valid bound when the deadlines differ. In particular, when we remove task 1, the makespan decreases by 2 unit despite the fact that the processing time of task 1 is equal to 1. This happens because the deadline of task 1 forces us to schedule it before task 3. In the absence of task 1, task 2 and 3 can be processed simultaneously, hence the makespan decreases more than the processing time of task 1.

Observe that for this instance, the bound given in (2.7) yields

$$(p_1 + p_2 + p_3) - p_3 \leq p_1 + (p_1 + p_2, p_3) - d_2.$$

We have that

$$-p_3 = -2 < p_1 - d_2 = 0 \tag{2.13}$$

is not tight.

We next show that the difference between the left and the right hand side of (2.13) can get arbitrarily small. Consider the above instance. We keep d_2 and p_2 the same, but we simultaneously decrease p_1, d_1 and p_3 by the same amount. We observe that task 1 still needs to be processed before task 3, because the deadline and the processing time of task 1 are both getting smaller, and the processing time of task 1 and 3 are decreasing at the same rate. Under this construction, we still have $M^* = p_1 + p_2 + p_3$, $\hat{M} = p_3$, $\Delta = p_1$, $d^+ = p_1 + p_2 + p_3$, and $d^- = d_2 = p_2$.

We see that as $p_1 \to 0$, $d_1 \to 1$, and $p_3 \to 1$, both the left and right hand side of (2.13) approach to -1. This shows that the cut is (almost/asymptotically) tight. □

We next derive another class of logic-based Benders cuts for the makespan problem studied in Elçi and Hooker (2022).

Lemma 14. *Consider the same problem in Lemma 7. We have that*

$$M^* - \hat{M} \leq \sum_{j \in \bar{J}} \gamma_j \tag{2.14}$$

where $\gamma_j = p_j + C_j$ *with* $C_j = (r_j - r^- - p^-)^+$ *for all* $j \in \bar{J}$.

Proof. Consider the same construction in the proof of Lemma 7-improved.

Suppose without loss of generality that each task has a distinct release time. Let $k = \arg\max_{j \in J}\{r_j\}$. If $k \in \bar{J}$, then the bound (2.14) is dominated by the bound (2.6). Otherwise, we know that $k \in \hat{J}$. This implies that $\hat{M} > r^+$. We schedule all

the tasks in $\bar{J}$ sequentially starting from time $\hat{M}$. This is a feasible solution for P, and we have $M^* \leq \hat{M} + \sum_{j \in \bar{J}} p_j$ that implies the bound (2.14). □

The above lemma establishes the following logic-based Benders cuts.

Theorem 15. *The following is a valid Benders cut for the minimum makespan problem with no deadlines*

$$\beta_{i\omega} \geq \mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} \gamma_j (1 - x_{ij}) \tag{2.15}$$

where $\gamma_j = p_j + C_j$ *with* $C_j = (r_j - r^- - p^-)^+$ *for all* $j \in J_i(\bar{\mathbf{x}})$.

Remark 16. *Neither cut dominates the other. Clearly,* (2.9) *is tight by Theorem 12, therefore cannot be dominated. On the other hand, the cut* (2.15) *can be better than* (2.9) *when fewer tasks are removed, see Example 17 below. We show in our computational study that it can be advantageous to use both cuts.*

Example 17. *Let* $J = \{1, 2, 3, 4\}$. *Let* $p_1 = p_2 = p_3 = p_4 = p$. *Suppose that* $r_1 = r_2 = r_3 = 0$ *and* $r_4 > p$. *We have that the constant* C *in the cut* (2.9) *is equal to* $(r_4 - 0 - p)^+ = r_4 - p$. *Furthermore, the constants in the cut* (2.15) *are given by*

$$C_1 = (r_1 - 0 - p)^+ = 0$$
$$C_2 = (r_1 - 0 - p)^+ = 0$$
$$C_3 = (r_1 - 0 - p)^+ = 0$$
$$C_4 = (r_4 - 0 - p)^+ = r_4 - p > 0$$

Let $\bar{J} = \{1\}$ *so that we remove task* 1 *from the set* J*. We see that the lower bound obtained from* (2.9) *is given by*

$$\mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - (r_4 - p) - p$$

is strictly lower than the bound

$$\mathrm{SP}_{i\omega}(\bar{\mathbf{x}}) - p$$

that is obtained from (2.15)*.*

2.5 Makespan Problem with Sequence-Dependent Setup Times

In this section, we analyze the sequence-dependent scheduling problem with time windows. The objective of the scheduling problem is makespan minimization. The analysis of this problem is crucial to derive logic-based Benders cuts for PMSP-TW. We use these logic-based Benders cuts within our branch-and-check algorithm.

Suppose that at a given iteration of our branch-and-check algorithm, we have $\bar{\mathbf{x}}$ as a candidate incumbent solution. Recall that $J_i(\bar{\mathbf{x}})$ denotes the set of tasks assigned to facility i. Let $\mathrm{SP}_i(\bar{\mathbf{x}})$ denote the optimal makespan on facility i. Theorem 18 establishes our main result.

Theorem 18. *The following is a valid Benders cut for facility* i*.*

$$\beta_i \geq \begin{cases} \mathrm{SP}_i(\bar{\mathbf{x}}) - \left(\sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij})\gamma_j + C^1 + C^2\right), & \textit{if one or more tasks are removed} \\ \mathrm{SP}_i(\bar{\mathbf{x}}), & \textit{otherwise} \end{cases}$$

where $\gamma_j = p_{ij} + \alpha_j$, $C^1 = (r^+ - r^- - p^-)^+$, *and* $C^2 = d^+ - d^-$ *with* $\alpha_j = \max_{k \in J_i(\bar{\mathbf{x}}) \setminus \{j\}} \{s_{kj}\}$ *for all* $j \in J_i(\bar{\mathbf{x}})$, $r^+ = \max_{j \in J_i(\bar{\mathbf{x}})} \{r_j\}$, $r^- = \min_{j \in J_i(\bar{\mathbf{x}})} \{r_j\}$, $d^+ = \max_{j \in J_i(\bar{\mathbf{x}})} \{d_j\}$, $d^- = \min_{j \in J_i(\bar{\mathbf{x}})} \{d_j\}$, $p^- = \min_{j \in J_i(\bar{\mathbf{x}})} \{p_{ij}\}$.

Proof. Let P be a minimum makespan problem where the tasks in N are assigned to facility i at a given iteration, i.e., $N = J_i(\bar{\mathbf{x}})$. Denote the optimal makespan with C_N. Suppose that the cut is not valid for a particular assignment decision where the tasks in N^* are assigned to facility i. We consider the case where N^* and N are not equal to each other and they are not mutually exclusive, because it is trivial to see that the cut is valid in these two cases.

Let $\hat{N} = N \setminus N^*$ denote the set of tasks that are removed from facility i. Because the cut is not valid, we have that

$$C_{N^*} < C_N - \sum_{j \in \hat{N}} \gamma_j - C^1 - C^2 \tag{2.16}$$

We take the optimal solution of the makespan problem where the tasks N^* are scheduled on facility i. Remove the tasks in $N^* \setminus N$ from this solution. The remaining tasks are only the ones in $\bar{N} = N \cap N^*$. This solution is a feasible scheduling of tasks in $\bar{N}$. Denote the makespan of this solution by $C_{\bar{N}}$. We have that $C_{\bar{N}} \leq C^*_N$ by the triangular inequality. This implies that

$$C_{\bar{N}} < C_N - \sum_{j \in \hat{N}} \gamma_j - C^1 - C^2 \tag{2.17}$$

Rearranging the above inequality, we have that

$$C_N > C_{\bar{N}} + \sum_{j \in \hat{N}} \gamma_j + C^1 + C^2 \tag{2.18}$$

We will show that (2.18) leads to a contradiction.

Case 1. $C_{\bar{N}} > r^+$. We schedule the tasks in $\hat{N}$ sequentially after time $C_{\bar{N}}$.

Case 1-1. If $C_{\bar{N}} + \sum_{j \in \hat{N}} \gamma_j \leq d^-$, then this is a feasible solution for problem P. Denote the makespan of this feasible solution by C'_N. We have that

$$C'_N \leq C_{\bar{N}} + \sum_{j \in \hat{N}} \gamma_j \tag{2.19}$$

by our choice of γ_j for each task j. We see that (2.19) contradicts with (2.18) because C_N is the optimal makespan and $C^1 + C^2$ is non-negative.

Case 1-2. If $C_{\bar{N}} + \sum_{j \in \hat{N}} \gamma_j > d^-$, we have that

$$C_{\bar{N}} + \sum_{j \in \hat{N}} \gamma_j + d^+ > d^- + d^+$$

Because $C_N < d^+$, this implies that

$$C_N < C_{\bar{N}} + \sum_{j \in \hat{N}} \gamma_j + d^+ - d^- \tag{2.20}$$

We see that (2.20) contradicts with (2.18) because C_N is the optimal makespan and C^1 is non-negative.

Case 2. If $C_{\bar{N}} < r^+$. We schedule the tasks in $\hat{N}$ sequentially after time r^+. Note

the $C^1 = r^+ - r^- - p^-$ in this case, because $C_{\bar{N}} \geq r^- + p^-$. Therefore, (2.18) can be written as

$$C_N > C_{\bar{N}} + \sum_{j \in \hat{N}} \gamma_j + r^+ - r^- - p^- + d^+ - d^- \tag{2.21}$$

Case 2-1. If $r^+ + \sum_{j \in \hat{N}} \gamma_j \leq d^-$. This is a feasible solution for problem P. Denote the makespan of this feasible solution by C'_N. We have that

$$C'_N \leq r^+ + \sum_{j \in \hat{N}} \gamma_j \tag{2.22}$$

by our choice of γ_j for each task j.

We conclude that (2.22) contradicts with (2.21) because C_N is the optimal makespan, $C_{\bar{N}}$ is at least $r^- + p^-$, and $C^2 = d^+ - d^-$ is non-negative.

Case 2-2. If $r^+ + \sum_{j \in \hat{N}} \gamma_j > d^-$, we have that

$$r^+ + \sum_{j \in \hat{N}} \gamma_j + d^+ > d^- + d^+$$

Because $C_N \leq d^+$, this implies that

$$C_N \leq r^+ + \sum_{j \in \hat{N}} \gamma_j + d^+ - d^- \tag{2.23}$$

We see that (2.23) contradicts with (2.21) because C_N is the optimal makespan, and $C_{\bar{N}}$ is at least $r^- + p^-$ □

Remark 19. *The following are true for Theorem 18.*

1. *It generalizes the Benders cut given in Tran et al. (2016) when there are no time windows.*

2. *It captures Theorem 9 as a special case. Therefore, it improves the Benders cut given in Elçi and Hooker (2022) when there are no setup time and deadlines.*

3. *It generalizes the Benders cuts given in Hooker (2007) and Elçi and Hooker (2022) to include different release times and deadlines simultaneously. In other words, our Benders cut can be used for the planning and scheduling problem introduced by Hooker (2007) in the presence of time windows with different release time and deadlines.*

Corollary 20. *We can linearize the cut* (18) *due to Lemma 10 as follows:*

$$\begin{aligned} \beta_i &\geq \mathrm{SP}_i(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij})(\gamma_j + C^1 + C^2) \qquad (a) \\ \beta_i &\geq \mathrm{SP}_i(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} (1 - x_{ij})\gamma_j - C^1 - C^2 \qquad (b) \end{aligned} \tag{2.24}$$

In the remainder of this section, we present two more Benders cuts for the parallel machine scheduling problem in the same spirit of Theorem 15.

Theorem 21. *The following is a valid Benders cut for the minimum makespan problem with setup times when there are no deadlines.*

$$\beta_i \geq \mathrm{SP}_i(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} \theta_j (1 - x_{ij}) \tag{2.25}$$

where $\theta_j = p_j + \alpha_j + C_j^1$ *with* $C_j^1 = (r_j - r^- - p^-)^+$ *and* $\alpha_j = \max_{k \in J_i(\bar{\mathbf{x}})}\{s_{kj}\}$ *for all*

$j \in J_i(\bar{\mathbf{x}})$.

Proof. Consider the same construction in the proof of Theorem 18. Let $k^1 = \arg\max_{j \in N}\{r_j\}$. In this proof, we consider two cases depending on whether k^1 is an element of $\hat{N}$. Suppose that the cut is not valid. We have that

$$C_N > C_{\bar{N}} + \sum_{j \in \hat{N}} \theta_j \tag{2.26}$$

Case 1 - ($k^1 \in \hat{N}$). In this case, we have that

$$C_N > C_{\bar{N}} + \sum_{j \in \hat{N}} \theta_j \tag{2.27}$$

$$= C_{\bar{N}} + \sum_{j \in \hat{J}} \gamma_j + \sum_{j \in \hat{J} \setminus \{k^1\}} C_j^1 + C^1 \tag{2.28}$$

We see that (2.28) is greater than the right hand side of (2.18). Thus, we reach the same contradictions under all possible cases. (A longer argument would go through two cases depending on whether $C_{\bar{N}} > r^+$ or not. We don't need this longer argument.)

Case 2 - ($k^1 \notin \hat{N}$). In this case, we have that $k^1 \in \bar{N}$. This implies that $C_{\bar{N}} > r^+$. We schedule the tasks in $\hat{N}$ sequentially after time $C_{\bar{N}}$. This is a feasible solution for problem P. Denote the makespan of this feasible solution by C'_N. We have that

$$C'_N \leq C_{\bar{N}} + \sum_{j \in \hat{N}} \gamma_j \tag{2.29}$$

by our choice of γ_j for each task j. We see that (2.29) contradicts with (2.26) because

C_N is the optimal makespan and C^1 is non-negative.

□

When there are deadlines, the following theorem can be used.

Theorem 22. *The following is a valid Benders cut for the minimum makespan problem with setup times.*

$$\beta_i \geq \text{SP}_i(\bar{\mathbf{x}}) - \sum_{j \in J_i(\bar{\mathbf{x}})} \theta_j (1 - x_{ij}) \tag{2.30}$$

where $\theta_j = p_j + \alpha_j + C_j^1 + C^2$ *with* $C_j^1 = (r_j - r^- - p^-)^+$, $C^2 = (d^+ - d^-)$, *and* $\alpha_j = \max_{k \in J_i(\bar{\mathbf{x}})}\{s_{kj}\}$ *for all* $j \in J_i(\bar{\mathbf{x}})$.

Proof. The proof if analogous to the above Theorem. □

2.6 Computational Experiments

In this section, we present the results of our computational experiments. The goal of this section is two-fold. Our first goal is the demonstrate the impact of using improved Benders cuts (2.12) together with (2.14) for solving the stochastic planning and scheduling problem studied in Elçi and Hooker (2022). Our second goal is to demonstrate the effectiveness of the proposed branch-and-check method for solving PMSP-TW.

All experiments presented in this section are conducted on a personal computer with a 2.80 GHz Intel® Core™ i7-7600 processor and 24 GB memory running on a Microsoft

Windows 10 Pro. We use CPLEX and CP Optimizer engines of IBM® ILOG® CPLEX® 12.7 Optimization Studio for all MILP and CP formulations, respectively. The formulations are implemented in C++ API using a single thread. We use the Lazy Constraint Callback function of CPLEX to implement branch-and-check algorithm.

2.6.1 Instance Generation for PMSP-TW

In order to generate instances for PMSP-TW, we follow a similar approach proposed in Elçi and Hooker (2022). Let $|I| = m$ and $|J| = n$. Integer release times are drawn from a uniform distribution on $[0,\ 2.5n(m+1)/m]$. For each facility $i \in I$, integer processing times p_{ij} are drawn from a uniform distribution on $[2,\ 25-10(i-1)/(m-1)]$. We also use the same distribution to generate setup times. In particular, integer setup times s_{ijk} are drawn from a uniform distribution on $[2,\ 25-10(i-1)/(m-1)]$. As a result, we have that facilities with a higher index are more efficient both in terms of processing tasks and setup time. Furthermore, we have a balance between the setup and processing times, since they are generated from the same distribution (see, e.g., Ying et al., 2012). Lastly, the deadline d_j of task j is obtained as follows. Let $L = 20 \times n/m$. We set $d_j = r_j + \beta$ where r_j is the release time of task j and β is drawn from a uniform distribution on $[0.75 \times \alpha L,\ 1.25 \times \alpha L]$ with $\alpha = 2$.

2.6.2 Impact of the Analysis in Section 4

We use the same data set from Elçi and Hooker (2022) (see Table 5). We modify the B&C algorithm in Elçi and Hooker (2022) in two ways: (i) We use the improved Benders cuts (2.12). (ii) We add the Benders cuts (2.14) in addition to the cuts (2.12) whenever a violated Benders cut is identified during the course of the algorithm. Each cell in Table 2.1 represents an average computation time over three instances.

Table 2.1: Average computation time for planning and scheduling problem.

		2 facilities			4 facilities		
Tasks	Scenarios	L-shaped integer cuts only	B&Ch analytic cuts	B&Ch enhanced	L-shaped integer cuts only	B&Ch analytic cuts	B&Ch enhanced
10	1	18.5	0.3	0.2	3246.3	0.9	0.4
	5	141.3	2.6	1.0	*	1.9	1.5
	10	292.3	2.7	1.4	*	2.8	2.2
	50	2425.0	11.6	6.5	*	15.1	11.9
	100	*	22.3	13.4	*	21.5	26.7
	500	*	129.3	73.0	*	149.1	128.9
14	1	1971.2	5.2	1.3	1368.6	0.8	0.5
	5	*	18.1	6.4	*	3.7	2.2
	10	*	43.3	12.9	*	5.6	4.3
	50	*	241.8	66.3	*	26.2	25.5
	100	*	704.4	169.1	*	60.1	47.4
	500	*	3503.4	1899.2	*	407.4	236.2
18	1	*	213.3	13.3	1770.1	2.7	1.3
	5	*	2190.4	169.0	*	10.1	8.9
	10	*	2932.3	239.8	*	31.5	15.0
	50	*	*	1771.3†	*	316.6	95.5
	100	*	*	2326.3†	*	634.3	138.3
	500	*	*	*	*	3162.8	1525.2

†Average excludes one instance that exceeded an hour in computation time.

††Average excludes two instances that exceeded an hour.

*All three instances exceeded an hour.

We see from Table 2.1 that our enhancements are significant. The impact of our enhancements is more pronounced on the harder instances with 2 facilities and 14 tasks or 18 tasks. We see that the enhanced branch-and-check method is at least an order of magnitude faster on the hardest instances with 2 facilities and 18 tasks, solving 4 instances that the previous version could not solve within the given time limit.

2.6.3 The Performance of the Branch-and-Check Algorithm

In this section, we perform experiments to assess the effectiveness of the proposed branch-and-check method in solving PMSP-TW. To this end, we generate instances as described in Section 2.6.1. Each cell in Table 2.2 is an average over 5 instances that represents the computational time required to prove optimality within one hour of time limit.

We benchmark our branch-and-check algorithm against the MIP formulation given in (2.1). We use the default settings of `CPLEX` solver for the MIP formulation. In our implementation of the branch-and-check method we add two sets of optimality cuts. Namely, we add both (2.24) and (2.30) whenever executing line 1 of Algorithm 1.

We see from Table 2.2 that the branch-and-check method clearly outperforms the MIP formulation. For example, we observe that the instances with 15 tasks cannot be solved using the MIP formulation, whereas the branch-and-check method can solve those instances within 100 seconds, on average.

Table 2.2: Average computation time for PMSP-TW.

	2 facilities		4 facilities	
Tasks	MIP	B&Ch	MIP	B&Ch
5	0.2	0.1	0.2	0.2
10	3054.5†	1.0	117.7	1.0
15	*	91.0	*	82.4
20	*	*	*	3222.0††

†Average excludes two instances that exceeded an hour.
††Average excludes four instances that exceeded an hour.
*All three instances exceeded an hour.

2.7 Conclusion

In this paper, we make methodological contributions to the literature of logic-based Benders decomposition framework. We begin our analysis by focusing on the planning and scheduling problem, a problem well studied in the literature. We propose two new classes of logic-based Benders cuts for a variant of this problem with no deadlines. We show that our cuts and the cuts for the variant with no release times are both tight. The computational experiments show the effectiveness of the two new classes of cuts we introduce.

We then focus on a sequence-dependent parallel machine scheduling problem. We derive novel logic-based Benders cuts that generalizes several well known cuts in the literature. We show that these cuts can be used to devise a branch-and-check algorithm. Our computational study shows that the proposed algorithm performs better than the benchmark.

We conclude by providing several directions for future work. The proposed branch-and-check method can be used to solve a stochastic variant of the sequence-dependent

scheduling problem. Furthermore, it is an interesting research direction to explore the use of logic-based Benders cuts for vehicle routing problems as this problem can be cast as a sequence-dependent scheduling problem.

Chapter 3

Portfolio Optimization in the Presence of Estimation Errors on the Expected Asset Returns

This chapter is a joint work with Gérard Cornuéjols and Matthias Köppe.

3.1 Introduction

Consider a portfolio optimization problem where we want to invest in n assets. If the return vector $\mathbf{r} \in \mathbb{R}^n$ is given, we formulate the problem as maximize$\{\mathbf{r}^\mathsf{T}\mathbf{x} : \ \mathbf{x} \in \Delta\}$ where $\mathbf{x} \in \mathbb{R}^n$ denotes the fraction of investment in each asset and Δ denotes the feasible region. In this paper we consider $\Delta := \{\mathbf{x} \in \mathbb{R}^n : \ \sum_{i=1}^n x_i = 1, \ \mathbf{x} \geq 0\}$. The constraints $\mathbf{x} \geq 0$ restrict the model to portfolios with long positions only. This problem has a trivial optimal solution: Only invest in the asset that has the highest

return.

In practice, however, the assets are risky and the return vector $\mathbf{r}$ is random. The classical mean-variance portfolio optimization problem introduced by Markowitz (1952) addresses this issue by maximizing the expected return of the portfolio subject to a constraint on the risk modeled as the variance of the portfolio return.

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \boldsymbol{\mu}^{\mathsf{T}}\mathbf{x} && \text{(3.1a)} \\ \text{subject to} \quad & \mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x} \leq v && \text{(3.1b)} \\ & \mathbf{x} \in \Delta. && \text{(3.1c)} \end{aligned}$$

Here, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the expectation vector and covariance matrix of the asset returns, respectively. In practice, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are estimated, and it has been observed that the Markowitz model tends to amplify estimation errors. In particular, small errors in $\boldsymbol{\mu}$ may produce large changes in portfolio holdings (see, e.g., Best and Grauer, 1991; Chopra and Ziemba, 1993; Michaud, 2008). Approaches to mitigate the effect of estimation errors in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ on portfolio construction have lead to a vast literature. Among these we name ter Horst et al. (2006), Chopra (1993), Jagannathan and Ma (2003), Goldfarb and Iyengar (2003), Tutuncu and Koenig (2004), Ceria and Stubbs (2006), Scherer (2007), Garlappi et al. (2007), Kan and Zhou (2007), DeMiguel et al. (2009a), DeMiguel et al. (2009b), Lim et al. (2012), and Ban et al. (2018). In this paper, we pursue this line of work, focusing on the uncertainty in the $\boldsymbol{\mu}$ estimates.

We assume that the expected return vector $\boldsymbol{\mu}$ is unknown and belongs to an ellipsoidal

uncertainty set given by

$$\mathcal{U} \coloneqq \left\{ \boldsymbol{\mu} \in \mathbb{R}^n \ : \ (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^{\mathsf{T}} \boldsymbol{\Xi}^{-1} (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}) \leq \kappa^2 \right\} \tag{3.2}$$

where $\hat{\boldsymbol{\mu}}$ is the estimated expected return and the positive definite matrix $\boldsymbol{\Xi}$ is referred to as the estimation-error matrix. Using the uncertainty set $\mathcal{U}$, we formulate the robust portfolio optimization problem as follows

$$\max_{\mathbf{x} \in \mathcal{X}} \ \min_{\boldsymbol{\mu} \in \mathcal{U}} \left\{ \boldsymbol{\mu}^{\mathsf{T}} \mathbf{x} \right\} \tag{3.3}$$

where $\mathcal{X}$ denotes the feasible set (3.1b) – (3.1c). We assume that the covariance matrix $\boldsymbol{\Sigma}$ is known. Consequently, $\mathcal{X}$ is the intersection of an ellipsoid with a simplex.

This problem can be reformulated following Ben-Tal and Nemirovski (1999) as

$$\max_{\mathbf{x} \in \mathcal{X}} \left\{ \hat{\boldsymbol{\mu}}^{\mathsf{T}} \mathbf{x} - \kappa \sqrt{\mathbf{x}^{\mathsf{T}} \boldsymbol{\Xi} \mathbf{x}} \right\}. \tag{3.4}$$

We refer the reader to Goldfarb and Iyengar (2003), Tutuncu and Koenig (2004) and Fabozzi et al. (2007) for more detailed discussions on robust portfolio optimization. The *risk-like* term $\kappa \sqrt{\mathbf{x}^{\mathsf{T}} \boldsymbol{\Xi} \mathbf{x}}$ in formulation (3.4) can be interpreted as an estimation risk that must be considered by risk-averse investors on top of the risk caused by the variance $\mathbf{x}^{\mathsf{T}} \boldsymbol{\Sigma} \mathbf{x}$ of the portfolio return (Fabozzi et al., 2007, p 371).

Our contribution. In this paper, we provide a theoretical analysis on the choice for the estimation-error matrix $\boldsymbol{\Xi}$ in robust portfolio optimization. The literature on selecting/constructing an estimation-error matrix is scarce (Gotoh and Takeda, 2011). Stubbs and Vance (2005) provide a comprehensive overview on practical

approaches for computing estimation-error matrices. They also recommend the use of diagonal estimation-error matrices to practitioners. On the other hand, there are several studies in which a scalar multiple of $\mathbf{\Sigma}$ is used as the estimation-error matrix (see, e.g., Scherer, 2007; Garlappi et al., 2007; Olivares-Nadal and DeMiguel, 2018). Among these, Scherer (2007) has a skeptical take on robust optimization and shows that such a choice for $\mathbf{\Xi}$ is equivalent to some other well known shrinkage approaches.

In our work, we begin by discussing the difference between true and actual frontier - a well known approach that is used to quantify the impact of the estimation errors. We show that when $\mathbf{\Xi}$ is a multiple of $\mathbf{\Sigma}$, robust optimization cannot improve the actual performance of the Markowitz model. This analysis supports the recommendation of Stubbs and Vance (2005) where they claim that such a choice is not used in practice and is not recommended. We then focus on the use of diagonal estimation-error matrices and show that the class of diagonal estimation-error matrices can achieve an arbitrarily small loss in the expected portfolio return as compared to the optimum. We accomplish this by reformulating the optimality conditions of the robust portfolio problem when we have a single expected return vector.

We then investigate diagonal estimation-error matrices in the presence of multiple expected returns. To this end, we propose a bilevel model that computes the loss and we show that the diagonal estimation-error matrices can achieve an arbitrarily small loss even when there are multiple estimates for the expected return. The proposed bilevel model also provides a principled way to construct estimation-error matrices using data, though it is a non-convex optimization problem and it is difficult to solve in practice.

We finally focus on the use of an identity matrix for $\mathbf{\Xi}$. This choice requires

calibrating a single parameter (κ) for the robust problem. We perform computational experiments to test whether robust optimization can perform better than the Markowitz model. Our results demonstrate that a good choice for κ can significantly improve the performance of the Markowitz model, especially when the expected return estimates are not reliable.

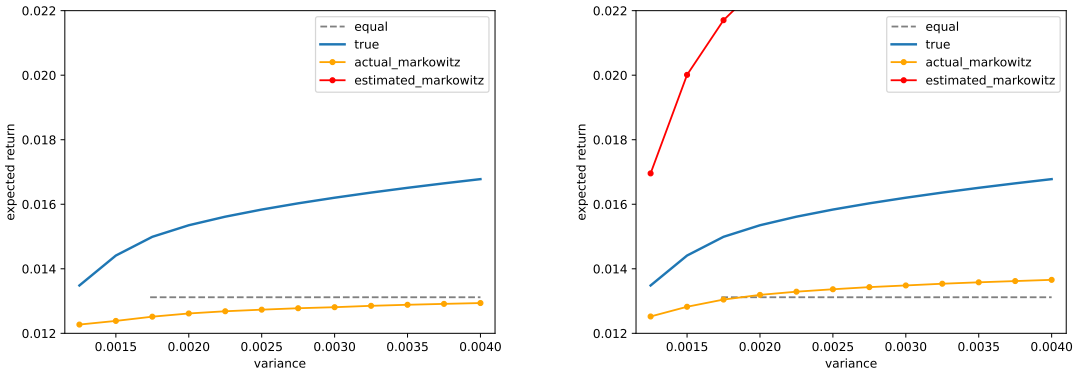
The rest of the paper is organized as follows. We introduce true, estimated and actual frontiers in Section 3.2. In Section 3.3, we show that robust optimization can improve on the actual performance. We present the analysis of the robust optimization problem in Section 3.4. Section 3.5 generalizes the analysis of the robust model to incorporate multiple expected returns. We analyze the use of an identity matrix as estimation-error matrix in Section 3.6. Section 3.7 concludes the paper.

3.2 True, Estimated, and Actual Frontiers

The sensitivity of mean-variance portfolio optimization models to estimation errors on the expected asset returns is well documented in the literature (see, e.g., Best and Grauer, 1991; Chopra, 1993; Michaud, 2008). It is sometimes referred to as the error maximization tendency of mean-variance optimization. In order to quantify the effect of estimation errors, Broadie (1993) made a distinction between true, estimated, and actual frontiers. A *frontier* plots the maximum expected return of a portfolio of assets as a function of the risk threshold (Markowitz, 1952). The *true frontier* is computed by using the true expected returns of the assets, a quantity in fact unknown to the decision maker. The *estimated frontier* is computed by using the estimated expected returns. It describes what appears will be the expected return of a portfolio optimized

based on the estimated parameters. The *actual frontier* plots the expected return one actually achieves (using the true expected returns) when one invests in the above portfolio (constructed using estimated expected returns). We next describe how we compute these three frontiers.

We solve the problem maximize$\{\boldsymbol{\mu}^\mathsf{T}\mathbf{x} : \ (3.1b)-(3.1c)\}$ to obtain the optimal solution $\mathbf{x}^*$ where $\boldsymbol{\mu}$ is the true (but unknown to the investor) vector of expected asset returns. Using $\mathbf{x}^*$, we construct the true frontier $\boldsymbol{\mu}^\mathsf{T}\mathbf{x}^*$. Let $\hat{\boldsymbol{\mu}}$ be the vector of estimated expected asset returns. We solve maximize$\{\hat{\boldsymbol{\mu}}^\mathsf{T}\mathbf{x} : \ (3.1b) - (3.1c)\}$ to obtain the *Markowitz solution estimate* $\hat{\mathbf{x}}^M$. Using $\hat{\mathbf{x}}^M$, we construct the *estimated Markowitz frontier* $\hat{\boldsymbol{\mu}}^\mathsf{T}\hat{\mathbf{x}}^M$ and the *actual Markowitz frontier* $\boldsymbol{\mu}^\mathsf{T}\hat{\mathbf{x}}^M$.



(a) sample size = 1

(b) sample size = 24

Figure 3-1: True, actual Markowitz, and estimated Markowitz frontiers.

In Figure 3-1, we illustrate *the gap* between the true and the actual frontiers on real-world data. Details on the data set can be found in Appendix A.2.1. We assume that the asset returns are normally distributed with distribution Normal$(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The vector $\hat{\boldsymbol{\mu}}$ is the sample average of N random samples. Equivalently, we generate $\hat{\boldsymbol{\mu}}$ from Normal$(\boldsymbol{\mu}, \boldsymbol{\Sigma}/N)$. We illustrate the cases $N = 1$ and $N = 24$. Each point in the figures represents the average value over 10000 trials. The dashed-line represents the

performance of the equal-weight portfolio where we invest equally in all the assets. The expected return of the equal-weight portfolio is 0.0131. Another interesting portfolio for comparison is the minimum variance portfolio, which has an expected return of 0.0122.

We see in Figure 3-1a that the actual performance of the Markowitz solution estimate is poor when the sample size is equal to 1. In fact, the average expected return values are worse than the equal-weight portfolio. On the other hand, we see that the performance of the Markowitz solution estimate improves significantly in Figure 3-1b with more accurate return estimates with a sample size of 24. As N goes to $+\infty$, the actual Markowitz frontier converges to the true frontier. Furthermore, we plot the estimated frontier in Figures 3-1a and 3-1b. We see that the estimated frontier is far away from the true frontier so much that it is off the chart in Figure 3-1a.

Similarly, solving the robust optimization problem (3.4), we obtain the *robust solution estimate* $\hat{\mathbf{x}}^R$. Using $\hat{\mathbf{x}}^R$, we construct the *estimated robust frontier* $\hat{\boldsymbol{\mu}}^\mathsf{T}\hat{\mathbf{x}}^R$ and the *actual robust frontier* $\boldsymbol{\mu}^\mathsf{T}\hat{\mathbf{x}}^R$. A key aspect here is the modeler's choice of the estimation-error matrix $\boldsymbol{\Xi}$.

3.3 Robust Optimization Can Improve the Actual Performance

A possible choice for $\boldsymbol{\Xi}$ is to make this matrix proportional to $\boldsymbol{\Sigma}$, namely $\boldsymbol{\Xi} = \rho\boldsymbol{\Sigma}$. Such an idea has been proposed by ter Horst et al. (2006) and Garlappi et al. (2007). Notice however that, for solutions $\mathbf{x}$ that satisfy the risk constraint $\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{x} \leq v$ at equality (which is the most interesting case), the objective $\hat{\boldsymbol{\mu}}^\mathsf{T}\mathbf{x} - \kappa\sqrt{\mathbf{x}^\mathsf{T}\boldsymbol{\Xi}\mathbf{x}}$ of the

robust problem (3.4) becomes

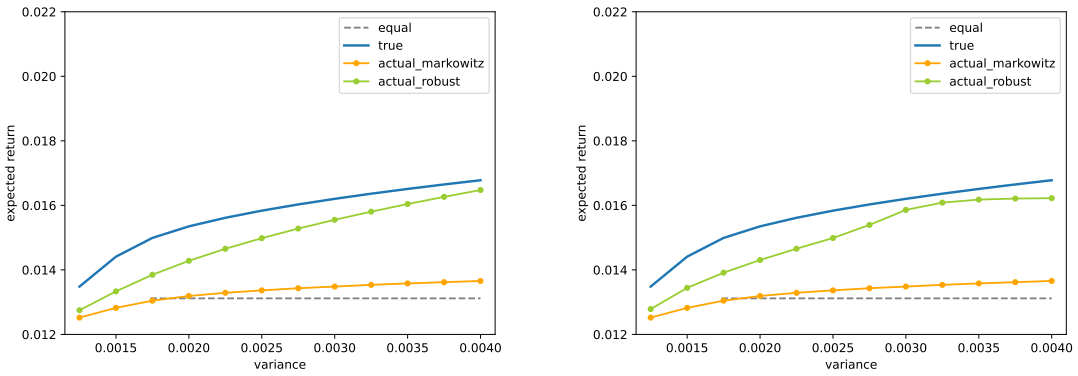
$$\hat{\boldsymbol{\mu}}^{\mathsf{T}}\mathbf{x} - \kappa\sqrt{\mathbf{x}^{\mathsf{T}}\rho\boldsymbol{\Sigma}\mathbf{x}} = \hat{\boldsymbol{\mu}}^{\mathsf{T}}\mathbf{x} - \kappa\sqrt{\rho v}.$$

The last term is just a constant, and therefore the robust problem reduces to the Markowitz model (3.1). The estimation errors do not affect the optimal portfolio!

Garlappi et al. (2007) consider a variation of model (3.1) where the risk is not a hard constraint $\mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x} \leq v$ but instead a penalty in the objective, leading to the model $\max_{x\in\Delta} \boldsymbol{\mu}^{\mathsf{T}}\mathbf{x} - \gamma\ \mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x}$ for a given penalty term $\gamma \in \mathbb{R}$. Robustifying as in (3.4), this model becomes $\max_{x\in\Delta} \boldsymbol{\mu}^{\mathsf{T}}\mathbf{x} - \gamma\ \mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x} - \kappa\sqrt{\mathbf{x}^{\mathsf{T}}\boldsymbol{\Xi}\mathbf{x}}$. Garlappi et al. (2007) consider this model with $\boldsymbol{\Xi} = \boldsymbol{\Sigma}$. When $\kappa = 0$, this reduces to the Markowitz model and when $\kappa \to +\infty$, it reduces to the minimum variance portfolio. In their computational experiments, Garlappi et al. (2007) find that the minimum variance portfolio gives the best results in terms of measures such as the expected return or the Sharpe ratio.

In the remainder of this paper, we consider choices for $\boldsymbol{\Xi}$ that can outperform both the Markowitz portfolio and the minimum variance portfolio, using actual expected return as a measure.

To illustrate the potential of robust optimization, we design an experiment showing that good matrices $\boldsymbol{\Xi}$ exist, even among diagonal matrices with only two different diagonal entries. We construct $\boldsymbol{\Xi}$ based on partial information on the true expected return $\boldsymbol{\mu}$. In particular, in Figures $3-2a-3-2b$, we let some entries of the estimation-error matrix equal to $\epsilon = 0.001$ ($\kappa = 1$, sample size $= 24$). In Figure 3-2a, the entry corresponding to the asset with the highest true return is set to ϵ, and in



(a) $\boldsymbol{\Xi} = \mathrm{diag}\big([1, 1, 1, 1, 1, 1, 1, \epsilon, 1, 1, 1]\big)$

(b) $\boldsymbol{\Xi} = \mathrm{diag}\big([1, \epsilon, 1, 1, 1, 1, 1, \epsilon, 1, 1, 1]\big)$

Figure 3-2: Robust frontiers of several Ξ matrices.

Figure 3-2b, the entries corresponding to the four assets with highest true return are set to ϵ.

The overall picture that we see in Figure 3-2 is that there is room for improvement if one is able to choose a good matrix $\boldsymbol{\Xi}$. It seems promising that the robust optimization can close the gap between the actual Markowitz frontier and the true frontier.

3.4 The Class of Diagonal Estimation-Error Matrices

In this section, we focus on the class of diagonal estimation-error matrices. This choice is partly motivated by the study of Stubbs and Vance (2005). The authors argue that it is difficult to generate estimation-error matrices accurately, and they suggest that the use of simple diagonal matrices is beneficial. A natural question to ask is: Do we loose in performance by restricting our attention to diagonal matrices

$\boldsymbol{\Xi}$?

The issue boils down to the following: Given an estimate $\hat{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$, can we always find a diagonal matrix $\boldsymbol{\Xi}$ such that the resulting robust portfolio $\hat{\mathbf{x}}^R$ has an actual expected return very close to the true expected return? To this end, we start examining the optimality conditions of (3.4).

3.4.1 Analysis of the Robust Portfolio Optimization Problem

We first write (3.4) as a convex optimization problem of the following form.

$$\begin{aligned}
\underset{\mathbf{x}}{\text{minimize}} \quad & -\hat{\boldsymbol{\mu}}^{\mathsf{T}}\mathbf{x} + \sqrt{\mathbf{x}^{\mathsf{T}}\boldsymbol{\Xi}\mathbf{x}} && \text{(3.5a)}\\
\text{subject to} \quad & \mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x} \leq v, && \text{(3.5b)}\\
& \mathbf{1}^{\mathsf{T}}\mathbf{x} = 1, && \text{(3.5c)}\\
& -\mathbf{x} \leq \mathbf{0}. && \text{(3.5d)}
\end{aligned}$$

We next make some observations about (3.5). The proof of these observations are deferred to the Appendix.

Proposition 23. *Suppose that* $\boldsymbol{\Xi}$ *and* $\boldsymbol{\Sigma}$ *are positive definite. Then, the following statements are true about* (3.5).

1. *It is a convex optimization problem with a strictly convex objective function. Therefore, the optimal solution* $\hat{\mathbf{x}}^R$ *is unique.*

2. *It satisfies Slater's condition.*

3. *It is sufficient to consider the optimality conditions for differentiable functions.*

We next derive the optimality conditions of (3.5). Let $\lambda^1 \in \mathbb{R}$, $\lambda^2 \in \mathbb{R}$, and $\boldsymbol{\lambda}^3 \in \mathbb{R}^n$ respectively denote the Lagrangian multipliers of the constraints of (3.5). The Lagrangian function associated with this problem is given below.

$$L\left(\mathbf{x}, \lambda^1, \lambda^2, \boldsymbol{\lambda}^3\right) = -\hat{\boldsymbol{\mu}}^{\mathsf{T}}\mathbf{x} + \sqrt{\mathbf{x}^{\mathsf{T}}\boldsymbol{\Xi}\mathbf{x}} + \lambda^1\left(\mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x} - v\right) + \lambda^2\left(\mathbf{1}^{\mathsf{T}}\mathbf{x} - 1\right) - \left(\boldsymbol{\lambda}^3\right)^{\mathsf{T}}\mathbf{x}$$

Using the Lagrangian function, we can write the optimality conditions of (3.5).

$$-\hat{\boldsymbol{\mu}} + \frac{\boldsymbol{\Xi}\mathbf{x}}{\sqrt{\mathbf{x}^{\mathsf{T}}\boldsymbol{\Xi}\mathbf{x}}} + 2\lambda^1\boldsymbol{\Sigma}\mathbf{x} + \lambda^2\mathbf{1} - \boldsymbol{\lambda}^3 = \mathbf{0} \tag{3.6a}$$

$$\mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x} \leq v \tag{3.6b}$$

$$\mathbf{1}^{\mathsf{T}}\mathbf{x} = 1 \tag{3.6c}$$

$$-\mathbf{x} \leq \mathbf{0} \tag{3.6d}$$

$$\lambda^1 \geq 0 \tag{3.6e}$$

$$\boldsymbol{\lambda}^3 \geq \mathbf{0} \tag{3.6f}$$

$$\lambda^1\left(\mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x} - v\right) + \left(\boldsymbol{\lambda}^3\right)^{\mathsf{T}}\mathbf{x} = 0 \tag{3.6g}$$

When $\boldsymbol{\Xi}$ is a diagonal matrix, (3.6a) can be written as

$$-\hat{\mu}_i + \frac{\xi_i x_i}{\sqrt{\sum_{j=1}^n \xi_j x_j^2}} + 2\lambda^1\sum_{j=1}^n \sigma_{ij}x_j + \lambda^2 - \lambda_i^3 = 0, \quad i \in [1, n] \tag{3.7}$$

Here, $\boldsymbol{\xi}$ denotes the vector of diagonal entries of $\boldsymbol{\Xi}$ and $[1, n] \coloneqq \{1, \ldots, n\}$. Now, we introduce additional variables and constraints to rewrite the square-root term in (3.7). Let

$$z_i = \frac{\xi_i x_i}{\sqrt{\sum_{j=1}^n \xi_j x_j^2}}, \quad i \in [1, n]. \tag{3.8}$$

Using this substitution, we rewrite (3.7) as

$$-\hat{\mu}_i + z_i + 2\lambda^1 \sum_{j=1}^{n} \sigma_{ij} x_j + \lambda^2 - \lambda_i^3 = 0, \quad i \in [1, n], \tag{3.9a}$$

$$\sum_{i=1}^{n} x_i z_i = \alpha, \tag{3.9b}$$

$$\xi_i x_i = z_i \alpha, \quad i \in [1, n], \tag{3.9c}$$

$$z_i \geq 0, \quad i \in [1, n], \tag{3.9d}$$

$$\xi_i > 0, \quad i \in [1, n], \tag{3.9e}$$

$$\alpha > 0 \tag{3.9f}$$

We have the following result, its proof is deferred to Appendix A.2.2.

Proposition 24. *The system* $\{(3.6b) - (3.6g), (3.9)\}$ *is a correct reformulation of the system* (3.6).

3.4.2 Analysis of the Loss Due to Estimation Error

Given a solution estimate $\hat{\mathbf{x}}$, we define the *loss* in objective value in the following way.

$$\text{loss}(\hat{\mathbf{x}}) = \boldsymbol{\mu}^{\mathsf{T}} \mathbf{x}^* - \boldsymbol{\mu}^{\mathsf{T}} \hat{\mathbf{x}}$$

A solution $\hat{\mathbf{x}}^R$ to the robust problem (3.4) depends on the expected return estimate $\hat{\boldsymbol{\mu}}$ and the estimation-error matrix $\boldsymbol{\Xi}$. In this case we write

$$\text{loss}(\hat{\boldsymbol{\mu}}, \boldsymbol{\Xi}) = \boldsymbol{\mu}^{\mathsf{T}} \mathbf{x}^* - \boldsymbol{\mu}^{\mathsf{T}} \hat{\mathbf{x}}^R$$

The first question we want to answer is the following: Given any $\hat{\boldsymbol{\mu}}$, does there exist a diagonal $\boldsymbol{\Xi}$ matrix such that loss$(\hat{\boldsymbol{\mu}}, \boldsymbol{\Xi})$ is equal to zero. The following example shows that it is not always possible to find such a diagonal matrix $\boldsymbol{\Xi}$.

Example 25. *Consider a two asset portfolio optimization problem. Let* $\sigma_{11} < v$, $\sigma_{22} > v$. *Let* $\mu_1 > \mu_2$. *Clearly* $(x_1 = 1, x_2 = 0)$ *is the true optimal solution. Suppose that* $\hat{\mu}_2 > \hat{\mu}_1$.

It follows that we have $\lambda^1 = 0$ *because the risk constraint is not tight. Furthermore,* $\lambda_1^3 = 0$ *because the non-negativity constraint of* x_1 *is not active. We see that the optimality conditions are satisfied except* (3.6a). *Using our reformulation idea, we must find a feasible solution to the following system.*

$$
\begin{aligned}
&-\hat{\mu}_1 + z_1 + \lambda^2 = 0 \\
&-\hat{\mu}_2 + z_2 + \lambda^2 - \lambda_2^3 = 0 \\
&x_1 z_1 + x_2 z_2 = \alpha \\
&\xi_1 x_1 = z_1 \alpha \\
&\xi_2 x_2 = z_2 \alpha \\
&\xi_1,\ \xi_2 > 0 \\
&z_1,\ z_2 \geq 0 \\
&\alpha > 0
\end{aligned}
$$

We see that because $x_2 = 0$ *and* $\alpha > 0$, *it must be that* $z_2 = 0$. *Furthermore, we can rewrite the first two equations in the following way by treating* z_1 *and* λ_2^3 *as slack*

variables.

$$\lambda^2 \leq \hat{\mu}_1$$
$$\lambda^2 \geq \hat{\mu}_2$$

This is impossible since $\hat{\mu}_2 > \hat{\mu}_1$. Because the true optimal solution is unique, we conclude there does not exist $\boldsymbol{\xi}$ such that the robust portfolio optimization problem yields zero loss.

Our next result shows that even though it is not possible to reach zero loss, we can get arbitrarily close to it.

Theorem 26. *Given $\epsilon > 0$, for every $\hat{\boldsymbol{\mu}}$ there exists a diagonal $\boldsymbol{\Xi}$ matrix such that loss$(\hat{\boldsymbol{\mu}}, \boldsymbol{\Xi}) \leq \epsilon$.*

Proof. Let $\mathbf{x}^{\text{mv}} := \arg\min_{\mathbf{x}}\{\mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x} : \mathbf{x} \in \Delta\}$ denote the minimum variance portfolio. Let $\mathbf{x}^{\text{eq}}$ denote the equal-weight portfolio, i.e. $x_i = 1/n,\ i \in [1, n]$. Pick $\delta \in (0, 1)$ such that $\tilde{\mathbf{x}} = \delta\mathbf{x}^{\text{mv}} + (1 - \delta)\mathbf{x}^{\text{eq}}$ has a lower variance than the true optimal solution $\mathbf{x}^*$. Clearly, all components of $\tilde{\mathbf{x}}$ are strictly greater than zero. Let $\tilde{\mathbf{x}}^* = (1-\bar{\epsilon})\mathbf{x}^* + \bar{\epsilon}\tilde{\mathbf{x}}$ for some $\bar{\epsilon} \in (0, 1)$. Note that $\tilde{\mathbf{x}}^* \in \mathcal{X}$ and has a lower variance than $\mathbf{x}^*$, therefore it is a feasible solution to the portfolio problem.

Now, observe that the loss corresponding to $\tilde{\mathbf{x}}^*$ is given by

$$\begin{aligned}
\text{loss}(\tilde{\mathbf{x}}^*) &= \sum_{i=1}^{n} \mu_i \Big(x_i^* - \tilde{x}_i^* \Big) \\
&= \sum_{i=1}^{n} \mu_i \Big(x_i^* - \big[(1-\bar{\epsilon}) x_i^* + \bar{\epsilon} \tilde{x}_i \big] \Big) \\
&= \bar{\epsilon} \sum_{i=1}^{n} \mu_i \Big(x_i^* - \tilde{x}_i \Big) \\
&= \bar{\epsilon} \Big(\boldsymbol{\mu}^\mathsf{T} \mathbf{x}^* - \boldsymbol{\mu}^\mathsf{T} \tilde{\mathbf{x}} \Big) > 0.
\end{aligned}$$

Therefore, given ϵ, we can pick $\bar{\epsilon} \in (0, 1)$ such that

$$\bar{\epsilon} \leq \frac{\epsilon}{\boldsymbol{\mu}^\mathsf{T} \mathbf{x}^* - \boldsymbol{\mu}^\mathsf{T} \tilde{\mathbf{x}}}$$

This pick ensures that $\text{loss}(\tilde{\mathbf{x}}^*) \leq \epsilon$.

It suffices to find $\boldsymbol{\xi}$, λ^1, λ^2 and $\boldsymbol{\lambda}^3$ such that together with $\tilde{\mathbf{x}}^*$ we have a solution that satisfies the optimality conditions.

Let $\lambda^1 = 0$, and $\lambda_i^3 = 0$ for all $i \in [1, n]$. We see that all optimality conditions are satisfied except (3.6a). Using the reformulation idea, we see that finding such $\boldsymbol{\xi}$ is

equivalent to finding a solution to the below system.

$$
\begin{aligned}
-\hat{\mu}_i + z_i + \lambda^2 &= 0, \quad i \in [1, n] \\
\sum_{i=1}^{n} z_i \tilde{x}_i^* &= \alpha \\
\xi_i \tilde{x}_i^* &= z_i \alpha, \quad i \in [1, n] \\
\xi_i &\geq 0, \quad i \in [1, n] \\
z_i &\geq 0, \quad i \in [1, n] \\
\alpha &\geq 0
\end{aligned}
$$

Let $\lambda^2 < \min_i\{\hat{\mu}_i\}$. Let $z_i = \hat{\mu}_i - \lambda^2$ for all $i \in [1, n]$. Then $\alpha = \sum_{i=1}^{n} z_i \tilde{x}_i^*$. We let $\xi_i = \frac{z_i \alpha}{\tilde{x}_i^*}$ for all $i \in [1, n]$. Clearly, all ξ_i, z_i, α are greater than zero. Therefore, $\boldsymbol{\xi}$ satisfies the optimality conditions and this concludes the proof. □

Finally we conclude this section by stating a sufficient condition for obtaining zero loss.

Theorem 27. *If all the assets are active in the true optimal solution, then for every* $\hat{\boldsymbol{\mu}}$ *there exists a diagonal* $\boldsymbol{\Xi}$ *matrix such that loss*$(\hat{\boldsymbol{\mu}}, \boldsymbol{\Xi}) = 0$.

Proof. Without loss of generality, we assume that all elements of the given $\hat{\boldsymbol{\mu}}$ are greater than 0. It suffices to find $\boldsymbol{\xi}$, λ^1, λ^2 and $\boldsymbol{\lambda}^3$ such that together with $\mathbf{x}^*$ we have a solution that satisfies the optimality condition.

Let $\lambda^1 = 0$, $\lambda^2 = 0$, and $\lambda_i^3 = 0$ for all $i \in [1, n]$. Using the reformulation idea, we

see that finding such $\boldsymbol{\xi}$ is equivalent to finding a solution to the below system.

$$
\begin{aligned}
-\hat{\mu}_i + z_i &= 0, \quad i \in [1, n] \\
\sum_{i=1}^{n} z_i x_i^* &= \alpha \\
\xi_i x_i^* &= z_i \alpha, \quad i \in [1, n] \\
\xi_i &> 0, \quad i \in [1, n] \\
z_i &\geq 0, \quad i \in [1, n] \\
\alpha &> 0
\end{aligned}
$$

Let $z_i = \hat{\mu}_i$ for all $i \in [1, n]$. Then we have that $\alpha = \sum_{i=1}^{n} \hat{\mu}_i x_i^*$. We let $\xi_i = \frac{z_i \alpha}{x_i^*}$ for all $i \in [1, n]$. Clearly, all ξ_i, z_i, α are greater than zero. Therefore, $\boldsymbol{\xi}$ satisfies the optimality conditions and this concludes the proof. □

3.5 Finding the Best Estimation-Error Matrix

In this section, we generalize the results of the previous section to the situation where we have several estimates $\hat{\boldsymbol{\mu}}^1, \ldots, \hat{\boldsymbol{\mu}}^T$ of $\boldsymbol{\mu}$. We consider a setup in the same spirit with the computational experiments conducted to measure the performance of the robust optimization via an out-of-sample simulation in Scherer (2007). In particular, we investigate the ability of the robust model to produce portfolios that are close to the true optimal portfolio even when different estimated expected returns are used as an input to the robust model.

We show that we can always find a diagonal matrix $\boldsymbol{\Xi}$ such that the resulting robust portfolios $\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^T$ all have an actual expected return very close to the optimal

expected return $\boldsymbol{\mu}^{\mathsf{T}} x^*$ where x^* is the optimal portfolio computed using the true $\boldsymbol{\mu}$. To this end, we write a mathematical program that outputs a matrix $\boldsymbol{\Xi}$ that achieves the minimum loss. We define the loss given a collection of return estimates $\{\hat{\boldsymbol{\mu}}^t\}_{t=1}^T$ and estimation-error matrix $\boldsymbol{\Xi}$ as

$$\text{loss}(\{\hat{\boldsymbol{\mu}}^t\}_{t=1}^T, \boldsymbol{\Xi}) = \sum_{t\in[1,T]} (\boldsymbol{\mu}^{\mathsf{T}}\mathbf{x}^* - \boldsymbol{\mu}^{\mathsf{T}}\hat{\mathbf{x}}^{R,t})$$

where $\hat{\mathbf{x}}^{R,t}$ denotes the optimal robust portfolio for $t \in [1, T]$.

3.5.1 A Bilevel Programming Formulation

In this section, we describe the bilevel programming formulation that computes the best estimation-error matrix $\boldsymbol{\Xi}$ for a given collection of return estimates $\{\hat{\boldsymbol{\mu}}^t\}_{t=1}^T$.

The following parameters are used in the description of the bilevel model.

- $\boldsymbol{\mu}$: True expected return vector.
- $\boldsymbol{\Sigma}$: True covariance matrix of the asset returns.
- $\hat{\boldsymbol{\mu}}^t$: Estimated return vector under trial $t \in [1, T]$.
- v : Risk threshold.
- $\mathbf{x}^*(v) = \arg\max_{\mathbf{x}\in\Delta^n} \left\{\boldsymbol{\mu}^{\mathsf{T}}\mathbf{x} \;:\; \mathbf{x}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x} \leq v\right\}$, The portfolio with maximum expected return for a given risk threshold value v.

Let $\hat{\mathbf{x}}^t(v, \boldsymbol{\Xi}, \hat{\boldsymbol{\mu}}^t)$ denote the optimal portfolio given that the estimated return vector is $\hat{\boldsymbol{\mu}}^t$, estimation-error matrix is $\boldsymbol{\Xi}$, and the risk threshold is v under trial $t \in [1, T]$. Note that this value is obtained by solving a convex optimization problem of the

form (3.4). Therefore, we have the following bilevel program to compute the best $\boldsymbol{\Xi}$ that minimizes the loss.

$$\underset{\boldsymbol{\Xi}}{\text{minimize}} \quad \sum_{t=1}^{T} \Big(\boldsymbol{\mu}^{\mathsf{T}} \mathbf{x}^*(v) - \boldsymbol{\mu}^{\mathsf{T}} \hat{\mathbf{x}}^t(v, \boldsymbol{\Xi}, \hat{\boldsymbol{\mu}}^t) \Big) \tag{3.10a}$$

$$\text{subject to} \quad \hat{\mathbf{x}}^t(v, \boldsymbol{\Xi}, \hat{\boldsymbol{\mu}}^t) = \underset{\mathbf{x} \in \Delta^n}{\arg\max} \Big\{ (\hat{\boldsymbol{\mu}}^t)^{\mathsf{T}} \mathbf{x} - \sqrt{\mathbf{x}^{\mathsf{T}} \boldsymbol{\Xi} \mathbf{x}} \ : \ \mathbf{x}^{\mathsf{T}} \boldsymbol{\Sigma} \mathbf{x} \leq v \Big\}, \quad t \in [1, T], \tag{3.10b}$$

$$\boldsymbol{\Xi} \in \mathcal{S}_{++}^n. \tag{3.10c}$$

In this formulation, $\boldsymbol{\Xi}$ is the upper-level decision variable, and each $\hat{\mathbf{x}}^t(v, \boldsymbol{\Xi}, \hat{\boldsymbol{\mu}}^t)$ is a lower-level decision variable. Note that we have as many lower-level problems as the number of trials we have.

The above bilevel program minimizes the sum of the difference between the value of the true frontier, and the value of the actual frontier under each trial. Alternatively, one can use other performance measures, for example minimizing the maximum of the differences. In this formulation, we assume the risk threshold v is large enough that the lower-level problem is feasible. Our next goal is to reformulate problem (3.10) as a single level optimization program.

3.5.2 Reformulating (3.10) as a Single Level Program

We know that robust portfolio optimization problem is a convex optimization program, it satisfies Slater's condition, and it has a unique solution, see Proposition 23. Therefore, we can use the optimality conditions (3.6) to reformulate (3.10) as a single level program (Bard, 1998).

$$\underset{\boldsymbol{\Xi},\hat{\mathbf{x}}^{\mathsf{T}},\lambda^{1,t},\lambda^{2,t},\boldsymbol{\lambda}^{3,t}}{\text{minimize}} \quad \sum_{t=1}^{T}\left(\boldsymbol{\mu}^{\mathsf{T}}\mathbf{x}^{*}-\boldsymbol{\mu}^{\mathsf{T}}\mathbf{x}^{t}\right) \tag{3.11a}$$

$$\text{subject to} \quad -\hat{\boldsymbol{\mu}}^{t}+\frac{\boldsymbol{\Xi}\mathbf{x}^{t}}{\sqrt{(\mathbf{x}^{t})^{\mathsf{T}}\boldsymbol{\Xi}\mathbf{x}}}+2\lambda^{1,t}\boldsymbol{\Sigma}\mathbf{x}^{t}+\lambda^{2,t}\mathbf{1}-\boldsymbol{\lambda}^{3,t}=\mathbf{0}, \quad t\in[1,T], \tag{3.11b}$$

$$(\mathbf{x}^{t})^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x}^{t}\leq v, \quad t\in[1,T], \tag{3.11c}$$

$$\mathbf{1}^{\mathsf{T}}\mathbf{x}^{t}=1, \quad t\in[1,T], \tag{3.11d}$$

$$-\mathbf{x}^{t}\leq\mathbf{0}, \quad t\in[1,T], \tag{3.11e}$$

$$\lambda^{1,t}\geq 0, \quad t\in[1,T], \tag{3.11f}$$

$$\boldsymbol{\lambda}^{3,t}\geq\mathbf{0}, \quad t\in[1,T], \tag{3.11g}$$

$$\lambda^{1,t}\left((\mathbf{x}^{t})^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{x}^{t}-v\right)+\left(\boldsymbol{\lambda}^{3,t}\right)^{\mathsf{T}}\mathbf{x}^{t}=0, \quad t\in[1,T], \tag{3.11h}$$

$$\boldsymbol{\Xi}\in\mathcal{S}_{++}. \tag{3.11i}$$

Note that in the above formulation, the optimality conditions are appended for each trial $t\in[1,T]$.

The bilevel model (3.11) is a non-convex optimization problem due to the stationary point equations (3.11b) and the complementary slackness conditions (3.11h). We next present a reformulation for (3.11) when $\boldsymbol{\Xi}$ is a positive definite diagonal matrix.

Using the earlier reformulation idea in Section 3.4, we can reformulate (3.11) so as

to avoid the square-root terms.

$$\underset{\boldsymbol{\xi},\mathbf{x}^t,\lambda^{1,t},\lambda^{2,t},\boldsymbol{\lambda}^{3,t}}{\text{minimize}} \quad \text{(3.11a)} \tag{3.12a}$$

$$\text{subject to} \quad \text{(3.11c)} - \text{(3.11h)} \tag{3.12b}$$

$$-\hat{\mu}_i^t + z_i^t + 2\lambda^{1,t}\sum_{j=1}^{n}\sigma_{ij}x_j^t + \lambda^{2,t} - \lambda_i^{3,t} = 0, \quad i \in [1,n],\ t \in [T], \tag{3.12c}$$

$$\sum_{i=1}^{n} x_i^t z_i^t = \alpha^t, \quad t \in [T], \tag{3.12d}$$

$$\xi_i x_i^t = z_i^t \alpha^t, \quad i \in [n],\ t \in [T], \tag{3.12e}$$

$$z_i^t \geq 0, \quad i \in [1,n],\ t \in [T], \tag{3.12f}$$

$$\alpha^t > 0, \quad t \in [T], \tag{3.12g}$$

$$\xi_i > 0. \tag{3.12h}$$

3.5.3 Analysis of the Bilevel Model

In this section, we show that the bilevel model (3.12) has always an optimal solution that yields arbitrarily small loss.

Theorem 28. *For every $\epsilon > 0$ and $\{\hat{\boldsymbol{\mu}}^t\}_{t=1}^T$, there exists a diagonal matrix $\boldsymbol{\Xi}$ such that loss$(\{\hat{\boldsymbol{\mu}}\}_{t=1}^T, \boldsymbol{\Xi}) \leq \epsilon$.*

Proof. It suffices to find $\boldsymbol{\xi}$, $\lambda^{1,t}$, $\lambda^{2,t}$, $\boldsymbol{\lambda}^{3,t}$, and $\mathbf{x}^t$ such that they are feasible for (3.12), and all the vectors $\mathbf{x}^t$ are within a small neighborhood of $\mathbf{x}^*$.

Let $\lambda^{1,t} = 0$, and $\lambda_i^{3,t} = 0$ for all $i \in [1,n]$ and $t \in [1,t]$. Then (3.12c) can be written

as

$$-\hat{\mu}_i^t + z_i^t + \lambda^{2,t} = 0, \quad i \in [1,n],\ t \in [T], \tag{3.13}$$

Using (3.12h), we substitute $x_i^t = z_i^t \alpha^t / \xi_i$ in (3.12g). We see that it suffices to solve the following system to complete the proof.

$$\sum_{i \in [1,n]} \frac{(\hat{\mu}_i^t - \lambda^{2,t})^2}{\xi_i} = 1, \quad t \in [1,T]. \tag{3.14a}$$

$$z_i^t \geq 0, \quad i \in [1,n],\ t \in [T], \tag{3.14b}$$

$$\alpha^t > 0, \quad t \in [T], \tag{3.14c}$$

$$\xi_i > 0. \tag{3.14d}$$

Let $\tilde{\mathbf{x}}^*$ be as in the proof of Theorem 26.

That is, $\tilde{\mathbf{x}}^*$ is a feasible solution of (3.1) that satisfies the constraints $\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{x} \leq v$ and $\mathbf{x} \geq \mathbf{0}$ strictly and is within a small ball of the true optimum $\mathbf{x}^*$.

Pick $\xi_i = M/\tilde{x}_i^*$ for some large M. With this choice, the equations in (3.14a) are written as

$$\sum_{i \in [1,n]} \frac{(\hat{\mu}_i^t - \lambda^{2,t})^2}{\tilde{x}_i^*} = M, \quad t \in [1,T]. \tag{3.15}$$

For each $t \in [1,T]$, start by letting $\lambda^{2,t} = \min_{i \in [1,n]} \{\hat{\mu}_i^t\}$ and decrease $\lambda^{2,t}$ until the equation in (3.15) is satisfied. This will happen, because the left hand side in (3.15) is continuous and monotonically increasing.

Now, let $z_i^t = \hat{\mu}_i^t - \lambda^{2,t}$. Let $\eta > 0$ be any positive real. Note that by choosing M sufficiently large, we can guarantee that, for each t, all z_i^t are within $1 + \eta$ of the smallest. Namely, for all $i \in [1,n]$, $k_t \leq z_i^t \leq (1+\eta)k_t$, where $k_t = \min_j z_j^t$. This is

because the $\hat{\mu}_i^t$ are fixed whereas $-\lambda^{2,t}$ increases as M increases.

For each $t \in [1,T]$, set $\alpha^t = \frac{1}{\sum_i \frac{z_i^t}{\xi_i}}$. Now let $x_i^t = \alpha^t \times \frac{z_i^t}{\xi_i}$. Note that we have $\sum_{i\in[1,n]} x_i^t = 1$ and $x_i^t > 0$. To show that $\mathbf{x}^t$ is feasible, we only need to show that it satisfies the variance constraint. This will follow from showing that $\mathbf{x}^t$ is in a small neighborhood of $\tilde{\mathbf{x}}^*$, which satisfies it strictly. We have $x_i^t = \alpha^t \times \frac{z_i^t}{\xi_i} = \alpha^t \times \frac{z_i^t \tilde{x}_i^*}{M}$. Using our bounds on z_i^t, we get

$$\frac{\alpha^t k_t}{M}\tilde{x}_i^* \leq x_i^t \leq (1+\eta)\frac{\alpha^t k_t}{M}\tilde{x}_i^*.$$

Adding over i we get $\frac{\alpha^t k_t}{M} \leq 1 \leq (1+\eta)\frac{\alpha^t k_t}{M}$. This implies

$$(1-\eta)\tilde{x}_i^* \leq x_i^t \leq (1+\eta)\tilde{x}_i^*.$$

This translates into a vanishing loss in the objective value as η goes to 0. And this concludes the proof. □

We conclude this section by noting that despite the exciting theoretical results of Sections 3.4 and 3.5, the non-convex non-linear bilevel model is intractable to solve using the current state-of-the-art software for global optimization. Developing an efficient method to solve the bilevel model presented in this section remains a future work.

3.6 Using the Identity Matrix as Estimation-Error Matrix

The results of the previous sections demonstrate that one can focus on diagonal estimation-error matrices and consequently deal with calibrating only n parameters in constructing the uncertainty sets. In this section, we restrict our attention to the simplest diagonal estimation-error matrix, the identity matrix. In this case, we only need to calibrate one parameter(κ), which is advantageous for decision makers.

Our goal is to investigate whether using an identity matrix as the estimation-error matrix can improve the performance of the classical Markowitz model. To this end, we perform a simulation where we compare the out-of-sample performances of the two approaches. We generate samples from the distribution Normal($\boldsymbol{\mu}, \boldsymbol{\Sigma}/N$). The value of N determines the accuracy of the samples we generate (increasing N results in better estimates of the expected return). Then, we solve the classical Markowitz model and the robust portfolio model for varying κ values and compare their actual performances. When $\kappa = \infty$, the robust portfolio becomes the equal-weight portfolio.

		$\kappa \times N$										
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	∞
N	1	13.0	18.3	19.3	19.4	19.3	19.2	19.1	19.0	18.9	18.9	18.4
	3	7.8	13.7	16.3	17.2	17.4	17.4	17.2	17.0	16.8	16.7	15.1
	6	5.3	10.1	13.1	14.6	15.2	15.3	15.2	15.0	14.8	14.6	11.4
	12	3.6	7.0	9.6	11.3	12.2	12.5	12.6	12.4	12.2	11.9	5.8
	24	2.4	4.6	6.4	7.8	8.7	9.1	9.2	9.0	8.7	8.3	-3.4
	120	1.2	2.0	2.7	3.1	3.3	3.2	2.8	2.3	1.6	0.8	-59.2

Table 3.1: Percentage gap closed by the robust model when $\boldsymbol{\Xi} = \mathbf{I}$ ($v = 0.002$).

Table 3.1 gives the result of our simulation. Each cell in Table 3.1 contains the percentage gap closed by the robust solution compared to the Markowitz solution. Specifically, in each cell of Table 3.1, we report $(\bar{R} - \bar{M})/(T - \bar{M})$ where $\bar{R}$ denotes the actual performance of the robust solution estimates, $\bar{M}$ denotes the actual performance of the Markowitz solution estimates, and T denotes the true return. For example, the entry 19.4 for $N = 1$ and $\kappa \times N = 0.4$ is obtained by first computing the estimates $\bar{M} = 0.01262$ and $\bar{R} = 0.01314$. Similarly, for $N = 120$ and $\kappa \times N = 0.5$, we have that $\bar{M} = 0.01395$ and $\bar{R} = 0.01399$ which yields the entry 3.3. The true return T is equal to 0.01535. Note that for each cell, the values $\bar{R}$ and $\bar{M}$ are averages over 10000 trials; consequently, the standard error for each cell in Table 3.1 is less than 0.5.

We see from Table 3.1 that the robust model can significantly outperform the Markowitz model when κ lies in a wide range of values. This is interesting, because the identity matrix contains no information about the problem or the relationship between the assets. Furthermore, the equal-weight portfolio outperforms the Markowitz solution estimates up to $N = 12$. On the other hand, the Markowitz solution estimates are better than equal-weight portfolio when we have more accurate samples (i.e., $N = 24$ and $N = 120$). It is important to note that the robust solution estimates are better than the Markowitz solution estimates even when the samples are very accurate (i.e., $N = 120$).

We repeat the experiments on a different data set based on four international equity indices. We use the monthly returns for Canada, Switzerland, the United Kingdom and the United States between 2004 and 2022. The data is obtained from Morgan Stanley Capital International (MSCI).

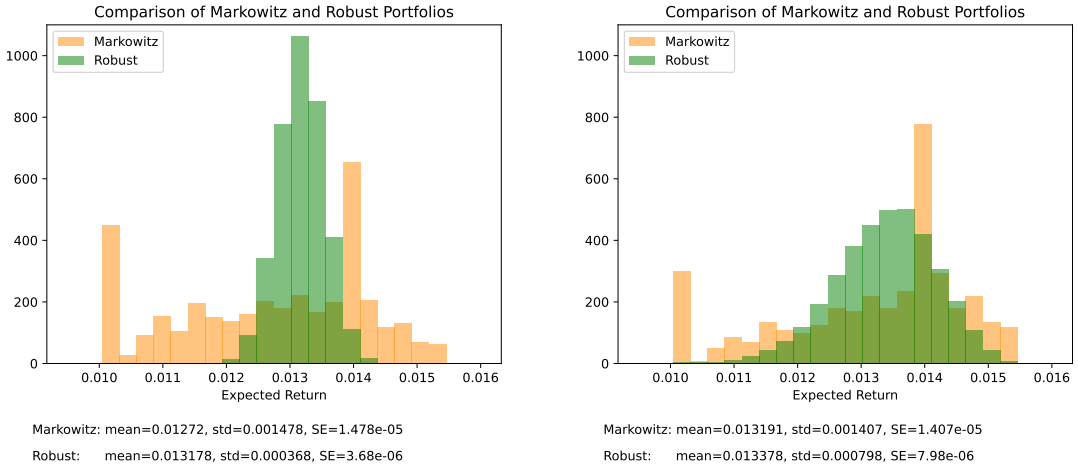
		$\kappa \times N$										
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	∞
N	1	32.9	31.5	30.8	30.5	30.3	30.2	30.1	30.0	30.0	29.9	29.6
	3	31.2	31.9	30.1	29.1	28.5	28.1	27.9	27.7	27.5	27.4	26.3
	6	28.2	32.8	30.6	28.6	27.4	26.5	26.0	25.5	25.2	24.9	22.6
	12	24.8	33.1	32.6	29.9	27.4	25.6	24.3	23.4	22.7	22.1	17.1
	24	21.5	32.0	34.4	32.8	29.7	26.6	23.9	21.8	20.1	18.8	7.4
	120	14.7	25.4	32.4	36.4	38.0	37.7	35.8	32.9	29.2	25.2	-63.3

Table 3.2: Percentage gap closed by the robust model when $\mathbf{\Xi} = \mathbf{I}$ ($v = 0.0013$).

We see from Table 3.2 that the robust portfolios perform better than the portfolios obtained from the classical Markowitz model. In this data set, the expected returns of the minimum variance portfolio and of the equal-weight portfolio are 0.00108 and 0.00473, respectively. The true expected return T is equal to 0.00559. We note that the Markowitz model beats the equal-weight portfolio only when $N = 120$, whereas the robust model beats the equal-weight portfolio under all $N - \kappa \times N$ combinations where $\kappa \times N$ is finite in Table 3.2. Furthermore, the minimum variance portfolio is outperformed by both the Markowitz model and the robust model under all $N-\kappa\times N$ combinations.

The results in Table 3.1 and 3.2 illustrate that the $\kappa \times N$ values that perform best vary between 0.2 and 0.7. We observe that keeping $\kappa \times N$ constant when N varies works well for improving the actual performance of the portfolio. Not surprisingly, the need for robustification diminishes when we have better estimates as N increases.

In order to understand the results better, we focus on two particular distributions $N = 3$ and $N = 24$ when $\kappa \times N = 0.5$ in Table 3.1. We present the averages $\bar{M}$ and $\bar{R}$ as a histogram. Figure 3-3 provides a clearer picture for the superior performance



(a) $n = 3$ (b) $n = 24$

Figure 3-3: Histograms of the actual performances of Robust and Markowitz models.

of the robust model. We see that the empirical distributions of the actual returns under the robust model have a smaller variance. On the other hand, the actual returns under the Markowtiz model features outliers that perform very poorly. As a result, the robust model outperforms the Markowitz model, and the difference is significant.

3.7 Conclusion

In this paper, we take a theoretical look at the robust portfolio optimization problem. Our main contribution is to show that the family of diagonal estimation-error matrices is sufficiently rich to achieve an arbitrarily small loss compared to the optimum expected return. We perform illustrative computational experiments to show that even using an identity matrix can improve on the actual performance of the Markowitz model when the size of the uncertainty set is correctly calibrated.

Our work is a step towards constructing better estimation-error matrices for robust portfolio optimization. Our bilevel model can be used to construct good estimation-error matrices using the available data. This will require developing a more efficient method to solve the bilevel model. Another research direction would be a theoretical analysis on the use of identity matrices as estimation-error matrices.

Chapter 4

Structural Properties of Equitable and Efficient Distributions

This chapter is a joint work with John Hooker and Peter Zhang.

4.1 Introduction

Optimization offers a powerful tool for identifying an efficient and equitable allocation of resources. By maximizing a suitable objective function subject to resource limits and other constraints, one can find the best possible allocation of resources as measured by that function. Far and away the most widely used objective is the maximization of total utility, which can take the form of minimizing cost or maximizing revenue or some other benefit. Yet a purely utilitarian criterion lacks an explicit measure of equity. When maximized subject to typical resource constraints, it can lead to extreme and unsatisfactory solutions that benefit a very few at the

expense of the many. If this result does not occur for many optimization models used in practice, it is because the complexity of the constraint set excludes extreme solutions—not because they are recognized as unjust, but simply because they do not happen to satisfy all of the constraints. In effect, the constraints conceal the inherent inadequacy of the objective function.

This poses the challenge of identifying an objective function that incorporates within itself a criterion for equity as well as efficiency. We address this challenge by examining the structure of optimal solutions that result from maximizing a variety of social welfare functions (SWFs), subject to basic resource limitations. We find that not only a utilitarian objective, but some of the best-known fairness measures, can result in extreme and often unacceptable solutions. Again, if these extreme solutions are not often observed, it is because they are excluded by side constraints that reflect the exigencies of the situation rather than some underlying concept of fairness. This indicates that one must look further for an SWF that encapsulates an adequate concept of equity.

With this goal in mind, we examine a series of increasingly sophisticated social welfare functions. They include utilitarian, Rawlsian maximin, and leximax SWFs, as well as alpha fairness, proportional fairness (the Nash bargaining solution), the Kalai–Smorodinsky bargaining solution, and recently proposed threshold functions that combine utilitarian with maximin or leximax fairness. We focus primarily on SWFs that combine efficiency and equity criteria in some fashion, partly because this is an obvious strategy for avoiding the extreme outcomes, and partly because efficiency is as important as equity in most practical applications. We derive the structural properties of optimal solutions that result when these SWFs are maximized subject to simple but generic constraints that form the core of a wide variety of

applications. Specifically, we suppose there is a budget limit that constrains total available resources, optional bounds on each party's utility, and a linear (or concave piece-wise linear) utility function that links each party's utility to the resources provided. To our knowledge, very few of these structural results appear in the literature. We find that while each SWF avoids some of the extreme solutions associated with the previous ones, it introduces anomalies of its own. Only the last criterion seems to avoid these difficulties, although it may itself require further refinement.

We also examine the structure of solutions in a hierarchical distribution network. This represents the common situation in which a national authority allocates resources to regions, which in turn combine these with their own resources for distribution to its subregions or institutions. We find that the more sophisticated SWFs are more likely to be regionally nondecomposable, as perhaps they should be. This means that the regional authorities must take into account the national picture before they can equitably allocate resources within their own territory.

The paper begins with a statement of the generic optimization problem, followed by a healthcare example that illustrates how this type of problem can occur in practice. It then states the optimization problem on a hierarchical network and defines concepts of collapsibility, monotone separability, and regional decomposability. Next, incentives from the perspectives of individuals are discussed. Concretely, such questions are studied by looking at the behavior of the optimal solutions as the model parameters change. Following this, it considers a sequence of SWFs and derives properties of the optimal solutions they deliver. Proofs may be found in the Appendix. The paper concludes by drawing lessons from these results.

4.2 The Optimization Problem

We wish to maximize social welfare subject to a budget constraint. If $\mathbf{x} = (x_1, \ldots, x_n)$ are the resources allotted to individuals $1, \ldots, n$, a general distribution problem may be stated

$$\max_{\mathbf{x} \in \mathbb{R}^n} \left\{ W\big(\mathbf{U}(\mathbf{x})\big) \;\middle|\; \sum_j x_j \leq B,\ \bar{\mathbf{c}} \leq \mathbf{x} \leq \bar{\mathbf{d}} \right\} \tag{4.1}$$

The social welfare function $W(\mathbf{u})$ measures the desirability of a distribution of utilities $\mathbf{u} = (u_1, \ldots, u_n)$ across individuals $1, \ldots, n$. The utility function $\mathbf{U}$ determines the vector $\mathbf{u} = \mathbf{U}(\mathbf{x})$ of utilities resulting from resource allotment $\mathbf{x} = (x_1, \ldots, x_n)$. The budget constraint $\sum_j x_j \leq B$ limits total resource consumption to B. The bounds $\bar{\mathbf{c}} \leq \mathbf{x} \leq \bar{\mathbf{d}}$ constrain the resources allotted to individuals by requiring that $\bar{c}_j \leq x_j \leq \bar{d}_j$ for each j.

We will focus on linear utility functions of the form

$$\mathbf{U}(\mathbf{x}) = (x_1/a_1, \ldots, x_n/a_n) \tag{4.2}$$

where each $a_j > 0$ and $\mathbf{c} \geq \mathbf{0}$. This allows us to eliminate $\mathbf{x}$ and write (4.1) as

$$\max_{\mathbf{u} \in \mathbb{R}^n} \left\{ W(\mathbf{u}) \;\middle|\; \mathbf{a}^\mathsf{T}\mathbf{u} \leq B,\ \mathbf{c} \leq \mathbf{u} \leq \mathbf{d} \right\} \tag{4.3}$$

where $c_j = \bar{c}_j/a_j$ and $d_j = \bar{d}_j/a_j$ for all j. Thus a large coefficient a_j indicates that it is expensive to provide for the welfare of individual j, perhaps due to a disease that is costly to treat. The lower bounds impose a floor on the welfare of each individual, or may reflect a default utility level without resources. The upper bounds reflect the fact that greater resources can yield greater utility only up to a point; once the disease is cured, there is no need to provide more medical resources.

To simplify notation, we assume individuals are indexed so that $a_1 \leq \cdots \leq a_n$. This means that individual 1's welfare is the least costly to provide. We also assume without loss of generality that $0 \leq c_j \leq d_j \leq B/a_j$ for each j, since $0 \leq u_j \leq B/a_j$ is already enforced by the budget constraint and $\mathbf{u} \geq \mathbf{0}$.

Finally, it is often revealing to investigate problem (4.3) without upper and/or lower bounds on the utilities. When there are no positive lower bounds ($\mathbf{c} = \mathbf{0}$), (4.3) becomes

$$\max_{\mathbf{u}\in\mathbb{R}^n} \left\{ W(\mathbf{u}) \;\middle|\; \mathbf{a}^\mathsf{T}\mathbf{u} \leq B,\ \mathbf{0} \leq \mathbf{u} \leq \mathbf{d} \right\} \tag{4.4}$$

which we refer to as maximizing social welfare *subject to a budget constraint and upper bounds*. When, in addition, there are no nontrivial upper bounds ($d_j = B/a_j$ for all j), model (4.3) becomes

$$\max_{\mathbf{u}\in\mathbb{R}^n} \left\{ W(\mathbf{u}) \;\middle|\; \mathbf{a}^\mathsf{T}\mathbf{u} \leq B,\ \mathbf{u} \geq \mathbf{0} \right\} \tag{4.5}$$

which we refer to as maximizing social welfare *subject to a budget constraint*.

The model (4.3) accommodates a wide variety of resource allocation scenarios, one of which is described in the next section. Yet in some cases it may be desirable to measure utility as a nonlinear function of resources, as when there are decreasing returns to scale. In the latter case, the utility function $U(\mathbf{x})$ is concave and can be approximated by imposing a system $\mathbf{Au} \leq \mathbf{B}$ of linear budget constraints. In such cases one can solve the problem

$$\max_{\mathbf{u}\in\mathbb{R}^n} \left\{ F(\mathbf{u}) \;\middle|\; \mathbf{Au} \leq \mathbf{B},\ \mathbf{c} \leq \mathbf{u} \leq \mathbf{d} \right\} \tag{4.6}$$

where each budget constraint has the form $A_i\mathbf{u} \leq B_i$ for $i = 1, \ldots, m$.

4.3 A Motivating Example

A health provision problem solved by Hooker and Williams (2012) illustrates how the optimization model of the previous section can occur in practice. Hooker and Williams solved the problem using their threshold SWF (described in Section 4.9), but any of the SWFs we survey can be used. In this section, we focus on the problem constraints.

The problem is to allocate healthcare resources in a manner that is both equitable and efficient, subject to a budget limitation. We are given m treatment groups that are distinguished by the severity and prognosis of the disease. Each group i has size n_i. We let c_i be the cost per patient of administering the treatment, q_i the average net gain in quality-adjusted life years (QALYs) for a member of group i when the treatment is administered, and α_i is the average QALYs that results from medical management without the treatment in question. The budget constraint is

$$\sum_i n_i c_i y_i \leq \tilde{B} \tag{4.7}$$

where

$$u_i = q_i y_i + \alpha_i \tag{4.8}$$

and $0 \leq y_i \leq 1$. The variables y_i indicate the fraction of patients in group i provided treatment. In the event of partial funding, medical personnel make triage decisions based on individual situations. The expected utility in each group (measured in QALYs) is formulated as an affine function of resources, because the additional expected utility resulting from treatment is reasonably seen as proportional to the number of patients treated.

Since we have from (4.8) that $y_i = (u_i - \alpha_i)/q_i$, we substitute this into (4.7) to get the budget constraint

$$\sum_i \frac{n_i c_i}{q_i} u_i \leq \tilde{B} + \sum_i \frac{n_i c_i \alpha_i}{q_i} \tag{4.9}$$

Also (4.8) and $0 \leq y_i \leq 1$ imply the lower and upper bounds

$$\alpha_i \leq u_i \leq q_i + \alpha_i \tag{4.10}$$

Now the problem of maximizing ($\mathbf{u}$) subject to (4.9) and (4.10) has the form of our general model (4.3), where $a_i = n_i c_i / q_i$ and $B = \tilde{B} + \sum_i n_i c_i \alpha_i / q_i$. This example illustrates how model (4.3) can encompass a broad class of resource allocation problems with linear utility functions.

4.4 Hierarchical Distribution

We also examine distribution on a hierarchical network, a type of allocation problem that frequently arises in real-world applications (Simchi-Levi et al., 2019). The resulting optimization problem is a special case of (4.6). We show analysis in this section with a two-level network setup. But all definitions and results can be generalized to a hierarchical network with any finite number of levels, which we briefly comment on at the end of the section. Each region k has an existing resource budget B_k, and the national government must decide how much resources y_k to allocate to

each region. If there are r regions, the distribution problem (4.1) becomes

$$\max_{\mathbf{x},\mathbf{y}} \left\{ W\big(\mathbf{U}(\mathbf{x})\big) \;\middle|\; \begin{array}{l} \displaystyle\sum_{k=1}^{r} y_k \le B,\ \bar{\mathbf{c}} \le \mathbf{x} \le \bar{\mathbf{d}},\ \mathbf{y} \ge \mathbf{0} \\ \displaystyle\sum_{j \in J_k} x_j \le B_k + y_k,\ k = 1, \ldots, r \end{array} \right\}$$

where J_k is the index set for the subregions in region k. Again using the linear utility function (4.2), this problem becomes

$$\max_{\mathbf{y},\mathbf{u}} \left\{ W(\mathbf{u}) \;\middle|\; \begin{array}{l} \mathbf{e}^{\mathsf{T}}\mathbf{y} \le B,\ \mathbf{c} \le \mathbf{u} \le \mathbf{d},\ \mathbf{y} \ge \mathbf{0} \\ \mathbf{a}^k \mathbf{u}^k \le B_k + y_k,\ k = 1, \ldots, r \end{array} \right\} \tag{4.11}$$

where $\mathbf{e} = (1, \ldots, 1)$ and vector $\mathbf{u}^k$ contains the utilities of subregions of region k.

Interestingly, if we drop the requirement $\mathbf{y} \ge \mathbf{0}$ (i.e, we allow the national government to take resources from the regions), the model collapses into a single-level problem:

$$\max_{\mathbf{u}} \left\{ W(\mathbf{u}) \;\middle|\; \mathbf{a}^{\mathsf{T}}\mathbf{u} \le B + \sum_{k=1}^{r} B_k,\ \mathbf{c} \le \mathbf{u} \le \mathbf{d} \right\} \tag{4.12}$$

We will say that a hierarchical problem is *collapsible* if it can be solved by solving its collapsed version (4.12). A problem is collapsible if each region's allocation in the collapsed problem (4.12) is no less than its stock already on hand.

Proposition 29. *A hierarchical problem* (4.11) *is collapsible if for any optimal solution* $\bar{\mathbf{u}}$ *of* (4.12)*,* $\mathbf{a}^k \bar{\mathbf{u}}^k \ge B_k$ *for* $k = 1, \ldots, r$*.*

Paradoxically, the individual regions may not compute the same distribution for their subregions as recommended by national planners, even when they use the same

social welfare function. Examples of this are given in Sections 4.9 and 4.10. We will say that a problem is *regionally decomposable* when this issue does not arise; that is, when the optimal distribution within any region, given the resource subsidy provided by the national solution, is the same as in the national solution. More precisely, (4.11) is regionally decomposable if, given any optimal solution $(\bar{\mathbf{u}}, \bar{\mathbf{y}})$ of (4.11), and given any set of solutions $\hat{\mathbf{u}}^1, \ldots, \hat{\mathbf{u}}^r$ that are respectively optimal in the regional distribution problems

$$\max_{\mathbf{u}^k} \left\{ W(\mathbf{u}^k) \;\middle|\; \mathbf{a}^k \mathbf{u}^k \leq B_k + \bar{y}_k,\ \mathbf{c}^k \leq \mathbf{u}^k \leq \mathbf{d}^k \right\}, \quad k = 1, \ldots, r \tag{4.13}$$

the solution $(\mathbf{u}, \mathbf{y}) = (\hat{\mathbf{u}}^1, \ldots, \hat{\mathbf{u}}^r, \bar{\mathbf{y}})$ is optimal in (4.11).

A key to regional decomposability is *monotonic separability* of $W(\mathbf{u})$. This means that for any partition $\mathbf{u} = (\mathbf{u}^1, \mathbf{u}^2)$, $W(\bar{\mathbf{u}}^1) \geq W(\mathbf{u}^1)$ and $W(\bar{\mathbf{u}}^2) \geq W(\mathbf{u}^2)$ imply $W(\bar{\mathbf{u}}) \geq W(\mathbf{u})$. In particular, a separable function[1] $W(\mathbf{u}) = \sum_j W_j(u_j)$ is monotonically separable. Then we have

Proposition 30. *If $W(\mathbf{u})$ is monotonically separable, problem* (4.11) *is regionally decomposable.*

Thus if the SWF is monotonically separable, the regions will distribute their allotment in a way consistent with a nationally optimal solution. However, if the SWF is not monotonically separable, or some regions use a different social welfare criterion, a region's distribution to its subregions may depart from the national plan. If so, the nation's allocation of resources to regions is based on the false assumption

[1] A common assumption made by early research in this area, such as Dalton (1920) and Atkinson (1970). For more discussion on such a welfare function, we refer readers to Section 4.7.

that the resources will be distributed within regions as prescribed by the national solution.

In a network with more than two levels, the definition of collapsibility and Proposition 29 essentially remain the same. For regional decomposability, one needs to generalize the definition in a natural way to reflect the consistency of resource allocations at *all* levels. Proposition 30 remains the same.

4.5 Incentives and Sharing

A social welfare solution raises two important and related questions. One is whether an individual i receives more utility allotment by having a more efficient utility conversion; that is, by reducing a_i in the resource constraint. Another question is how such an efficiency improvement affects the welfare of other individuals. Using concepts from Sen's capabilities approach (Sen, 1995), we call a_i the *conversion efficiency* of individual i. In most cases, we simply refer to it as efficiency, unless there is ambiguity between "conversion efficiency" and "sum of everyone's utilities".

Many factors contribute to conversion efficiencies. For the purpose of our discussion, we categorize them into personal, technological, and others. Personal factors of individual i refer to the factors that each individual i has control over, such as individual i's exercise routine and diet that could affect the treatment effectiveness of certain drugs. Technological factors refer to the factors that our social planner has control over, such as the development of a new drug that is cheaper and more effective, therefore can convert the same healthcare resources to a larger increase in overall societal QALYs. While personal factors affect one a_i at a time, technological factors can affect one or more a_i's simultaneously.

We can ask questions regarding both personal and technological factors. For personal factors, we ask whether there is a personal incentive to become more efficient, and whether doing so benefits others or pits one in competition against them. One might hypothesize that a greater emphasis on equity in a SWF results in more sharing of benefits and therefore encourages cooperation. For technological factors, we ask whether the social planner has an incentive to improve the conversion efficiencies for a certain subset of individuals, and who is in this subset. This is especially interesting if we think about two different ways of investing additional societal resource: directly investing the resource in the form of increased budget B, or indirectly investing the resource by improving conversion efficiencies, i.e., decreasing a_i. Investing directly into B usually benefits every individual i at the rate of $1/a_i$ or 0. On the other hand, as we will see, indirect investment via conversion efficiencies leads to a different set of rates under each social welfare functions. In general, the indirect investment could lead to outcomes that are more balanced than $1/a_i$ ($1/a_N^2$ rate for each a_i) or less balanced than $1/a_i$ ($1/a_i^2$ for each a_i), or somewhere in between.

We address these questions mathematically by observing how reducing an individual i's coefficient a_i in the budget constraint changes a socially optimal solution $\mathbf{u}^*$. For purposes of this analysis, we assume the optimum is obtained subject only to a resource constraint, without upper bounds on utilities. We focus on effects at the margin, which means that we investigate only the effect of small perturbations in a_i. To simplify the analysis without undermining basic insights, we ignore points at which at which the solution $\mathbf{u}^*$ is a discontinuous function of $\mathbf{a}$. This allows us to express the sensitivity of an individual's utility u_j^* to reductions in a_i as a negative partial derivative: the rate of increase of u_j^* with respect to decreases in a_i. In addition, because we are dealing with small perturbations, the effect of technology

improvement can be directly approximated as the sum of these partial derivatives.

4.6 Utilitarian, Maximin, and Leximax Criteria.

4.6.1 Socially Optimal Distributions

An analysis of utilitarian, maximin and leximax social welfare functions is straightforward but a useful starting point. The utilitarian SWF

$$W(\mathbf{u}) = \sum_j u_j$$

stems ultimately from Jeremy Bentham's idea that one should maximize the greatest good for the greatest number (Bentham, 1789). While many optimization models are designed to achieve this goal, a consistently utilitarian objective can lead to extreme distributions. This is borne out by solutions of the optimization problem (4.3).

Suppose that utilities are indexed so that $a_1 \leq \cdots \leq a_n$, which means that that utility is least expensive for individual 1. Then utility is maximized by allocating all available utility to individual 1, up to d_1, then all remaining utility to individual 2 up to d_2, and so forth. Thus we have the simple result,

Proposition 31. *If* $a_1 \leq \cdots \leq a_n$ *and*

$$\phi_j = \frac{1}{a_j}\Big(B - \sum_{\ell=1}^{j-1} a_\ell d_\ell\Big), \text{ for } j = 1, \ldots, n$$

then maximizing utility subject to a resource constraint and upper bounds yields the

optimal solution

$$u_j^* = \begin{cases} d_j & \text{for } j = 1, \ldots, k \\ \phi_{k+1} & \text{for } j = k+1 \\ 0 & \text{for } j = j+2, \ldots, n \end{cases}$$

where $k = \max\{j \in \{0, \ldots, n\} \mid d_j \leq \phi_j\}$. *In particular, if* $B/a_1 \leq d_1$, *then person 1 receives all available utility, and everyone else receives nothing.*

While a utilitarian policy of lavishing all resources on a single or a few individuals is often unacceptable, a maximin criterion can also lead to questionable outcomes. It is inspired by the Difference Principle of John Rawls (Rawls, 1999), which has been discussed in a large literature (surveyed in Richardson and Weithman (1999); Freeman (2003)). It states roughly that inequality is justified only to the extent that it improves the lot of the worst-off. The criterion therefore seeks to maximize the minimum utility and corresponds to the simple SWF

$$W(\mathbf{u}) = \min_j \{u_j\}$$

While Rawls intended the principle only to apply to the design of social institutions and the distribution of "primary goods," it can be investigated as a possible rule for fair distribution in general.

The maximin solution subject to a budget constraint splits utility equally among individuals. Thus if we define $a_J = \sum_{j \in J} a_j$ and $N = \{1, \ldots, n\}$, we have $u_j^* = B/a_N$ for all j, so that person j receives resource allotment $B(a_j/a_N)$. This can require a very unequal distribution of resources when $a_1 \ll a_n$. For example, when devoting all available resources to a seriously ill patient yields only a slight improvement, nearly all resources must be devoted to that person, and the welfare of everyone else

reduced to the same low level. The outcome is similar if there are upper bounds. If we let $d_{\min} = \min_j\{d_j\}$, then we have

Proposition 32. *A maximin solution subject to a resource constraint and upper bounds allots equal utility* $u_j^* = \min\{d_{\min}, B/a_N\}$ *to all persons* j.

The proposition reveals another problem with the maximin criterion. A solution in which the utilities are equal to $d_{\min}$ can waste much of the available resources, specifically $\max\{0, B - a_N d_{\min}\}$.

This anomaly is avoided by moving to a *leximax* criterion (lexicographic maximization). It maximizes the smallest utility $\min_j\{u_j\}$, then while fixing $\min_j\{u_j\}$ at that level, maximizes the second smallest utility, and so forth. The social contract argument with which Rawls defends the maximin criterion can reasonably be extended to a leximax criterion.

This is a different sense of lexicographic maximization than is often used. When comparing utility distributions $\mathbf{u}$, it assumes that the utilities in $\mathbf{u}$ are ordered by size rather than their positions in the sequence $u_1, \ldots, u_n$. To define this sense precisely, we say that utility distribution $\mathbf{u}$ *dominates* $\mathbf{u}'$ when there is permutation $\boldsymbol{\pi}$ of $1, \ldots, n$ for which $u_{\pi_1} \leq \cdots \leq u_{\pi_n}$, a permutation $\boldsymbol{\pi}'$ for which $u'_{\pi'_1} \leq \cdots \leq u'_{\pi'_n}$, and an index $k \in N$, such that the following holds: $u_{\pi_j} = u'_{\pi'_j}$ for $j = 1, \ldots, k-1$ and $u_{\pi_k} > u'_{\pi'_k}$. Thus, while $\mathbf{u} = (1, 3, 2)$ dominates $\mathbf{u}' = (1, 2, 4)$ in the usual sense, $\mathbf{u}'$ dominates $\mathbf{u}$ in the sense defined here. We now say that $\mathbf{u}^* \in U$ is a leximax solution over a feasible set U if no element of U dominates $\mathbf{u}^*$.

A leximax solution subject to a budget constraint and upper bounds can be obtained as follows: set utilities with the k smallest upper bounds to those upper bounds, and

then set the remaining utilities to an equal value that is selected to exhaust the remaining resources. The number k depends on the specific problem data. Thus we have the following.

Proposition 33. *Suppose that utilities $u_1, \ldots, u_n$ are indexed so that their upper bounds satisfy $d_1 \leq \cdots \leq d_n$, and that*

$$\phi_j = \Big(B - \sum_{\ell=1}^{j-1} a_\ell d_\ell\Big)\Big(a_N - \sum_{\ell=1}^{j-1} a_\ell\Big)^{-1}, \quad \textit{for } j = 1, \ldots, n$$

Then a leximax solution $\mathbf{u}^*$ subject to a resource constraint and the upper bounds is given by

$$u_j^* = \begin{cases} d_j & \textit{for } j = 1, \ldots, k \\ \phi_{k+1} & \textit{for } j = k+1, \ldots, n \end{cases}$$

where $k = \max\{j \in \{0, \ldots, n\} \mid d_j \leq \phi_j\}$ and $d_0 = \phi_0 = 0$. Furthermore, this solution consumes all available resources.

4.6.2 Hierarchical Distributions

The hierarchical problem (4.11) is regionally decomposable for the utilitarian criterion, because the SWF is separable and therefore monotonically separable. The maximin SWF is not separable, but it is monotonically separable and therefore regionally decomposable due to Proposition 30.

Proposition 30 cannot be applied to the leximax criterion, because it is not represented by a single SWF. However, a property analogous to monotone separability can be defined for lexicographic comparisons as shown in the Appendix, and it allows one to establish that the hierarchical problem is regionally decomposable for the

leximax criterion. Regional decomposability is defined for this problem by replacing $\max\{W(\mathbf{u})\}$ with $\text{leximax}\{\mathbf{u}\}$ in (4.11) and (4.13). Thus we have

Proposition 34. *The hierarchical problem* (4.11) *is regionally decomposable for a leximax criterion.*

The utilitarian criterion is collapsible only in the degenerate case where $B = B_k = 0$ for all $k \neq 1$. The utilitarian solution is again extreme. If there are no upper bounds, for example, at most one subregion in each region receives positive utility. A straightforward argument shows that the maximin problem is collapsible when

$$B + \sum_i B_i \geq \min\left\{a_N d_{\min},\ \max_k \left\{(a_N/a_{J_k})B_k\right\}\right\}$$

where $d_{\min} = \min_j\{d_j\}$. An optimal solution of the collapsed problem is

$$u_j^* = \min\left\{d_{\min},\ (1/a_N)\Big(B + \sum_k B_k\Big)\right\},\ \text{all } j$$

4.6.3 Incentives and Sharing

Finally, we examine the effect on a socially optimal solution $\mathbf{u}^*$ (subject to a resource constraint) when an individual i improves personal efficiency by reducing a_i. In the utilitarian problem, only the individual who receives all resources ($i = 1$) benefits, and everyone else continues to receive zero utility. A reduction in a_1 increases this individual's utility at the rate $B/a_1^2 = u_1^*/a_1$. In the maximin problem, everyone shares the surplus equally when one individual improves efficiency. A reduction in any a_i increases every individual j's utility at the rather small rate $(a_i/a_N^2)B$.

Since one's efficiency improvements benefit oneself as well as others, one's incentive to improve is inversely proportional to one's current efficiency. This implies a leveling tendency that complements the equal distribution of utility in a maximin solution. Those who waste more resources have a stronger incentive to cut waste. Those who require greater resources due to ill health, or some other debilitating condition, benefit more in a maximin solution from reductions in the cost of treatment or therapy.

We can define the societal utility increase rate from investment into i as $-\partial \sum_j u_j^* / \partial a_i$, and define the gradient of technology investment as the change in total utility as we change different a_i's, $(-\partial \sum_j u_j^* / \partial a_1, -\partial \sum_j u_j^* / \partial a_2, \dots)$. It is straightforward to see that under a utilitarian framework, investment in technological efficiency leads to a utility increase rate proportional to $1/a_i^2$ for the most efficient individual i, and 0 otherwise. So the gradient of technology investment is in the form of $(0, \dots, a_{\min}^{-2}, 0, 0, \dots)$. Under a maximin SWF, the utility increase rate is nB/a_N^2 for every i, therefore the gradient of technology investment is proportional to $(a_N^{-2}, a_N^{-2}, \dots)$.

4.7 Alpha Fairness

4.7.1 Socially Optimal Distribution

Alpha fairness is perhaps the most popular criterion for balancing equity and efficiency Mo and Walrand (2000); Lan et al. (2010); Bertsimas et al. (2012). It has the advantage of allowing one to regulate the balance with a parameter α, where larger values of α place a greater emphasis on fairness. In particular, $\alpha = 0$ corresponds to a purely utilitarian criterion, and $\alpha = \infty$ to a maximin criterion. An

important special case is proportional fairness, also known as the Nash bargaining solution Nash (1950), which is frequently used in such engineering contexts as telecommunications and traffic signal timing Mazumdar et al. (1991); Kelly et al. (1998).

The alpha fairness SWF is

$$W_\alpha(\mathbf{u}) = \begin{cases} \dfrac{1}{1-\alpha}\displaystyle\sum_j u_j^{1-\alpha}, & \text{if } \alpha \geq 0 \text{ and } \alpha \neq 1 \\ \displaystyle\sum_j \log(u_j), & \text{if } \alpha = 1 \end{cases}$$

The special case of $\alpha = 1$ corresponds to the Nash bargaining solution. Nash provided an axiomatic justification for this solution, but it rests on a strong axiom of interpersonal noncomparability that arguably rules out the possibility of assessing distributive justice. The Nash solution is also the outcome of certain "rational" bargaining procedures, but they, too, rely on strong assumptions.

Since the alpha fairness SWF is concave (strictly concave for $\alpha > 0$), classical optimality conditions yield a simple closed-form solution for the optimization problem without utility bounds.

Proposition 35. *Maximizing alpha fairness subject to a resource constraint yields an optimal solution in which*

$$u_i^* = \frac{B}{a_i^{1/\alpha} \displaystyle\sum_j a_j^{1-1/\alpha}}, \; i = 1, \ldots, n$$

In the case of proportional fairness ($\alpha = 1$), each individual i receives utility $u_i^* =$

$B/(na_i)$.

It is evident from Proposition 35 that alpha fairness gives some priority to individuals with a smaller budget coefficient a_i, but without giving everything to one individual (if $\alpha > 0$) as in a utilitarian context. As α increases, the solutions transform smoothly from utilitarian to maximin, and the allotments become more egalitarian. Proportional fairness ($\alpha = 1$) is appropriately named, because it results in utility allotments that are exactly proportional to efficiency. A solution can be derived for the case when upper bounds are present, but it is complicated to state and yields little structural insight.

A difficulty with alpha fairness is that the parameter α is difficult to interpret in practice. It is somewhat helpful to characterize mathematically a welfare-preserving transfer of utility from one individual to another. If $u_k > u_j$, then individual k's utility must be reduced by $(u_k/u_j)^\alpha$ to compensate for a one-unit increase in individual j's utility, if total social welfare is to remain constant. Thus equality is a stronger imperative for larger α, but it is not obvious what particular value of α is appropriate in a given context.

Alpha fairness is also capable of extreme solutions in a nonconvex feasible set, because it can assign equality the same social welfare as arbitrarily extreme inequality. In a 2-player situation, for example, the distribution $\mathbf{u} = (1, 1)$ has the same social welfare value as (t, T), where

$$t = \begin{cases} 1/T & \text{if } \alpha = 1 \\ \left(2 - T^{1-\alpha}\right)^{1/(1-\alpha)} & \text{if } \alpha > 1 \text{ and } T^{1-\alpha} < 2 \end{cases}$$

Thus for $\alpha = 1$, we have $t \to 0$ has $T \to \infty$, and for $\alpha > 1$, $t \to 2^{1/(1-\alpha)}$ as $T \to \infty$, even when social welfare is held fixed. If the feasible set is the union of the box $[0,1] \times [0,1]$ with the box $[0,t] \times [0,T]$, both $[1,1]$ and $[t,T]$ are optimal. Alpha fairness can judge an egalitarian solution to be no better than a solution in which there is arbitrarily extreme inequality. This anomaly does not arise when $0 \leq \alpha < 1$.

4.7.2 Hierarchical Distribution

The hierarchical problem is regionally decomposable with an alpha fairness criterion, since the SWF is separable and therefore monotonically separable (Proposition 30). It is collapsible under the condition stated in Proposition 29.

4.7.3 Incentives and Sharing

An alpha fairness solution responds to increases in efficiency in ways that might be expected: there is more sharing when α is larger. For a given $\alpha > 0$, a reduction in coefficient a_i changes another individual j's allotment u_j^* at the rate

$$\frac{(u_j^*)^2}{B}\Big(1 - \frac{1}{\alpha}\Big)\Big(\frac{a_j}{a_i}\Big)^{1/\alpha} \tag{4.14}$$

It increases individual i's own utility at the rate

$$\frac{(u_i^*)^2}{B}\Big[1 + \frac{1}{\alpha}\Big(a_i^{1/\alpha - 1}\sum_j a_j^{1-1/\alpha} - 1\Big)\Big] \tag{4.15}$$

We see from (4.14) that an improvement in individual i's utility increases the utility of others only when $\alpha > 1$, since $1 - 1/\alpha > 0$ in this case. Thus as one might anticipate, the surplus utility is shared with others only when equity is viewed as

relatively important ($\alpha > 1$). Resources are diverted from others when equity is less important ($\alpha < 1$), due to the relative emphasis on utilitarianism and individual i's enhanced ability to make use of those resources. Proportional fairness ($\alpha = 1$) dictates that one's efficiency improvements have no marginal effect on the utility of other individuals. Finally, (4.15) tells us that one always benefits personally from one's own efficiency improvements, regardless of the value of α, so long as $\alpha > 0$. This is because

$$\sum_j a_j^{1-1/\alpha} > a_i^{1-1/\alpha}$$

implies that $1/\alpha$ is multiplied by a positive number in (4.15). When $\alpha = 0$, the solution $\mathbf{u}^*$ is utilitarian and behaves as described in the previous section.

For the social planner, the gradient of technology investment becomes more involved with an arbitrary α, but we can observe that as α changes from 0 to 1 to ∞, the gradient goes from $(0, \dots, a_{\min}^{-2}, 0, 0, \dots)$, to $\approx$ to $(a_N^{-2}, a_N^{-2}, \dots)$.

4.8 Kalai–Smorodinsky Bargaining

4.8.1 Socially Optimal Distribution

The Kalai–Smorodinsky (K–S) bargaining solution minimizes each person's relative concession (Kalai and Smorodinsky, 1975) . It is defined as the feasible vector $\mathbf{u}$ of utilities that maximizes a scalar β subject to $\mathbf{u} = \beta \mathbf{u}^{\max}$, where $u_j^{\max}$ is the "ideal" utility for individual j (i.e., the maximum of u_j over all feasible utility distributions). The K–S solution therefore maximizes each individual's fraction of his or her ideal utility, subject to the condition that this fraction is the same for all individuals. This can be interpreted geometrically as the furthest feasible point from the origin on the

line segment connecting the origin and $\mathbf{u}^{\max}$. A curious feature of the K–S criterion is that it proposes no SWF in the usual sense. The social welfare of a distribution $\mathbf{u}$ that lies even slightly off this line segment is undefined.

The K–S solution is easily derived for a budget constraint, with or without upper bounds. Recall that we assume (without loss of generality) that the upper bounds d_j satisfy $d_j \leq B/a_j$ for all j, which means that $\mathbf{u}^{\max} = \mathbf{d}$. We assume in this section that $\mathbf{a}^\mathsf{T}\mathbf{d} \geq B$, since otherwise the budget constraint plays no role, and the K–S solution simply sets each utility equal to its upper bound ($\mathbf{u} = \mathbf{d}$).

Proposition 36. *If $\mathbf{a}^\mathsf{T}\mathbf{d} \geq B$, then the Kalai–Smorodinsky bargaining solution subject to a budget constraint and upper bounds $\mathbf{d}$ is $\mathbf{u}^* = B\mathbf{d}/\mathbf{a}^\mathsf{T}\mathbf{d}$. Otherwise, the solution is $\mathbf{u}^* = \mathbf{d}$. If there are no upper bounds, the solution is $u_i^* = (1/n)(B/a_i)$ for all i.*

Thus, in the absence of upper bounds, each individual j receives $1/n$ of his or her ideal utility B/a_j. This is the same solution as obtained under proportional fairness (alpha fairness with $\alpha = 1$). Yet we now see that this solution can allocate far more utility to an individual whose welfare is easily improved than to one who is less fortunate. For example, it may divert treatment resources from cancer patients to persons suffering from the common cold to provide them the same fraction of their maximum health potential. In general, it favors individuals who happen to enjoy favorable circumstances, perhaps through no merit of their own. This rules out any notion that justice should compensate for the capriciousness of fate. The K–S model offers no means to prevent this kind of outcome by adjusting the trade-off between equity and efficiency, as is possible with alpha fairness.

4.8.2 Hierarchical Distribution

The hierarchical problem may or not be collapsible, and a sufficient condition is given in the following proposition. A collapsible problem is regionally decomposable.

Proposition 37. *The hierarchical problem* (4.11) *with a Kalai–Smorodinsky SWF is collapsible if*

$$\frac{\mathbf{a}^k\mathbf{d}^k}{\mathbf{a}^\mathsf{T}\mathbf{d}}\left(B+\sum_i B_i\right)\geq B_k \tag{4.16}$$

for all k. Furthermore, the problem is regionally decomposable if it is collapsible.

4.8.3 Incentives and Sharing

An improvement in individual i's efficiency increases that individual's utility at the rate $(1/n)(B/a_i^2) = u_i^*/a_i$. The analysis of personal efficiency and technological efficiency is identical to the case of $\alpha = 1$ in alpha fairness SWF. The change of personal efficiency has no effect on the utility of others. This poses what might be viewed as a classical bargaining situation for independent, self-interested individuals. One's efforts to make better use of resources benefits only oneself, with no necessity to share the surplus with others, and one cannot expect any benefits from the efforts of others. On the other hand, one's incentive to improve efficiency is proportional to the square of one's current efficiency. As a result, those who consume more resources to achieve a given welfare level have significantly less incentive to improve their efficiency. From the social planner's perspective, the gradient of technology investment is $(a_1^{-2}, a_2^{-2}, \dots)$.

4.9 Utility Threshold Criterion

Williams and Cookson (2000) propose a pair of 2-person social welfare criteria based on thresholds. One uses a *utility threshold*: it employs a maximin criterion until the utility cost becomes excessive, at which point it begins to switch to a utility criterion. The other uses am *equity threshold*: it employs a utilitarian criterion until inequality becomes excessive, at which point it switches to a maximin criterion. We study the former in this section and the latter in the next.

4.9.1 Socially Optimal Distribution

Williams and Cookson illustrate contours of the 2-person utility-based threshold criterion as in Fig. 4-1. The contours are based on a maximin criterion but switch to a utilitarian criterion when $|u_1 - u_2| > \Delta$. Given the feasible region shown, a maximin solution (small open circle) requires great sacrifice from individual 2. It may therefore be desirable to use a utilitarian solution (solid dot), whose social welfare is slightly greater than that of the maximin solution.

Hooker and Williams (2012) generalize the utility threshold criterion to n individuals, formulate a mixed integer programming model for it, and apply it to a health resources problem. Their SWF is given by

$$W_\Delta(\mathbf{u}) = (n-1)\Delta + \sum_{j=1}^{n} \max\{u_j - \Delta,\ u_{\min}\} \tag{4.17}$$

where $u_{\min} = \min_j\{u_j\}$. They propose a practically meaningful interpretation of the parameter Δ that goes as follows. Utilities within Δ of the lowest utility are regarded as belonging to the *fair region*, and the corresponding individuals receive

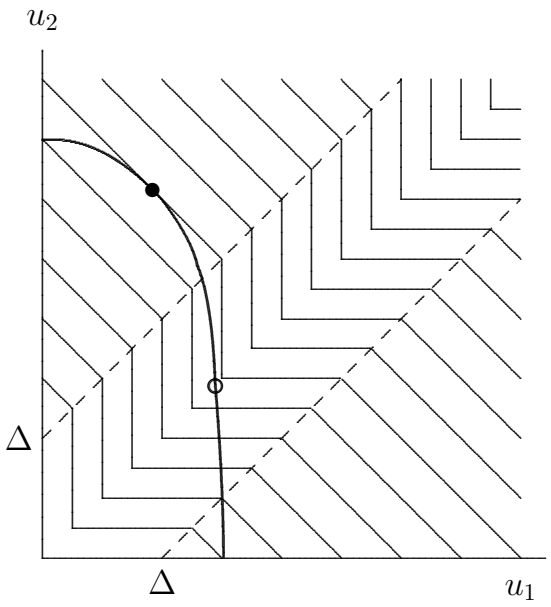


Figure 4-1: Contours for a 2-person utility threshold SWF.

special priority. The remaining individuals belong to the *utilitarian region*. The SWF treats utilities in the fair region as though they were equal to the smallest utility, which therefore receives weight in (4.17) equal to the number of utilities in the fair region. Utilities in the utilitarian region receive unit weight. The function becomes purely utilitarian when $\Delta = 0$ and maximin as $\Delta \to \infty$. The parameter Δ is chosen so as to locate utilities in the fair region when the corresponding individuals should be seen as disadvantaged enough to deserve higher priority. Larger values of Δ therefore place a greater emphasis on fairness as measured by the Rawlsian maximin criterion.

The threshold function $W_\Delta(\mathbf{u})$ escapes an anomaly that, as noted earlier, characterizes alpha fairness. It cannot assign equality the same social value as arbitrarily extreme inequality. In a 2-person context, for example, an egalitarian distribution $\mathbf{u} = (1, 1)$ can have the same social value as a distribution in which one party has no

utility and the other $\Delta + 2$, but the gap can be no greater than this.

When the utility-based threshold SWF is maximized subject to a budget constraint, the solution is either purely utilitarian or purely maximin, depending on the value of Δ. In particular,

Proposition 38. *Maximizing the utility threshold SWF subject to a budget constraint yields a purely utilitarian optimal solution* $\mathbf{u}^* = (B/a_1)\mathbf{e}_1$ *when*

$$\Delta \leq B\left(\frac{1}{a_1} - \frac{n}{a_N}\right) \tag{4.18}$$

and otherwise a purely maximin solution $\boldsymbol{u}^* = (B/a_N)\boldsymbol{e}$.

When there are lower and upper bounds on individual utilities, as often occurs in practice, the solutions are less extreme and perhaps more useful. They have an interesting structure as well.

Proposition 39. *Maximizing a utility threshold SWF subject to a budget constraint and upper and lower utility bounds yields an optimal solution in which at most one utility* u_i *is strictly between* $u_{\min}$ *and* d_i. *Furthermore, if there is such a utility, then some other utility* u_j *that is equal to* $u_{\min}$ *is at its lower or upper bound.*

This result says that nearly all utilities will be at their upper bound or equal to the lowest utility. It is illustrated for two persons in Fig. 4-2, where u_1^* is strictly between the upper bound d_1 and $u_{\min}^*$, but $u_2^* = u_{\min}^*$. This kind of structure can simplify implementation and provide managerial insight. In the healthcare example described earlier, it tells us that nearly all patients (all but those suffering from one

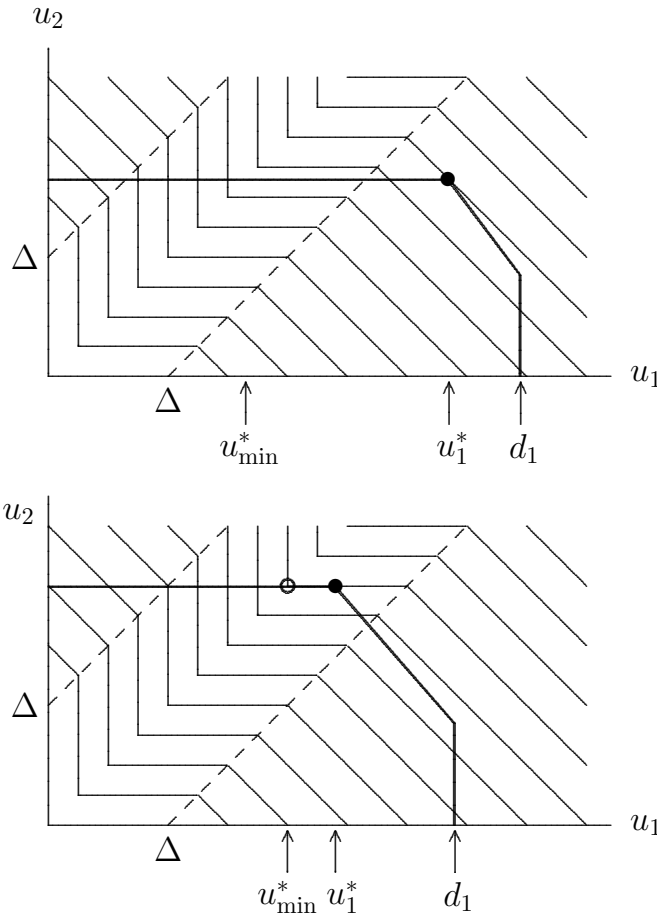


Figure 4-2: Two instances of the utility threshold problem in which u_1^* lies strictly between d_1 and $u_{\min}^*$ in an optimal solution (u_1^*, u_2^*) (black dot). Note that in the second instance, there is another optimal solution in which $u_1^* = u_{\min}^*$ (open circle).

particular disease) will either receive their maximum possible utility or else end up as one of the worst-off patients, who are given the highest priority. We can also specialize Proposition 39 to a hierarchical network with r regions, each of which has subregions.

Proposition 40. *Any optimal solution* $\bar{\mathbf{u}}$ *of* (4.11) *with a utility-based threshold function* $W_\Delta(\mathbf{u})$ *satisfies the following:*

(i) At most r utilities lie strictly between $\bar{u}_{\min}$ and their upper bounds.

(ii) At most one utility in any region lies strictly between $\bar{u}_{\min}$ and its upper bound.

4.9.2 Hierarchical Distribution

The threshold function $W_\Delta(\mathbf{u})$ is not monotonically separable, and so there is no assurance that a given instance of the problem is regionally decomposable. As a simple example, suppose there are two regions, one with subregional utilities u_1, u_2 and the other with a single utility u_3, and let $\Delta = 1$. Then if $\mathbf{a} = (1, 1, 4)$ and $(B, B_1, B_2) = (1, 1, 0)$, Proposition 38 tells us that the solution of the collapsed problem (4.12) is $\mathbf{u} = (2, 0, 0)$. This instance of the problem is, in fact, collapsible by Proposition 29 because $(B_1, B_2) \leq (2, 0)$, and so $\mathbf{u} = (2, 0, 0)$ solves the original problem (4.11). However, the regionally optimal solution for (u_1, u_2) is $(1, 1)$, which is suboptimal in the national problem. This instance of the problem is therefore not regionally decomposable.

4.9.3 Incentives and Sharing

The benefits of improving one's efficiency are distributed as in a purely utilitarian or purely maximin solution, depending on which one obtains for the chosen value of Δ.

4.10 Equity Threshold Criteria

4.10.1 Socially Optimal Solution

Contours of the 2-person equity threshold SWF of Williams and Cookson (2000) are illustrated in Fig. 4-3. They are based on a utilitarian function but switch to a

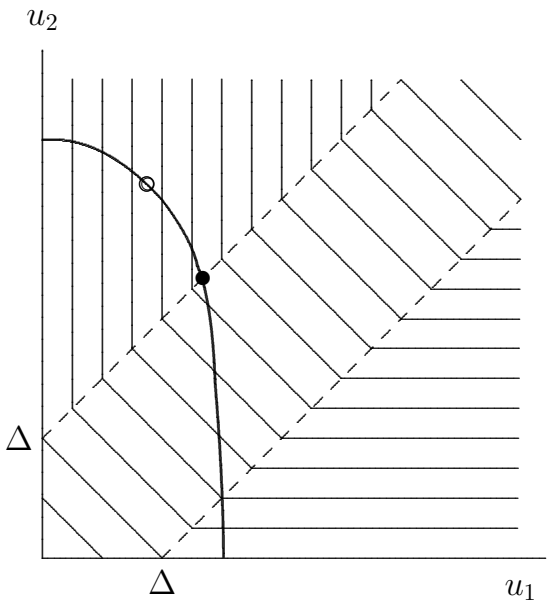


Figure 4-3: Contours for a 2-person equity threshold SWF.

maximin criterion when $|u_1 - u_2| > \Delta$. The SWF is generalized to n persons in Chen and Hooker (2021) as follows:

$$W_\Delta(\mathbf{u}) = n\Delta + \sum_{j=1}^{n} \min\left\{u_j - \Delta, u_{\min}\right\} \tag{4.19}$$

The interpretation of Δ is different than for a utility threshold. Increasing a utility u_j that already exceeds the lowest by more than Δ is viewed as adding nothing to social welfare, unless the lowest is increased an equal amount. Thus larger values of Δ correspond to more nearly utilitarian solutions, rather than smaller values as in the utility threshold problem. In particular, $\Delta = 0$ yields a pure maximin solution, and sufficiently large Δ a pure utilitarian solution.

The analysis of the equity threshold SWF is also quite different because, unlike the utility threshold SWF, it is concave. The welfare maximizing solution also has a

more complex structure. When maximizing welfare subject to a budget constraint, larger values of Δ result in an allocation of utility Δ to a few of the more efficient individuals, while all other individuals (with one possible exception) receive nothing. As Δ increases, fewer individuals benefit, with only one individual receiving utility in the pure utilitarian case. Smaller values of Δ allocate utility to everyone, but at two levels. Less efficient individuals receive utility u_0 and more efficient individuals utility $u_0 + \Delta$, where u_0 depends on the problem data. The precise result is stated in the following theorem, whose proof uses the fact that the optimization problem has a linear programming model:

Proposition 41. *Suppose $a_1 \leq \cdots \leq a_n$, and let k^* be the largest k for which $a_N/a_k > n$. Then for values of Δ with*

$$0 \leq \Delta < B\Big(\sum_{i=1}^{k^*} a_i\Big)^{-1} \tag{4.20}$$

an optimal solution subject to a budget constraint for the equity threshold criterion is given by

$$u_i^* = \begin{cases} u_0 + \Delta, & i = 1, \ldots, k^* \\ u_0, & i = k^* + 1, \ldots, n \end{cases} \tag{4.21}$$

where

$$u_0 = \frac{1}{a_N}\Big(B - \Delta \sum_{i=1}^{k^*} a_i\Big) \tag{4.22}$$

For values of Δ with

$$\Delta \geq B\Big(\sum_{i=1}^{k^*} a_i\Big)^{-1} \tag{4.23}$$

an optimal solution is given by

$$u_i^* = \begin{cases} \Delta, & i = 1, \ldots, k-1 \\ \dfrac{1}{a_k}\Big(B - \Delta \displaystyle\sum_{j=1}^{k-1} a_j\Big), & i = k \\ 0, & i = k+1, \ldots, n \end{cases} \tag{4.24}$$

when $k \leq k^*$ *and* k *satisfies*

$$B\Big(\sum_{i=1}^{k} a_i\Big)^{-1} < \Delta \leq B\Big(\sum_{i=1}^{k-1} a_i\Big)^{-1}$$

4.10.2 Hierarchical Distribution

The equity threshold SWF is not regionally decomposable. The counterexample exhibited in the previous section serves as a counterexample here. In this case, the unique optimal solution of the national problem is $(\bar{u}_1, \bar{u}_2, \bar{u}_3) = (1, 1, 0)$ and $(\bar{y}_1, \bar{y}_2) = (1, 0)$, while the only two optimal solutions of the region 1 subproblem are $(u_1, u_2) = (1.5, 0.5)$ and $(0.5, 1.5)$. Thus an optimal solution of the regional problem cannot be part of an optimal solution of the national problem.

4.10.3 Incentives and Sharing

The effects of efficiency improvements have an interesting structure. For the more egalitarian values of Δ that satisfy (4.20), an improvement in any individual's efficiency increases every utility u_j^* at the rate u_j^*/a_N. Thus an individual's rate of improvement is proportional to that individual's current utility allotment, so that everyone has the same *percentage* rate of increase. Since one's efficiency

improvements benefit oneself as well as others, one has a personal incentive to improve that is proportional to one's current utility allotment.

For the less egalitarian values of Δ that satisfy (4.23), only the individual k who has utility u_k^* strictly between 0 and Δ is affected by efficiency changes. Efficiency changes by the less efficient individuals $i > k$ have no effect at all. Efficiency improvements by the more efficient individuals $i < k$ increase individual k's utility at the rate Δ/a_k. Efficiency improvements obtained personally by individual k increase utility at the lesser rate u_k^*/a_k (recall that $u_k^* < \Delta$). Thus only the transitional individual k is personally incentivized to improve efficiency. Indeed, only individual k benefits from *anyone's* efficiency improvements, and even then only from those of more efficient individuals. However, the benefit from any of their improvements is greater than from individual k's own improvements.

For small values of Δ satisfying 4.20, the gradient of technology investment is $(g, g, \ldots, g, 0, \ldots)$, where the first k^* entries are $g = (nB/a_N^2 - na_{[k^*]}/a_N + \Delta/a_N)$. For larger values of Δ satisfying (4.23), the gradient is $(0, \ldots, 0, g', 0, \ldots)$, where the only non-zero element is at the kth position with $g' = (B - \Delta a_{[k-1]})/a_i^2$.

4.11 A Threshold Criterion with Leximax Fairness

4.11.1 Socially Optimal Distribution

While a utility-based threshold criterion with maximin fairness tends to avoid extreme solutions, at least in the presence of utility bounds, the maximin component continues to ignore all but the lowest utility value in the fair region. This can result in solutions that are insensitive to the plight of disadvantaged individuals other than

the very worst off. Chen and Hooker (2020b) and Chen and Hooker (2020a) avoid this problem by combining utilitarianism with a leximax rather than maximin criterion. An added benefit of this approach is that it avoids extreme solutions (i.e., purely utilitarian or purely leximax) even when there is a single budget constraint with no upper bounds on utilities.

The Chen–Hooker approach sequentially maximizes social welfare functions $W_1, \ldots, W_n$, where W_1 is the Hooker–Williams SWF. The first maximization problem P_1 is (4.3). The remaining maximization problems P_k for $k = 2, \ldots, n$ are

$$\max_{\mathbf{u}} \left\{ W_k(\mathbf{u}_K) \;\middle|\; \begin{array}{l} u_i \geq \bar{u}_{i_{k-1}},\ \max\{c_i,\ \bar{u}_{i_k}\} \leq u_i \leq d_i,\ i \in I_k \\ \displaystyle\sum_{i \in I_k} a_i u_i \leq B_k \end{array} \right\} \tag{4.25}$$

where

$$W_k(\mathbf{u}_K) = (n-k+1)u_{\min} + \sum_{i \in I_k} \big(u_i - \bar{u}_{i_1} - \Delta\big)^+$$

and where

$$u_{\min} = \min_{i \in I_k}\{u_i\}, \quad B_k = B - \sum_{j=1}^{k-1} a_{i_j} \bar{u}_{i_j}$$

Problem P_k is solved over the variable set $\{u_i \mid i \in I_k\}$, where $I_k = \{1, \ldots, n\} \setminus \{i_1, \ldots, i_{k-1}\}$ and $u_{i_1}, \ldots, u_{i_{k-1}}$ are the variables whose values in the socially optimal solution are determined by solving $P_1, \ldots, P_{k-1}$ respectively. Thus the vector $\mathbf{u}_K$ consists of the elements u_i of vector $\mathbf{u}$ for I_k. Solving P_k determines the value of the kth smallest utility in the socially optimal solution $\bar{u}$. That is, P_k sets $\bar{u}_{i_k} = \tilde{u}_{i_k}$, where $i_k = \arg\min_{i \in I_k}\{\tilde{u}_i\}$ and $\tilde{u}_K$ is an optimal solution of P_k. Actually, one need only solve $P_1, \ldots, P_{k^*}$ (rather than $P_1, \ldots, P_n$) where k^* is the smallest k for which $\bar{u}_{i_k} - \bar{u}_{i_1} \geq \Delta$. At this point, all the remaining utilities (i.e., all u_i for $i \in I_{k^*}$) are

in the utilitarian region, and one can set $\bar{u}_i = \tilde{u}_i$ for $i \in I_{k^*}$, where $\tilde{u}_i$ is the optimal value of u_i in the solution of P_{k^*}. Chen and Hooker state mixed integer programming models of $P_1, \ldots, P_n$ that can be readily solved in practice.

We showed earlier (Proposition 38) that when the social welfare problem is solved subject only to a budget constraint, the solution of P_1 is purely utilitarian or purely maximin. The socially optimal solution $\bar{\mathbf{u}}$ that results from solving the sequence of problems $P_1, \ldots, P_n$ is, in general, neither utilitarian nor leximax and has an interesting structure.

Proposition 42. *Suppose $a_1 \geq \cdots \geq a_n$. Let $m = 1$ if (4.18) is violated, and otherwise let m be the smallest index $k \geq 2$ for which the following (4.26) is violated:*

$$\frac{1}{n-k} B \left(2\frac{n-k+1}{a_K} - \frac{1}{a_{\ell_k}} \right) \leq \Delta \leq B \left(\frac{1}{a_{\ell_k}} - \frac{n-k+1}{a_K} \right) \tag{4.26}$$

where $a_{\ell_k} = \min\{a_\ell \mid \ell = k, \ldots, n\}$ and $a_K = \sum_{i=k}^n a_i$. We regard (4.26) as violated when $n = k$. Then a socially optimal solution $\bar{\mathbf{u}}$ of the leximax threshold problem subject to a budget constraint is given by the following:

$$\bar{u}_i = \begin{cases} 0, & \text{for } i = 1, \ldots, m-1 \\ B/a_M, & \text{for } i = m, \ldots, n \end{cases} \tag{4.27}$$

where $a_M = \sum_{i=m}^n a_i$.

Thus, several of the more efficient individuals receive equal shares of utility, with the remaining individuals receiving nothing. Utility can be spread more broadly by increasing Δ, even to a point where everyone receives an equal share. A threshold criterion that combines utilitarian and leximax elements can therefore yield a variety

of solutions, even when there is only a single budget constraint. The variety is still greater when there are lower and/or upper bounds on utilities.

4.11.2 Incentives and Sharing

When individual i improves efficiency, individual j's utility increases at a rate Ba_i/a_M if $i, j \geq m$, with no effect if $i < m$ or $j < m$. Thus an individual who receives utility in the socially optimal solution is incentivized to improve efficiency, and all others with positive utility benefit as well. This means there is sharing rather than competition, although this effect exists at the margin only among those who already benefit from the optimal distribution.

We can also look beyond marginal analysis to note that an improvement in any individual's efficiency (with the possible exception of the most efficient individual) causes the lower bound in (4.26) to increase and the upper bound to decrease. This a sufficiently large improvement can result in a larger m and a greater number of individuals who benefit.

4.12 Conclusion

One might construct a narrative from the foregoing observations as follows. The inadequacy of popular optimization objectives becomes evident when they are applied to a generic constraint set consisting of a budget limitation and perhaps bounds on the utilities of each party concerned. A utilitarian objective is by far the most widely used but leads to results that almost any observer would find unacceptable. It allocates all utility to a single party, an outcome that is only marginally ameliorated by placing bounds on individual utilities. While this extreme result is not evident in

most practical optimization models, due to the complexity of the constraint set, this complexity only serves to conceal the basic unreasonableness of a purely utilitarian criterion.

Objectives based solely on equity can yield equally extreme and unacceptable solutions. Perhaps the most famous fairness criterion, the maximin objective that derives from the Rawlsian difference principle, forces all parties to accept the same level of utility, except, again, where this is blocked by other constraints—constraints that may reflect only the situation and no coherent understanding of what is just. For example, if one person is difficult to accommodate, due to unfortunate circumstances such as incurable disease, it is necessary to lavish resources on that person to the point that all others are reduced to the same level of suffering. Even if we prevent this outcome by placing a low upper bound on the disadvantaged party's utility, the maximin objective allows us to allot others that same low level of utility, when they could receive much more. A leximax objective largely removes this second anomaly, but the extreme solutions remain.

A natural strategy for avoiding extreme solutions is to combine equity and utilitarian considerations in some fashion. A simple convex combination is both unprincipled and difficult to calibrate, particularly since equity and efficiency tend to be measured in different units. Alpha fairness, perhaps the best known composite criterion, allows a parameterized balancing of fairness and efficiency that avoids the extreme solutions just descibed. Yet it creates an extreme result of its own, because it can regard an egalitarian distribution as no better than one in which there is extreme inequality. This does not become evident in optimal solutions subject to simple budgetary and bounding constraints, but it can emerge in nonconvex constraint sets. It is also unclear how to select and interpret the balancing parameter. The Kalai–Smorodinsky

bargaining solution avoids the extreme outcomes of alpha fairness, but at the cost of another extreme that is opposite to that of the maximin criterion. It awards wealthy and privileged individuals the same fraction of their potential utility as individuals who have far less potential, perhaps due to some physical or mental impairment. There is also no parameter for regulating the equity/efficiency balance.

Threshold functions provide an alternate means for combining equity and efficiency, where the balance is governed a parameter Δ that is more easily interpreted in practice. A utility threshold function employs a maximin criterion but switches to a utilitarian criterion when utility cost fairness crosses a specified threshold, while an equity threshold function does the reverse. Yet threshold criteria can lead to extreme solutions, at least in the presence of the simplest constraints. The utility threshold function, for example, yields a purely utilitarian or purely maximin solution in the presence of a single budget constraint without utility bounds, although one can state in closed form which values of Δ produce one or the other. The addition of utility bounds results in much more reasonable solutions, unlike the situation with a simple utilitarian objective. In addition, the resulting solutions have an interesting structure that can ease implementation and provide managerial insights.

Nonetheless, threshold functions that combine utilitarian with maximin objectives inherit a shortcoming of the latter, if only in attenuated form. This is the tendency to give insufficient attention to disadvantaged parties other than the very worst off. The problem can again be addressed by replacing a maximin with a leximax criterion, in this case by optimizing a certain sequence of threshold functions rather than a single one. The resulting solutions avoid all the extremes described here, even in the presence of a simple budget constraint without utility bounds. While the individual threshold functions share the structural properties mentioned earlier, the

socially optimal solutions that result are more complex in nature. A threshold-based combination of utilitarian and leximax criteria doubtless has shortcomings of its own, but it illustrates the thesis that we must move beyond the naïveté of simpler social welfare functions if we are to avoid unacceptable results.

This narrative is enriched by observing the behavior of the various criteria on hierarchical networks in which resources can be passed from a national to a regional level. Each region combines its own resources with those received from above and distributes them to subregions (which can be interpreted as hospitals or other institutions). All of the extremes discussed above persist in this context. However, the simpler models are more likely to be regionally decomposable, due to a technical property (monotone separability) of the corresponding social welfare functions. Regional decomposability means that if each region distributes its allotted resources using the same social welfare criterion as used at the national level, it will obtain an allocation to subregions that is consistent with that prescribed at the national level. The allocation computed by the national authority assumes, in fact, that the allocation within regions will follow this pattern. When there is no regional decomposability, the national solution is valid only if the regions follow the national prescription for intra-regional distribution rather than computing the distribution themselves.

To be specific, the pure utilitarian and maximin criteria are regionally decomposable, as is alpha fairness. The Kalai–Smorodinsky model is regionally decomposable when it is collapsible, meaning that the multilevel problem can be solved as a single-level problem. A sufficient condition for collapsibility can be checked by applying a simple test. The utility threshold model can be regionally nondecomposable even when it is collapsible. Thus the more sophisticated models are progressively less prone to be

decomposable. This might be regarded as an inconvenience, but it can also signal greater adequacy and subtlety as an equity measure. Perhaps local decisions should reflect a larger perspective if they are to be truly fair.

Conclusion

In this dissertation, we study a variety of topics in stochastic programming, decomposition-based methods, robust optimization, and fairness in resource allocation. In the first two chapters, we explore the use of logic-based Benders decomposition for solving a variety of scheduling problems. In Chapter 3, we study a robust portfolio optimization model that attempts to address the uncertainty in portfolio returns. In the last chapter, we focus on the use of social welfare functions for fair allocation of resources. In this section, we conclude by summarizing our contributions.

In Chapter 1, our goal is to devise an exact algorithm to solve two-stage stochastic programs with integer recourse. We propose utilizing the logic-based Benders decomposition framework to accomplish this goal. To this end, we focus on a stochastic version of the planning and scheduling problem. The classical Benders decomposition algorithm cannot be used for this problem, because the subproblem is a difficult cumulative scheduling problem that cannot be modeled as a linear problem. We summarize our contribution in this chapter as follows:

- We derive novel Benders cuts for the planning and scheduling problem with

different release times.

- We devise a branch-and-check method that can solve the planning and scheduling problem exactly.

- We benchmark our method against a mixed-integer programming formulation and the integer L-shaped method.

- Our computational study shows that our branch-and-check method can be faster by several orders of magnitude, allowing significantly larger instances to be solved.

In Chapter 2, we extend our analysis to a class of sequence-dependent parallel machine scheduling problems. For further generality, we assume that there are strict time windows associated with each task. A wide range of problems can be formulated as sequence-dependent scheduling problems including the canonical vehicle routing problem. However, sequence-dependent scheduling problems are notoriously very difficult to solve precisely due to the sequence-dependent nature of the setup times. We summarize our contribution in this chapter as follows:

- We extend our analysis in Chapter 1 by introducing new Benders cuts and improving some of the cuts proposed for the planning and scheduling problem. We show that these cuts are tight and yield a computationally more efficient branch-and-check method.

- We introduce novel Benders cuts for the sequence-dependent parallel machine scheduling problem. These cuts generalize some of the other well-known Benders cuts in the literature.

- We devise a branch-check method that solves the sequence-dependent parallel machine scheduling problem exactly.

- Our computational study shows the effectiveness of the proposed solution method.

In Chapter 3, we study the classical Markowitz model for portfolio optimization. In the classical model, it is assumed that the expected portfolio returns are known exactly. In practice, however, the portfolio returns are uncertain. To this end, robust optimization is one of the techniques that addresses this uncertainty. We summarize our contribution in this chapter as follows:

- We focus on ellipsoidal uncertainty sets around a point estimate of the expected asset returns and the choice of the error-covariance matrix that specifies this ellipsoid.

- We show that the class of diagonal estimation-error matrices can achieve an arbitrarily small loss in the expected portfolio return as compared to the optimum portfolio return.

- We propose a bilevel program that finds the best estimation error matrix. The bilevel model also allows us to numerically analyze the error when there are multiple estimates for the expected return and/or when there are additional restrictions on the structure of the estimation error matrix.

- We perform simulations to test the use of an identity matrix as the estimation-error matrix. Our simulations show that the robust portfolio models featuring an identity matrix as an estimation-error matrix outperform the classical Markowitz model when the size of the uncertainty set is chosen properly.

In the last chapter, we focus on fair allocation of resources. Optimization offers a powerful tool for identifying an efficient and equitable allocation of resources. This is usually accomplished by using a social welfare function as an objective function. To this end, we investigate a plethora of social welfare functions. We summarize our contribution in this chapter as follows:

- We focus primarily on social welfare functions that combine efficiency and equity criteria.
- We derive the structural properties of optimal solutions that result when these social welfare functions are maximized subject to simple but generic constraints.
- We examine the structure of solutions in a hierarchical distribution network that represents a typical situation in which a national authority allocates resources to regions, which in turn combine these with their own resources for distribution to their subregions or institutions.
- We discuss the implications of selecting a social welfare function with respect to the incentives it creates for both the players and the social planner.
- We show that several well known social welfare functions can result in extreme and often unacceptable solutions.

Appendix A

Appendix

A.1 Chapter 1

In this appendix, we present the results of additional computational experiments concerning the details of our implementation of the algorithms used in Chapter 1.

A.1.1 CP Parameters

In this section, we test the impact of using different CP parameters on the overal solution time of the LBBD algorithm for solving the minimum makespan problem. We use the same instances used in Tables $1.1 - 1.4$. The results are presented in Table A.1.

We see from Table A.1 that using different CP parameters does not change the overall picture of the performance of the two solution methods. Extended inference level and DFS search look to be good choices for the CP solver, therefore, we use these

Table A.1: Average computation time in seconds over 3 instances for different CP parameter values, based on 10 tasks and 2 facilities.

		INT-L				LBBD			
	Inference level Search type	Default Default	Extended Default	Default DFS	Extended DFS	Default Default	Extended Default	Default DFS	Extended DFS
$\|S\|$	1	2.25	2.26	1.77	2.84	1.44	1.58	1.22	0.91
	5	11.85	13.40	8.43	9.00	3.73	4.33	3.17	2.12
	10	19.75	23.92	15.92	16.00	4.65	5.44	5.54	3.09
	50	116.16	116.97	92.09	87.19	23.92	26.46	23.16	17.33
	100	247.71	254.05	192.95	209.51	50.78	55.73	45.29	36.72
	500	1430.95	1322.95	1151.25	1166.55	363.55	450.47	391.01	279.07

setting in all the experiments we perform in this paper.

A.1.2 Lower Bound for the Integer L-shaped Method

In this section, we perform experiments to see the impact of using better lower bounds on the performance of the integer L-shaped method. We tested two different sets of lower bounds as shown in Table A.2. The results in the "integer bounding" column correspond to the global lower bound obtained by solving (1.25) for fixed scenario ω without relaxing the integrality constraints. The results in the "relaxed bounding" column correspond to the results where we us the bounds obtained by solving the LP relaxation of (1.25). We again use the same makespan instances used in Tables $1.1 - 1.4$.

We see from Table A.2 that better lower bounds from integer programming yield modest improvements in the average solution times (indicated in columns labeled "INT-L time"). Yet this improvement is substantially offset by the much longer time required to compute the integer programming bound. We therefore opted to use relaxed bounding in all computational experiments.

Table A.2: Average computation time in seconds over 3 instances for two different lower bounds, based on 10 tasks and 2 facilities.

	integer bounding			relaxed bounding		
$\|S\|$	Preprocessing time	INT-L time	total	Preprocessing time	INT-L time	total
1	2.2	0.9	3.0	0.2	1.9	2.1
5	9.0	8.2	17.2	0.6	8.1	8.7
10	14.0	12.0	26.0	1.1	15.0	16.1
50	64.1	80.2	144.3	5.3	88.4	93.6
100	139.7	186.2	325.9	12.0	212.3	224.2
500	707.4	1227.3	1934.6	53.3	1473.7	1527.0

A.1.3 Accessing Problem Instances

The readers can access all the problem instances used in our computational experiments via `https://github.com/ozgunelci/Stochastic-Scheduling-With-LBBD`.

A.2 Chapter 3

A.2.1 Data Set

We use the data set provided in Kocuk and Cornuéjols (2020). The data set includes 360 monthly returns of 11 sectors based on the Global Industrial Classification Standard. We calculate the true return vector $\boldsymbol{\mu}$ and the true covariance matrix $\boldsymbol{\Sigma}$ as the sample average and the sample covariance matrix of these returns. We refer the reader to Kocuk and Cornuéjols (2020) for more details about the dataset.

A.2.2 Additional Proofs

In this section, we present the additional proofs.

Proof of Proposition 23

1. We know that there exists a matrix $\mathbf{L}$ such that $\boldsymbol{\Xi} = \mathbf{L}\mathbf{L}^\mathsf{T}$, because $\boldsymbol{\Xi}$ is a positive definite matrix. Therefore, the objective function f of (3.5) can be written as

$$\begin{aligned} f(\mathbf{x}) &= -(\hat{\boldsymbol{\mu}})^\mathsf{T}\mathbf{x} + \sqrt{\mathbf{x}^\mathsf{T}\boldsymbol{\Xi}\mathbf{x}} \\ &= -(\hat{\boldsymbol{\mu}})^\mathsf{T}\mathbf{x} + \|\mathbf{x}\mathbf{L}\|_2 \end{aligned} \tag{A.1}$$

   To conclude the proof, it suffices to show that the norm function $\|\cdot\|_2$ is strictly convex on the feasible region of (3.5), because the composition with an affine mapping of a strictly convex function is strictly convex.

   To this end, we first note that any two distinct point in the feasible region of (3.5) are linearly independent because of $\mathbf{1}^\mathsf{T}\mathbf{x} = 1$. Let $\lambda \in (0, 1)$. We want to

show that

$$\|\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2\|_2 < \lambda\|\mathbf{x}_1\|_2 + (1-\lambda)\|\mathbf{x}_2\|_2$$

We take the square of both sides and observe that

$$\begin{aligned}
\|\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2\|_2^2 &= \big(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2,\ \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2\big) \\
&= \lambda^2\big(\mathbf{x}_1, \mathbf{x}_1\big) + (1-\lambda)^2\big(\mathbf{x}_2, \mathbf{x}_2\big) + 2\lambda(1-\lambda)\big(\mathbf{x}_1, \mathbf{x}_2\big) \\
&= \lambda^2\|\mathbf{x}_1\|_2^2 + (1-\lambda)^2\|\mathbf{x}_2\|_2^2 + 2\lambda(1-\lambda)\big(\mathbf{x}_1, \mathbf{x}_2\big) \\
&< \lambda^2\|\mathbf{x}_1\|_2^2 + (1-\lambda)^2\|\mathbf{x}_2\|_2^2 + 2\lambda(1-\lambda)\|\mathbf{x}_1\|_2\|\mathbf{x}_2\|_2 \\
&= \left(\lambda\|\mathbf{x}_1\|_2 + (1-\lambda)\|\mathbf{x}_2\|_2\right)^2
\end{aligned}$$

The strict inequality above is true due to the fact that Cauchy-Schwarz inequality is strict when $\mathbf{x}_1$ and $\mathbf{x}_2$ are linearly independent.

2. It is clear that the robust portfolio optimization problem (3.5) is a convex program. It suffices to exhibit an interior point of the feasible region. We assume that v is strictly greater than the minimum variance portfolio. Let $\mathbf{x}^{\text{mv}}$ and $\mathbf{x}^{\text{eq}}$ denote the minimum variance and the equal-weight portfolios, respectively. There exists $\delta \in (0,1)$ such that the variance of $\tilde{\mathbf{x}} = \delta\mathbf{x}^{\text{mv}} + (1-\delta)\mathbf{x}^{\text{eq}}$ is less than v. Clearly, all components of $\tilde{\mathbf{x}}$ is greater than zero. Thus, $\tilde{\mathbf{x}}$ is an exterior point of the feasible region of (3.5).

3. The square-root function is non-differentiable only at the origin. We have that the objective function (A.1) is always differentiable in the feasible region of (3.5), because $\boldsymbol{\Xi}$ is positive definite, and $\mathbf{1}^{\mathsf{T}}\mathbf{x} = 1$.

Proof of Proposition 24

We want to show the equivalence of the system (3.6) and the system $\{(3.6b) - (3.6g), (3.9)\}$. We only need to show given $(\boldsymbol{\Xi}, \mathbf{x}, \lambda^1, \lambda^2, \boldsymbol{\lambda}^3)$ that satisfies (3.6) with $\boldsymbol{\Xi}$ is positive definite and diagonal, there exists $(\mathbf{z}, \alpha, \boldsymbol{\xi}, \mathbf{x}, \lambda^1, \lambda^2, \boldsymbol{\lambda}^3)$ that satisfies $\{(3.6b) - (3.6g), (3.9)\}$. The other direction is trivial since the $\boldsymbol{\xi}$ vector in any feasible solution to $\{(3.6b) - (3.6g), (3.9)\}$ is a diagonal estimation-error matrix that satisfies the optimality conditions.

Let $\boldsymbol{\xi}$ denote the diagonal entries of $\boldsymbol{\Xi}$. Because $\boldsymbol{\Xi}$ is positive definite, we know that all elements of $\boldsymbol{\xi}$ are strictly greater than zero. It suffices to find $(\mathbf{z}, \alpha)$ that satisfies (3.9), since $(\mathbf{x}, \lambda^1, \lambda^2, \boldsymbol{\lambda}^3)$ automatically satisfies $(3.6b) - (3.6g)$.

Consider any given $\mathbf{x} >= 0$. Let $P \coloneqq \{i \in [1, n] : x_i > 0\}$. Let $Z = [1, n] \setminus P$. Since $\mathbf{1}^\mathsf{T}\mathbf{x} = 1$, we know that P is non-empty. This implies that the right hand side of (3.9c) is positive for all $i \in P$. Therefore, α must be positive.

Now note that we can write (3.9b) as

$$\begin{aligned}
\alpha &= \sum_{i \in P} x_i z_i + \sum_{i \in Z} x_i z_i \\
&= \sum_{i \in P} x_i z_i \\
&= \sum_{i \in P} x_i \frac{\xi_i x_i}{\alpha}
\end{aligned}$$

Thus we have that $\alpha = \sqrt{\sum_{i \in P} x_i \xi_i x_i} > 0$. Accordingly, each z_i assumes the value $\frac{\xi_i x_i}{\alpha}$ for all $i \in P$. Letting each $z_i = 0$ for all $i \in Z$ concludes the proof.

A.2.3 The Importance of the Choice of Horizon Length in Dynamic Analysis of the Portfolio Models

Rolling-horizon-based experiments allow decision makers to perform dynamic analyses of their portfolio optimization models. Such experiments are typically conducted by following a historical sample path of assets prices. In this section, we present an important observation on the performance of the Markowitz model when analyzed through a historical path.

We use the same data described in Appendix A.2.1 and perform a rolling horizon analysis. In particular, for a given horizon length (n), the portfolio weights are determined each month using the expected return estimated from a rolling window of length n, and these portfolio weights are then used the calculate the returns of the month $n + 1$.

horizon length (n)	the average return of the Markowitz model
1	0.01071
3	0.01083
6	0.01116
12	0.01338
24	0.01507
36	0.01417
60	0.01366
90	0.01396
120	0.01117
180	0.00948

Table A.3: Impact of horizon length in a dynamic analysis.

The results presented in Table A.3 are very interesting. We see that the horizon length has a very significant impact on the average actual return. In particular,

the horizon length of 24 significantly outperforms the other choices. These results show the elaborate dynamics of performing rolling-horizon-based experiments and the importance of the choice for horizon length when the returns are correlated over a time horizon.

A.3 Chapter 4

A.3.1 Proofs of Results

Proof of Proposition 29. Let $\bar{\mathbf{u}}$ be an optimal solution of (4.12) that satisfies $\mathbf{a}^k\bar{\mathbf{u}}^k \geq B_k$ for $k = 1, \ldots, r$. If we let $\bar{y}_k = \mathbf{a}^k\bar{\mathbf{u}}^k - B_k \geq 0$, then $(\bar{\mathbf{y}}, \bar{\mathbf{u}})$ is feasible in (4.11) because

$$\sum_{k=1}^{r} \bar{y}_k = \sum_{k=1}^{r} \mathbf{a}^k\bar{\mathbf{u}}^k - B_k \leq B \tag{A.2}$$

where the inequality is due to the constraint in (4.12). Also any $\mathbf{u}$ feasible in (4.11) is feasible in (4.12), which implies $F(\mathbf{u}) \leq F(\bar{\mathbf{u}})$ since $\bar{\mathbf{u}}$ is optimal in (4.12). Thus $(\bar{\mathbf{y}}, \bar{\mathbf{u}})$ is optimal in (4.11). □

Proof of Proposition 30. Let $(\bar{\mathbf{y}}, \bar{\mathbf{u}})$ be optimal in (4.11) and $\hat{\mathbf{u}}^k$ optimal in (4.13) for $k = 1, \ldots, r$. We wish to show that $(\hat{\mathbf{u}}^1, \ldots, \hat{\mathbf{u}}^r, \bar{\mathbf{y}})$ is optimal in (4.11). We first note that $(\hat{\mathbf{u}}^1, \ldots, \hat{\mathbf{u}}^r, \bar{\mathbf{y}})$ is feasible in (4.11), by hypothesis. To show that $(\hat{\mathbf{u}}^1, \ldots, \hat{\mathbf{u}}^r, \bar{\mathbf{y}})$ is optimal, it suffices to show

$$W(\hat{\mathbf{u}}^1, \ldots, \hat{\mathbf{u}}^r) \geq W(\mathbf{u}^1, \ldots, \mathbf{u}^r) \tag{A.3}$$

for any $(\mathbf{u}^1, \ldots, \mathbf{u}^r, \mathbf{y})$ that is feasible in (4.11). But if $(\mathbf{u}^1, \ldots, \mathbf{u}^r, \mathbf{y})$ is feasible in (4.11), then $\mathbf{u}^k$ is feasible in (4.13) with $\bar{y}_k = y_k$ for each k. Since $\hat{\mathbf{u}}^k$ is optimal in (4.13), we have $W(\hat{\mathbf{u}}^k) \geq W(\mathbf{u}^k)$ for each k. Now (A.3) follows from the fact that $W(\mathbf{u})$ is monotonically separable. □

Proof of Proposition 31. Since (4.3) is a linear programming problem, it suffices to exhibit a feasible solution of the dual problem whose objective function value is

equal to the optimal value $\sum_j u_j^*$ of the primal. The dual problem is

$$\min\left\{Bv_0 + \sum_{j\in N} d_j v_j \;\middle|\; v_0 \geq 0;\ a_j v_0 + v_j \geq 1,\ v_j \geq 0 \ \text{ for } j \in N\right\}$$

If $u_n^* = d_n$, the desired dual solution is $v_0 = 0$, $v_j = 1$ for $j \in N$. Its objective value is the same as the optimal value $\sum_j d_j$ of (4.3). Otherwise the desired dual solution is

$$v_j = \begin{cases} B/a_k & \text{for } j = 0 \\ 1 - a_j/a_k & \text{for } j = 1, \ldots, k-1 \\ 0 & \text{for } j = k, \ldots, n \end{cases}$$

where $k = \min\{j \in N \mid u_j^* < d_j\}$. It is easily verified that this solution is dual feasible and has same objective function value as the optimal solution of (4.3), namely $B/a_k + \sum_{j=1}^{k-1}(1 - a_j/a_k)d_j$. □

Proof of Proposition 32. Let $d_k = d_{\min}$. If the given solution $\mathbf{u}^*$ is not maximin, there must be a feasible solution $\bar{\mathbf{u}}$ such that $\bar{u}_j > u_j^*$ for all j. In particular, since $\bar{u}_k > u_k^* = \min\{d_k, B/a_N\}$ and $\bar{u}_k \leq d_k$, we must have $B/a_N < d_k$. Now since each $a_j > 0$,

$$\sum_j a_j \bar{u}_j > \sum_j a_j u_j^* = a_N \min\left\{d_k, B/a_N\right\} = a_N B/a_N = B$$

which implies that $\bar{\mathbf{u}}$ is infeasible. □

Proof of Proposition 33. We first show that the given solution $\mathbf{u}^*$ consumes all available resources B. The resource consumption is

$$\sum_{j=1}^{n} a_j u_j^* = \sum_{j=1}^{k} a_j d_j + \sum_{j=k+1}^{n} a_j \phi_{k+1} \tag{A.4}$$

Substituting the definition of ϕ_{k+1} and observing that

$$\sum_{j=k+1}^{n} a_j = a_N - \sum_{j=1}^{k} a_j$$

we find that (A.4) simplifies to B. We also note that by definition of k, we have $u_j^* = \phi_j < d_j$ for $j = k+1, \ldots, n$. Thus all and only the k smallest utilities in $\mathbf{u}^*$ are set to their upper bounds. We now suppose, contrary to the claim, that $\mathbf{u}^*$ is dominated by some feasible solution $\mathbf{u}'$. Since the k smallest utilities in $\mathbf{u}^*$ are set to their upper bounds, the k smallest utilities in $\mathbf{u}'$ cannot exceed the k smallest utilities in $\mathbf{u}^*$, respectively, and must therefore be equal to these utilities. Thus if $\mathbf{u}'$ is to dominate $\mathbf{u}^*$, some utility among the $n-k$ largest in $\mathbf{u}'$ must exceed the common value ϕ_{k+1} of $u_{k+1}^*, \ldots, u_n^*$. But this forces another of the $n-k$ largest utilities in $\mathbf{u}'$ to be smaller than ϕ_{k+1}, because the utilities $\mathbf{u}^*$ consume all available resources. This is inconsistent with assumption that $\mathbf{u}'$ dominates $\mathbf{u}^*$. □

To prove Proposition 34, we establish a property for lexicographic comparisons that is analogous to monotone separability. We write $\bar{\mathbf{u}} \succcurlyeq \mathbf{u}$ when $\bar{u}_{\langle i \rangle} \geq u_{\langle i \rangle}$ for $i = 1, \ldots, n$, where $(\bar{u}_{\langle 1 \rangle}, \ldots, \bar{u}_{\langle n \rangle})$ is $\bar{\mathbf{u}}$ arranged in nondecreasing order, and similarly for $(u_{\langle 1 \rangle}, \ldots, u_{\langle n \rangle})$.

Lemma 43. *If* $\bar{\mathbf{u}} \succcurlyeq \mathbf{u}$ *and* $\bar{\mathbf{v}} \succcurlyeq \mathbf{v}$*, then* $(\bar{\mathbf{u}}, \bar{\mathbf{v}}) \succcurlyeq (\mathbf{u}, \mathbf{v})$.

Proof. Let $(\bar{w}_1, \ldots, \bar{w}_n)$ consist of the components of $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ in nondecreasing order, and similarly for $(w_1, \ldots, w_n)$. It suffices to show that $\bar{w}_k \geq w_k$ for $k = 1, \ldots, n$. We first note that there is a perfect matching between $\bar{w}_1, \ldots, \bar{w}_n$ and $w_1, \ldots, w_n$ in which each $\bar{u}_{\langle i \rangle}$ is matched with $u_{\langle i \rangle}$ and each $\bar{v}_{\langle i \rangle}$ with $v_{\langle i \rangle}$. For any given k, $\bar{w}_k$ must

be matched with w_ℓ either for $\ell \geq k$ or for $\ell < k$. If $\ell \geq k$, then $\bar{w}_k \geq w_\ell \geq w_k$, so that $\bar{w}_k \geq w_k$. If $\ell < k$, then $\bar{w}_i$ for some $i < k$ must be matched with w_j for some $j \geq k$. Now we have $\bar{w}_k \geq \bar{w}_i \geq w_j \geq w_k$, which again implies $\bar{w}_k \geq w_k$. □

Proof of Proposition 34. Let $(\bar{\mathbf{y}}, \bar{\mathbf{u}})$ be a leximax solution of (4.11), which means that $\bar{\mathbf{u}} \succcurlyeq \mathbf{u}$ for all $(\mathbf{u}, \mathbf{y})$ that are feasible in (4.11). Let $\hat{\mathbf{u}}^k$ a leximax solution of (4.13) for each k, so that

$$\hat{\mathbf{u}}^k \succcurlyeq \mathbf{u}^k, \ k = 1, \ldots, r \tag{A.5}$$

for all $\mathbf{u}^k$ feasible in (4.13). We note that by construction, $(\hat{\mathbf{u}}^1, \ldots, \hat{\mathbf{u}}^r, \bar{\mathbf{y}})$ is feasible in (4.11). We wish to show that

$$(\hat{\mathbf{u}}^1, \ldots, \hat{\mathbf{u}^r}) \succcurlyeq (\mathbf{u}^1, \ldots, \mathbf{u}^r) \tag{A.6}$$

for any $(\mathbf{u}^1, \ldots, \mathbf{u}^r, \mathbf{y})$ feasible in (4.11). But for any such solution of (4.11), $\mathbf{u}^k$ is feasible in (4.13) with $\bar{y}_k = y_k$. Thus (A.5) holds, which implies (A.6) by Lemma 43. □

Proof of Proposition 36. The K–S problem is

$$\max_{\beta, \mathbf{u}} \left\{ \beta \;\middle|\; \mathbf{u} = \beta \mathbf{u}^{\max}, \ \mathbf{a}^{\mathsf{T}} \mathbf{u} \leq B, \ 0 \leq \mathbf{u} \leq \mathbf{d}, \ 0 \leq \beta \leq 1 \right\}$$

Substituting $\beta \mathbf{u}^{\max} = \beta \mathbf{d}$ for $\mathbf{u}$, this becomes

$$\max_{\beta} \left\{ \beta \mid \beta \mathbf{a}^{\mathsf{T}} \mathbf{d} \leq B, \ 0 \leq \beta \mathbf{d} \leq \mathbf{d}, \ 0 \leq \beta \leq 1 \right\}$$

The constraint $0 \leq \beta \mathbf{d} \leq \mathbf{d}$ is redundant, and $B / \mathbf{a}^{\mathsf{T}} \mathbf{d} \leq 1$ since we are given

that $\mathbf{a}^\mathsf{T}\mathbf{d} \geq B$. The optimal solution is therefore $\beta^* = B/\mathbf{a}^\mathsf{T}\mathbf{d} \leq 1$, and $\mathbf{u}^* = \beta^*\mathbf{d} = B\mathbf{d}/\mathbf{a}^\mathsf{T}\mathbf{d}$. If there are no upper bounds, we can set $d_i = B/a_i$ for each i, so that $\mathbf{a}^\mathsf{T}\mathbf{d} = nB$. Thus we have the optimal solution $u_j^* = (1/n)B/a_j$ for all j. $\square$

Proof of Proposition 37. The collapsed problem (4.12) is

$$\max_{\beta,\mathbf{u}} \left\{ \beta \;\middle|\; \mathbf{u} = \beta\mathbf{d},\ \mathbf{a}^\mathsf{T}\mathbf{u} \leq B + \sum_i B_i,\ 0 \leq \beta \leq 1 \right\}$$

By Proposition 36, the solution of this problem is

$$\bar{\mathbf{u}} = \Big(B + \sum_i B_i\Big)\mathbf{d}/\mathbf{a}^\mathsf{T}\mathbf{d}, \quad \text{or} \quad \bar{\beta} = \big(1/\mathbf{a}^\mathsf{T}\mathbf{d}\big)\Big(B + \sum_i B_i\Big) \tag{A.7}$$

where the latter expression is due to $\bar{\mathbf{u}} = \bar{\beta}\mathbf{d}$. By Proposition 29, this solves the original problem (4.11) if $\mathbf{a}^k\bar{\mathbf{u}}^k \geq B_k$ for all k. Substituting the value of $\bar{\mathbf{u}}$, we obtain (4.16).

To show that the hierarchical problem is regionally decomposable, we note that region k's problem (4.13) is

$$\max_{\beta_k} \big\{ \beta_k \;\big|\; \mathbf{a}^k\mathbf{d}^k\beta_k \leq \mathbf{a}^k\bar{\mathbf{u}}^k,\ 0 \leq \beta_k \leq 1 \big\}$$

where $\mathbf{u}^k = \beta_k\mathbf{d}^k$. Substituting the value of $\bar{\mathbf{u}}^k$ in (A.7), this becomes

$$\max_{\beta_k} \left\{ \beta_k \;\middle|\; \mathbf{a}^k\mathbf{d}^k\beta_k \leq \frac{\mathbf{a}^k\mathbf{d}^k}{\mathbf{a}^\mathsf{T}\mathbf{d}}\Big(B + \sum_i B_i\Big),\ 0 \leq \beta_k \leq 1 \right\}$$

By Proposition 36, the solution of this problem is

$$\hat{\mathbf{u}}^k = \hat{\beta}_k \mathbf{d} = \Big(B + \sum_i B_i\Big)\mathbf{d}/\mathbf{a}^\mathsf{T}\mathbf{d}, \;\text{ or }\; \hat{\beta}_k = \big(1/\mathbf{a}^\mathsf{T}\mathbf{d}\big)\Big(B + \sum_i B_i\Big)$$

Thus we have $\hat{\beta}_k = \bar{\beta}$ for all k, and $\hat{\mathbf{u}} = \hat{\beta}\mathbf{d} = \bar{\beta}\mathbf{d} = \bar{u}$ solves the original problem (4.11). □

To prove Proposition 38, it is useful to reformulate (4.5) as follows:

$$\max_{v_0, \mathbf{v}} \big\{ \overline{W}(v_0, \mathbf{v}) \;\big|\; a_N v_0 + \mathbf{a}^\mathsf{T}\mathbf{v} \le B,\; \mathbf{v} \ge \mathbf{0} \big\} \tag{A.8}$$

where $\boldsymbol{v} = (v_1, \ldots, v_n)$ and

$$\overline{W}(v_0, \boldsymbol{v}) = (n-1)\Delta + n v_0 + \sum_{i=1}^{n} \big(v_i - \Delta\big)^+$$

Lemma 44. *Formulation* (A.8) *has the same optimal value as* (4.5).

Proof. It suffices to show that for any feasible solution of (4.5), there is a feasible solution of (A.8) with value at least as large as that of (4.5), and vice-versa. First consider any feasible solution $\boldsymbol{u}$ of (4.5). If we let $v_0 = u_{\min}$ and $v_j = u_j - u_{\min}$ for all j, the solution $(v_0, \mathbf{v})$ is feasible in (A.8), given the constraints of (4.5). Also the objective function $\overline{W}$ is identical to W, and so $\overline{W}(v_0, \mathbf{v}) = W(\mathbf{u})$.

Now suppose that $(v_0, \mathbf{v})$ is feasible in (A.8). Set $u_j = v_0 + v_j$ for all j, which implies $u_{\min} = v_0 + v_{\min}$, where $v_{\min} = \min_j\{v_j\}$. The constraint $\boldsymbol{a}^\mathsf{T}\boldsymbol{u} \le B$ of (4.5) becomes $a_N v_0 + \mathbf{a}^\mathsf{T}\mathbf{v} \le B$ in (A.8), so that $\boldsymbol{u}$ is feasible in (4.5). The objective function of

(4.5) becomes

$$(n-1)\Delta + nv_{\min} + nv_0 + \sum_j \big(v_j - v_{\min} - \Delta\big)^+$$

which can be written

$$(n-1)\Delta + nv_0 + \sum_j \max\big\{v_j - \Delta,\ v_{\min}\big\}$$

This is no smaller than the objective function of (A.8) because $v_{\min} \geq 0$. □

Proof of Proposition 38. We will show that $(v_0^*, \boldsymbol{v}^*) = (0, (B/a_1)\boldsymbol{e}_1)$ is an optimal solution of (A.8) if (4.18) holds, and $(v_0^*, \boldsymbol{v}^*) = (B/a_N, \mathbf{0})$ is an optimal solution otherwise. This proves the theorem because (4.5) and (A.8) have the same optimal value by Lemma 44, and because

$$W\big((B/a_1)\boldsymbol{e}_1\big) = (n-2)\Delta + B/a_1 = \overline{W}\big(0, (B/a_1)\boldsymbol{e}_1\big)$$
$$W\big((B/a_N)\boldsymbol{e}\big) = (n-1)\Delta + nB/a_N = \overline{W}\big(B/a_N, \mathbf{0}\big)$$

We first observe that the objective function $\overline{W}(v_0, \boldsymbol{v})$ is convex because $(v_j - \Delta)^+$ is a convex function of v_j, and a sum of convex functions is convex. It follows that some extreme point of the feasible set of (A.8) is optimal. Yet every extreme point is the solution of some linearly independent subset T of $n+1$ of the following equations:

$$a_N v_0 + \boldsymbol{a}\boldsymbol{v} = B \qquad (a)$$
$$v_0 = 0 \qquad (b)$$
$$v_i = 0,\ i = 1, \ldots, n \qquad (c)$$

We can suppose T contains (a), since otherwise the corresponding extreme point $(v_0, \boldsymbol{v}) = (0, \mathbf{0})$ is clearly dominated by $(0, (B/a_1)\boldsymbol{e}_1)$ and $(B/a_N, \mathbf{0})$. Then T either contains (b) and all but one equation $v_j = 0$ in (c), or else all equations in (c). The former yields extreme point $(v_0, \boldsymbol{v}) = (0, (B/a_j)\boldsymbol{e}_j)$ and the latter $(v_0, \boldsymbol{v}) = (B/a_N, \mathbf{0})$. Now if (4.18) holds, both $\overline{W}(0, (B/a_j)\boldsymbol{e}_j) = (n-2)\Delta + B/a_t$ and $\overline{W}(B/a_N, \mathbf{0}) = (n-1)\Delta + nB/a_N$ are less than or equal to $\overline{W}(0, (B/a_t)\boldsymbol{e}_t)$ because $a_1 \leq a_j$. If (4.18) does not hold, both of these expressions are less than or equal to $\overline{W}(B/a_N, \mathbf{0})$. The proposition follows. □

Proof of Proposition 39. We first show that maximizing a utility threshold SWF subject to a budget constraint and upper and lower utility bounds yields an optimal solution $\mathbf{u}^*$ in which at most one utility u_i^* is strictly between $u_{\min}^*$ and d_i.

Let S be the feasible set of (4.6), and define

$$S_i = S \cap \{\boldsymbol{u} \mid u_i \leq u_j, \text{ all } j\}$$

Since S is the union of all S_i, the maximum of $W_\Delta(\boldsymbol{u})$ over some S_i is optimal in (4.3). Suppose without loss of generality that the maximum of $W_\Delta(\boldsymbol{u})$ over S_1 is optimal in (4.3). For $\boldsymbol{u} \in S_1$, the function $W_\Delta(\boldsymbol{u})$ can be written

$$W'_\Delta(\boldsymbol{u}) = (n-1)\Delta + \sum_{i=1}^{n} \max\left\{u_i - \Delta,\ u_1\right\}$$

Since $W'_\Delta(\boldsymbol{u})$ is convex, some extreme point $\boldsymbol{u}^*$ of S_1 maximizes $W'_\Delta(\boldsymbol{u})$ and therefore $W_\Delta(\boldsymbol{u})$ over S_1. Since $\boldsymbol{u}^*$ is an extreme point of S_1, it is the solution of some linearly

independent[1] subset E of n of the equations

$$
\begin{aligned}
& u_1 - u_j = 0,\ j = 2, \ldots, n && (a) \\
& \mathbf{a}^{\mathsf{T}}\mathbf{u} = B && (b) \\
& \mathbf{u} = \mathbf{c} && (c) \\
& \mathbf{u} = \mathbf{d} && (d)
\end{aligned}
$$

Let T be the subset of equations in E that appear in (c) or (d). This means that $n - |T|$ variables are not fixed to one of their bounds by (c) and (d). Suppose that 2 or more of these variables are not set equal to u_1 by (a). Then the only nonzero in each of the corresponding columns must appear in (b). These columns must therefore be linearly dependent, which is impossible because E is nonsingular. We conclude that at most one variable u_i^* is fixed neither to a bound by (c) and (d) nor to $u_1^* = u_{\min}^*$ by (a). Since u_i^* cannot be strictly between its lower bound c_i and $u_{\min}^*$, the proposition follows.

We next show that if there is utility strictly between $u_{\min}^*$ and its upper bound, then some other utility u_j^* that is equal to $u_{\min}^*$ is at its lower or upper bound. Let Q be the set of inequalities (a) in E. Then Q contains $|Q| + 1$ variables, and T contains $|T|$ variables. Since m variables are strictly between $u_{\min}^*$ and their upper bounds, these variables are in neither Q nor T, and E must therefore contain all m rows of (b). If Q and T have no variables in common, the number of variables is at least $|Q| + |T| + m + 1$. But $|Q| + |T| + m + 1 > n$ because $|Q| + |T| + m = n$ due to the linear independence of the equations in E. Thus one variable u_j^* belongs to both Q and T, which implies that $u_j^* = u_1^* = u_{\min}^*$ and u_j^* is at its lower or upper bound. □

[1]We consider equations to be linearly independent when their coefficient rows are linearly independent.

Proof of Proposition 40. Let S be the feasible set of (4.11), and define

$$S_i = S \cap \{(\mathbf{y}, \mathbf{u}) \mid u_i \leq u_j, \text{ all } j\}$$

Since S is the union of all S_i, the maximum of $F(\boldsymbol{u})$ over some S_i is optimal in (4.11). Suppose without loss of generality that the maximum of $F(\boldsymbol{u})$ over S_1 is optimal in (4.11). For $(\mathbf{y}, \mathbf{u}) \in S_1$, the function $F(\boldsymbol{u})$ can be written

$$F_1(\boldsymbol{u}) = (n-1)\Delta + nu_1 + \sum_{i=1}^{n} \big(u_i - u_1 - \Delta\big)^+$$

Since $F_1(\boldsymbol{u})$ is convex, some extreme point $(\bar{\mathbf{y}}, \bar{\mathbf{u}})$ of S_1 maximizes $F_1(\boldsymbol{u})$ and therefore $F(\boldsymbol{u})$ over S_1. Since $(\bar{\mathbf{y}}, \bar{\mathbf{u}})$ is an extreme point of S_1, it is the solution of some linearly independent subset E of $n + r$ of the equations

$$\begin{aligned}
&u_1 - u_j = 0, \ j = 2, \ldots, n && (a)\\
&\mathbf{e}^{\mathsf{T}}\mathbf{y} = B && (b)\\
&\mathbf{a}^k\mathbf{u} - y_k = B_k, \ k = 1, \ldots, r && (c)\\
&u_i = c_i, \ i = 1, \ldots, n && (d)\\
&u_i = d_i, \ i = 1, \ldots, n && (e)\\
&y_k = 0, \ k = 1, \ldots, r && (f)
\end{aligned}$$

We first demonstrate (i). Let T be the subset of equations in E that appear in (d) or (e). This means that $n - |T|$ variables $\bar{u}_j$ are not fixed to one of their bounds by (d) and (e). Suppose that $r+1$ of these variables are not set equal to $\bar{u}_1$ by (a). Then all of the nonzeros in the corresponding columns must occur in the r rows of (c). The $r+1$ columns must therefore be linearly dependent, which is impossible because is E

is nonsingular. We conclude that at most r variables $\bar{u}_j$ are fixed neither to a bound by (d) and (e) nor to $\bar{u}_1 = \bar{u}_{\min}$ by (a). Since these $\bar{u}_j$s cannot be strictly between their lower bound c_j and $\bar{u}_{\min}$, (i) follows.

We now demonstrate (ii). Let T be as before. This means that $n - |T|$ variables $\bar{u}_j$ are not fixed to one of their bounds by (d) and (e). Suppose for any given k that 2 of these variables that are in $\bar{\mathbf{u}}^k$ are not set equal to $\bar{u}_1$ by (a). Then all of the nonzeros in the corresponding columns must occur in row k of (c). The 2 columns must therefore be linearly dependent, which is impossible because is E is nonsingular. We conclude that at most one variable $\bar{u}_j$ in $\bar{\mathbf{u}}^k$ is strictly between its lower bound c_j and $\bar{u}_{\min}$, and (ii) follows. □

To prove Proposition 41, we first show that the social welfare problem has a linear programming model.

Lemma 45. *If $W_\Delta(\mathbf{u})$ is the equity threshold SWF given by* (4.19)*, the optimization problem* (4.3) *has the linear programming formulation*

$$\max_{\mathbf{u},\mathbf{v},w} \left\{ n\Delta + \sum_{j=1}^{n} v_j \;\middle|\; \begin{array}{l} w \le u_j,\ v_j \le u_j - \Delta,\ v_i \le w,\ j = 1, \dots n \\ \sum_j a_j u_j \le B,\ \ w \ge 0,\ \ \mathbf{c} \le \mathbf{u} \le \mathbf{d} \end{array} \right\} \tag{A.9}$$

where $\mathbf{v}$ *and* w *are auxiliary variables.*

Proof. It suffices to show that for any feasible solution of (A.9), some feasible solution of (4.3) has an objective function value at least as large, and vice-versa. For the first claim, if we let $(\mathbf{u}, \mathbf{v}, w)$ be a feasible solution of (A.9), then $\mathbf{u}$ is obviously feasible

in (4.3), and it suffices to show that

$$n\Delta + \sum_j \min\left\{u_j - \Delta,\ u_{\min}\right\} \geq n\Delta + \sum_j v_j$$

But this follows because $v_j \leq u_j - \Delta$ and $v_j \leq w \leq u_{\min}$ for each j, the latter due to the fact that $w \leq u_j$ for each j. To show the converse, suppose $\mathbf{u}$ is a feasible solution of (4.3). It suffices to exhibit a feasible solution $(\mathbf{u}, \mathbf{v}, w)$ of (A.9) for which

$$n\Delta + \sum_j v_j \geq n\Delta + \sum_j \min\left\{u_j - \Delta,\ u_{\min}\right\} \tag{A.10}$$

The solution given by $w = u_{\min}$ and $v_j = \min\{u_j - \Delta, u_{\min}\}$ for each j is feasible on inspection. We also see that (A.10) follows immediately from the definition of v_j. □

Proof of Proposition 41. From Lemma 45, it suffices to show that (4.21) and (4.24) hold under their respective conditions for some optimal solution $\mathbf{u}^*$ of (A.9) with $\mathbf{c} = \mathbf{0}$ and $\mathbf{d} = \boldsymbol{\infty}$. Since (A.9) is an LP, we can show this by exhibiting a feasible solution $(\mathbf{u}^*, \mathbf{v}^*, w^*)$ of (A.9) that is consistent with (4.21) or (4.24), along with a feasible dual solution that yields the same objective function value as $(\mathbf{u}^*, \mathbf{v}^*, w^*)$. The dual of (A.9) (with $\mathbf{c} = \mathbf{0}$ and $\mathbf{d} = \boldsymbol{\infty}$) is

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\delta} \left\{ n\Delta + B\delta - \Delta \sum_{j=1}^{n} \beta_j \;\middle|\; \begin{array}{l} a_j\delta \geq \alpha_j + \beta_j,\ \beta_j + \gamma_j = 1,\ j \in N \\ \sum_j \alpha_j \geq \sum_j \gamma_j,\quad \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \geq \mathbf{0},\ \delta \geq 0 \end{array} \right\} \tag{A.11}$$

where dual variables $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \delta$ correspond respectively to the first four constraints of (A.9). We first consider the Δ range defined by (4.20). It suffices to show that the

optimal solution $(\mathbf{u}^*, \mathbf{v}^*, w^*)$ is optimal when $\mathbf{u}^*$ is as given by (4.21), $\mathbf{v}^*$ is given by

$$v_i^* = \begin{cases} u_0, & \text{for } i = 1, \ldots, k^* \\ u_0 - \Delta, & \text{for } i = k^* + 1, \ldots, n \end{cases}$$

and $w^* = u_0$. We note that $u_0 \geq 0$ due to (4.20). Given this, it is easily checked that $(\mathbf{u}^*, \mathbf{v}^*, w^*)$ is feasible in (A.9). To show it is optimal, we exhibit the dual solution $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \delta)$ given by

$$(\alpha_i, \beta_i) = \begin{cases} \big(0,\ na_j/a_N\big), & \text{for } i = 1, \ldots, k^* \\ \big(na_j/a_N - 1,\ 1\big), & \text{for } i = k^* + 1, \ldots, n \end{cases}$$

as well as $\gamma_i = 1 - \beta_i$ for $i = 1, \ldots, n$ and $\delta = n/a_N$. We note that $\alpha_i \geq 0$ because $na_i/a_N \geq 1$ for $i = k^* + 1, \ldots, n$ due to the definition of k^*. Given this, direct substitution confirms that $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \delta)$ is dual feasible. Furthermore, the objective value of both primal and dual is

$$k\Delta + \frac{n}{a_N}\Big(B - \Delta \sum_{j=1}^{k^*} a_j\Big)$$

This shows that $(\mathbf{u}^*, \mathbf{v}^*, w^*)$ is optimal.

We now consider the Δ range defined by (4.23). It suffices to show that the optimal solution $(\mathbf{u}^*, \mathbf{v}^*, w^*)$ is optimal when $\mathbf{u}^*$ is given by (4.24) for values of $k \leq k^*$ satisfying (4.22), when $v_i^* = u_i^* - \Delta$ for all i, and when $w^* = 0$. We note that $u_k^* \geq 0$ due to the second inequality in (4.22). Given this, it is easily checked that $(\mathbf{u}^*, \mathbf{v}^*, w^*)$ is feasible in (A.9). To show it is optimal, we exhibit the dual solution

$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \delta)$ given by

$$(\alpha_i, \beta_i) = \begin{cases} \big(0,\ a_i/a_k\big), & \text{for } i = 1, \ldots, k \\ \big(a_i/a_k - 1,\ 1\big), & \text{for } i = k+1, \ldots, n \end{cases}$$

as well as $\gamma_i = 1 - \beta_i$ for $i = 1, \ldots, n$ and $\delta = 1/a_k$. We note that $\alpha_i \geq 0$ because $a_i \geq a_k$ for $i = k+1, \ldots, n$. We also note that the dual constraint $\sum_i \alpha_i \geq \sum_i \gamma_i$ reduces to $a_k \leq a_N/n$, which holds because $k \leq k^*$ and the definition of k^* implies $a_{k^*} \leq a_N/n$. Given these facts, direct substitution confirms that $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \delta)$ is dual feasible. Furthermore, the objective value of both primal and dual is

$$(k-1)\Delta + \frac{1}{a_k}\Big(B - \Delta \sum_{i=1}^{k-1} a_i\Big)$$

This shows that $(\mathbf{u}^*, \mathbf{v}^*, w^*)$ is optimal. $\square$

To prove Proposition 42, we note that when the social welfare problem has only a budget constraint, problem P_k has the form

$$\min_{\mathbf{u}_K} \left\{ W_k(\mathbf{u}_K) \;\middle|\; \begin{array}{l} u_i \geq \bar{u}_{i_{k-1}},\ i \in I_k \\ \displaystyle\sum_{i \in I_k} a_i u_i \leq B_k \end{array} \right\} \tag{A.12}$$

where

$$W_k(\mathbf{u}_K) = (n-k+1)u_{\min} + \sum_{i \in I_k} (u_i - \bar{u}_{i_1} - \Delta)^+$$

and $u_{\min} = \min_{i \in I_k}\{u_i\}$. To simplify notation, we offset the utilities with the change

of variable $u'_i = u_i - \bar{u}_{i_{k-1}}$. Problem P_k now becomes a problem P'_k of the form

$$\max_{\mathbf{u}'_K} \left\{ W'_k(\mathbf{u}'_K) \;\middle|\; \begin{array}{l} u'_i \geq 0,\ i \in I_k \\ \displaystyle\sum_{i \in I_k} a_i u'_i \leq B'_k \end{array} \right\} \tag{A.13}$$

where $B'_k = B_k - a_K \bar{u}_{i_{k-1}}$ and

$$W'_k(\mathbf{u}'_K) = (n-k+1)u'_{\min} + \sum_{i \in I_k} (u'_i - \Delta')^+$$

with $\Delta' = \Delta - (\bar{u}_{i_{k-1}} - \bar{u}_{i_1})$. It is useful to reformulate (A.13) as follows:

$$\max_{v_0, \mathbf{v}} \left\{ \overline{W}_k(v_0, \mathbf{v}) \;\middle|\; \begin{array}{l} v_0 \geq 0,\ v_i \geq 0,\ i \in I_k \\ \displaystyle a_K v_0 + \sum_{i \in I_k} a_i v_i \leq B'_k \end{array} \right\} \tag{A.14}$$

where

$$\overline{W}(v_0, \boldsymbol{v}) = (n-k+1)v_0 + \sum_{i \in I_k} \left(v_i + v_0 - \Delta'\right)^+$$

Lemma 46. *Formulation* (A.14) *has the same optimal value as* (A.13)*, and solution* $\mathbf{u}'_K$ *is optimal in* (A.13) *if and only if* $(v_0, \mathbf{v})$ *is optimal in* (A.14)*, where* $v_0 = u'_{\min}$ *and* $v_i = u'_i - u'_{\min}$ *for* $i \in I_k$*.*

Proof. It suffices to show that (a) if $\mathbf{u}'_K$ is optimal in (A.13), then $(v_0, \mathbf{v})$ is optimal in (A.14), where $v_0 = u'_{\min}$ and $v_i = u'_i - u'_{\min}$, and (b) if $(v_0, \mathbf{v})$ is optimal in (A.14), then $\mathbf{u}'_K$ is optimal in (A.13), where $u'_i = v_0 + v_i$ for $i \in I_k$. To show (a), suppose that $\mathbf{u}'_K$ is optimal in (A.13). Then $(v_0, \mathbf{v})$ is feasible in (A.14), given the constraints of (A.13). Also $\overline{W}_k(v_0, \mathbf{v}) = W'_k(\mathbf{u}'_K)$, and so $(v_0, \mathbf{v})$ is optimal in (A.14).

To show (b), suppose that $(v_0, \mathbf{v})$ is optimal in (A.14). Set $u'_i = v_0 + v_i$ for $i \in I_k$, which implies $u'_{\min} = v_0 + v_{\min}$, where $v_{\min} = \min_{i \in I_k}\{v_i\}$. It is easily checked that $\mathbf{u}'_K$ is feasible in (A.13). Then $W'_k(\mathbf{u}'_K)$ becomes

$$(n-k+1)(v_{\min} + v_0) + \sum_{i \in I_k} \big(v_i + v_0 - \Delta'\big)^+$$

This is no smaller than $\overline{W}(v_0, \mathbf{v})$ because $v_{\min} \geq 0$. Since $\overline{W}(v_0, \mathbf{v})$ is the optimal value of (A.14), $\mathbf{u}'_K$ is optimal in (A.13). □

We next show that (A.13) has a utilitarian solution when the following holds, and a maximin solution otherwise:

$$\frac{1}{n-k} B'_k \left(2\frac{n-k+1}{a_K} - \frac{1}{a_{\ell_k}} \right) \leq \Delta' \leq B'_k \left(\frac{1}{a_{\ell_k}} - \frac{n-k+1}{a_K} \right) \tag{A.15}$$

where we regard (A.15) as false when $n = k$.

Lemma 47. *If $B'_k \geq 0$, problem* (A.13) *has a optimal solution $\mathbf{u}'_K$ in which*

$$\mathbf{u}'_K = \begin{cases} (B'_k/a_{\ell_k})\mathbf{e}_{\ell_k}, & \textit{if (A.15) holds} \\ (B'_k/a_K)\mathbf{e}_K, & \textit{otherwise} \end{cases}$$

where $\ell_k = \arg\min_{\ell \in I_k}\{a_\ell\}$ and $\mathbf{e}_K$ is a vector of $|I_k|$ ones.

Proof. Due to Lemma 46, it suffices to show that problem (A.14) has an optimal solution in which

$$(v_0, \mathbf{v}) = \begin{cases} \big(0, (B'_k/a_{\ell_k})\mathbf{e}_{\ell_k}\big), & \text{if (A.15) holds} \\ \big(B'_k/a_K, \mathbf{0}\big), & \text{otherwise} \end{cases}$$

Since we solve P_k only for $k \leq k^*$, we know that $\bar{u}_{i_k} - \bar{u}_{i_1} \leq \Delta$, which implies $\Delta' \geq 0$. Thus we have

$$\overline{W}_k\big(0, (B'_k/a_{\ell_k})\mathbf{e}_{\ell_k}\big) = \left(\frac{B'_k}{a_{\ell_k}} - \Delta'\right)^+ \tag{A.16}$$

$$\overline{W}_k\big(B'_k/a_K, \mathbf{0}\big) = (n-k+1)\frac{B'_k}{a_K} + (n-k+1)\left(\frac{B'_k}{a_K} - \Delta'\right)^+ \tag{A.17}$$

We will show that $(v_0, \mathbf{v}) = (0, (B'_k/a_{\ell_k})\boldsymbol{e}_{\ell_k})$ is an optimal solution of (A.14) if (A.15) holds, and $(v_0, \mathbf{v}) = (B'_k/a_K, \mathbf{0})$ is an optimal solution otherwise. We first observe that the objective function $\overline{W}_k(v_0, \mathbf{v})$ is convex. It follows that some extreme point of the feasible set of (A.14) is optimal. Yet every extreme point is the solution of some linearly independent subset T of $n+1$ of the following equations:

$$a_K v_0 + \sum_{i \in I_k} v_i = B'_k \quad (a)$$

$$v_0 = 0 \quad (b)$$

$$v_i = 0,\ i \in I_k \quad (c)$$

We can suppose T contains (a), since otherwise the corresponding extreme point $(v_0, \boldsymbol{v}) = (0, \mathbf{0})$ is clearly dominated by $(0, (B'_k/a_{\ell_k})\mathbf{e}_{\ell_k})$ and $(B'_k/a_K, \mathbf{0})$. Then T either contains (b) and all but one equation $v_j = 0$ in (c), or else all equations in (c). The former yields extreme point $(v_0, \mathbf{v}) = (0, (B'_k/a_j)\mathbf{e}_j)$ and the latter $(v_0, \mathbf{v}) = (B'_k/a_K, \mathbf{0})$.

We first suppose that (A.15) holds and show that $(v_0, \mathbf{v}) = (0, (B'_k/a_{\ell_k})\mathbf{e}_{\ell_k})$ is optimal.

For this, it suffices to show

$$\overline{W}_k\big(0,(B'_k/a_j)\mathbf{e}_j\big) \le \overline{W}_k\big(0,(B'_k/a_{\ell_k})\mathbf{e}_{\ell_k}\big),\ \text{all } j \in I_k \tag{A.18}$$

$$\overline{W}_k\big(B'_k/a_K,\mathbf{0}\big) \le \overline{W}_k\big(0,(B'_k/a_{\ell_k})\mathbf{e}_{\ell_k}\big) \tag{A.19}$$

Due to (A.16)–(A.17), these are respectively equivalent to

$$\left(\frac{B'_k}{a_j}-\Delta'\right)^+ \le \left(\frac{B'_k}{a_{\ell_k}}-\Delta'\right)^+,\ \text{all } j \in I_k \tag{A.20}$$

$$(n-k+1)\frac{B'_k}{a_K}+(n-k+1)\left(\frac{B'_k}{a_K}-\Delta'\right)^+ \le \left(\frac{B'_k}{a_{\ell_k}}-\Delta'\right)^+ \tag{A.21}$$

But (A.20) holds simply because $a_{\ell_k} \le a_j$ for all $j \in I_k$, by definition of ℓ_k. To show (A.21), we distinguish three cases. *Case 1:* $\Delta' \le B'_k/a_K$. In this case, to establish (A.21) it suffices to show that

$$(n-k+1)\frac{B'_k}{a_K}+(n-k+1)\left(\frac{B'_k}{a_K}-\Delta'\right) \le \frac{B'_k}{a_{\ell_k}}-\Delta'$$

But this follow from the first inequality in (A.15). *Case 2:* $B'_k/a_K < \Delta' \le B_k/a_{\ell_k}$. Here it suffices to show that

$$(n-k+1)\frac{B'_k}{a_K} \le \frac{B'_k}{a_{\ell_k}}-\Delta'$$

This follows from the second inequality in (A.15). *Case 3:* $\Delta' > B'_k/a_{\ell_k}$. This case cannot occur because it violates the second inequality in (A.15). Since case 1 or 2 must obtain, (A.21) follows.

We now suppose that (A.15) is violated, which means that at least one of the two inequalities in (A.15) is violated. We wish to show that $(v_0,\mathbf{v}) = (B'_k/a_K,\mathbf{0})$ is

optimal. For this it suffices to show

$$\overline{W}_k\big(0,(B'_k/a_j)\mathbf{e}_j\big) \le \overline{W}_k\big(\frac{B'_k}{a_K},\mathbf{0}\big),\ \text{all } j \in I_k \tag{A.22}$$

which is equivalent to

$$\left(\frac{B'_k}{a_j}-\Delta'\right)^+ \le (n-k+1)\frac{B'_k}{a_K}+(n-k+1)\left(\frac{B'_k}{a_K}-\Delta'\right)^+,\ \text{all } j \in I_k \tag{A.23}$$

We again consider 3 cases.

Case 1: $\Delta' \le B'_k/a_K$. In this case, it suffices to show

$$\frac{B'_k}{a_j}-\Delta' \le (n-k+1)\frac{B'_k}{a_K}+(n-k+1)\left(\frac{B'_k}{a_K}-\Delta'\right),\ \text{all } j \in I_k \tag{A.24}$$

This follows if the first inequality in (A.15) is false, because $a_{\ell_k} \le a_j$ for all $j \in I_k$. It therefore suffices to show that the first inequality is false, given the case hypothesis. We will demonstrate that the first inequality cannot be true when the second is false, which means that the first implies the second. But we are given that the first and second inequalities cannot both be true because (A.15) is false, which means that the first inequality implies the negation of the second. It follows that the first inequality must be false. Now it remains to show that the first inequality cannot be true if the second is false. The case hypothesis $\Delta' \le B'_k/a_N$ together with the first inequality imply

$$\frac{1}{n-k}B'_k\left(2\frac{n-k+1}{a_K}-\frac{1}{a_{\ell_k}}\right) \le \Delta' \le \frac{B'_k}{a_K}$$

which implies

$$\frac{n-k+2}{a_K} \le \frac{1}{a_{\ell_k}} \tag{A.25}$$

The case hypothesis together with the falsehood of the second inequality imply

$$B'_k\left(\frac{1}{a_{\ell_k}}-\frac{n-k+1}{a_K}\right)<\Delta'\leq\frac{B'_K}{a_K}$$

which implies the negation of (A.25). Thus we have a contradiction, and we conclude that the first inequality in (A.15) cannot be true when the second is false, as claimed.

Case 2: $B'_k/a_K<\Delta'\leq B'_k/a_j$. In this case, it suffices to show

$$\frac{B'_k}{a_j}-\Delta'\leq(n-k+1)\frac{B'_k}{a_K},\ \text{ or }\ \Delta'\geq B_k\left(\frac{1}{a_j}-\frac{n-k+1}{a_K}\right),\ \text{all } j\in I_k$$

This follows if the second inequality in (A.15) is false, because $a_{\ell_k}\leq a_j$ for all $j\in I_k$. It therefore suffices to show that the second inequality is false, given the case hypothesis. We will demonstrate that the second inequality cannot be true when the first is false, which means that the second implies the first. But we are given that the first and second inequalities cannot both be true because (A.15) is false, which means that the second inequality implies the negation of the first. It follows that the second inequality must be false. Now it remains to show that the second inequality cannot be true if the first is false. The case hypothesis $\Delta'>B'_k/a_N$ together with the second inequality imply

$$\frac{B'_k}{a_K}<\Delta'\leq B'_k\left(\frac{1}{a_{\ell_k}}-\frac{n-k+1}{a_K}\right)$$

which implies the negation of (A.25). The case hypothesis $\Delta'>B'_k/a_N$ together with the falsehood of the first inequality imply

$$\frac{B'_k}{a_K}<\Delta'<\frac{1}{n-k}b'_k\left(2\frac{n=k+1}{a_K}-\frac{1}{a_{\ell_k}}\right)$$

which implies (A.25). Thus we have a contradiction, and we conclude that the second inequality in (A.15) cannot be true when the first is false, as claimed.

Case 3. $\Delta' > B'_k/a_{\ell_k}$. Since $a_{\ell_k} \leq a_j$ for any $j \in I_k$, it suffices in this case to show that

$$0 \leq (n-k+1)\frac{B'_k}{a_K}$$

But this is true because we are given that $B'_k \geq 0$. Since one of the 3 cases must obtain, the proof is complete. □

Proof of Proposition 42. Let $i_j = j$ for $j = 1, \ldots, n$. It suffices to prove that the following claims are true in some socially optimal solution $\bar{\mathbf{u}}$.

$$\bar{u}_k = 0 \text{ for } k = 1, \ldots, m-1 \tag{A.26}$$

$$\bar{u}_k = B/a_M \text{ for } k = m, \ldots, n \tag{A.27}$$

Proof of (A.26). The argument is by induction. For $k = 1$, we know that (4.18) is satisfied, by definition of m. Thus by Proposition 38, an optimal solution of P_1 is $\tilde{\mathbf{u}} = (0, \ldots, 0, B/a_N)$, and we can set $\bar{u}_1 = \min\{\tilde{u}_i \mid i = 1, \ldots, n\} = 0$. We now suppose that (A.26) is true for $k = 1, \ldots, \ell - 1$ and show it is also true for $k = \ell$ in some optimal solution. We note first that $\bar{u}_i = 0$ for $i = 1, \ldots, \ell - 1$ implies that $B_\ell = B$ and $B'_\ell = B_\ell - a_L \bar{u}_{\ell-1} = B_\ell = B$. Also $\Delta' = \Delta - (\bar{u}_{\ell-1} - \bar{u}_1) = \Delta$. Thus (A.15) reduces to (4.26). But (4.26), and therefore (A.15), are satisfied by definition of m. Since $B'_\ell \geq 0$, Lemma 47 implies that $\tilde{\mathbf{u}}'_L = (0, \ldots, 0, B'_\ell/a_n) = (0, \ldots, 0, B/a_n)$ solves P'_ℓ. This implies that $\tilde{u} = \tilde{u}' - \bar{u}_{\ell-1}\mathbf{e} = (0, \ldots, 0, B/a_n)$ solves P_ℓ, and we can let $\bar{u}_\ell = 0$.

Proof of (A.27). The argument is again by induction. First suppose (A.26) is true and consider $k = m$ in (A.27). We have $B'_m = B$ and $\Delta' = \Delta$ for the same reason as above, so that (4.26) again reduces to (A.15). But (4.26), and therefore (A.15), are violated, by definition of m. Due to this and $B'_m \geq 0$, Lemma 47 implies that P'_m has a solution $\tilde{\mathbf{u}}'_M = (B/a_M, \ldots, B/a_M)$, which implies that P_m has a solution $\tilde{\mathbf{u}}_M = \tilde{\mathbf{u}}'_M - \bar{u}_{m-1}\mathbf{e} = (B/a_M, \ldots, B/a_M)$ since $\bar{u}_{m-1} = 0$. We can therefore set $\bar{u}_m = B/a_M$. We now suppose (A.26) is true, and (A.27) is true for $k = m, \ldots, \ell - 1$, and we show (A.27) is also true for $k = \ell$ in some optimal solution. By the induction hypothesis,

$$B_\ell = B - \sum_{i=m}^{\ell-1} a_i \bar{u}_m = B - \frac{B}{M} \sum_{i=m}^{\ell-1} a_i$$

and

$$B'_\ell = B_\ell = \bar{u}_{\ell-1} \sum_{i=\ell}^{n} a_i = B - \frac{B}{a_M} \left(\sum_{i=m}^{\ell-1} a_i + \sum_{i=\ell}^{n} a_i \right) = B - \frac{B}{a_M} \cdot a_M = 0$$

Since $B'_\ell = 0$, the only feasible solution of P'_ℓ is $(\tilde{u}_\ell, \ldots, \tilde{u}_n) = (0, \ldots, 0)$. We can therefore set $\bar{u}'_\ell = 0$, which means $\bar{u}_\ell = B/a_M$, as claimed. $\square$

Bibliography

Aggoun, A. and Beldiceanu, N. (1993). Extending CHIP in order to solve complex scheduling and placement problems. *Mathematical and Computer Modelling*, 17:57–73.

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

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, 246(2):345–378.

Allahverdi, A. and Soroush, H. (2008). The significance of reducing setup times/setup costs. *European Journal of Operational Research*, 187(3):978–984.

Angulo, G., Ahmed, S., and Dey, S. S. (2016). Improving the integer L-shaped method. *INFORMS Journal on Computing*, 28(3):483–499.

Atakan, S., Bülbül, K., and Noyan, N. (2017). Minimizing value-at-risk in single-machine scheduling. *Annals of Operations Research*, 248(1-2):25–73.

Atkinson, A. B. (1970). On the measurement of inequality. *Journal of Economic Theory*, 2(3):244–263.

Ban, G., El Karoui, N., and Lim, A. (2018). Machine learning and portfolio optimization. *Management Science*, 64(3):1136–1154.

Baptiste, P., Le Pape, C., and Nuijten, W. (2001). *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*. Kluwer.

Bard, J. F. (1998). *Practical Bilevel Optimization*, volume 30 of *Nonconvex Optimization and Its Applications*. Springer US, Boston, MA.

Ben-Tal, A. and Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations research letters*, 25(1):1–13.

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

Bentham, J. (1789). *An Introduction to the Principles of Morals and Legislation*. reprinted by Oxford University Press (1907).

Bertsimas, D., Farias, V. F., and Trichakis, N. (2012). On the efficiency-fairness trade-off. *Management Science*, 58(12):2234–2250.

Best, M. J. and Grauer, R. R. (1991). On the sensitivity of mean-variance efficient portfolios to changes in asset means: Some analytical and computational results. *Review of Financial Studies*, 4(2):315–342.

Binato, S., Pereira, M. V. F., and Granville, S. (2001). A new Benders decomposition approach to solve power transmission network design problems. *IEEE Transactions on Power Systems*, 16(2):235–240.

Birge, J. R. and Louveaux, F. (2011). *Introduction to Stochastic Programming*. Springer Science & Business Media.

Birge, J. R. and Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392.

Broadie, M. (1993). Computing efficient frontiers using estimated parameters. *Annals of Operations Research*, 45(1):21–58.

Bülbül, K., Kücükyavuz, S., Noyan, N., and Şen, H. (2016). A two-stage chance-constrained mean-risk stochastic programming model for single-machine scheduling. Working Paper.

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

Ceria, S. and Stubbs, R. A. (2006). Incorporating estimation errors into portfolio selection: Robust portfolio construction. *Journal of Asset Management*, 7(2):109–127.

Chen, V. and Hooker, J. (2021). A guide to formulating equity and fairness in an optimization model.

Chen, V. and Hooker, J. N. (2020a). Balancing fairness and efficiency in an optimization model. *ArXiv preprint 2006.05963*.

Chen, V. and Hooker, J. N. (2020b). A just approach balancing Rawlsian leximax fairness and utilitarianism. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 221–227.

Chopra, V. and Ziemba, W. (1993). The effect of errors in means, variances, and covariances on optimal portfolio choice. *Journal of Portfolio Management*, 19(2):6–11.

Chopra, V. K. (1993). Mean-variance revisited: Near-optimal portfolios and sensitivity to input variations. *Journal of Investing*.

Ciré, A., Çoban, E., and Hooker, J. N. (2016). Logic-based Benders decomposition for planning and scheduling: A computational analysis. *Knowledge Engineering Review*, 31:440–451.

Contreras, I., Cordeau, J.-F., and Laporte, G. (2011). Benders decomposition for large-scale uncapacitated hub location. *Operations Research*, 59(6):1477–1490.

Cordeau, J.-F., Stojković, G., Soumis, F., and Desrosiers, J. (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4):375–388.

Dalton, H. (1920). The measurement of the inequality of incomes. *The Economic Journal*, 30(119):348.

DeMiguel, V., Garlappi, L., Nogales, F., and Uppal, R. (2009a). A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, 55(5):798–812.

DeMiguel, V., Garlappi, L., and Uppal, R. (2009b). Optimal versus naive diversification: How inneficient is the 1/n portfolio strategy? *Review of Financial Studies*, 22(5):1915–1953.

Elçi, Ö. and Hooker, J. (2022). Stochastic Planning and Scheduling with Logic-Based Benders Decomposition. *INFORMS Journal on Computing*, pages 773–781.

Elçi, Ö. and Noyan, N. (2018). A chance-constrained two-stage stochastic programming model for humanitarian relief network design. *Transportation Research Part B: Methodological*, 108:55–83.

Fabozzi, F. J., Kolm, P. N., Pachamanova, D. A., and Focardi, S. M. (2007). *Robust portfolio optimization and management*. John Wiley & Sons.

Fazel-Zarandi, M. M. and Beck, J. C. (2012). Using logic-based Benders decomposition to solve the capacity-and distance-constrained plant location problem. *INFORMS Journal on Computing*, 24(3):387–398.

Fazel-Zarandi, M. M., Berman, O., and Beck, J. C. (2013). Solving a stochastic facility location/fleet management problem with logic-based Benders' decomposition. *IIE Transactions*, 45(8):896–911.

Freeman, S., editor (2003). *The Cambridge Companion to Rawls*. Cambridge University Press.

Gade, D., Küçükyavuz, S., and Sen, S. (2014). Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, 144(1-2):39–64.

Garlappi, L., Uppal, R., and T., W. (2007). Portfolio selection with parameter and model uncertainty: A multi-prior approach. *Review of Financial Studies*, 20(1):41–81.

Geoffrion, A. M. (1972). Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260.

Geoffrion, A. M. and Graves, G. W. (1974). Multicommodity distribution system design by Benders decomposition. *Management Science*, 20(5):822–844.

Goldfarb, D. and Iyengar, G. (2003). Robust Portfolio Selection Problems. *Mathematics of Operations Research*, 28(1):1–38.

Gotoh, J. and Takeda, A. (2011). On the role of norm constraints in portfolio selection. *Computational Management Science*, 8(4):323–353.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of discrete mathematics*, volume 5, pages 287–326. Elsevier.

Grötschel, M. (2012). *Optimization stories*. Dt. Mathematiker-Vereinigung.

Guo, C., Bodur, M., Aleman, D. M., and Urbach, D. R. (2019). Logic-based Benders decomposition and binary decision diagram based approaches to stochastic distributed operating room scheduling. *arXiv*, 1907.13265v1.

Heching, A., Hooker, J. N., and Kimura, R. (2019). A logic-based Benders approach to home healthcare delivery. *Transportation Science*, 53:510–522.

Hooker, J. (2000a). *Logic-based methods for optimization: combining optimization and constraint satisfaction*. John Wiley & Sons.

Hooker, J. N. (2000b). *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. Wiley, New York.

Hooker, J. N. (2007). Planning and scheduling by logic-based Benders decomposition. *Operations Research*, 55(3):588–602.

Hooker, J. N. (2012). *Integrated Methods for Optimization, 2nd ed.* Springer.

Hooker, J. N. (2019a). Logic-based Benders decomposition for large-scale optimization. In Velasquez-Bermúdez, J., Khakifirooz, M., and Fathi, M., editors, *Large Scale Optimization Applied to Supply Chain and Smart Manufacturing: Theory and Real-Life Applications*, pages 1–26. Springer.

Hooker, J. N. (2019b). Logic-based benders decomposition for large-scale optimization. In *Large Scale Optimization in Supply Chains and Smart Manufacturing*, pages 1–26. Springer.

Hooker, J. N. and Ottosson, G. (2003). Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60.

Hooker, J. N. and Williams, H. P. (2012). Combining equity and utilitarianism in a mathematical programming model. *Management Science*, 58(9):1682–1693.

Jagannathan, R. and Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. *Journal of Finance*, 58:1651–1684.

Jula, P. and Rafiey, A. (2012). Coordinated scheduling of a single machine with sequence-dependent setup times and time-window constraints. *International Journal of Production Research*, 50(8):2304–2320.

Kalai, E. and Smorodinsky, M. (1975). Other solutions to Nash's bargaining problem. *Econometrica*, 43:513–518.

Kan, R. and Zhou, G. (2007). Optimal portfolio choice with parameter uncertainty. *Journal of Financial and Quantitative Analysis*, 42(3):621–656.

Kelly, F. P., Maulloo, A. K., and Tan, D. K. H. (1998). Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252.

Kocuk, B. and Cornuéjols, G. (2020). Incorporating black-litterman views in portfolio construction when stock returns are a mixture of normals. *Omega*, 91:102008.

Kohl, N., Desrosiers, J., Madsen, O. B. G., Solomon, M. M., and Soumis, F. (1999). 2-Path Cuts for the Vehicle Routing Problem with Time Windows. *Transportation Science*, 33(1):101–116.

Küçükyavuz, S. and Sen, S. (2017). An introduction to two-stage stochastic mixed-integer programming. In *Leading Developments from INFORMS Communities*. INFORMS TutORials in Operations Research.

Laborie, P., Rogerie, J., Shaw, P., and Vilím, P. (2018). IBM ILOG CP optimizer for scheduling: 20+ years of scheduling with constraints at IBM/ILOG. *Constraints*, 23(2):210–250.

Lan, T., Kao, D., Chiang, M., and Sabharwal, A. (2010). An axiomatic theory of fairness in network resource allocation. In *Conference on Information Communications (INFOCOM 2010)*, pages 1343–1351. IEEE.

Laporte, G., Louveaux, F., and Mercure, H. (1992). The vehicle routing problem with stochastic travel times. *Transportation Science*, 26(3):161–170.

Laporte, G. and Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142.

Lenstra, J. K., Kan, A. R., and Brucker, P. (1977). Complexity of machine scheduling problems. In *Annals of discrete mathematics*, volume 1, pages 343–362. Elsevier.

Li, C. and Grossmann, I. E. (2018). An improved L-shaped method for two-stage convex 0–1 mixed integer nonlinear stochastic programs. *Computers & Chemical Engineering*, 112:165–179.

Lim, A., Shanthikumar, J., and Vahn, G. (2012). Robust portfolio choice with learning in the framework of regret: Single-period case. *Management Science*, 58(9):1732–1746.

Lin, S.-W. and Ying, K.-C. (2014). ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times. *Computers & Operations Research*, 51:172–181.

Lombardi, M., Milano, M., Ruggiero, M., and Benini, L. (2010). Stochastic allocation and scheduling for conditional task graphs in multi-processor systems-on-chip. *Journal of Scheduling*, 13:315–345.

Markowitz, H. (1952). Portfolio analysis. *Journal of Finance*, 8:77–91.

Mazumdar, R., Mason, L., and Douligeris, C. (1991). Fairness in network optimal flow control: Optimality of product forms. *IEEE Transactions on Communications*, 39(5):775–782.

Michaud, R. (2008). *Efficient asset management*. Oxford University Press.

Mo, J. and Walrand, J. (2000). Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8:556–567.

Nash, J. (1950). The bargaining problem. *Econometrica*, 18:155–162.

Noyan, N., Balcik, B., and Atakan, S. (2015). A stochastic optimization model for designing last mile relief networks. *Transportation Science*, 50(3):1092–1113.

Olivares-Nadal, A. V. and DeMiguel, V. (2018). Technical Note—A Robust Perspective on Transaction Costs in Portfolio Optimization. *Operations Research*, 66(3):733–739.

Prékopa, A. (2013). *Stochastic Programming*, volume 324. Springer Science & Business Media.

Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817.

Rawls, J. (1999). *A Theory of Justice* (revised). Harvard University Press (original edition 1971).

Richardson, H. S. and Weithman, P. J., editors (1999). *The Philosophy of Rawls* (5 volumes). Garland.

Rocha, P. L., Ravetti, M. G., Mateus, G. R., and Pardalos, P. M. (2008). Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. *Computers & Operations Research*, 35(4):1250–1264.

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

Scherer, B. (2007). Can robust portfolio optimisation help to build better portfolios? *Journal of Asset Management*, 7(6):374–387.

Sen, A. (1995). *Inequality Reexamined*. Oxford University Press.

Sen, S. and Higle, J. L. (2005). The C3 theorem and a D2 algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming*, 104(1):1–20.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2009). *Lectures on Stochastic Programming: Modeling and Theory*. SIAM.

Simchi-Levi, D., Trichakis, N., and Zhang, P. Y. (2019). Designing response supply chain against bioattacks. *Operations Research*, 67(5):1246–1268.

Solak, S., Scherrer, C., and Ghoniem, A. (2014). The stop-and-drop problem in nonprofit food distribution networks. *Annals of Operations Research*, 221(1):407–426.

Stubbs, R. A. and Vance, P. (2005). Computing return estimation error matrices for robust optimization. *Axioma Research Paper*.

ter Horst, J., De Roon, F., and Werker, B. J. (2006). Incorporating estimation risk in portfolio choice. *Advances in Corporate Finance and Asset Pricing. Elsevier, Amsterdam*, pages 449–472.

Thorsteinsson, E. (2001). Branch and check: A hybrid framework integrating mixed integer programming and constraint logic programming. In Walsh, T., editor, *Principles and Practice of Constraint Programming (CP 2001)*, volume 2239 of *Lecture Notes in Computer Science*, pages 16–30. Springer.

Toth, P. and Vigo, D., editors (2002). *The vehicle routing problem*. SIAM monographs on discrete mathematics and applications. Society for Industrial and Applied Mathematics, Philadelphia, Pa.

Tran, T. T., Araujo, A., and Beck, J. C. (2016). Decomposition Methods for the Parallel Machine Scheduling Problem with Setups. *INFORMS Journal on Computing*, 28(1):83–95.

Tutuncu, R. and Koenig, M. (2004). Robust asset allocation. *Annalas of Operations Research*, 132:157–187.

Van Slyke, R. M. and Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663.

Williams, A. and Cookson, R. (2000). Equity in Health. *Culyer, A.J. and Newhouse, J.P. editors, Handbook of Health Economics.*

Ying, K.-C., Lee, Z.-J., and Lin, S.-W. (2012). Makespan minimization for scheduling unrelated parallel machines with setup times. *Journal of Intelligent Manufacturing*, 23(5):1795–1803.

Ying, K.-C. and Lin, S.-W. (2012). Unrelated parallel machine scheduling with sequence-and machine-dependent setup times and due date constraints. *International Journal of Innovative Computing, Information and Control*, 8(5):3279–3297.